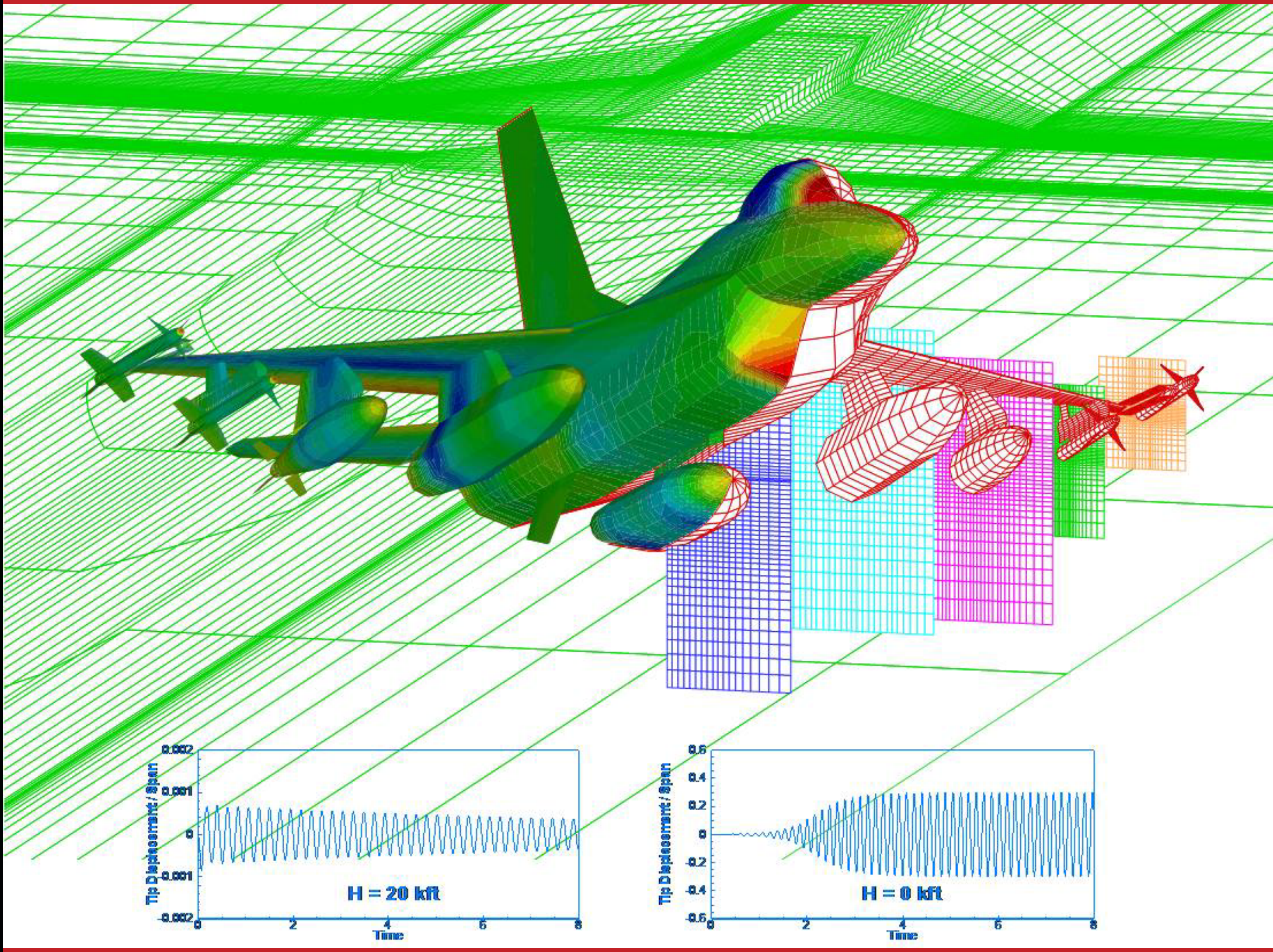
ZEUS v. 4.0

User's Manual

A CFD Code for Aeroelasticians



ZONA TECHNOLOGY INC

ZEUS

Version 4.0

USER'S MANUAL

ZONA 09 – 18
March 2020

Fifth Edition 06-19

ZONA TECHNOLOGY INC

ZONA Technology, Inc. • 9489 E. Ironwood Square Drive • Scottsdale, AZ 85258-4578
Tel: (480) 945-9988 • Fax: (480) 945-6588 • E-mail: info@zonatech.com

TABLE OF CONTENTS

PAGE

TABLE OF CONTENTS (CONT.)

PAGE

TABLE OF CONTENTS (CONT.)

PAGE

This page is intentionally left blank.

FOREWORD

ZEUS UPGRADES

This section lists the enhancements made to the ZEUS software system.

- ***Version 4.0 Enhancements***

1. A new bulk data card called **STABDRV** is introduced to compute the aerodynamic force, moment, and damping stability derivatives using the linearized Euler solver. The **STABDRV** bulk data card is referred to by a new case control command called **STABDRV**. A new testcase for the STABDRV module is created in Chapter 11 of the application manual.

2. Flutter solution for anti-symmetric modes of half-span structural and aerodynamic models can be obtained using the linearized Euler solver. This can be achieved using the executive control command "**ASSIGN FEM**=" with entry BOUNDARY=ANTI to import the anti-symmetric modal solution. Then, specifying the entry SYM=ANTI in the **FLUTTER** bulk data card and the entry METHOD=3 in the **MKPARAM** bulk data card via the **MKAEROZ** bulk data card yield the anti-symmetric flutter solution.

3. A new testcase is added to Chapter 8.8 of the application manual. This testcase shows how to perform a symmetric and an anti-symmetric flutter analysis of a half-span F-16 with stores model using the linearized Euler solver.

4. A new bulk data card called **FEMSAVE** is available to save the structural modal solution imported by the "**ASSIGN FEM**=" executive control commend. This saved structural modal solution can be retrieved by specifying the entry FORM="ACQUIRE" in the "**ASSIGN FEM=**" executive control command.

5. ZONA's Design Variable Linking Scheme (ZLINK) module is included within the ZEUS software. ZLINK is used to describe, and is completely responsible for, the geometrical shape change (i.e., morph) that takes place. It allows for the input of design variables, independent and dependent variable definitions, and arbitrary general functions of virtually any complexity. Five new bulk data cards called **DESDEP**, **DESFUN**, **DESIND**, **DESSEN**, and **DESVAR** are introduced to execute the ZLINK module.

6. A new bulk data card called **PLTSLP** is introduced that generates a Tecplot file for plotting the slope distribution on the computational surface mesh. If the program encounters a convergence problem, it is highly recommended to use the **PLTSLP** bulk data card for verifying the correct slopes on the computational surface mesh.

7. A new strategy for improving the efficiency of the Message Passing Interface (MPI) is implemented for the linearized Euler solver. According to the number of MPI processes specified by the **MPICPU** executive control command, the new MPI strategy can automatically ensure that all MPI processes are always fully loaded to execute the linearized Euler solver for computing the frequency-domain generalized aerodynamic forces of all reduced frequency-mode pairs.

8. For two surfaces physically connected together such as a wing-pylon or fin-body configuration but these two surfaces belong to different blocks of mesh, the previous version might have the flow "leakage problem". For instance, the overset mesh at the wing-pylon junction might not be completely sealed since the wing and pylon are in different blocks of mesh due to which the flow could pass through the root of the pylon and create a non-physical vortex roll-up. This flow leakage problem is mitigated by the new version. If the user wishes to obtain the results from the previous version, he/she can specify the bulk data card **PARAM** with entry NAME="VROLLUP" and VALUE=0.

9. Two new options of the entry FORM=RFORMAT and FORM=RUNFORM are added to the "**ASSIGN MATRIX**=" executive control command to import a non-spare matrix in ASCII format and binary format, respectively.

- ***Version 3.9 Enhancements***

1. The transpiration boundary condition for the linearized Euler solver has been slightly modified so that the real part of the unsteady pressure due to rigid body pitch mode at zero reduced frequency is proportional to the imaginary part of the unsteady pressure due to rigid body plunge mode at small but non-zero reduced frequency.

2. Because the modification of the transpiration boundary condition, the unsteady aerodynamic solution computed by the linearized Euler solver in Version 3.9 is slightly different from that of the previous version. The result of the testcases for the linearized Euler solver documented in Applications Manual is slightly different between Version 3.9 and the previous version.

- ***Version 3.8 Enhancements***

1. The MPICPU Executive Control Command is intended to parallelize to both linearized Euler solver and time accurate transient analysis.

2. Two new bulk data cards are introduced. The first is called **BLKMPI** to specify block assignment for MPI load balance strategy for time accurate transient analysis such as MLOADS, GLOADS, ELOADS, NLFLTR, and NANSI analysis. The second is called **FINDMPI** to determine automated load balancing for MPI.

- ***Version 3.7 Enhancements***

1. The Message Passing Interface (MPI) is incorporated in ZEUS to perform parallel computation for generating the frequency-domain generalized aerodynamic forces using the linearized Euler solver. A new executive control command called "MPICPU" is added to the executive control command section to activate this parallel computation.

- ***Version 3.6 Enhancements***

1. A new bulk data card called **KEXPAN** has been added to the bulk data card section that can circumvent the inaccurate frequency-domain unsteady aerodynamics problem generated by the linearized Euler solver at low reduced frequencies.

2. A summary for checking the convergence of the linearized Euler solver for each reduced frequency and mode pair is printed on the standard output file. The user can use this summary table to verify the accuracy of the frequency-domain generalized aerodynamic forces computed by the linearized Euler solver.

3. A new bulk data card called **RESTART** has been added to the bulk data section that allows the transient analysis by the MLOADS, GLOADS, ELOADS, and NLFLTR module to save or retrieve the entire flow solution at every incremental time steps specified by the **RESTART** bulk data card.

4. A new entry “MODEZRO” is added to the **MLDTIME** bulk data card that can force the generalized coordinates of selected modes to be zero during the transient analysis.

5. For ZEUS run performing FLUTTER/GENGAF linearized Euler analysis, an additional log file (as yourcase_prog.log) will be generated in working directory together with standard output file (as yourcase.out) and log file (as yourcase.log) where yourcase.inp is the ZEUS input file name.

6. A bug in the "H2F" option in the **AEROZ** bulk data card has been fixed. This bug may introduce memory over-write that could lead to incorrect mode shapes on the full span model.

Chapter 1

INTRODUCTION

1.1 WHAT IS ZEUS

ZEUS is a ZONA's Euler Unsteady Aerodynamic Solver that integrates the essential disciplines required for aeroelastic design/analysis. It uses a Euler equation solver with/without viscous effects as the underlying aerodynamic force generator coupled with structural finite element modal solution to solve various aeroelastic problems such as flutter, maneuver loads, store ejection loads, gust loads, and static aeroelastic/trim analysis. Structural nonlinearities can also be included to perform a nonlinear aeroelastic analysis.

The Euler equation solver solves the Euler equations on a Cartesian grid system using a cell-centered finite volume method with dual-time stepping algorithm for unsteady solutions. The viscous effects are included by coupling the Euler solution with a steady boundary-layer equation. For turbulence closure, the Green's lag entrainment is employed. Because of solving the Euler equations with boundary layer coupling option, the requirement of large computing resources by a Navier-Stokes code can be avoided by ZEUS. Therefore, ZEUS provides a good balance between the complete modeling of the flow physics and the computational efficiency.

ZEUS uses the bulk data input format that is very similar to that of NASTRAN and ZAERO. In fact, the majority of the input bulk data cards of ZEUS are identical to those of ZAERO. This can minimize the learning curve for an experienced ZAERO user to learn ZEUS. The major difference in input between ZEUS and ZAERO is the mesh generation because ZEUS requires the mesh of the entire volume domain whereas ZAERO only needs the surface mesh. However, ZEUS equips an automated mesh generation scheme that can automatically generate a volume mesh by growing the mesh from the surface mesh. In addition, ZEUS has an overset mesh capability to handle very complex configurations such as whole aircraft with external stores.

In order to solve the Euler equation but retain the ease in setting up a computational mesh, ZEUS employs approximate boundary conditions applied on a Cartesian grid where the full Euler boundary condition is replaced by its first-order expansion on the mean plane of the lifting surface. For bodies such as fuselage and stores, ZEUS approximates the exact surface geometry of the body by a rectangular box. Using a slender body theory, the exact surface geometry is mapped onto the surface of the rectangular box where the boundary condition of the bodies is applied. This approximate boundary condition for lifting surfaces and bodies significantly reduces the grid generation effort for ZEUS because no conformal mesh is required. It is this approximate boundary condition that enables the development of an automated mesh generation scheme in ZEUS.

As an option, the body can be modeled by a body-fitted mesh where the exact Euler boundary condition is applied on the body surface mesh. However, this body-fitted mesh can only contain one body component and cannot contain other components like the wing or fin. Other components must be modeled by the Cartesian mesh so that this wing-body configuration is handled by the overset mesh scheme.

The transferal of displacements and aerodynamic forces between the structural and aerodynamic girds in ZEUS is accomplished by a 3D spline module. The transferred displacement at the aerodynamic grids is used to define the unsteady boundary condition that, again, is applied on the mean plane of the lifting surface and the rectangular box of the bodies using a transpiration boundary condition technique. Therefore, there is no moving mesh involved in the ZEUS unsteady aerodynamic computation which further reduces the computational resource.

ZEUS does not provide the structural finite element solutions. It imports externally computed structural free vibration solutions (or, the normal modes solutions) generated by other structural finite element codes. The Modal Data Importer module of ZEUS is developed to directly process the output files of five commercial finite element programs: MSC.NASTRAN, NE/NASTRAN, NX/NASTRAN, ASTROS, and I-DEAS. For other finite element codes, a "free format" for the modal data input is available in the Modal Data Importer module. The Modal Data Importer processes the finite element output file to obtain the structural grid point locations for spline, the coordinate transformations for relating the local/global to the basic coordinate system, the modes, the natural frequencies, the generalized mass matrix and the generalized stiffness matrix of the structural finite element model. As an option, the user can also directly input the generalized mass and stiffness matrices as well as the mode shapes into ZEUS. Thus, there is virtually no burden to the user for importing the structural finite element solutions to ZEUS. In the following subsections, the main features of ZEUS listed above are discussed and presented.

1.2 AUTOMATED MESH GENERATION

The purpose of the automated mesh generation capability in ZEUS is to automate the manual volume mesh generation effort. This is accomplished by a ZAERO-to-ZEUS model converter incorporated in ZEUS that adopts the surface mesh of ZAERO as input and automatically generates a single block of volume mesh into which a set of the components of the aircraft such as fuselage, strake, wing, and horizontal tail can be fitted. Other components such as the underwing stores can be modeled by additional blocks of mesh. Communication between different blocks of mesh is accomplished by the overset mesh scheme. The volume mesh consists of orthogonal grid lines on the Y-Z and X-Y planes; i.e., the vertical and horizontal grid lines are perpendicular to each other. The gridlines projected on the X-Y plane (called the chordwise Y-lines) and the Y-Z plane (called the horizontal Y-line) can be further sheared to accommodate the sweep angle of the leading and trailing edges and the dihedral angle, respectively, of the lifting surfaces. This sheared mesh on the X-Y plane is automatically generated by a Y-Zone technique. The Y-Zone technique projects all components such as the fuselage, strake, wing, tip launcher and horizontal tail such as those shown in Figure 1.1 on the X-Y plane. On this X-Y plane, all components are divided into several spanwise zones, called the Y-Zones. Within each Y-Zone, numerous fictitious chordwise Y-lines are first generated whose slopes start from that of the leading and trailing edges of those components in the same Y-Zone and gradually decrease to zero as they move towards downstream and upstream. Then a line tracing method is activated where each chordwise Y-line of the surface mesh traces the fictitious chordwise Y-lines across all Y-Zones thereby, connecting all the chordwise Y-lines. The final step of the Y-Zone technique is to check duplicate spanwise lines using a small tolerance, within this tolerance only one chordwise Y-line is kept.

- ZONE 1: Left Launcher
- ZONE 2: Output left wing
- ZONE 3: Inboard left wing and outboard left horizontal tail
- ZONA 4: Outboard left strake and inboard left horizontal tail
- ZONE 5: Inboard left stake
- ZONE 6: Fuselage
- ZONE 7: Inboard right strake
- ZONE 8: Outboard right strake and inboard right horizontal tail
- ZONE 9: Inboard right wing and outboard right horizontal tail
- ZONE 10: Outboard right wing
- ZONE 11: Right launcher

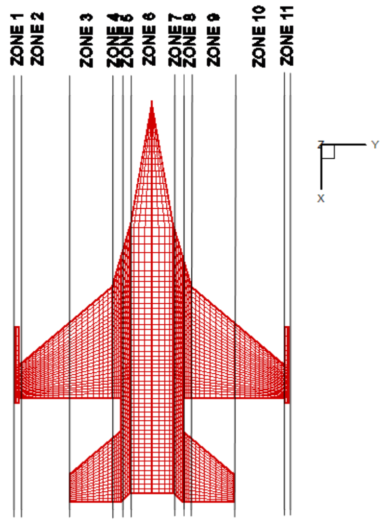


Figure 1.1 Y-Zone Technique for a Fuselage-Strake-Wing-Tip Launcher-Horizontal Tail Configuration

To extend the chordwise Y-lines to the far field, a cubic spline technique with two slope constraints is used. The lines leaving the right-most or the left-most point of the surface mesh should have consistent slope with that of the outer components. Also, they must be perpendicular to the right and the left far field boundaries when they reach them. The resulting automatic generated mesh for a fuselage-strake-wing-tip launcher-horizontal tail configuration is shown in Figure 1.2. It should be noted that the boundary condition of the fuselage is applied on a box-like prismatic surface that embodies the actual surface of the fuselage. ZEUS employs a correction parameter using a slender body theory to account for the spatial difference between the prismatic and the actual surfaces. It is this correction parameter that enables the body-like components such as fuselage and stores to be modeled by a Cartesian mesh.

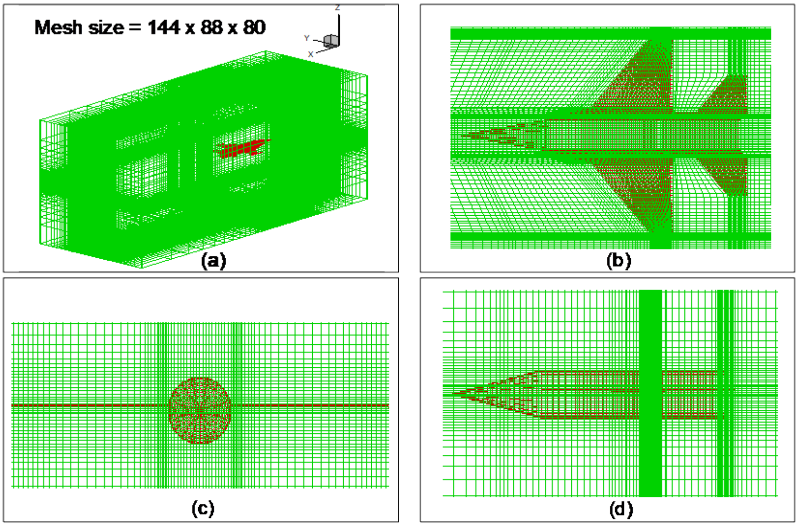
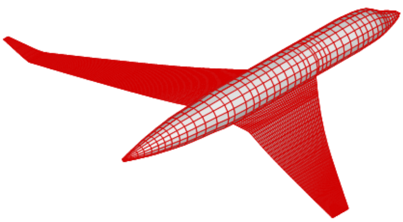
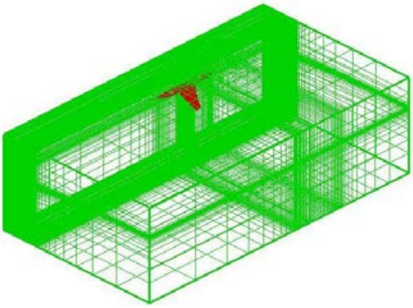


Figure 1.2 Automatic Generated Mesh for a Fuselage-Strake-Wing-Tip Launcher-Horizontal Tail Configuration

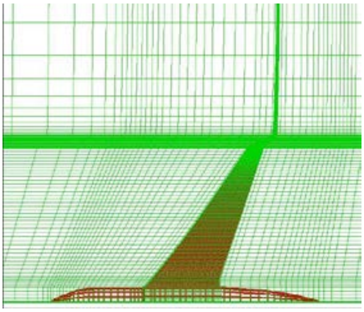
The Y-zone technique can be also applied to the shear grid lines on the Y-Z plane (horizontal Y-lines) to accommodate the dihedral angle of the lifting surfaces. Shown in Figure 1.3(a) is the ZAERO panel model of a wing-body configuration with a wing-let. Because of the wing-let, the grid lines on the Y-Z plane must be sheared so that the grid lines above and below the wing-let are nearly parallel to the wing-let mean surface. Applying the Y-zone techniques to the grid lines on the X-Y plane and the Y-Z plane, the resulting sheared grid lines are shown in Figure 1.3(c) and 1.3(d), respectively.



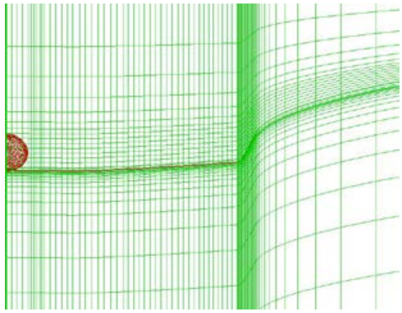
(a) ZAERO Panel Model



(b) 3-D View of the Mesh



(c) Sheared Gridlines on X-Y Plane



(d) Sheared Gridlines on Y-Z Plane

Figure 1.3 Sheared Mesh of a Wing-Body Configuration with Wing-let

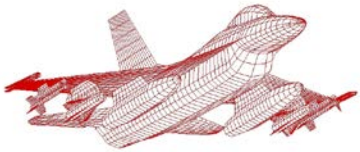
The automated mesh generation capability greatly reduces the mesh generation effort and allows a rapid change of the grid size for an optimum mesh density. In fact, the majority of the ZEUS input is identical to that of ZAERO so that only the surface mesh defined by the ZAERO panel model is required for input; rendering ZEUS as a user-friendly computational aeroelastic tool.

1.3 OVERSET MESH SCHEME

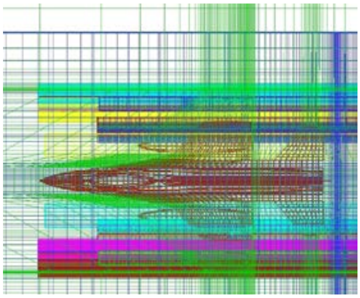
For a complex configuration, sometimes it is very difficult to fit all components of the configuration into a single block of mesh. In this situation, the overset mesh scheme can be employed to fit the components into multiple blocks of mesh. For instance, for a wing with underwing store configuration, the wing can be fitted into a block of mesh, whereas the store can be fitted into another block of mesh. The communication of the flow solution among the blocks can be achieved by interpolating the flow solution in the overlapped-mesh region.

The overset mesh scheme, also known as the Chimera scheme is a matured technology widely used by the CFD community to model complex configurations using structured grids. Because the ZEUS mesh is structured, the Chimera scheme can be applied directly. The overset mesh scheme incorporated in ZEUS consists of three steps: (1) Hole cutting, (2) Donor cell search, and (3) Finding the interpolation factors. Hole cutting is also referred to as blanking that nullifies the contribution from those non-physical points in the overlapping region. The Donor cell search finds the cells for the Chimera boundary points in the neighboring body meshes. The flow solutions are interpolated from those donor cells onto the Chimera boundary points. This interpolation is achieved using interpolation factors based on the topology of the meshes in the overlapping regions. Solution convergence within each time step is achieved by sub-iteration.

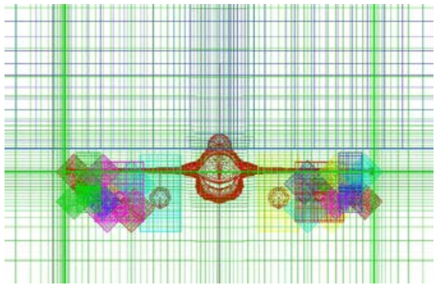
Figure 1.4 presents an overset mesh for the modeling of a fighter with underwing stores. The surface mesh of this fighter with the underwing stores is shown in Figure 1.4(a) which is described by the ZAERO aerodynamic panel model. There are 24 blocks of mesh employed to model this configuration where block 1 models the fuselage, wings, tip launchers, and horizontal tails, block 2 models the vertical tail and blocks 3 to 24 model the underwing store bodies and fins. The resulting overset mesh projected on the X-Y plane and the Y-Z plane are shown in Figure 1.4(b) and 1.4(c).



(a) ZAERO Surface Panel Model



(b) X-Y Plane



(c) Y-Z Plane

Figure 1.4 Overset Mesh of a Fighter with Underwing Stores

1.4 THE EULER-SOLVER MODULE

The Euler-Solver module is essentially an unsteady aerodynamic force generator that solves the Euler equations with/without boundary-layer coupling. The Euler-solver module occupies the majority of the computational time of a ZEUS analysis. Apparently, the computational time of the Euler-solver module depends on the number of grid points in the mesh. To accelerate the computational time, ZEUS allows the parallel computation using open MP on a multiple-processor computer that is activated by an executive control command called CPU. Several types of unsteady aerodynamic forces can be provided by the Euler-Solver module:

a. Time-domain unsteady aerodynamic computation
The Euler-Solver module is coupled with the state-space equation of the structural equations. At each time step, the state-space solution is first solved then the solution of the generalized coordinates is applied to the boundary condition of the Euler equations to compute the unsteady aerodynamic forces for the next time step. Thus, this is a tightly coupled aeroelastic simulation for transient aeroelastic responses. The time-domain unsteady aerodynamic computation is used by the maneuver loads and store ejection loads modules to generate the time-domain flight loads.

b. Frequency-domain unsteady aerodynamic computation
Three options are available to convert the time-domain solution to the frequency domain: the sinusoidal excitation technique, the composite sinusoidal excitation technique, and the linearized Euler solver. The sinusoidal excitation technique loops the Euler-Solver module with a sinusoidal oscillation at each reduced frequency then performs a Fourier Transform to obtain the frequency domain unsteady aerodynamic forces. The composite sinusoidal

excitation technique applies all reduced frequencies simultaneously to the boundary condition of the Euler-Solver module then performs a Fourier Transform for the frequency-domain unsteady aerodynamic forces. The linearized Euler solver solves the frequency-domain linearized Euler equation to directly generate the frequency-domain unsteady pressures and generalized aerodynamic forces. These unsteady aerodynamic forces are then plugged into the frequency domain flutter equation. The flutter solution is obtained using the g-method or the *k*-method. The frequency-domain unsteady aerodynamic forces can be considered as a linearized aerodynamic solution with respect to the structural oscillating amplitude. Thus, they are only capable of providing linear flutter boundary.

c. Pseudo time-domain aerodynamic computation
The pseudo time-domain aerodynamic forces are computed using the time-domain computation but with pseudo time step. All time-dependent terms in the Euler equation and the boundary condition are inactive. The pseudo time-domain computation is used for the static aeroelastic analysis where the steady aerodynamic forces on the flexible structure are obtained when the pseudo time-domain computation converges. The pseudo time-domain aerodynamic computation is also performed prior to a transient response analysis where the static aeroelastic solution is used as an initial condition for the time-domain unsteady aerodynamic computation. The pseudo time-domain aerodynamic computation is also used by the trim module as the underlying aerodynamic force generator to solve for the trim solution and trim loads.

d. Steady aerodynamic computation on rigid aircraft
The steady aerodynamic forces on the rigid aircrafts are computed also using the pseudo time-domain computation but without the structural flexibility. This steady aerodynamic computation is performed prior the frequency-domain unsteady aerodynamic computation. This is to say that the converged steady aerodynamic solution is used as the initial solution for generating the frequency-domain unsteady aerodynamic forces. In addition, the aerodynamic stability derivatives can be obtained from the frequency-domain generalized aerodynamic forces computed by the linearized Euler solver of six rigid-body modes.

e. Discrete gust aerodynamic computation
The aerodynamic forces due to the discrete gust are computed by introducing a time-domain traveling gust profile in the boundary condition of the Euler-Solver module. Similar to the time-domain unsteady aerodynamic computation, these aerodynamic forces are coupled with the state-space structural equations to compute the structural response due to a discrete gust excitation.

1.5 HIGH-FIDELITY GEOMETRY (HFG) MODULE

The HFG module is capable of modeling complex configurations such as the full aircraft with external stores. The surface mesh of a complex configuration can be represented by the wing-like and body-like components. Internally, the HFG module automatically generates a volume mesh by growing the mesh from the surface mesh to the outer boundary of the computational domain. In addition, multiple blocks of mesh can be used to model different components of a complex configuration. Those multiple block meshes are allowed to intercept with each other. An overset mesh scheme is employed to establish the interpolation of flow solutions in the overlapping region of the blocks. Solution convergence is achieved by sub-iterations.

A graphic output file can be generated by the HFG module to display the surface mesh as well as the volume mesh.

1.6 3D SPLINE MODULE

The 3D Spline module establishes the displacement/force transferal between the structural Finite Element Method (FEM) model and the ZEUS aerodynamic model. It consists of four spline methods that jointly assemble a spline matrix. These four spline methods include: (a) Thin Plate Spline; (b) Infinite Plate Spline; (c) Beam Spline, and (d) Rigid Body Attachment methods. The spline matrix provides the x, y and z displacements and slopes at all aerodynamic grids.

Graphic output file can be generated by the 3D spline module to store the deformed surface mesh of each structural mode. This is a very useful feature that allows the users to verify the spline input.

1.7 ZONA DYNAMIC MEMORY AND DATABASE MANAGEMENT SYSTEM

The ZONA Dynamic Memory and Database Management (ZDM) System serves as the fundamental software building block for ZEUS. It consists of the following five parts:

- *Matrix Entity Manager*

The matrix entity manager is designed to store and retrieve very large, often sparse, matrices. It minimizes the disk storage requirements while allowing algorithms developed by ZONA Technology, Inc. to perform matrix operations of virtually unlimited size.

- *Relational Entity Manager*

Relational entities are essentially tables. Each relation has data stored in rows (called entries) and columns (called attributes). Each attribute is given a descriptive name, a data type, and constraints on the values that the attributes may assume (i.e., integer, real or character data). These definitions are referred to as the schema of the relation.

- *Unstructured Entity Manager*

There are many times that a software module requires temporary, or scratch, disk space while performing tasks. The data generated within these tasks are generally "highly-local" and, due to the modular nature of the software, are not to be passed through arguments to other modules within the system. To effectively accommodate the transfer of this type of data, ZDM supports an unstructured database entity type composed of "records" that may contain any arbitrary collection of data.

- *Dynamic Memory Manager*

The Dynamic Memory Manager consists of a suite of utility routines to allocate and release blocks of dynamic memory. The Dynamic Memory Manager provides the capability of developing an engineering software system that allows operations to be performed on the data that would normally exceed the size of available memory.

- *Engineering Utility Modules*

Engineering utility modules contain a pool of routines that perform operations on the matrix database entities. These operations include matrix decomposition, eigenvalue solver, matrix multiplication, matrix partitioning/merging, etc. These routines first check the property of a given matrix and then select the appropriate numerical technique to perform a particular matrix operation.

1.8 THE FLUTTER MODULE

The flutter module first uses the Euler-Solver module to obtain the frequency-domain generalized aerodynamic forces by invoking the frequency-domain unsteady aerodynamic computation of the Euler-solver module. Then it constructs a frequency-domain flutter equation that can be solved either using the g-method or the *k*-method. The flutter boundary is calculated by varying the flight speed and air density for the non-matched point flutter solution or by varying the altitudes with a build-in atmospheric table for the matched point flutter solution.

Flutter mode animation can be done by exporting a graphic file from the flutter module. In addition, the unsteady pressure distribution due to a particular structural mode at a given reduced frequency can be also visualized using a graphic output file.

1.9 THE MLOADS MODULE

The MLOADS module performs a transient maneuver loads analysis due to the pilot input command. The input of the MLOADS module is a time history of the control surface deflections due to a pilot input command or the initial perturbation of the generalized coordinates. The time domain response of the structural generalized coordinates is computed by solving a state-space equation that involves the generalized mass, stiffness, and aerodynamic forces. The generalized aerodynamic forces are computed at each time step by invoking the time-domain unsteady aerodynamic computation of the Euler-Solver module.

As an option, the user can transform the structural rigid body modes to the airframe states so that the rigid body submatrix of the state space equation is in the same definition as the flight dynamics. For the output, the MLOADS module computes the transient response at the structural grid points as well as the sectional loads. It also outputs the forces at each structural degree of freedom in terms of NASTRAN **FORCE** and **MOMENT** bulk data cards for subsequent detailed stress analysis.

The MLOADS module also generates a graphic file that allows the visualization of the time-domain pressure variation on an oscillating aerodynamic surface mesh.

1.10 THE ELOADS MODULE

The ELOADS module performs the transient ejection loads analysis due to store ejections. It allows multiple store ejections in a sequential schedule while the aircraft is maneuvering due to a pilot input command. The computational procedure of the ELOADS module is very similar to that of the MLOADS module except:

a. The input is a set of time-domain forcing functions at structural grid points as the external excitation due to store ejection forces.

b. It accounts for the effects of the sudden reduction in the aircraft weight due to the separation of the stores from aircraft.

All output options of the MLOADS module are also available for the ELOADS module.

1.11 THE GLOADS MODULE

The GLOADS module performs a transient aeroelastic analysis when the aircraft structures encounter a discrete gust. The computational procedure of the GLOADS module is very similar to that of the MLOADS module except the time-domain aerodynamic forces are replaced by the discrete gust aerodynamic computation of the Euler-Solver module. The input of the GLOADS module is the discrete gust profile; such as one-minus-cosine, sine, sharp edge, and arbitrary profiles. All the output options of the MLOADS module are also available for the GLOADS module.

1.12 THE NLFLTR MODULE

The NLFLTR module performs a transient aeroelastic analysis including the structural nonlinearities such as free-play, nonlinear stiffness, and nonlinear damping. These nonlinearities can be specified as a function of multiple user-defined nonlinear parameters such as displacement, velocities, accelerations, element forces, and modal values. Based on this function, the NLFLTR module divides the nonlinear domain into many sub-linear domains and constructs the state-space equation of each sub-linear domain. At each time step, the solution of the nonlinear parameters is first computed to identify the corresponding sub-linear domain whose state-space equation is then used for the next time step. The aerodynamic forces are computed by invoking the time-domain unsteady aerodynamic computation of the Euler-Solver module.

All output options of the MLOADS module are also available for the NLFLTR module.

1.13 THE TRIM MODULE

The TRIM module performs a static aeroelastic/trim analysis for solving the trim system and computing the flight loads. It employs the modal approach to compute the aerodynamic stability derivatives of each trim variable by invoking the pseudo time-domain aerodynamic computation of the Euler-Solver module. The TRIM module is capable of dealing with the determined trim system as well as the over-determined trim system (more unknown than the trim equations). The solution of the over-determined trim system is obtained by using an optimization technique which minimizes a user-defined objective function while satisfying a set of constraint functions.

The TRIM module generates the flight loads in terms of the NASTRAN **FORCE** and **MOMENT** bulk data cards at each structural degree of freedom for subsequent detailed stress analysis. It also outputs a graphic file that allows the visualization of the deformed aerodynamic model with pressure distribution at the trim conditions.

1.14 THE GENGAF MODULE

The GENGAF generates the frequency-domain generalized aerodynamic forces using a sinusoidal excitation technique or a composite sinusoidal excitation technique. This Generalized Aerodynamic Forces (GAF) includes the GAF due to structural modes, control surface modes, and sinusoidal gust. The user can import these GAF into ZAERO and use ZAERO to perform aeroelastic analysis.

1.15 THE STABDRV MODULE

The STABDRV generates the aerodynamic stability derivatives including damping derivatives. The aerodynamic stability derivatives are computed from the frequency-domain generalized aerodynamic forces of six rigid-body modes that are generated by the linearized Euler solver at small reduced frequency.

Chapter 2

HOW TO RUN ZEUS

The ZEUS software system is available for both the workstation (UNIX operating system) and the personal computer (Windows/DOS and Linux) platforms. The execution of ZEUS (after proper installation of the code [See Installation Notes for instructions]) is described as follows:

UNIX / Linux

In the directory where the input file and the structural Finite Element Method (FEM) output file (the free vibration solutions of the FEM model) reside, type the following command:

ZEUS <INPUTFILENAME> <OUTPUTFILENAME>

where <outputfilename> is optional. An example is shown as follows:

ZEUS MYJOB.INP MYJOB.OUT

All output files will be placed in the same directory where the job was submitted after the program terminates. See Section 2.6 *The ZEUS Script File* for a detailed description of this process that takes place during code execution.

Windows/DOS

1. Open a MS-DOS command prompt window (under Start / Programs / MS-DOS Prompt).
2. In the directory where the input file and the FEM output file reside, type the following command:

ZEUS <INPUTFILENAME> <OUTPUTFILENAME>

where <outputfilename> is optional. An example is shown as follows:

ZEUS MYJOB.XXX MYJOB.OUT

All output files will be placed in the same directory where the job was submitted after the program terminates. See Section 2.6 *The ZEUS Script File* for a detailed description of this process that takes place during code execution.

2.1 ZEUS SOFTWARE SYSTEM

Figure 2.1 shows the ZEUS software system file processing that occurs during program execution.

Four files are required to run the code, namely; the input file which contains the executive control, case control and bulk data sections that describe the aerodynamic model, flight conditions, etc.; the structural Finite Element Method (FEM) output file containing the structure natural frequencies and mode shapes; DIRNAME.FIX which contains the pathname where the ZEUS run-time database files are to be located. The ZONA License Server (ZLS) (see Section 2.6 for detail) manages the license key provided by ZONA. This key specifies the number of tokens licensed from ZONA. During each ZEUS execution, the ZLS is contacted to provide a token which is then "checked out" from the server. Once the ZEUS job completes, the submitted token is then "checked back in" to the ZLS. Therefore, the total number of ZEUS jobs that can be submitted simultaneously is limited to the number of tokens purchased from ZONA.

A minimum of two output files are generated for each ZEUS run. These are the output file of the job and the logfile which contains the elapsed and step CPU times for each module call during the execution of ZEUS. Additional output plot files can be generated through the bulk data input requests (see Section 2.4 *ZEUS Output Files*).

Additional details relating to these files and details on execution of the ZEUS software system are described in the following sections.

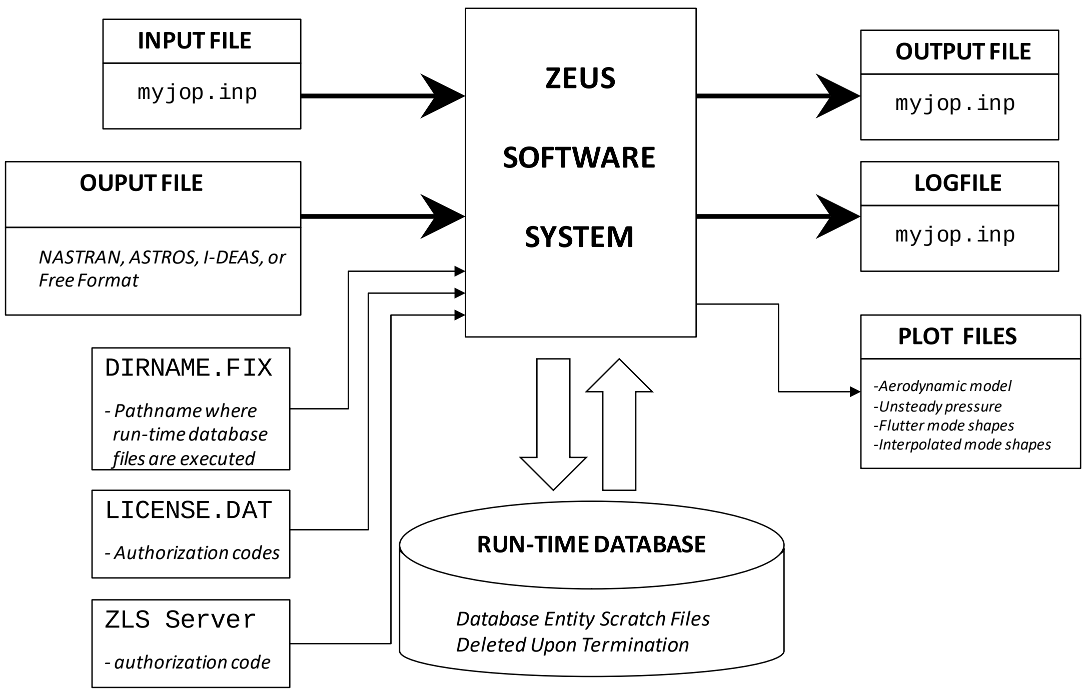


Figure 2.1 The ZEUS Software System File Processing

2.2 INPUT FILES

The ZEUS input file is made up of three sections that describe the aeroelastic problem to be analyzed. These are the following:

1. Executive Control Section
2. Case Control Section
3. Bulk Data Section

Figure 2.2 shows the ZEUS input data structure format.

Leading comments (initiated with a $) are allowed

Executive Control Section

```
ASSIGN  FEM = <filename>, FORM = <form>, BOUNDARY = <type>, PRINT = <print>
DIAG <values>
CEND
```

Case Control Section

```
TITLE = <title>
ECHO = <sort/nosort>
SUBCASE = <number>
   SUBTITLE = <subtitle>
   LABEL = <label>
   FLUTTER = <number>
      .
      .
      .
BEGIN BULK
```

Bulk Data Section

```
$       ACSID   XZSYM   FLIP    FMMUNIT FMLUNIT REFC    REFB    REFS    $
AEROZ   0       YES     NO      SLIN    IN      100.0   200.    10500.
+AERO
$       REFX    REFY    REFZ
+AERO   33.333  0.      0.
      .
      .
      .
ENDDATA
```

Figure 2.2 ZEUS Input Data Structure Format

Executive Control Section

The executive control section must be the first section of any ZEUS input deck. The `ASSIGN` and `CEND` are required delimiters. The keyword `ASSIGN` triggers the input file processing performed by the software. This section contains information such as the filename of the structural Finite Element Method (FEM) output to be read in, the type of analysis to be performed (i.e., symmetric, anti-symmetric boundary condition, etc.), and print options. Finally, diagnostic routines, useful in programming in the ZEUS environment, are specified in this section *(See Section 3 for details of the Executive Control Section).*

Case Control Section

The case control section, which must be in the second section of any ZEUS input deck, is used to define the disciplines to be performed. Each case is defined by a subcase that lists the flutter disciplines to be performed for that particular subcase. A title for the entire input deck and subtitles/labels for each subcase are defined in this section. The `BEGIN BULK` statement designates the end of the case control section. *(See Section 3 for details of the Case Control Section).*

Bulk Data Section

The last section of any ZEUS input deck is the bulk data section. The `BEGIN BULK` and `ENDDATA` are required delimiters. This section provides the complete engineering data required to perform the disciplines specified in the case control section. This includes the geometry of the aerodynamic model, spline instructions for displacement and force transferal between the structural finite element grid points and the aerodynamic boxes, flight conditions, and other parameters such as reference density, lengths, etc. *(See Section 4 for details of the Bulk Data Section).*

2.3 RUN-TIME DATABASE

A ZEUS run-time database is generated for each job that is submitted under the ZEUS script file. The database contains relational, unstructured and matrix entities (stored in separate scratch files) that are created by ZEUS during execution of the software. The location of the run-time database is dependent on the pathname specified in the 'DIRNAME.FIX' file that is stored in the ZEUS home directory. Temporary database folders under this pathname are created for each job and are removed upon normal termination of the ZEUS script file. The 'DIRNAME.FIX' file is setup during initial installation of the ZEUS software and can be modified by the user to change the location where the ZEUS database folders are executed.

Note: The location specified by 'DIRNAME.FIX' should be a very large scratch space with sufficient size to accommodate all jobs submitted under the ZEUS script file. There is no rule of thumb for how large this space should be since the capability of ZEUS, in terms of the size of the input model, is only limited by the memory and disk space of the hardware.

2.4 OUTPUT FILES

Output File

A minimum of two output files are generated for a given ZEUS job. The first output file contains the standard output from ZEUS program. The name of the output file will either be the name provided to the ZEUS script file or will be the input filename with an extension of '.out'. For example,

```
ZEUS  testcase.inp
```

would generate an output filename of `testcase.out`, while

```
ZEUS  testcase.inp  job1.txt
```

would generate an output filename of `job1.txt`.

The output file contains information, such as sorted bulk data input, interpolated modes on aerodynamic boxes, steady and unsteady pressure results, flutter results, etc.

Logfile

The second output file is a logfile that contains the run-times of the ZEUS engineering module calls. A sample of this output is shown in Figure 2.3. The logfile provides the elapsed time, Central Processing Unit (CPU) time and step CPU time for all module calls made during execution of ZEUS. The logfile name will always be the input filename with an extension of '`.log`.' For example,

```
ZEUS  testcase.inp  job1.txt
```

would generate a logfile filename of `testcase.log`.

The logfile information is very useful in instances where the program terminates due to input errors. Although error messages are generated and printed in the output file, the specific module in which the program terminated can be ascertained. It is also useful to see the relative CPU costs of each phase of execution.

.

The output format for times are [hours : minutes : seconds . hundredths of a second] .

```
                          * * *  Z E U S   L O G F I L E  * * *

 ELASPED     TOTAL                                                         STEP
  TIME        CPU                                                           CPU
---------  -----------                                                 -----------

000:00:00  000:00:00.0     *** BEGIN   ZEUS  ***
000:00:00  000:00:00.0     INIT MODULE: INITIALIZATION                 000:00:00.0
000:00:00  000:00:00.0     CNTL MODULE: PROCESS CASE CONTROL           000:00:00.0
000:00:00  000:00:00.3     IFP  MODULE: INPUT FILE PROCESSOR           000:00:00.3
000:00:00  000:00:00.4     FEM  MODULE: IMPORT FEM MODAL DATA          000:00:00.0
000:00:07  000:00:07.0     HFG  MODULE: HIGH FIDELITY GEOMETRY MODULE  000:00:06.6
000:00:08  000:00:08.5     SPLINE MODULE: SPLINE MATRIX GENERATION     000:00:01.5
000:00:08  000:00:08.6     CONMOD MODULE: CONTROL MODES FOR DYNAMIC LOADS  000:00:00.0
```

```
 000:00:08  000:00:08.8     GENDYN MODULE: STRUCTURAL DYNAMIC MATRICES          000:00:00.1
 000:00:08  000:00:08.8     CAPMODE MODULE: MAPPING OF FEM MODES TO ZEUS MESH   000:00:00.0
 000:00:08  000:00:08.8     CAP2ZRO MODULE: ZEUS MESH TO ZAERO MESH FOR CP      000:00:00.0
 000:00:08  000:00:08.9         SUBCASE NO.        1                            000:00:00.0
 000:00:10  000:00:10.6           FLUTTER MODULE: FLUTTER DISCIPLINE            000:00:01.6

 000:00:10  000:00:10.6     *** END   ZEUS  ***                                 000:00:00.0
```

Figure 2.3   ZEUS Logfile Containing the Execution Summary

Note that both the logfile and output file are overwritten upon resubmission of a ZEUS job with the same input filename which are located within the same directory where the logfile and output already reside. Therefore, the user is cautioned to rename these files in the event they should be permanently saved.

One exception to the output file being overwritten is if an output filename is specified when submitting a ZEUS job that already exists. For example, if the file `testcase.out` exists in the current directory and a ZEUS job is requested as follows:

```
ZEUS  testcase.inp  testcase.out
```

then the script file will prompt the user if the output file should be overwritten.

Plot Files

ZEUS provides a number of output plot files that can be viewed by several plotting programs. Filenames for all output plot files are specified via the bulk data entries **PLTAERO**, **PLTCP**, **PLTFLUT**, **PLTMODE**, **PLTVG,** and **PLTTIME**. Table 2.1 lists the output plot file capability of ZEUS.

Table 2.1   ZEUS Output Plot File Capability

Category	**Associated Bulk Data Card**	**Description**	**Software Compatibility**
Aerodynamic Model	**PLTAERO**	Generates an ASCII text file for plotting the aerodynamic model	- PATRAN - TECPLOT - I-DEAS - FEMAP - ANSYS - NASTRAN
Unsteady Pressures	**PLTCP**	Generates an ASCII text file for plotting the unsteady pressure coefficients	- PATRAN - TECPLOT - I-DEAS - FEMAP - ANSYS - NASTRAN

			- PEGASUS
Flutter Mode Shapes	**PLTFLUT**	Generates an ASCII text file for plotting the flutter	- PATRAN - TECPLOT - I-DEAS - FEMAP - ANSYS - NASTRAN
Interpolated Structural Modes	**PLTMODE**	Generates an ASCII text file for plotting the interpolated structural mode on the aerodynamic model	- PATRAN - TECPLOT - I-DEAS - FEMAP - ANSYS - NASTRAN
Slopes of the Aerodynamic Surface	**PLTSLP**	Generates an ASCII text file for plotting the slopes of the aerodynamic surface mesh	TECPLOT
Control Surface Deflection	**PLTSURF**	ASCII Text File Generation for Plotting the Aerodynamic Control Surface	- PATRAN - TECPLOT - I-DEAS - FEMAP - ANSYS - NASTRAN
Transient Response Analysis	**PLTTIME**	Generates an ASCII text file for plotting the transient response	- PATRAN - TECPLOT - I-DEAS - FEMAP - ANSYS - NASTRAN
Static Aeroelastic / Trim Analysis Results	**PLTTRIM**	Generates an ASCII text file for the post-processing of the static aeroelastic/trim analysis	- PATRAN - TECPLOT - I-DEAS - FEMAP - ANSYS - NASTRAN - PEGASUS
Flutter Damping and Frequency Results	**PLTVG**	Generates an X-Y plot file (ASCII text) for plotting the flutter frequency and damping curves	- I-DEAS - FEMAP - TABLE - PEGASUS *and most other spreadsheet applications*

2.5 ZEUS SCRIPT FILE

The ZEUS script file is used to submit jobs to be run by the ZEUS software system and is located in the ZEUS home directory. Multiple jobs can be submitted at one time on both UNIX and PC systems. This script file can be executed from any directory on the host system with the appropriate environment variables set. The environment variables are normally set-up automatically during installation of the ZEUS software system, but can be set-up manually (see the ZEUS Installation Notes for details on how to adjust environment variables).

Two versions of the ZEUS script file are available. The first, developed for the UNIX environment, is written in the C shell scripting language. The second, developed for the PC environment, is written in FORTRAN and is provided in executable format.

The following two sections provide instructions on submitting multiple ZEUS jobs and step-by-step descriptions of the steps taken by UNIX / Linux and PC versions of the ZEUS script file.

UNIX / Linux

Multiple jobs can be submitted in the UNIX environment (submitting multiple jobs is optional, not a requirement). Simply initiate the ZEUS script file multiple times in succession. For example, to submit two jobs called `test1.inp` and `test2.inp` type the following at the command prompt:

`ZEUS test1.inp` *and press the return key*

followed by

`ZEUS test2.inp` *and press the return key*

Two jobs will be submitted each with a unique process id (type `ps –a` to see a listing of all running jobs on the system).

Multiple jobs can be submitted from either the same directory or different directories. Associated output files will be placed in the directories from which the input jobs were submitted. Any AIC files to be read in for a restart run process must also be located in the directory from which the input job is submitted. At the end of each batch job process, the script file will notify the user of job termination by a *beep* sound.

As a final note, the input/output decks are in ASCII text format and can be viewed and/or modified with any editor on the host system (such as the ‘vi’ editor).

- *UNIX / Linux Script File Process*

1. Acquire the input filename from standard input.
2. Check if input file exists locally.
3. Acquire the output filename, if specified in the command line.
4. Establish an output filename (if not found in step #3) and a logfile filename.

5. Check the run-time database directory path specified in file ‘DIRNAME.FIX’ located in the ZEUS home directory.
6. Establish a run-time database folder (i.e., directory) using the current process id as an extension. For example: ZEUS0001.
7. Copy the complete pathname (pathname specified in DIRNAME.FIX along with the current run-time database folder name) to file ‘DIRNAME.TMP’. This temporary file is read by the ZEUS software system to know where the database files of the current job are to be executed.
8. Execute the ZEUS software system using the input/output filenames.
9. Copy the logfile from the run-time database folder to the local directory.
10. Delete the database folder and scratch files.
11. Notify user of job termination by a *beep* sound.

Windows/DOS

Multiple jobs can be submitted in the PC environment (submitting multiple jobs is optional, not a requirement). The exception to this case would be if the host system is operating under MS-DOS and does not utilize the Windows operating system. In this situation, only one job can be submitted at a time which will tie up the machine until the job terminates (unless the user can utilize multiple command interpreters with the option to toggle between them). For a system with Windows installed, multiple jobs can be submitted by opening up multiple MS-DOS command prompt windows and submitting one job per Window. For example, to submit two jobs in Windows called `test1.inp` and `test2.inp`, perform the following:

1. From the [Start] menu select [Programs/MS-DOS Prompt]. Note: The MS-DOS window can be maximized or minimized. Also Note: Terminating an MS-DOS window during execution of a job will terminate that ZEUS job!
2. Change the directory to where the input deck resides.
3. Type in the following at the command prompt.

`ZEUS  test1.inp` *and press the return key*

4. Open a second MS-DOS window as described in step #1 above.
5. Repeat step #2 from above.
6. Type in the following at the command prompt.

`ZEUS  test2.inp` *and press the return key*

Two jobs will be submitted each with a unique folder designation (e.g., ZEUS001) and will be located in the run-time database directory specified by the pathname in file ‘DIRNAME.FIX’.

Multiple jobs can be submitted from either the same directory or different directories. Associated output files will be placed in the directories from which the input jobs were submitted. Any AIC files to be read in for a restart run process must also be located in the directory from which the input job is submitted. At the end of each batch job process, the script file will notify the user of job termination by a *beep* sound.

As a final note, the input/output decks are in ASCII text format and can be viewed and/or modified with any editor on the host system (such as the DOS editor – initiated in an MS-DOS Window by 'edit').

- *PC Script File Process*

This is identical to the *UNIX Script File Process* described earlier, except step #6, as follows:

7. Establish a run-time database folder (i.e., directory) using the first available (i.e., lowest number) folder to obtain a new folder extension. For example, if two jobs were already submitted that occupy folders ZEUS001 and ZEUS004 and a third job is to be submitted, then a folder name of ZEUS002 would be used. Note that a maximum of 999 jobs can be submitted at one time on the PC system.

Command Line Options

To view the available script file command line options, please use the `-help` switch, e.g.

```
zeus -help
```

Run-time Database Directory

The ZEUS software system run-time database directory location is specified in the file 'DIRNAME.FIX' which is set-up upon installation of the software. Folders (i.e., directories) are set-up under this location for each job submitted via the ZEUS script file (as described earlier in this section).

Upon normal termination of a job, the run-time database folder is deleted, except under the following conditions:

- If the computer is shut down or if a power failure occurs during execution of a job.
- If a ZEUS script file job is terminated by the user (e.g., by closing the MS-DOS prompt window) or is terminated by some other means (e.g., by the Windows operating system).

In such situations, the run-time database folders are left in the run-time database directory and can occupy tremendous amounts of disk space. Therefore, the user should manually remove any run-time database folders of jobs that are no longer running.

2.6 THE ZONA LICENSE SERVER (ZLS)

The ZONA License Server (ZLS) has been developed by ZONA Technology, Inc. (ZONA) to act as the security license server for ZONA's software products. The ZLS operates with the Sentinel Protection Installer™ SuperPro hardware key that is developed by SafeNet (http://www.safenet-inc.com). The ZLS is described in detail in the ZLS Users Manual that is installed with the ZLS software.
ZEUS is a "network ready" version of ZEUS that requires the ZLS to be installed. During each ZEUS execution, a token is "checked out" from the server and "checked back in" to the server when the job terminates.

There are two types of ZEUS installations that can be made.

1. Node-Locked:
   The ZLS is installed on the same machine where ZEUS is installed. If ZEUS runs on a stand-alone machine, both ZEUS and ZLS must be installed as node-locked.

2. Floating License:
   ZEUS and the ZLS are installed on separate machines connected to a network.

Note that, if desired, tokens managed by the node-locked ZLS can also be checked out by ZEUS jobs executed from any machines that can access the node-locked machine running the ZLS.

2.6.1 THE JAVA ENVIRONMENT

Java JRE 1.3.1 (or later versions) is required to run both the ZLS and ZEUS.

For Windows, UNIX or Linux platforms, download and installation instructions can be found on the Internet. ZONA can provide this download/installation information if requested.

2.6.2 SERVER INSTALLATION AND OPERATIONS

For details regarding the ZLS installation and operation, please refer to Section 3 of the ZLS User's Manual.

2.6.3 ENVIRONMENT VARIABLES

To run ZEUS, the following environment variables are required.

1. [PATH] variable needs to include ZEUS home directory, which is specified at installation.
2. [ZEUSEXE] is set to the ZEUS home directory location. It should end with \ for Windows and end with / for UNIX and Linux.
3. [ZLS_ZEUS] is set to the IP of the machine hosting ZLS. If ZEUS is run on the same machine that hosts the ZLS (i.e., a node-locked setup), the value of ZLS_ ZEUS should be set to `localhost`.
4. [ZLS_SERVER] is set to the ZLS home directory for node-locked installations.

2.6.4 THE ZONA LICENSE MONITOR

The ZONA License Monitor is a Windows program that provides a convenient interface for ZLS operations, including the ability to Start or Stop the ZLS, to load a new license file, and to view the status of the current token usage (i.e., what's checked-out). The ZONA License Monitor is only available on the machine hosting the ZLS. In the case of a node-locked installation of ZEUS, both the ZLS and the ZONA License Monitor will exist on the same machine. For details on usage of the ZONA License Monitor, please refer to Section 6 of ZLS User's Manual.

2.6.5 LOCKED TOKENS AND THE CLEANUP UTILITY

ZEUS is designed to operate in the following way. When a ZEUS job is submitted, the ZLS is contacted for checkout of a token. With a successful checkout (i.e., tokens available for the requested modules in the ZEUS job), a token file is saved under [ZLS\log] directory, the ZLS adjusts the token count, and then the ZEUS job proceeds. After the ZEUS job is finished, the ZLS is contacted for a check-in. With a successful check-in, the token file is deleted, and the ZLS adjusts the token count accordingly.

In the event of an abnormal ZEUS termination (e.g., a power failure during a job) token(s) can become locked. To release locked token(s), a cleanup utility is provided. The utility program `zeus_cleanup.exe` (or `zeus_cleanup` for Unix or Linux) can be found in the ZEUS home directory under the [ZLS\log] directory. To run the cleanup utility, open a command prompt window (UNIX and Linux) or an MS-DOS prompt window (Windows); change the directory to *ZEUS home directory*\ZLS\log and type **`zeus_cleanup`**. When executing `zeus_cleanup`, if a locked-token is found, you will be prompted whether you wish to release the token back to the ZLS.

Token file names are in the format of log-nnn-DD-MMM-YY-hh-mm-ss (e.g., log-001-14-MAY-09-17-22-14). The time stamp in the log file name shows submission time of ZEUS job and nnn indicates its tmp directory. Therefore, token file names can be used to judge if corresponding tokens should be freed while running `zeus_cleanup`.

Instead of using cleanup utility to release locked token(s), re-starting the ZLS will also free up locked token(s). However, doing this will also release the token(s) that might be checked out by other job(s), and **all on-going job(s) will terminate due to ZLS restart**. Therefore, it is strongly recommended to check if there is any job running before re-starting the ZLS by either (1) Clicking on the 'List Current Jobs' button within the ZONA License Monitor Windows program (see Section 6.1 of the ZLS User's Manual), or (2) Executing a '`java zls_serverwhatsrunning`' from a prompt in an MS-DOS or command window (see Section 5.4 of the ZLS User's Manual). Both (1) and (2) will show information related to any on-going job executions.

If ZLS is re-started for any reason, including a reboot of the computer, any remaining token files found in the [ZLS\log] folder under the ZEUS home directory can be deleted before any new ZEUS job(s) are submitted. These old token files are no longer useful since the ZLS record is cleared upon the ZLS re-start.

2.6.6 HEARTBEAT

During execution of ZEUS, heartbeat signals are continuously sent back and forth between ZEUS and the ZLS. Failure in receiving a heartbeat signal by a ZEUS job will result in termination of that ZEUS job. To avoid such a termination, the ZLS needs to be up and running all the times during the execution of ZEUS job(s) and the network connection between the machines running ZEUS and hosting the ZLS must be operational.

2.6.7 ZLS ERROR CODES

The following is a list of the ZLS status and error codes (last one or last three digits) that are reported in the ZEUS output file or are displayed on the screen in the event of an error during submission and execution of a ZEUS job. If the encountered error cannot be resolved, please contact ZONA's technical support staff for assistance. Section 7.1 of ZLS User's Manual documents the error codes in more detail.

ZLS STATUS/ERROR CODES RELATED TO ZEUS:

0 - Success status: the operation succeeded with no warnings.

Related to direct interaction with zls_server:

101 - Exception occurred at opening socket. Don't know about host: provided_zlsIP.
102 - Exception occurred at opening socket. Couldn't get I/O connecting to: provided_zlsIP.
103 - Exception occurred at fillarray.
104 - Exception occurred at readLine. Be aware zls_server might be forced down.

Related to license file:

201 - License has expired.
202 - License product name check failed at reading license.

Related to software product operation:

601 - Needed module was not found in license.
602 - Needed module was not available.
603 - CheckoutID was not found in the record.
604 - Module inconsistency was found in the license.
605 - Token count inconsistency was found in the license.

Chapter 3

EXECUTIVE CONTROL AND CASE CONTROL SECTIONS

The Executive Control Section must be located at the beginning of the input file. Its major functions are:

- To define the filename that contains the free vibration output from the structural finite element methods
- To allow direct matrix input
- To turn on diagnostic routines

The Case Control Section must be located after the Executive Control Section and before the Bulk Data Section. Its major functions are:

- To input title cards that describe the ZEUS analysis.
- To select the disciplines (FLUTTER, TRIM, MLOADS, ELOADS, GLOADS, GENGAF, or NLFLTR) for the analysis.

A typical example of the Executive Control and Case Control Sections is shown as follows:

```
$ Begin Executive Control Section
ASSIGN FEM = demo1.f06,    FORM = MSC,    BOUND = SYM,    PRINT = 1
ASSIGN FEM = demo2.f06,    FORM = MSC,    BOUND = ANTI
ASSIGN MATRIX = demo1.mgh,  MNAME = AMGH
ASSIGN MATRIX = demo1.kgh,  MNAME = AKGH
DOUBLE
DIAG 1, 3
CEND
$ Begin Case Control Section
TITLE = DEMO WING-BODY CASE
ECHO = SORT
SUBCASE = 1
        SUBTITLE = Matched point flutter analysis
        LABEL = at Mach 0.8
$ match point flutter analysis
        FLUTTER = 10
SUBCASE = 2
        SUBTITLE = Trim analysis
        LABEL = at Mach 1.2
$ TRIM analysis
        TRIM = 30
SUBCASE = 3
        SUBTITLE = Transient Maneuver Loads Analysis
        MLOADS = 40
BEGIN BULK
$ Begin Bulk Data Section
```

3.1 Executive Control Section

The Executive Control Section allows the following Executive Control Commands:

'ASSIGN FEM='	Structural modal data importer.	Required
'ASSIGN MATRIX='	Direct matrix input by INPUTT4 format.	Optional
CEND	End of Executive Control Section.	Required
CPU	Defines the number of processors for parallel computation.	Optional
DESSEN	Activates the linking of the dependent geometry parameters.	Optional
DIAG	Diagnostic output options.	Optional
MEMORY	Maximum memory in terms of megabytes that is allocable by ZEUS from the heap space.	Optional
MPICPU	Specify the total number of CPU cores used for MPI	Optional
SOLUTION	Alter the solution sequence.	Optional
SUBAC50	Shift the center of pressure from 25% chord to 50% chord on each panel	Optional
$	Comment statement.	Optional

All Executive Control Commands can be written either in lower case or upper case.

Each command must start from the first column and it must lie within 80 columns. For example:

```
ASSIGN FEM = demo1.f06,    FORM = UAI,    PRINT = -2
```

80 columns

As an added option, one and only one continuation line can be used when entering the '**ASSIGN FEM=**' and **'ASSIGN MATRIX='** Executive Control Commands. The continuation line is active if the first line ends in a comma (,) as shown in the following example:

```
ASSIGN FEM = demo1.f06, FORM=MSC, BOUNDARY=SYM,
PRINT=1, SUPORT =123
```

continuation line active if ending in (,)

CEND must be the last command in the Executive Control Section. Other commands can be located arbitrarily in the Executive Control Section.

‘ASSIGN FEM=’ Structural Modal Data Importer

Description: Assigns an external file that contains the free vibration solutions of the finite element model. It should be noted that the data entities created by the 'ASSIGN FEM=' executive control command can be saved using the **FEMSAVE** bulk data card.

Format:

ASSIGN FEM ='a', **FORM** ='b', **BOUNDARY** ='c', **PRINT** = n, **SUPORT** = m/L, **ASET** ='d'

Example 1:

```
ASSIGN FEM=demo1.f06, FORM=MSC, BOUNDARY=SYM, PRINT=1, SUPORT=123
```

Example 2:

```
ASSIGN FEM=/export/home/ZEUS/demo2.f06, BOUNDARY=ANTI,
SUPORT=-246/3000, ASET=YES
```

Describer	Meaning
FEM = ' a '	FEM indicates that ' a ' is the filename of the external file that contains the free vibration solution of the structural finite element model. ' a ' is a character string specifying the name of the external file. (Required) UNIX systems are case sensitive; therefore, lower/upper case characters must identically match the name of the file. DOS and WINDOWS systems are not case sensitive (see Remarks 1 and 2).
FORM = ' b '	FORM indicates the name of the structural finite element code that generates the output file ' a ' by a free vibration analysis where ' b ' is a character string specifying the name of the structural finite element code. (Optional) Seven options are available for ' b ' : Data of the free vibration solution is: ' MSC ' generated by MSC.NASTRAN or NXNASTRAN (see Remark 3) ' NE ' generated by NE/NASTRAN (see Remark 3) ' ASTROS ' generated by ASTROS (see Remark 5) ' IDEAS ' generated by I-DEAS (see Remark 6) ' ELFINI ' generated by ELFINI (see Remark 7) 'GENESIS' generated by GENESIS 'ABAQUS generated by ABAQUS 'ALTAIR' generated by ALTAIR's RADIOSS (see Remark 8) ' FREE ' stored according to the input instruction described in Remark 9. 'ACQUIRE' Retrieves the free vibration solution from the file 'a' that must match the file name specified in the **FEMSAVE** bulk data card. Importing the modal solution of a large FEM by the “**ASSIGN FEM**=” command could be time consuming but that can be saved using the **FEMSAVE** bulk data card and

	retrieved by the FORM=”ACQUIRE” option. This usually can drastically reduce CPU time. If no FORM is specified, 'MSC' is used as default. Note: For NASTRAN-type of finite element code, the scalar points (**SPOINT**) will be internally expanded from one degree of freedom to six..
`BOUNDARY='c'`	BOUNDARY indicates the boundary condition of the structural finite element model. (Optional) ' c ' is a character string that has 3 options: ' SYM ' for symmetric boundary condition ' ANTI ' for anti-symmetric boundary condition ' ASYM ' for asymmetric boundary condition If no BOUNDARY is specified, ' SYM ' is used as default (See Remark 10).
`SUPORT = m/L`	Optional input to specify the degrees of freedom of the rigid body modes of the structural finite element model. *m* is an integer representing the component numbers of the rigid body degrees of freedom. It contains any unique combination of the integer 1 through 6 with no embedded blanks, where 1, 2 and 3 represent the translational rigid body modes along the x, y and z axes of the finite element basic coordinates, respectively. 4, 5 and 6 are the rotation rigid body modes about the x, y and z axes, respectively. m can also be a negative integer that activates the program to perform the following tasks: - Replaces the imported rigid body modes by the program-computed rigid body modes. - Forces the natural frequency and the generalized stiffness of the rigid body modes to be zero. The “negative m” option is useful for the cases where the structural finite element analysis fails to provide well-behaved rigid body modes or zero rigid body natural frequency. “/L” is optional where L is an integer representing the identification number of a grid point in the structural finite element model where the rigid body modes are referred to. Note that there is a slash (“/”) that separates m and L. If no “/L” is specified, the program will search for a grid point in the structural finite element model that can be best referred to by the rigid body modes. For NASTRAN type of finite element codes, m should be the R-set degrees of freedom (please see MSC.NASTRAN User’s Manual for the definition of the R-set degrees of freedom) and L is the grid identification numbers that are specified in the NASTRAN **SUPORT** bulk data card. However, if the displacement of the grid point specified in the NASTRAN SUPORT bulk data card is defined in a local coordinate system, the user must transform the component numbers in the NASTRAN SUPORT bulk data card from the local coordinate system to the basic coordinate system. *Note that the spelling of SUPORT contains only one P.* (Optional, Default = 0) (See Remark 11)

Describer	Meaning
`PRINT = n`	Print options to the standard output file; where n is an integer. (Optional) n = 0 no printout of the imported structural free vibration solution $\| n \| \geq 1$ print out the structural grid point locations in the <u>aerodynamic coordinate system</u> $n \geq 2$ print out the modal data (mode shapes) at the structural grid points in the <u>aerodynamic coordinate system</u> $n \leq -2$ print out the interpolated modal data at the control points of the aerodynamic boxes in the <u>aerodynamic coordinate system</u> n = 3 print all of the above If no PRINT is specified, n = 0 is used as a default.
`ASET`	Uses the A-set (Analysis Set) degrees of freedom (d.o.f.) of the finite element model for the ZEUS static and dynamic aeroelastic loads analysis where ‘d’ is a character string either “YES” or “NO” (Optional) (Default = “NO”) If ASET = “YES”, then 1. Only those structural grid points specified in the ASET and ASET1 NASTRAN input are used by ZEUS 2. SVECTOR = ALL must be specified in the NASTRAN Case Control Section. 3. To compute the inertial loads in the static or dynamic aeroelastic analysis, it is required to import the SMAA matrix for the symmetric/asymmetric analysis and AMAA matrix for the anti-symmetric analysis by the ZEUS **‘ASSIGN MATRIX=’** Executive Control Command into ZEUS where SMAA/AMAA is the A-set mass matrix computed by NASTRAN. Note that this option is active only if FORM = MSC. (See Remark 12)

<u>Remarks:</u>

<u>Remark 1</u> of **‘ASSIGN FEM=’:**

At least one **‘ASSIGN FEM=’** Executive Control Command must exist in the Executive Control Section. If the user wishes to perform the aeroelastic analysis for both symmetric and anti-symmetric boundary conditions of the structural finite element model, two **‘ASSIGN FEM=’** Executive Control Commands can be specified, one with BOUNDARY = SYM and the other with BOUNDARY = ANTI. For example:

```
ASSIGN FEM = demo1.f06,    FORM = MSC,    BOUNDARY = SYM

ASSIGN FEM = demo2.f06,    FORM = MSC,   BOUNDARY = ANTI
```

However, no more than two **‘ASSIGN FEM=’** commands can be specified. Furthermore, if both symmetric and anti-symmetric boundary conditions are specified, the number of structural grid points and their locations must be identical between these two finite element models.

Remark 2 of **'ASSIGN FEM=':**

ZEUS reads the file 'a' to obtain the free vibration solutions computed by the structural finite element code 'b'. Specifically, ZEUS searches for the following data in the file 'a':

- The structural grid point locations of the finite element model. These grid point locations and their identification numbers are used for spline.
- The coordinate transformations that relate the local or global coordinates to the basic coordinates. These coordinate transformations are used to transform the structural grid point locations from the local coordinates to the basic coordinates as well as the modal data from the global coordinates to the basic coordinates (for the definition of local, global and basic coordinates, please see a NASTRAN User's Manual).
- The natural frequencies, the generalized masses, the generalized stiffness and the mode shapes.

Remark 3 of ′**'ASSIGN FEM=':**

For MSC.NASTRAN, UAI/NASTRAN, CSA/NASTRAN or NE/NASTRAN, the following command must exist in the case control deck of the NASTRAN input (as well as output) file that generates the NASTRAN solution output file 'a'.

ECHO = SORT

The user must ensure that the structural finite element analysis is a free vibration analysis (or normal modes analysis). For MSC.NASTRAN, the solution sequence:

SOL 103

must be selected.

In addition, in order to import the eigenvectors, ZEUS searches for one of the following **NASTRAN** Case Control Commands:

(1) If ASET = "NO"
DISP = ALL or,
DISP = n or,
(2) If ASET = "YES"
SVECTOR = ALL

For DISP = ALL, all structural grid points can be referred to by all ZEUS bulk data cards; for instance, the structural grid points for spline. However, for a larger size finite element model, including all grid points in the ZEUS analysis may require large computational time. In this case, DISP = n or SVECTOR = ALL is recommended to reduce the number of the structural grid points for a ZEUS analysis. Consequently, only those grid points listed in the eigenvector output can be referred to by ZEUS bulk data cards.

In what follows, the "G-set" is defined as 6x (number of grid points in the structural finite element model), and "A-set" is defined as 6x (number of grid points specified by DISP=n or SVECTOR = ALL).

It should be noted that omitting structural grids by using DISP = n or SVECTOR = ALL may also exclude some structural masses if masses are attached to these omitted structural grids. This may give discrepancy in the distributed inertial loads computation that is normally a part of trim analysis and dynamic loads analysis. Therefore, it is the user's responsibility to ensure all structural masses are included in a ZEUS analysis.

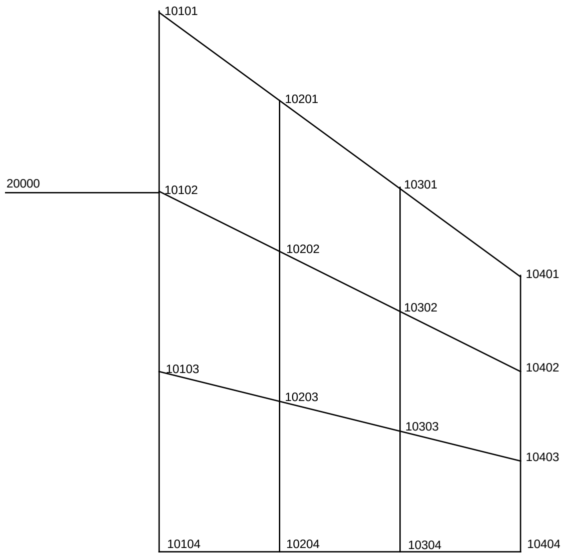
Please see a NASTRAN User's Manual for a description of the above NASTRAN case control.

<u>Remark 4</u> of '**ASSIGN FEM=**':

A single continuation line can be used in the **'ASSIGN FEM='** Executive Control Command if the first line ends in a comma (,)

<u>`MSC.NASTRAN Example`</u>

The following figure shows a plate type of finite element model:



The MSC.NASTRAN output file for normal modes analysis of the above model is listed as follows:

```
                                 N A S T R A N    E X E C U T I V E    C O N T R O L    D E C K    E C H O
0
SOL 103
CEND
0
0                                        C A S E    C O N T R O L    D E C K    E C H O
                 CARD
                 COUNT
                  1        ECHO=SORTED
                  2        DISP = ALL
                  3        METHOD = 20
                  4        SPC = 10
                  5        BEGIN BULK
0                              INPUT BULK DATA CARD COUNT =      43
```

```
0
0                                                   S O R T E D   B U L K   D A T A   E C H O
  CARD
  COUNT  .. 1      .. 2      .. 3      .. 4      .. 5      .. 6      .. 7      .. 8      .. 9      .. 10
    1-   ASET1     3         10101     THRU      10104
    2-   ASET1     3         10201     THRU      10204
    3-   ASET1     3         10301     THRU      10304
    4-   ASET1     3         10401     THRU      10404
    5-   CBAR      1010      1010      10102     20000     10101
    6-   CQUAD4    1001      1000      10101     10102     10202     10201
    7-   CQUAD4    1002      1000      10102     10103     10203     10202
    8-   CQUAD4    1003      1000      10103     10104     10204     10203
    9-   CQUAD4    1004      1000      10201     10202     10302     10301
   10-   CQUAD4    1005      1000      10202     10203     10303     10302
   11-   CQUAD4    1006      1000      10203     10204     10304     10303
   12-   CQUAD4    1007      1000      10301     10302     10402     10401
   13-   CQUAD4    1008      1000      10302     10303     10403     10402
   14-   CQUAD4    1009      1000      10303     10304     10404     10403
   15-   EIGRL     20                  5
   16-   GRID      10101     0.0       30.000    0.0
   17-   GRID      10102     33.333    30.000    0.0
   18-   GRID      10103     66.667    30.000    0.0
   19-   GRID      10104     100.000   30.000    0.0
   20-   GRID      10201     16.667    53.333    0.0
   21-   GRID      10202     44.444    53.333    0.0
   22-   GRID      10203     72.222    53.333    0.0
   23-   GRID      10204     100.000   53.333    0.0
   24-   GRID      10301     33.333    76.667    0.0
   25-   GRID      10302     55.555    76.667    0.0
   26-   GRID      10303     77.778    76.667    0.0
   27-   GRID      10304     100.000   76.667    0.0
   28-   GRID      10401     50.000    100.000   0.0
   29-   GRID      10402     66.667    100.000   0.0
   30-   GRID      10403     83.333    100.000   0.0
   31-   GRID      10404     100.000   100.000   0.0
   32-   GRID      20000     33.333    0.0       0.0
   33-   MAT1      1100      1.E+07    .3        .1
   34-   PARAM     COUPMASS1
   35-   PARAM     WTMASS .00259
   36-   PBAR      1010      1100      100.      .1E+04     .1E+04  .05E+04
   37-   PSHELL    1000      1100      1.5       1100
   38-   SPC1      10        126       10101     THRU      10104
   39-   SPC1      10        126       10201     THRU      10204
   40-   SPC1      10        126       10301     THRU      10304
   41-   SPC1      10        126       10401     THRU      10404
   42-   SPC1      10        123456    20000
                 ENDDATA
0      TOTAL COUNT=  43

            E I G E N V A L U E   A N A L Y S I S   S U M M A R Y   (READ MODULE)


                                     BLOCK SIZE USED  ..........................      7

                                     NUMBER OF DECOMPOSITIONS  .................      1

                                     NUMBER OF ROOTS FOUND    ..................      5

                                     NUMBER OF SOLVES REQUIRED  ................      5


                                                 R E A L   E I G E N V A L U E S
   MODE    EXTRACTION           EIGENVALUE            RADIANS                CYCLES              GENERALIZED       GENERALIZED
    NO.      ORDER                                                                                   MASS            STIFFNESS
    1          1              8.399865E+02        2.898252E+01          4.612711E+00          1.000000E+00      8.399865E+02
    2          2              5.401589E+03        7.349551E+01          1.169717E+01          1.000000E+00      5.401589E+03
    3          3              4.316370E+04        2.077587E+02          3.306583E+01          1.000000E+00      4.316370E+04
    4          4              7.341672E+04        2.709552E+02          4.312386E+01          1.000000E+00      7.341672E+04
    5          5              2.008154E+05        4.481243E+02          7.132120E+01          1.000000E+00      2.008154E+05

EIGENVALUE =  8.399865E+02
    CYCLES =  4.612711E+00         R E A L   E I G E N V E C T O R   N O .          1

    POINT ID.    TYPE          T1              T2              T3              R1              R2              R3
      10101       G          0.0             0.0          2.438916E-01   -1.046505E-03   1.388628E-02       0.0
      10102       G          0.0             0.0         -2.682208E-03   -1.641355E-04   6.512418E-04       0.0
      10103       G          0.0             0.0         -2.536763E-01   -1.233849E-02   1.385677E-02       0.0
      10104       G          0.0             0.0         -7.429644E-01   -1.128285E-02   1.528565E-02       0.0
      10201       G          0.0             0.0          4.281797E-02   -2.549300E-03   4.682182E-03       0.0
      10202       G          0.0             0.0         -2.088563E-01   -1.018326E-02   1.336013E-02       0.0
      10203       G          0.0             0.0         -6.013343E-01   -1.044415E-02   1.434578E-02       0.0
      10204       G          0.0             0.0         -1.016269E+00   -1.194192E-02   1.532579E-02       0.0
      10301       G          0.0             0.0         -2.736918E-01   -9.773146E-03   1.539033E-02       0.0
      10302       G          0.0             0.0         -6.082150E-01   -1.032532E-02   1.458461E-02       0.0
      10303       G          0.0             0.0         -9.465001E-01   -1.173877E-02   1.546269E-02       0.0
      10304       G          0.0             0.0         -1.292082E+00   -1.146654E-02   1.546714E-02       0.0
      10401       G          0.0             0.0         -7.746809E-01   -1.083210E-02   1.543930E-02       0.0
      10402       G          0.0             0.0         -1.037815E+00   -1.164881E-02   1.600222E-02       0.0
      10403       G          0.0             0.0         -1.304427E+00   -1.135552E-02   1.564861E-02       0.0
      10404       G          0.0             0.0         -1.566373E+00   -1.190655E-02   1.563803E-02       0.0
      20000       G          0.0             0.0          0.0             0.0             0.0                0.0

EIGENVALUE =  5.401589E+03
    CYCLES =  1.169717E+01         R E A L   E I G E N V E C T O R   N O .          2
```

POINT ID.	TYPE	T1	T2	T3	R1	R2	R3
10101	G	0.0	0.0	-2.248838E-01	-2.775307E-02	-1.129584E-02	0.0
10102	G	0.0	0.0	-5.626384E-03	-3.665544E-04	-1.306792E-03	0.0
10103	G	0.0	0.0	7.584442E-01	-2.015278E-02	-4.288727E-02	0.0
10104	G	0.0	0.0	2.027624E+00	-3.048569E-02	-3.250301E-02	0.0
10201	G	0.0	0.0	-6.131099E-01	-1.869456E-02	-7.974025E-03	0.0
10202	G	0.0	0.0	-2.231425E-01	-2.759980E-02	-1.968281E-02	0.0
10203	G	0.0	0.0	4.212290E-01	-2.518181E-02	-2.596193E-02	0.0
10204	G	0.0	0.0	1.302277E+00	-3.147953E-02	-3.677993E-02	0.0
10301	G	0.0	0.0	-1.083220E+00	-3.588441E-02	-1.368388E-02	0.0
10302	G	0.0	0.0	-6.909589E-01	-3.148519E-02	-2.168402E-02	0.0
10303	G	0.0	0.0	-1.315537E-01	-3.430069E-02	-2.810665E-02	0.0
10304	G	0.0	0.0	5.153657E-01	-3.505625E-02	-2.956726E-02	0.0
10401	G	0.0	0.0	-1.624408E+00	-3.543852E-02	-2.199665E-02	0.0
10402	G	0.0	0.0	-1.240496E+00	-3.705165E-02	-2.421410E-02	0.0
10403	G	0.0	0.0	8.081899E-01	-3.638719E-02	-2.723050E-02	0.0
10404	G	0.0	0.0	-3.356609E-01	-3.727286E-02	-2.898597E-02	0.0
20000	G	0.0	0.0	0.0	0.0	0.0	0.0

```
EIGENVALUE =  4.316370E+04
     CYCLES =  3.306583E+01        R E A L   E I G E N V E C T O R   N O .        3
```

POINT ID.	TYPE	T1	T2	T3	R1	R2	R3
10101	G	0.0	0.0	8.585293E-01	4.772991E-02	4.532157E-02	0.0
10102	G	0.0	0.0	4.092312E-02	2.285311E-03	-1.063501E-03	0.0
10103	G	0.0	0.0	6.611228E-01	-3.340588E-04	-2.785009E-02	0.0
10104	G	0.0	0.0	1.358505E+00	-4.635419E-02	-1.239622E-02	0.0
10201	G	0.0	0.0	1.117747E+00	2.147846E-02	3.023199E-02	0.0
10202	G	0.0	0.0	5.920185E-01	3.574188E-02	3.084039E-03	0.0
10203	G	0.0	0.0	5.350678E-01	-2.010099E-02	7.068093E-03	0.0
10204	G	0.0	0.0	2.898809E-01	-4.816738E-02	1.145165E-02	0.0
10301	G	0.0	0.0	1.326068E+00	2.839814E-02	1.657062E-02	0.0
10302	G	0.0	0.0	6.829033E-01	-9.244961E-03	3.839550E-02	0.0
10303	G	0.0	0.0	-1.475777E-01	-2.670205E-02	3.879387E-02	0.0
10304	G	0.0	0.0	-9.657666E-01	-5.703675E-02	3.529658E-02	0.0
10401	G	0.0	0.0	7.903178E-01	-1.441457E-02	6.219073E-02	0.0
10402	G	0.0	0.0	-2.571835E-01	-2.155319E-02	6.048384E-02	0.0
10403	G	0.0	0.0	-1.253393E+00	-4.181003E-02	5.970822E-02	0.0
10404	G	0.0	0.0	-2.218969E+00	-4.735287E-02	5.613626E-02	0.0
20000	G	0.0	0.0	0.0	0.0	0.0	0.0

```
EIGENVALUE =  7.341672E+04
   CYCLES =  4.312386E+01        R E A L   E I G E N V E C T O R   N O .        4
```

POINT ID.	TYPE	T1	T2	T3	R1	R2	R3
10101	G	0.0	0.0	-4.123309E+00	2.694899E-04	-2.266406E-01	0.0
10102	G	0.0	0.0	-2.318701E-02	-1.243020E-03	-5.314526E-03	0.0
10103	G	0.0	0.0	4.520594E-01	-6.215813E-03	-2.121695E-02	0.0
10104	G	0.0	0.0	2.440502E-01	-3.207996E-02	3.096157E-02	0.0
10201	G	0.0	0.0	-1.283839E+00	3.459100E-02	-6.163387E-02	0.0
10202	G	0.0	0.0	1.576902E-01	9.684119E-04	-3.170266E-02	0.0
10203	G	0.0	0.0	2.870014E-01	-6.443665E-03	2.288025E-02	0.0
10204	G	0.0	0.0	-4.325946E-01	-2.493853E-02	2.622103E-02	0.0
10301	G	0.0	0.0	2.637444E-01	3.015133E-02	-2.357039E-02	0.0
10302	G	0.0	0.0	4.917885E-01	1.411954E-02	9.793158E-03	0.0
10303	G	0.0	0.0	9.593081E-03	-4.840639E-03	3.343225E-02	0.0
10304	G	0.0	0.0	-9.164144E-01	-1.701009E-02	4.736346E-02	0.0
10401	G	0.0	0.0	9.870760E-01	2.619092E-02	2.147435E-02	0.0
10402	G	0.0	0.0	5.199889E-01	9.901542E-03	3.957807E-02	0.0
10403	G	0.0	0.0	-2.610622E-01	8.370100E-04	5.313112E-02	0.0
10404	G	0.0	0.0	-1.179375E+00	-7.177794E-03	5.455842E-02	0.0
20000	G	0.0	0.0	0.0	0.0	0.0	0.0

```
EIGENVALUE =  2.008154E+05
   CYCLES =  7.132120E+01        R E A L   E I G E N V E C T O R   N O .        5
```

POINT ID.	TYPE	T1	T2	T3	R1	R2	R3
10101	G	0.0	0.0	1.976632E+00	-9.472703E-02	1.147041E-01	0.0
10102	G	0.0	0.0	-2.276870E-02	-1.390012E-03	9.441336E-04	0.0
10103	G	0.0	0.0	8.776556E-01	-7.974713E-02	-5.213426E-02	0.0
10104	G	0.0	0.0	2.455836E+00	-1.372054E-01	-3.867267E-02	0.0
10201	G	0.0	0.0	-7.512134E-01	-4.105826E-02	-3.071724E-03	0.0
10202	G	0.0	0.0	-6.829810E-01	-3.796132E-02	-3.375415E-03	0.0
10203	G	0.0	0.0	-5.347021E-01	-3.837091E-02	-6.255790E-03	0.0
10204	G	0.0	0.0	-5.127236E-01	-9.846668E-02	7.154678E-03	0.0
10301	G	0.0	0.0	-7.428587E-01	2.160513E-02	-2.462126E-02	0.0
10302	G	0.0	0.0	-3.308228E-01	6.036545E-02	-1.134401E-02	0.0
10303	G	0.0	0.0	-4.807106E-01	4.840772E-02	2.519936E-02	0.0
10304	G	0.0	0.0	-1.450914E+00	2.149668E-02	6.291359E-02	0.0
10401	G	0.0	0.0	1.687299E+00	1.180740E-01	-3.960162E-02	0.0
10402	G	0.0	0.0	1.979366E+00	1.149720E-01	8.102916E-03	0.0
10403	G	0.0	0.0	1.461685E+00	1.163374E-01	5.380255E-02	0.0
10404	G	0.0	0.0	3.300396E-01	1.111649E-01	8.177724E-02	0.0
20000	G	0.0	0.0	0.0	0.0	0.0	0.0

Remark 5 of **‘ASSIGN FEM=’:**

If FORM = ASTROS, the following three commands must exist in the solution control section of the input (as well as output) file that generates the ASTROS solution output file ‘a’:

```
MODES
PRINT (MODES  = ALL),    DISP = ALL,    ROOT = ALL
BEGIN BULK (SORT)
```

Please see the ASTROS User’s Manual for a description of the above commands.

A sample output file of ASTROS free vibration analysis is shown below:

```
*****  ASTROS ASSIGN DATABASE COMMAND ECHO  *****

          *...10...**...20...**...30...**...40...**...50...**...60...**...70...**...80...*

          ASSIGN DATABASE DEMO,ZONA,NEW,DELETE

          *...10...**...20...**...30...**...40...**...50...**...60...**...70...**...80...*

          DATA BASE NAME                    =      DEMO
          DATA BASE PASSWORD                =      ZONA
          DATA BASE STATUS                  =      NEW
          USER PARAMETERS ARE :
          DELETE

                S O L U T I O N   C O N T R O L   S U M M A R Y

          ANALYZE
           BOUNDARY METHOD=20,REDUCE=30 ,SPC=10
            LABEL = DEMO CASE
            MODES
           PRINT (MODES=ALL) DISP=ALL,ROOT=ALL
         END

                 S O R T E D   B U L K   D A T A   E C H O
        CARD
        COUNT    *. . . 1 . .  **. . . 2 . . ** . . . 3 . . ** . . . 4 . . ** . . . 5 . . ** . . . 6 . . ** . . . 7 . . ** . . . 8 . . ** . . . 9 . . ** . . 1 0 . . *
           1 -    ASET1     30      3      10401    THRU    10404
           2 -    ASET1     30      3      10301    THRU    10304
           3 -    ASET1     30      3      10101    THRU    10104
           4 -    ASET1     30      3      10201    THRU    10204
           5 -    CBAR      1010    1010   10102    20000   10101
           6 -    CONVERT MASS      00259
           7 -    CQUAD4    1001    1000   10101    10102   10202   10201
           8 -    CQUAD4    1002    1000   10102    10103   10203   10202
           9 -    CQUAD4    1003    1000   10103    10104   10204   10203
          10 -    CQUAD4    1004    1000   10201    10202   10302   10301
          11 -    CQUAD4    1005    1000   10202    10203   10303   10302
          12 -    CQUAD4    1006    1000   10203    10204   10304   10303
          13 -    CQUAD4    1007    1000   10301    10302   10402   10401
          14 -    CQUAD4    1008    1000   10302    10303   10403   10402
          15 -    CQUAD4    1009    1000   10303    10304   10404   10403
          16 -    EIGR      20      MGIV            5.0             5                       +ABC
          17 -    +ABC      MAX
          18 -    GRID      10101          0.0      30.000  0.0
          19 -    GRID      10102          33.333   30.000  0.0
          20 -    GRID      10103          66.667   30.000  0.0
          21 -    GRID      10104          100.000  30.000  0.0
          22 -    GRID      10201          16.667   53.333  0.0
          23 -    GRID      10202          44.444   53.333  0.0
          24 -    GRID      10203          72.222   53.333  0.0
          25 -    GRID      10204          100.000  53.333  0.0
          26 -    GRID      10301          33.333   76.667  0.0
          27 -    GRID      10302          55.555   76.667  0.0
          28 -    GRID      10303          77.778   76.667  0.0
          29 -    GRID      10304          100.000  76.667  0.0
          30 -    GRID      10401          50.000   100.000 0.0
          31 -    GRID      10402          66.667   100.000 0.0
          32 -    GRID      10403          83.333   100.000 0.0
          33 -    GRID      10404          100.000  100.000 0.0
          34 -    GRID      20000          33.333   0.0     0.0
          35 -    MAT1      1100    1.E+07          .3      .1
          36 -    PBAR      1010    1100   100.     .1E+04  .1E+04          .05E+04
          37 -    PSHELL    1000    1100   1.5      1100
          38 -    SPC       10      20000  123456
          39 -    SPC1      10      126    10101    THRU    10104
          40 -    SPC1      10      126    10201    THRU    10204
          41 -    SPC1      10      126    10301    THRU    10304
          42 -    SPC1      10      126    10401    THRU    10404
          43 -    ENDDATA
 DEMO CASE                                                         MODES ANALYSIS: BOUNDARY 1
                                      S U M M A R Y   O F   R E A L   E I G E N   A N A L Y S I S

            16 EIGENVALUES AND 5 EIGENVECTORS EXTRACTED USING METHOD GIVENS

            MAXIMUM OFF DIAGONAL MASS TERM IS 2.081896235E-15 AT ROW 5 AND COLUMN 3
```

MODE	EXTRACTION ORDER	EIGENVALUE (RAD/S)**2	FREQUENCY (RAD/S)	FREQUENCY (HZ)	GENERALIZED MASS	GENERALIZED STIFFNESS
1	1	7.85673E+02	2.80299E+01	4.46109E+00	4.36633E-01	3.43051E+02
2	2	4.40079E+03	6.63384E+01	1.05581E+01	3.02067E-01	1.32933E+03
3	3	3.41515E+04	1.84801E+02	2.94120E+01	2.70232E-01	9.22883E+03
4	4	4.18786E+04	2.04643E+02	3.25699E+01	9.05048E-02	3.79021E+03
5	5	9.88844E+04	3.14459E+02	5.00477E+01	4.82243E-01	4.76864E+04
6	6	1.33059E+05	3.64773E+02	5.80554E+01	0.00000E+00	0.00000E+00
7	7	1.86616E+05	4.31991E+02	6.87535E+01	0.00000E+00	0.00000E+00
8	8	3.81747E+05	6.17857E+02	9.83350E+01	0.00000E+00	0.00000E+00
9	9	3.88298E+05	6.23135E+02	9.91751E+01	0.00000E+00	0.00000E+00
10	10	6.67839E+05	8.17214E+02	1.30064E+02	0.00000E+00	0.00000E+00
11	11	8.58100E+05	9.26337E+02	1.47431E+02	0.00000E+00	0.00000E+00
12	12	1.03265E+06	1.01619E+03	1.61732E+02	0.00000E+00	0.00000E+00
13	13	1.17125E+06	1.08224E+03	1.72245E+02	0.00000E+00	0.00000E+00
14	16	1.76139E+06	1.32717E+03	2.11226E+02	0.00000E+00	0.00000E+00
15	14	2.78933E+06	1.67013E+03	2.65809E+02	0.00000E+00	0.00000E+00
16	15	4.13498E+06	2.03347E+03	3.23636E+02	0.00000E+00	0.00000E+00

DEMO CASE MODES ANALYSIS: BOUNDARY 1, MODE 1

R E A L E I G E N V E C T O R F O R M O D E 1

EIGENVALUE = 7.85673E+02 (RAD/S)**2
CYCLIC FREQUENCY = 4.46109E+00 HZ

POINT ID.	TYPE	T1	T2	T3	R1	R2	R3
10101	G	0.00000E+00	0.00000E+00	-1.57724E-01	4.74973E-04	-9.04594E-03	0.00000E+00
10102	G	0.00000E+00	0.00000E+00	1.65129E-03	1.01365E-04	-4.18610E-04	0.00000E+00
10103	G	0.00000E+00	0.00000E+00	1.60828E-01	7.84750E-03	-8.90299E-03	0.00000E+00
10104	G	0.00000E+00	0.00000E+00	4.75673E-01	7.20763E-03	-9.88694E-03	0.00000E+00
10201	G	0.00000E+00	0.00000E+00	-3.11719E-02	1.54758E-03	-3.01484E-03	0.00000E+00
10202	G	0.00000E+00	0.00000E+00	1.30878E-01	6.42130E-03	-8.59521E-03	0.00000E+00
10203	G	0.00000E+00	0.00000E+00	3.82162E-01	6.65170E-03	-9.26838E-03	0.00000E+00
10204	G	0.00000E+00	0.00000E+00	6.49799E-01	7.61182E-03	-9.90462E-03	0.00000E+00
10301	G	0.00000E+00	0.00000E+00	1.69017E-01	6.16125E-03	-9.91356E-03	0.00000E+00
10302	G	0.00000E+00	0.00000E+00	3.84447E-01	6.54387E-03	-9.40881E-03	0.00000E+00
10303	G	0.00000E+00	0.00000E+00	6.02194E-01	7.47848E-03	-9.98555E-03	0.00000E+00
10304	G	0.00000E+00	0.00000E+00	8.25272E-01	7.31122E-03	-1.00018E-02	0.00000E+00
10401	G	0.00000E+00	0.00000E+00	4.88816E-01	6.88055E-03	-9.98074E-03	0.00000E+00
10402	G	0.00000E+00	0.00000E+00	6.58736E-01	7.41750E-03	-1.03422E-02	0.00000E+00
10403	G	0.00000E+00	0.00000E+00	8.30776E-01	7.23585E-03	-1.01161E-02	0.00000E+00
10404	G	0.00000E+00	0.00000E+00	1.00000E+00	7.59407E-03	-1.01071E-02	0.00000E+00
20000	G	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00

DEMO CASE MODES ANALYSIS: BOUNDARY 1, MODE 2

R E A L E I G E N V E C T O R F O R M O D E 2

EIGENVALUE = 4.40079E+03 (RAD/S)**2
CYCLIC FREQUENCY = 1.05581E+01 HZ

POINT ID.	TYPE	T1	T2	T3	R1	R2	R3
10101	G	0.00000E+00	0.00000E+00	-9.99443E-02	-1.37041E-02	-5.07218E-03	0.00000E+00
10102	G	0.00000E+00	0.00000E+00	-2.77751E-03	-1.81102E-04	-6.32624E-04	0.00000E+00
10103	G	0.00000E+00	0.00000E+00	3.69295E-01	-9.94513E-03	-2.12634E-02	0.00000E+00
10104	G	0.00000E+00	0.00000E+00	1.00000E+00	-1.52928E-02	-1.62984E-02	0.00000E+00
10201	G	0.00000E+00	0.00000E+00	-2.96668E-01	-9.29880E-03	-3.80525E-03	0.00000E+00
10202	G	0.00000E+00	0.00000E+00	-1.10681E-01	-1.36695E-02	-9.48454E-03	0.00000E+00
10203	G	0.00000E+00	0.00000E+00	2.01263E-01	-1.24942E-02	-1.27435E-02	0.00000E+00
10204	G	0.00000E+00	0.00000E+00	6.35580E-01	-1.58076E-02	-1.82668E-02	0.00000E+00
10301	G	0.00000E+00	0.00000E+00	-5.32187E-01	-1.78486E-02	-6.38841E-03	0.00000E+00
10302	G	0.00000E+00	0.00000E+00	-3.44961E-01	-1.56378E-02	-1.04377E-02	0.00000E+00
10303	G	0.00000E+00	0.00000E+00	-7.43806E-02	-1.71329E-02	-1.37092E-02	0.00000E+00
10304	G	0.00000E+00	0.00000E+00	2.41691E-01	-1.75982E-02	-1.45025E-02	0.00000E+00
10401	G	0.00000E+00	0.00000E+00	-8.07570E-01	-1.76693E-02	-1.05283E-02	0.00000E+00
10402	G	0.00000E+00	0.00000E+00	-6.22996E-01	-1.85493E-02	-1.16236E-02	0.00000E+00
10403	G	0.00000E+00	0.00000E+00	-4.14728E-01	-1.82571E-02	-1.31904E-02	0.00000E+00
10404	G	0.00000E+00	0.00000E+00	-1.85288E-01	-1.87568E-02	-1.41208E-02	0.00000E+00
20000	G	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00

DEMO CASE MODES ANALYSIS: BOUNDARY 1, MODE 3

R E A L E I G E N V E C T O R F O R M O D E 3

EIGENVALUE = 3.41515E+04 (RAD/S)**2
CYCLIC FREQUENCY = 2.94120E+01 HZ

POINT ID.	TYPE	T1	T2	T3	R1	R2	R3
10101	G	0.00000E+00	0.00000E+00	-1.75601E-01	-2.32465E-02	-9.02655E-03	0.00000E+00
10102	G	0.00000E+00	0.00000E+00	-1.71048E-02	-9.66031E-04	6.84568E-04	0.00000E+00
10103	G	0.00000E+00	0.00000E+00	-2.88296E-01	-1.42255E-03	1.34849E-02	0.00000E+00
10104	G	0.00000E+00	0.00000E+00	-6.34242E-01	2.18125E-02	6.83909E-03	0.00000E+00
10201	G	0.00000E+00	0.00000E+00	-4.50809E-01	-1.25424E-02	-1.14399E-02	0.00000E+00
10202	G	0.00000E+00	0.00000E+00	-2.79537E-01	-1.80782E-02	2.91872E-04	0.00000E+00
10203	G	0.00000E+00	0.00000E+00	-2.61509E-01	7.86821E-03	-3.22475E-03	0.00000E+00
10204	G	0.00000E+00	0.00000E+00	-1.33835E-01	2.23012E-02	-6.19697E-03	0.00000E+00
10301	G	0.00000E+00	0.00000E+00	-6.44802E-01	-1.60531E-02	-5.98641E-03	0.00000E+00
10302	G	0.00000E+00	0.00000E+00	-3.71222E-01	2.48809E-03	-1.76481E-02	0.00000E+00

```
    10303       G         0.00000E+00    0.00000E+00    2.27884E-02    1.12650E-02    -1.89721E-02     0.00000E+00
    10304       G         0.00000E+00    0.00000E+00    4.31970E-01    2.59648E-02    -1.80001E-02     0.00000E+00
    10401       G         0.00000E+00   0.00000E+00    -4.65978E-01    4.68261E-03    -2.96838E-02     0.00000E+00
    10402       G         0.00000E+00   0.00000E+00     3.61409E-02    9.01869E-03    -2.95462E-02     0.00000E+00
    10403       G         0.00000E+00   0.00000E+00     5.21803E-01    1.88400E-02    -2.95140E-02     0.00000E+00
    10404       G         0.00000E+00   0.00000E+00     1.00000E+00    2.15487E-02    -2.79225E-02     0.00000E+00
    20000       G         0.00000E+00   0.00000E+00     0.00000E+00    0.00000E+00     0.00000E+00     0.00000E+00

DEMO CASE                                                          MODES ANALYSIS: BOUNDARY 1, MODE 4


                                   R E A L   E I G E N V E C T O R   F O R   M O D E   4

          EIGENVALUE       =     4.18786E+04  (RAD/S)**2
          CYCLIC FREQUENCY =     3.25699E+01  HZ

   POINT ID.   TYPE          T1             T2             T3             R1             R2             R3
    10101       G      0.00000E+00     0.00000E+00     1.00000E+00   -2.01715E-03     5.64829E-02     0.00000E+00
    10102       G      0.00000E+00     0.00000E+00     4.92003E-03    2.67899E-04     1.21958E-03     0.00000E+00
    10103       G      0.00000E+00     0.00000E+00    -7.22189E-02    8.02059E-04     3.34307E-03     0.00000E+00
    10104       G      0.00000E+00     0.00000E+00     1.42217E-02    4.72348E-03    -8.03788E-03     0.00000E+00
    10201       G      0.00000E+00     0.00000E+00     2.91175E-01   -7.87643E-03     1.41986E-02     0.00000E+00
    10202       G      0.00000E+00     0.00000E+00    -2.42111E-02    8.48906E-04     6.83955E-03     0.00000E+00
    10203       G      0.00000E+00     0.00000E+00    -4.72883E-02    7.89466E-04    -5.12644E-03     0.00000E+00
    10204       G      0.00000E+00     0.00000E+00     1.03009E-01    2.83089E-03    -5.22758E-03     0.00000E+00
    10301       G      0.00000E+00     0.00000E+00    -1.28856E-02   -4.00481E-03     4.67938E-03     0.00000E+00
    10302       G      0.00000E+00     0.00000E+00    -6.90750E-02   -1.66587E-03    -9.81908E-04     0.00000E+00
    10303       G      0.00000E+00     0.00000E+00    -2.46962E-04    9.49520E-04    -5.16162E-03     0.00000E+00
    10304       G      0.00000E+00     0.00000E+00     1.50640E-01    1.48197E-03    -7.97554E-03     0.00000E+00
    10401       G      0.00000E+00     0.00000E+00    -1.25005E-01   -3.45775E-03    -2.60315E-03     0.00000E+00
    10402       G      0.00000E+00     0.00000E+00    -6.85205E-02   -9.94270E-04    -5.30369E-03     0.00000E+00
    10403       G      0.00000E+00     0.00000E+00     3.93305E-02   -3.41753E-04    -7.56189E-03     0.00000E+00
    10404       G      0.00000E+00     0.00000E+00     1.70948E-01    5.85901E-04    -7.80940E-03     0.00000E+00
    20000       G      0.00000E+00     0.00000E+00     0.00000E+00    0.00000E+00     0.00000E+00     0.00000E+00

DEMO CASE                                                          MODES ANALYSIS: BOUNDARY 1, MODE 5


                                   R E A L   E I G E N V E C T O R   F O R   M O D E   5

          EIGENVALUE       =     9.88844E+04  (RAD/S)**2
          CYCLIC FREQUENCY =     5.00477E+01  HZ
   POINT ID.   TYPE          T1             T2             T3             R1             R2             R3
    10101       G        0.00000E+00    0.00000E+00     7.30464E-01   -6.58403E-02    4.60747E-02     0.00000E+00
    10102       G        0.00000E+00    0.00000E+00    -1.01415E-02   -6.32742E-04   -6.43032E-04     0.00000E+00
    10103       G        0.00000E+00    0.00000E+00     5.49054E-01   -3.57703E-02   -3.07092E-02     0.00000E+00
    10104       G        0.00000E+00    0.00000E+00     1.00000E+00   -7.20460E-02    2.40956E-03     0.00000E+00
    10201       G        0.00000E+00    0.00000E+00    -6.66242E-01   -2.65721E-02   -1.53368E-02     0.00000E+00
    10202       G        0.00000E+00    0.00000E+00    -2.27467E-01   -1.97661E-02   -1.47552E-02     0.00000E+00
    10203       G        0.00000E+00    0.00000E+00    -7.48511E-02   -1.81559E-02    5.30764E-03     0.00000E+00
    10204       G        0.00000E+00    0.00000E+00    -4.51465E-01   -4.65767E-02    2.02751E-02     0.00000E+00
    10301       G        0.00000E+00    0.00000E+00    -4.88703E-01    9.74488E-03   -2.84319E-02     0.00000E+00
    10302       G        0.00000E+00    0.00000E+00    -4.13593E-02    2.35718E-02   -1.00285E-02     0.00000E+00
    10303       G        0.00000E+00    0.00000E+00    -1.31374E-01    1.94000E-02    1.94331E-02     0.00000E+00
    10304       G        0.00000E+00    0.00000E+00    -8.59769E-01    1.16691E-02    4.44581E-02     0.00000E+00
    10401       G        0.00000E+00    0.00000E+00     7.68068E-01    5.36557E-02   -2.24129E-02     0.00000E+00
    10402       G        0.00000E+00    0.00000E+00     9.32998E-01    5.02390E-02    4.81151E-03     0.00000E+00
    10403       G        0.00000E+00    0.00000E+00     6.32223E-01    5.15985E-02    3.23634E-02     0.00000E+00
    10404       G        0.00000E+00    0.00000E+00    -4.26008E-02    5.12637E-02    4.69097E-02     0.00000E+00
    20000       G        0.00000E+00    0.00000E+00     0.00000E+00    0.00000E+00    0.00000E+00     0.00000E+00
```

Remark 6 of **‘ASSIGN FEM=’:**

If FORM = IDEAS, the structural grids and modal results are read in from I-DEAS universal files. ZEUS supports both the older and newer versions of I-DEAS output formats. The following table lists allowable data sets for modal data input to ZEUS.

Data Set No.	Description
781 and/or 2411	Nodes (i.e. GRID points)
55 and/or 2414	Eigenvector information including frequency and modal mass (i.e., structure mode shapes)
18 and/or 2420	Coordinate systems

Data sets other than those in the table above may appear in the universal file and are ignored. ZEUS output plot files in universal file format are also supported which can be directly viewed by I-DEAS. Please see the PLTxxxx bulk data cards in Section 4.0 for descriptions.

A sample output file of the universal file format is shown below:

```
-1

2420

         8

Demo

         1         0         8

CS1

  1.0000000000000000D+000  0.0000000000000000D+000  0.0000000000000000D+000
  0.0000000000000000D+000  1.0000000000000000D+000  0.0000000000000000D+000
  0.0000000000000000D+000  0.0000000000000000D+000  1.0000000000000000D+000
  0.0000000000000000D+000  0.0000000000000000D+000  0.0000000000000000D+000
       111         0         8
CS111
  9.9999999999999989D-001  0.0000000000000000D+000  0.0000000000000000D+000
  0.0000000000000000D+000  0.0000000000000000D+000 -1.0000000000000000D+000
  0.0000000000000000D+000  1.0000000000000000D+000  0.0000000000000000D+000
  0.0000000000000000D+000  0.0000000000000000D+000  0.0000000000000000D+000
    -1
-1
  2411
     10101         1         1         1
   3.0000000000000000D+01   0.0000000000000000D+00   0.0000000000000000D+00
     10102         1         1         1
   3.0000000000000000D+01   3.3333000183105467D+01   0.0000000000000000D+00
     10103         1         1         1
   3.0000000000000000D+01   6.6666999816894527D+01   0.0000000000000000D+00
     10104         1         1         1
   3.0000000000000000D+01   1.0000000000000000D+02   0.0000000000000000D+00
     10201         1         1         1
   5.3333000183105472D+01   1.6666999816894532D+01   0.0000000000000000D+00
     10202         1         1         1
   5.3333000183105472D+01   4.4444000244140626D+01   0.0000000000000000D+00
     10203         1         1         1
   5.3333000183105472D+01   7.2222000122070308D+01   0.0000000000000000D+00
     10204         1         1         1
   5.3333000183105472D+01   1.0000000000000000D+02   0.0000000000000000D+00
     10301         1         1         1
   7.6666999816894527D+01   3.3333000183105467D+01   0.0000000000000000D+00
     10302         1         1         1
   7.6666999816894527D+01   5.5555000305175781D+01   0.0000000000000000D+00
     10303         1         1         1
   7.6666999816894527D+01   7.7777999877929691D+01   0.0000000000000000D+00
     10304         1         1         1
   7.6666999816894527D+01   1.0000000000000000D+02   0.0000000000000000D+00
     10401         1         1         1
   1.0000000000000000D+02   5.0000000000000000D+01   0.0000000000000000D+00
     10402         1         1         1
   1.0000000000000000D+02   6.6666999816894527D+01   0.0000000000000000D+00
     10403         1         1         1
   1.0000000000000000D+02   8.3333000183105490D+01   0.0000000000000000D+00
     10404         1       111         1
   1.0000000000000000D+02   1.0000000000000000D+02   0.0000000000000000D+00
     20000         1         1         1
   0.0000000000000000D+00   3.3333000183105467D+01   0.0000000000000000D+00
    -1
-1
  2414
         1
B.C. 0,MODE 1, DISPLACEMENT_1
         1
NONE
OUGV1    : REAL MODE SHAPE
ANALYSIS DATE 07/27/99
REAL EIGENVALUE SOLUTION
```

```
MODE SHAPE         1 : FREQUENCY (HERTZ) 4.61271E+000
         1         2         3         8         2         6
       -11         0         1         0         1         1         0         0
      1500         0
  0.00000E+00  4.61271E+00  0.00000E+00  1.00000E+00  0.00000E+00  0.00000E+00
  0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00
     10101
  0.00000E+00  0.00000E+00  2.43892E-01 -1.38863E-02  1.04650E-03  0.00000E+00
     10102
  0.00000E+00  0.00000E+00 -2.68221E-03 -6.51242E-04  1.64135E-04  0.00000E+00
     10103
  0.00000E+00  0.00000E+00 -2.53676E-01 -1.38568E-02  1.23385E-02  0.00000E+00
     10104
  0.00000E+00  0.00000E+00 -7.42964E-01 -1.52856E-02  1.12829E-02  0.00000E+00
     10201
  0.00000E+00  0.00000E+00  4.28180E-02 -4.68218E-03  2.54930E-03  0.00000E+00
     10202
  0.00000E+00  0.00000E+00 -2.08856E-01 -1.33601E-02  1.01833E-02  0.00000E+00
     10203
  0.00000E+00  0.00000E+00 -6.01334E-01 -1.43458E-02  1.04441E-02  0.00000E+00
     10204
  0.00000E+00  0.00000E+00 -1.01627E+00 -1.53258E-02  1.19419E-02  0.00000E+00
     10301
  0.00000E+00  0.00000E+00 -2.73692E-01 -1.53903E-02  9.77315E-03  0.00000E+00
     10302
  0.00000E+00  0.00000E+00 -6.08215E-01 -1.45846E-02  1.03253E-02  0.00000E+00
     10303
  0.00000E+00  0.00000E+00 -9.46500E-01 -1.54627E-02  1.17388E-02  0.00000E+00
     10304
  0.00000E+00  0.00000E+00 -1.29208E+00 -1.54671E-02  1.14665E-02  0.00000E+00
     10401
  0.00000E+00  0.00000E+00 -7.74681E-01 -1.54393E-02  1.08321E-02  0.00000E+00
     10402
  0.00000E+00  0.00000E+00 -1.03781E+00 -1.60022E-02  1.16488E-02  0.00000E+00
     10403
  0.00000E+00  0.00000E+00 -1.30443E+00 -1.56486E-02  1.13555E-02  0.00000E+00
     10404
  0.00000E+00  0.00000E+00 -1.56637E+00 -1.56380E-02  1.19065E-02  0.00000E+00
     20000
  0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00
    -1

                                  .
                                  .
                                  .
```

Remark 7 of **‘ASSIGN FEM=’:**

ZEUS supports the ELFINI neutral output file to acquire the free vibration solutions. Since ELFINI always outputs the free vibration solutions in SI (Standard International) format (metric - Length = meters, Mass = kilograms) the structural model and aerodynamic model may be in different unit systems for this option only. ***Again, different units between the structural and aerodynamic models is only allowed for this type of input*** (FORM=ELFINI). To convert the ELFINI metric units to the aerodynamic model units requires specifying the FMMUNIT and FMLUNIT entries of the **AEROZ** bulk data card to reflect the units of the aerodynamic model. The program will then convert the ELFINI free vibration solutions to match the units of the aerodynamic model. If the aerodynamic model is also in SI units, then the FMMUNIT and FMLUNIT entries should be set to KG and M, respectively.

A sample of the ELFINI neutral file is shown as follows:

```
HEADER=MODEL

 RELEASE=   1

 NAME=      E767

 DATE=30/10/96 AT=10.49.10
```

```
 TITLE

NO TITLE

  END TITLE

 END HEADER


CONTENT

 DEGREE     NBDOF=   30

 K-MATRIX   NBMATRIX=    1

 M-MATRIX   NBMATRIX=    1
 MONVAL     NBMON= 3177
 END CONTENT

UNIT
 SYSTEM= ISO
 END UNIT

DEGREE
 END DEGREE

K-MATRIX
 DOF=    1
    0.1230923559613264D+04    0.3686414966027151D-04    0.2419062914082075D-03
    0.6983785713017072D-04    -.2023261010701697D-04    0.7831696499717704D-04
    -.2457474668222889D-03    0.1130354829300728D-03    0.4604977332201705D-04
    -.1091649015086483D-03    -.3635281838273271D-04    -.9274910202744236D-04
    0.1309670841427416D-03    0.4488872714871719D-04    -.2327751708509681D-04
    -.7472762150226767D-04    0.4713285201889093D-04    0.3429857074005640D-04
    -.5447145357492463D-06    0.3868854191317292D-05    -.3624751739196352D-04
    0.8176428928352975D-05    -.1887274213933592D-05    0.1598938947169428D-05
    -.1808144454036312D-05    0.1165396547550420D-05    -.3689883842386831D-04
    -.2884186353798238D-04    0.8310472989868395D-05    -.3608598589997129D-05
 DOF=    2
    0.3686414966027151D-04    0.1358544160900551D+05    0.4296049769816649D-03
    0.1628147041567078D-03    -.5617118452227373D-04    0.1914983001579172D-03
                                          .

M-MATRIX

 DOF=    1

    0.5497212390994216D+00    -.9446850040509963D-14    -.4294748421913158D-14
    0.9254207643250023D-15    -.1457946990131992D-14    0.1109663982879969D-14
    -.6852211940217012D-14    0.6256388636327603D-15    -.1378020961229076D-15
    0.1480555993560104D-14    0.7164950056870456D-15    0.2515687853345272D-15
    -.3575698764857194D-15    -.1365281585702371D-14    0.1470923534883928D-15
    0.4089610594615323D-15    -.1493488492598782D-15    0.1576836534608606D-16
    -.4511975103434207D-16    -.2422683635736750D-16    0.3794707603699266D-17
    -.3460616633478488D-16    0.3662570463927595D-17    0.2534322578184867D-16
    0.1245350190700635D-17    0.3208899617378191D-16    0.3016114918583113D-16
    0.1008308020411519D-16    -.9849299110673004D-17    0.8090181085815273D-16
 DOF=    2
    -.9446850040509963D-14    0.1517800567316923D+01    -.1020468025402302D-14
    -.2718403166413960D-14    -.1067939139898222D-16    0.6759217887003857D-14
                                          .

MONVAL=     1
 TITLE
KIND=DISP    ;NAME=               ;UNIT=M         ;
SELECT=WING;NODE=        2;TYPE=TX;COORD.=0.2940557999999999D+00 0.0000000000000
000D+00 0.9113519999999997D-01 M;
  END TITLE
    -.6270452321170384D-04    -.3770965314259045D-03    0.1681525813481753D-02
    -.7268882809304363D-03    0.8641032168708783D-03    -.5617263119365574D-02
    -.4243476398549203D-03    0.1115622083120243D-01    -.2398198156757896D-02
    0.3264161266023014D-02    -.1459664532940326D-01    -.1179512947055147D-01
    0.6565833985373308D-02    -.1217856158745318D-01    -.7468784150169609D-02
    0.4677584975980958D-02    -.2097598934419818D-01    -.4019675274132945D-02
    0.5269019719698231D-03    -.2955775293472249D-02    0.7297341338640479D-03
    -.7193417522388700D-03    -.5516094061587379D-03    -.1730473266346012D-02
    -.1565697075052970D-03    -.9836643468589825D-04    0.3743652572097349D-02
    -.9436596571633759D-03    0.1392135637645999D-03    -.1182337803036962D-03
 END MONVAL

MONVAL=     2
 TITLE
KIND=DISP    ;NAME=               ;UNIT=M         ;
SELECT=WING;NODE=        2;TYPE=TY;COORD.=0.2940557999999999D+00 0.0000000000000
000D+00 0.9113519999999997D-01 M;
  END TITLE
    -.6631092559614345D-04    0.2057090096506267D-02    -.6501226652201383D-02
```

```
0.2216313171525446D-02    -.4468904163869089D-02    0.3136186294985838D-01
```

Remark 8 of **′ASSIGN FEM =′:**

Radioss/OptiStruct can output the modal solution in the ZEUS's ASSIGN FEM= 'FREE' format. To generate these output from Radioss/OptiStruct requires the following additional line be included in the Radioss/OptiStruct input deck:

PARAM,ZEUS,YES

The output eigenmode data are then output from Radioss/OptiStruct to an ASCII text file named

```
<input data file name>.zeus
```

The G-set mass matrix can also be output by Radioss/OptiStruct in the DMIG format. The DMIG can then simply be pasted into a ZEUS input deck to perform analyses that require the mass matrix (e.g., a Trim run to include inertial effects).

The output DMIG file (containing the MGG data) is named

```
<input data file name>_<DMIG matrix name>.pch
```

where <DMIG matrix name> can be defined using the following card in output control:

DMIGNAME=<DMIG matrix name> (Default is AX)

Radioss can directly process NASTRAN bulk data cards. Therefore, most NASTRAN input files (e.g., those from MSC.Nastran, NX.Nastran, etc.) can be run directly by Radioss. Note that Radioss only supports the Lanczos eigenvalue solver, so EIGRL needs to be used when defining METHOD in the case control.

Remark 9 of **‘ASSIGN FEM=’:**

If FORM = FREE, it is assumed that the free vibration solution of the finite element model is obtained by some other structural finite element code. In this case, it is the user’s responsibility to set up the modal data in file ‘a’ according to the following data format:

There are four input card sets required to construct the file ‘a’. Each card set may contain one or a group of input cards.

Card Set 1	**NGRID, NMODE** (Free Format)
NGRID NMODE	Number of structural grid points of the finite element model (Integer > 0) Number of structural modes (Integer > 0)
Example	17 5

Card Set 2	**ID, x, y, z** (Free Format)
ID X, y, z	Identification number of the structural grid points (Integer > 0) x, y and z locations of the grid points (Real)
Example	100 1.0 3.0 0.0

Repeat card set 2, **NGRID** times for all structural grid points.

Card Set 3	**FREQ, GENM** (Free Format)
FREQ GENM	Natural frequency of the mode (rad/sec) (Real) Generalized mass of the modes (Real)
Example	38.23 0.032

Card Set 4	**ID, T_1, T_2, T_3, R_1, R_2, R_3** (Free Format)
ID T_1, T_2, T_3 R_1, R_2, R_3	Identification number of the structural grid point; must exist in card set 2 (Integer > 0) Translational modal displacement in x, y and z directions (Real) Rotational modal displacement about x, y and z directions (Real)
Example	100 0.0 0.0 3.3 2.1 0.5 4.0

Repeat card set 4 **NGRID** times for the modal displacement at all grid points.

Go back to card set 3. Repeat this process **NMODE** times for all modes.

Comment cards may be used in a modal data file with FORM = FREE format and must be initiated with a "$" in the first column.

An example for FORM = FREE, is shown as follows:

```
$  EXAMPLE CASE WITH FORM = FREE
17  5
   10101        0.0000       30.0000      0.0000
   10102        33.3330      30.0000      0.0000
   10103        66.6670      30.0000      0.0000
   10104        100.0000     30.0000      0.0000
   10201        16.6670      53.3330      0.0000
   10202        44.4440      53.3330      0.0000
   10203        72.2220      53.3330      0.0000
   10204        100.0000     53.3330      0.0000
   10301        33.3330      76.6670      0.0000
   10302        55.5550      76.6670      0.0000
   10303        77.7780      76.6670      0.0000
   10304        100.0000     76.6670      0.0000
   10401        50.0000      100.0000     0.0000
   10402        66.6670      100.0000     0.0000
   10403        83.3330      100.0000     0.0000
   10404        100.0000     100.0000     0.0000
   20000        33.3330      0.0000       0.0000
$ MODE 1
0.28983E+02  0.10000E+01
  10101     0.00000E+00   0.00000E+00    0.24389E+00   -0.10465E-02   0.13886E-01   0.00000E+00
  10102     0.00000E+00   0.00000E+00   -0.26822E-02   -0.16414E-03   0.65124E-03   0.00000E+00
  10103     0.00000E+00   0.00000E+00   -0.25368E+00   -0.12338E-01   0.13857E-01   0.00000E+00
  10104     0.00000E+00   0.00000E+00   -0.74296E+00   -0.11283E-01   0.15286E-01   0.00000E+00
  10201     0.00000E+00   0.00000E+00    0.42818E-01   -0.25493E-02   0.46822E-02   0.00000E+00
  10202     0.00000E+00   0.00000E+00   -0.20886E+00   -0.10183E-01   0.13360E-01   0.00000E+00
  10203     0.00000E+00   0.00000E+00   -0.60133E+00   -0.10444E-01   0.14346E-01   0.00000E+00
  10204     0.00000E+00   0.00000E+00   -0.10163E+01   -0.11942E-01   0.15326E-01   0.00000E+00
  10301     0.00000E+00   0.00000E+00   -0.27369E+00   -0.97731E-02   0.15390E-01   0.00000E+00
  10302     0.00000E+00   0.00000E+00   -0.60821E+00   -0.10325E-01   0.14585E-01   0.00000E+00
  10303     0.00000E+00   0.00000E+00   -0.94650E+00   -0.11739E-01   0.15463E-01   0.00000E+00
```

```
10304    0.00000E+00   0.00000E+00   -0.12921E+01   -0.11467E-01    0.15467E-01   0.00000E+00
10401    0.00000E+00   0.00000E+00   -0.77468E+00   -0.10832E-01    0.15439E-01   0.00000E+00
10402    0.00000E+00   0.00000E+00   -0.10378E+01   -0.11649E-01    0.16002E-01   0.00000E+00
10403    0.00000E+00   0.00000E+00   -0.13044E+01   -0.11356E-01    0.15649E-01   0.00000E+00
10404    0.00000E+00   0.00000E+00   -0.15664E+01   -0.11907E-01    0.15638E-01   0.00000E+00
20000    0.00000E+00   0.00000E+00    0.00000E+00    0.00000E+00    0.00000E+00   0.00000E+00
$ MODE 2
0.73496E+02  0.10000E+01
10101    0.00000E+00   0.00000E+00   -0.22488E+00   -0.27753E-01   -0.11296E-01   0.00000E+00
10102    0.00000E+00   0.00000E+00   -0.56264E-02   -0.36655E-03   -0.13068E-02   0.00000E+00
10103    0.00000E+00   0.00000E+00    0.75844E+00   -0.20153E-01   -0.42887E-01   0.00000E+00
10104    0.00000E+00   0.00000E+00    0.20276E+01   -0.30486E-01   -0.32503E-01   0.00000E+00
10201    0.00000E+00   0.00000E+00   -0.61311E+00   -0.18695E-01   -0.79740E-02   0.00000E+00
10202    0.00000E+00   0.00000E+00   -0.22314E+00   -0.27600E-01   -0.19683E-01   0.00000E+00
10203    0.00000E+00   0.00000E+00    0.42123E+00   -0.25182E-01   -0.25962E-01   0.00000E+00
10204    0.00000E+00   0.00000E+00    0.13023E+01   -0.31480E-01   -0.36780E-01   0.00000E+00
10301    0.00000E+00   0.00000E+00   -0.10832E+01   -0.35884E-01   -0.13684E-01   0.00000E+00
10302    0.00000E+00   0.00000E+00   -0.69096E+00   -0.31485E-01   -0.21684E-01   0.00000E+00
10303    0.00000E+00   0.00000E+00   -0.13155E+00   -0.34301E-01   -0.28107E-01   0.00000E+00
10304    0.00000E+00   0.00000E+00    0.51537E+00   -0.35056E-01   -0.29567E-01   0.00000E+00
10401    0.00000E+00   0.00000E+00   -0.16244E+01   -0.35439E-01   -0.21997E-01   0.00000E+00
10402    0.00000E+00   0.00000E+00   -0.12405E+01   -0.37052E-01   -0.24214E-01   0.00000E+00
10403    0.00000E+00   0.00000E+00   -0.80819E+00   -0.36387E-01   -0.27230E-01   0.00000E+00
10404    0.00000E+00   0.00000E+00   -0.33566E+00   -0.37273E-01   -0.28986E-01   0.00000E+00
20000    0.00000E+00   0.00000E+00    0.00000E+00    0.00000E+00    0.00000E+00   0.00000E+00
$ MODE 3
0.20776E+03  0.10000E+01
10101    0.00000E+00   0.00000E+00    0.85853E+00    0.47730E-01    0.45322E-01   0.00000E+00
10102    0.00000E+00   0.00000E+00    0.40923E-01    0.22853E-02   -0.10635E-02   0.00000E+00
10103    0.00000E+00   0.00000E+00    0.66112E+00   -0.33406E-03   -0.27850E-01   0.00000E+00
10104    0.00000E+00   0.00000E+00    0.13585E+01   -0.46354E-01   -0.12396E-01   0.00000E+00
10201    0.00000E+00   0.00000E+00    0.11177E+01    0.21478E-01    0.30232E-01   0.00000E+00
10202    0.00000E+00   0.00000E+00    0.59202E+00    0.35742E-01    0.30840E-02   0.00000E+00
10203    0.00000E+00   0.00000E+00    0.53507E+00   -0.20101E-01    0.70681E-02   0.00000E+00
10204    0.00000E+00   0.00000E+00    0.28988E+00   -0.48167E-01    0.11452E-01   0.00000E+00
10301    0.00000E+00   0.00000E+00    0.13261E+01    0.28398E-01    0.16571E-01   0.00000E+00
10302    0.00000E+00   0.00000E+00    0.68290E+00   -0.92450E-02    0.38396E-01   0.00000E+00
10303    0.00000E+00   0.00000E+00   -0.14758E+00   -0.26702E-01    0.38794E-01   0.00000E+00


10304    0.00000E+00   0.00000E+00   -0.96577E+00   -0.57037E-01    0.35297E-01   0.00000E+00
10401    0.00000E+00   0.00000E+00    0.79032E+00   -0.14415E-01    0.62191E-01   0.00000E+00
10402    0.00000E+00   0.00000E+00   -0.25718E+00   -0.21553E-01    0.60484E-01   0.00000E+00
10403    0.00000E+00   0.00000E+00   -0.12534E+01   -0.41810E-01    0.59708E-01   0.00000E+00
10404    0.00000E+00   0.00000E+00   -0.22190E+01   -0.47353E-01    0.56136E-01   0.00000E+00
20000    0.00000E+00   0.00000E+00    0.00000E+00    0.00000E+00    0.00000E+00   0.00000E+00
$ MODE 4
0.27096E+03  0.10000E+01
10101    0.00000E+00   0.00000E+00   -0.41233E+01    0.26949E-03   -0.22664E+00   0.00000E+00
10102    0.00000E+00   0.00000E+00   -0.23187E-01   -0.12430E-02   -0.53145E-02   0.00000E+00
10103    0.00000E+00   0.00000E+00    0.45206E+00   -0.62158E-02   -0.21217E-01   0.00000E+00
10104    0.00000E+00   0.00000E+00    0.24405E+00   -0.32080E-01    0.30962E-01   0.00000E+00
10201    0.00000E+00   0.00000E+00   -0.12838E+01    0.34591E-01   -0.61634E-01   0.00000E+00
10202    0.00000E+00   0.00000E+00    0.15769E+00    0.96841E-03   -0.31703E-01   0.00000E+00
10203    0.00000E+00   0.00000E+00    0.28700E+00   -0.64437E-02    0.22880E-01   0.00000E+00
10204    0.00000E+00   0.00000E+00   -0.43259E+00   -0.24939E-01    0.26221E-01   0.00000E+00
10301    0.00000E+00   0.00000E+00    0.26374E+00    0.30151E-01   -0.23570E-01   0.00000E+00
10302    0.00000E+00   0.00000E+00    0.49179E+00    0.14120E-01    0.97932E-02   0.00000E+00
10303    0.00000E+00   0.00000E+00    0.95931E-02   -0.48406E-02    0.33432E-01   0.00000E+00
10304    0.00000E+00   0.00000E+00   -0.91641E+00   -0.17010E-01    0.47363E-01   0.00000E+00
10401    0.00000E+00   0.00000E+00    0.98708E+00    0.26191E-01    0.21474E-01   0.00000E+00
10402    0.00000E+00   0.00000E+00    0.51999E+00    0.99015E-02    0.39578E-01   0.00000E+00
10403    0.00000E+00   0.00000E+00   -0.26106E+00    0.83701E-03    0.53131E-01   0.00000E+00
10404    0.00000E+00   0.00000E+00   -0.11794E+01   -0.71778E-02    0.54558E-01   0.00000E+00
20000    0.00000E+00   0.00000E+00    0.00000E+00    0.00000E+00    0.00000E+00   0.00000E+00
$ MODE 5
0.44812E+03  0.10000E+01
10101    0.00000E+00   0.00000E+00    0.19766E+01    0.94727E-01    0.11470E+00   0.00000E+00
10102    0.00000E+00   0.00000E+00   -0.22769E-01   -0.13900E-02    0.94413E-03   0.00000E+00
10103    0.00000E+00   0.00000E+00    0.87766E+00   -0.79747E-01   -0.52134E-01   0.00000E+00
10104    0.00000E+00   0.00000E+00    0.24558E+01   -0.13721E+00   -0.38673E-01   0.00000E+00
10201    0.00000E+00   0.00000E+00   -0.75121E+00   -0.41058E-01   -0.30717E-02   0.00000E+00
10202    0.00000E+00   0.00000E+00   -0.68298E+00   -0.37961E-01   -0.33754E-02   0.00000E+00
10203    0.00000E+00   0.00000E+00   -0.53470E+00   -0.38371E-01   -0.62558E-02   0.00000E+00
10204    0.00000E+00   0.00000E+00   -0.51272E+00   -0.98467E-01    0.71547E-02   0.00000E+00
10301    0.00000E+00   0.00000E+00   -0.74286E+00    0.21605E-01   -0.24621E-01   0.00000E+00
10302    0.00000E+00   0.00000E+00   -0.33082E+00    0.60365E-01   -0.11344E-01   0.00000E+00
10303    0.00000E+00   0.00000E+00   -0.48071E+00    0.48408E-01    0.25199E-01   0.00000E+00
10304    0.00000E+00   0.00000E+00   -0.14509E+01    0.21497E-01    0.62914E-01   0.00000E+00
```

10401	0.00000E+00	0.00000E+00	0.16873E+01	0.11807E+00	-0.39602E-01	0.00000E+00
10402	0.00000E+00	0.00000E+00	0.19794E+01	0.11497E+00	0.81029E-02	0.00000E+00
10403	0.00000E+00	0.00000E+00	0.14617E+01	0.11634E+00	0.53803E-01	0.00000E+00
10404	0.00000E+00	0.00000E+00	0.33004E+00	0.11116E+00	0.81777E-01	0.00000E+00
20000	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00

Remark 10 of **‘ASSIGN FEM=’:**

Since the geometry of an aircraft is usually symmetric about a vertical plane passing through the center line of the fuselage, only half of the aircraft is required to be modeled structurally (as well as aerodynamically). The symmetric modes and anti-symmetric modes of the aircraft structure can be obtained by imposing the so-called ‘symmetric boundary condition’ and ‘anti-symmetric boundary condition’, respectively, at the structural grid points along the center line plane of the fuselage. Each boundary condition gives different natural frequencies and mode shapes.

Remark 11 of **‘ASSIGN FEM=’:**

If the SUPORT entry is activated, the program will transform the rigid body modes computed by the finite element analysis from the generalized coordinates to the body axis coordinates but leaves the elastic modes unaltered. In the body axis coordinates, all translational rigid body modes have a value of one in their respective translational degrees of freedom and zero in other degrees of freedom. Whereas all rotational rigid body modes have a unit rotation angle about their respective rotation degrees of freedom whose rotation center is located at REFX, REFY, and REFZ (specified by the **AEROZ** bulk data card). Consequently, the generalized mass matrix associated with the rigid body modes are also transformed into the body axis coordinates. This normally gives a non-diagonal generalized mass matrix that, in fact, contains the physical mass properties of the structure. Note that if only one half of the configuration is modeled (XZSYM = “YES” in the **AEROZ** bulk data card), these mass properties are only one half of the mass and mass moment of inertia of the whole configuration.

The impact of activating the SUPORT entry on flutter, static aeroelastic/trim and aeroservoelastic analysis are:

- *For flutter analysis*: the result of the flutter analysis is not altered.

- *For static aeroelastic/trim analysis*:

  If the structural finite element model involves rigid body modes, the SUPORT entry must be specified. Otherwise, the trim analysis may encounter a singularity in the static aeroelastic analysis. .This is because the stiffness matrix of a free-free structure is singular. The program can remove this singularity only if the SUPORT entry is specified.

- *For transient maneuver loads analysis*:

  The SUPORT entry is recommended. The rigid body modes are transformed into the airframe states to construct a set of state-space equations for transient response analysis.

- *For transient ejection loads analysis, discrete gust loads analysis, and nonlinear flutter analysis:*

Use of the SUPORT entry is highly recommended. In addition, if the entry SOF= "YES" in the **MLDPRNT** bulk data card is specified to compute the transient loads using the mode acceleration method, the SUPORT entry is a must if rigid body mode exists. Otherwise, the resulting loads could be incorrect due to the singular behavior of the time response of the rigid body displacement. This singular behavior can be removed by specifying the SUPORT entry.

<u>Remark 12</u> of ′**ASSIGN FEM =′:**

If ASET = "YES" is specified, only those grid points appearing in the ASET and ASET1 NASTRAN input can be referred to by the ZEUS bulk data cards (**SPLINEi**, **LOADMOD** …etc). This option reduces the degrees of freedom (d.o.f.) of the structural model from the G-set to A-set d.o.f. and can save CPU time for the computation of the inertial loads in the static and dynamic aeroelastic analysis (TRIM, MLOADS, ELOADS, GLOADS, and NLFLTR analyses). The inertial loads are computed using the [SMGH] for the symmetric/asymmetric analysis or the [AMGH] matrix for the anti-symmetric analysis where

[SMGH] = [SMAA][SPHI]
[AMGH] = [AMAA][APHI]

[SPHI] and [APHI] are the symmetric/asymmetric and anti-symmetric A-set modal matrices, respectively, imported from the NASTRAN modal output file with SVECTOR=ALL being specified in the NASTRAN Case Control Section and [SMAA] and [AMAA] are the symmetric/asymmetric and anti-symmetric A-set mass matrices, respectively, exported from NASTRAN using the NASTRAN DMAP ALTER command and imported to ZEUS using the ZEUS **'ASSIGN MATRIX='** Executive Control Command such as:

ASSIGN MATRIX = FILENAME, MNAME = SMAA
ASSIGN MATRIX = FILENAME, MNAME = AMAA

'ASSIGN MATRIX =' Direct Matrix Input by INPUTT4 Format

Description: Assign an external file that contains the ASCII or binary data of a matrix for direct matrix input. The format of the external file is the same as the INPUTT4 format of NASTRAN (see INPUTT4 or OUTPUT4 module description of MSC.NASTRAN DMAP Module Dictionary).

Format:

ASSIGN MATRIX = ' a ', **FORM** = ' b ', **MNAME** = ' c ', **PRINT** = n

Note that a single continuous line can be used if the first line ends in a comma (,).

Example 1:

```
ASSIGN MATRIX = demo1.mgh, MNAME=SMGH, FORM=FORMAT, PRINT = 1
```

Example 2:

```
ASSIGN MATRIX = /export/home/zeus/demo2.mgh,
FORM = UNFORMAT
```

Describer	Meaning
MATRIX = ' a '	MATRIX indicates that ' a ' is the filename of the external file that contains the data of a matrix for direct matrix input. ' a ' represents a character string specifying the name of the external file. (Required)
FORM = ' b '	FORM indicates the format of the data on the external file. (Optional) (default = FORMAT) ' b ' = FORMAT, for ASCII with 5E16.9 format. ' b ' = FORMAT23, for ASCII with 3D23.16 format. ' b ' = UNFORMAT, for binary format. Note that the matrix can be either sparse or non-sparse.
MNAME = ' c '	MNAME indicates that ' c ' is the name of the matrix (up to 8 characters). The matrix on the external file ' a ' is read in and written on the runtime database as a matrix entity with name = ' c '. If MNAME is not specified, the name of the matrix specified in the header record of the INPUTT4 format is used as the name of the matrix. (Optional)
PRINT = n	Print option to the output file; where n is an integer. $n = 0$ no printout of matrix. $n \neq 0$ print out the matrix. If no PRINT is specified, $n = 0$ is used as a default. (Optional)

'ASSIGN MATRIX=' is an optional Executive Control Command for direct matrix input, because trim and transient response analysis may require some additional structural matrices from the finite element analysis. These matrices can be exported from FEM analysis and imported into ZEUS using the **'ASSIGN MATRIX='** Executive Control Command. (See remark 1 for those structural matrices). The format of the matrix data stored on the external file is very similar to that of the INPUTT4 (or OUTPUT4) module of NASTRAN.

To output the matrix such as G-set mass (MGG) or stiffness (KGG) matrix from NASTRAN, one can use the following ALTER statements in the NASTRAN input file:

For ASCII and non-sparse matrix:

```
ASSIGN OUTPUT4='ha144ds.mgg' UNIT=13 FORM=FORMATTED
ASSIGN OUTPUT4='ha144ds.kgg' UNIT=14 FORM=FORMATTED
COMPILE SEMODES SOUIN=MSCSOU LIST NOREF $
ALTER 'STRAIN ENERGY' $
MATGEN EQEXINS/INTEXT/9//LUSETS $
OUTPUT4 MGG//-1/13///16 $
OUTPUT4 KGG//-1/14///16 $
ENDALTER
```

For ASCII and sparse matrix:

```
ASSIGN OUTPUT4='ha144ds.sparse.ascii.mgg' unit=29 form=formatted
ASSIGN OUTPUT4='ha144ds.sparse.ascii.kgg' unit=30 form=formatted
COMPILE SEMODES SOUIN=MSCSOU LIST NOREF $
ALTER 'STRAIN ENERGY' $
MATGEN EQEXINS/INTEXT/9//LUSETS $
OUTPUT4 MGG//-1/-29///16 $
OUTPUT4 KGG//-1/-30///16 $
ENDALTER
```

For binary and non-sparse matrix:

```
ASSIGN OUTPUT4='ha144ds.sparse.bin.mgg' unit=35 form=unformatted
ASSIGN OUTPUT4='ha144ds.sparse.bin.kgg' unit=36 form=unformatted
COMPILE SEMODES SOUIN=MSCSOU LIST NOREF $
ALTER 'STRAIN ENERGY' $
MATGEN EQEXINS/INTEXT/9//LUSETS $
OUTPUT4 MGG//-1/-35///16 $
OUTPUT4 KGG//-1/-36///16 $
ENDALTER
```

For binary and sparse matrix:

```
ASSIGN OUTPUT4='ha144ds.sparse.bin.bigmat.mgg' unit=37 form=unformatted
ASSIGN OUTPUT4='ha144ds.sparse.bin.bigmat.kgg' unit=38 form=unformatted
COMPILE SEMODES SOUIN=MSCSOU LIST NOREF $
ALTER 'STRAIN ENERGY' $
```

```
MATGEN EQEXINS/INTEXT/9//LUSETS $
OUTPUT4 MGG//-1/-37//TRUE/16 $
OUTPUT4 KGG//-1/-38//TRUE/16 $
ENDALTER
```

The options of format are presented as follows:

For Non-Sparse and ASCII Format (FORM = FORMAT)

Record 1	NCOL, NROW, NF, NTYPE, NAME (4I8, A8)
	NCOL Number of columns. NROW Number of rows. NF Form of matrix. NF=2 General rectangular matrix. NF=6 Symmetric matrix. Only the upper triangular part (including diagonals) is input. NTYPE Type of matrix. NTYPE=1 Real, single precision. NTYPE=2 Real, double precision. NTYPE=3 Complex, single precision. NTYPE=4 Complex, double precision. NAME Character string up to 8 characters. If no MNAME = ′c′ is specified, these characters are used as the name of the matrix.
Record 2	ICOL, IROW, NW (3I8)
	ICOL Column Number. IROW Row position of the first nonzero term. NW Number of words in the column. For a complex matrix, there are two words for each element of the matrix.
Record 3	A(J) , J = IROW, IROW + NW / NC - 1 (5E16.9) for FORM = FORMAT (3D23.16) for FORM = FORMAT23
	For NTYPE=1: NC=1 and A is a real, single precision array. NTYPE=2: NC=1 and A is a real, double precision array. NTYPE=3: NC=2 and A is a complex, single precision array. NTYPE=4: NC=2 and A is a complex, double precision array.

Records 2 and 3 are repeated for each column.

Record 2 with the last column number plus +1 and at least one dummy value in Record 3 must also be added at the bottom of the file. Thus, there are a total of (NCOL + 1) numbers of Records 2 and 3 in the file.
An example is shown as follows:

```
    5     102       2       2MGH     1P,5E16.9
       1       3      99
 6.855846336E-03 0.000000000E+00 0.000000000E+00 0.000000000E+00 0.000000000E+00
 0.000000000E+00-1.162878605E-02 0.000000000E+00-2.181833573E-03 0.000000000E+00
 0.000000000E+00 0.000000000E+00-5.625212629E-02 0.000000000E+00 0.000000000E+00
 0.000000000E+00 0.000000000E+00 0.000000000E+00-4.825029982E-02 0.000000000E+00
 0.000000000E+00 0.000000000E+00 0.000000000E+00 0.000000000E+00-6.989890183E-03
 0.000000000E+00 0.000000000E+00 0.000000000E+00 0.000000000E+00 0.000000000E+00
-6.215569848E-02 0.000000000E+00 0.000000000E+00 0.000000000E+00 0.000000000E+00
 0.000000000E+00-1.509172999E-01 0.000000000E+00 0.000000000E+00 0.000000000E+00
 0.000000000E+00 0.000000000E+00-1.093032792E-01 0.000000000E+00 0.000000000E+00
 0.000000000E+00 0.000000000E+00 0.000000000E+00-3.930833207E-02 0.000000000E+00
 0.000000000E+00 0.000000000E+00 0.000000000E+00 0.000000000E+00-1.210470133E-01
 0.000000000E+00 0.000000000E+00 0.000000000E+00 0.000000000E+00 0.000000000E+00
-1.884292515E-01 0.000000000E+00 0.000000000E+00 0.000000000E+00 0.000000000E+00
 0.000000000E+00-1.173323700E-01 0.000000000E+00 0.000000000E+00 0.000000000E+00
 0.000000000E+00 0.000000000E+00-2.918305947E-02 0.000000000E+00 0.000000000E+00
 0.000000000E+00 0.000000000E+00 0.000000000E+00-7.453748578E-02 0.000000000E+00
 0.000000000E+00 0.000000000E+00 0.000000000E+00 0.000000000E+00-9.896419781E-02
 0.000000000E+00 0.000000000E+00 0.000000000E+00 0.000000000E+00 5.763368009E-02
 0.000000000E+00 0.000000000E+00 0.000000000E+00 0.000000000E+00 0.000000000E+00
 0.000000000E+00-1.495288195E-04 0.000000000E+00 1.115356274E-03
       2       3      99
-1.847709670E-02 0.000000000E+00 0.000000000E+00 0.000000000E+00 0.000000000E+00
 0.000000000E+00-1.297997974E-03 0.000000000E+00-4.428561904E-03 0.000000000E+00
             ⋮
0.000000000E+00 0.000000000E+00 0.000000000E+00 0.000000000E+00 9.050600279E-03
 0.000000000E+00 0.000000000E+00 0.000000000E+00 0.000000000E+00 0.000000000E+00
 0.000000000E+00-2.976067064E-04 0.000000000E+00 2.228496429E-03
       3       3      99
-5.003520334E-02 0.000000000E+00 0.000000000E+00 0.000000000E+00 0.000000000E+00
 0.000000000E+00-6.703697305E-02 0.000000000E+00-3.474696162E-02 0.000000000E+00
             ⋮
0.000000000E+00 0.000000000E+00 0.000000000E+00 0.000000000E+00-6.193062631E-02
 0.000000000E+00 0.000000000E+00 0.000000000E+00 0.000000000E+00 0.000000000E+00
 0.000000000E+00-2.439368145E-03 0.000000000E+00 1.811090503E-02
       4       3      99
-1.515793658E-01 0.000000000E+00 0.000000000E+00 0.000000000E+00 0.000000000E+00
 0.000000000E+00-6.854211102E-02 0.000000000E+00-2.003283120E-02 0.000000000E+00
             ⋮
0.000000000E+00 0.000000000E+00 0.000000000E+00 0.000000000E+00 3.274867786E-02
 0.000000000E+00 0.000000000E+00 0.000000000E+00 0.000000000E+00 0.000000000E+00
 0.000000000E+00-1.419543836E-03 0.000000000E+00 1.052054613E-02
       5       3      99
 4.641623764E-02 0.000000000E+00 0.000000000E+00 0.000000000E+00 0.000000000E+00
 0.000000000E+00 8.665747224E-03 0.000000000E+00-1.854310840E-02 0.000000000E+00
             ⋮
0.000000000E+00 0.000000000E+00 0.000000000E+00 0.000000000E+00-2.731625756E-03
 0.000000000E+00 0.000000000E+00 0.000000000E+00 0.000000000E+00 0.000000000E+00
 0.000000000E+00-1.271699994E-03 0.000000000E+00 9.484510803E-03
       6       1       1
 1.310664892E+00
```

<u>*For Sparse and ASCII Format*</u> (FORM = SFORMAT)

Record 1	NCOL, NROW, NF, NTYPE, NAME (4I8, A8)
	NCOL Number of columns. NROW Number of rows. NF Form of matrix. NF=2 General rectangular matrix. NF=6 Symmetric matrix. Only the upper triangular part (including diagonals) is input. Type of matrix.

	NTYPE	NTYPE=1 Real, single precision. NTYPE=2 Real, double precision. NTYPE=3 Complex, single precision. NTYPE=4 Complex, double precision.
	NAME	Character string up to 8 characters. If no MNAME = ′c′ is specified, these characters are used as the name of the matrix.
Record 2	ICOL, IZERO, NW (3I8)	
	ICOL	Column Number.
	IZERO	Must be 0.
	NW	Number of words in the column. For a complex matrix, there are two values per row.
Record 3	IS (I8)	
	IS = IROW + 65536 * (L + 1) where IROW is the row position of the first term in the string and L is the length of the string. For example, a string of six words beginning in row 4 has IS = 458756. L and IROW may be derived from IS by: L = INT (IS / 65536) – 1 IROW = IS – 65536 * (L + 1)	
Record 4	A(J) , J = IROW, IROW + L / NC / ND - 1 (5E16.9)	
	For NTYPE=1: NC=1, ND=1 and A is a real, single precision array. NTYPE=2: NC=1, ND=2 and A is a real, double precision array. NTYPE=3: NC=2, ND=1 and A is a complex, single precision array. NTYPE=4: NC=2, ND=2 and A is a complex, double precision array.	

Records 3 and 4 are repeated for NW words.

Records 2, 3 and 4 are repeated for each column.

Record 2 with the last column number plus +1 and at least one dummy value in Records 3 and 4 must also be added at the bottom of the file. Thus, the total numbers of Record 2 in the file must be (NCOL + 1).

An example is shown as follows:

```
    5     102       2       2MGH     1P,5E16.9
       1       0      57
  196611
 6.855846336E-03
  196617
-1.162878605E-02
  196619
-2.181833573E-03
  196623
-5.625212629E-02
  196629
-4.825029982E-02
  196635
-6.989890183E-03
  196641
-6.215569848E-02
  196647
-1.509172999E-01
  196653
```

```
-1.093032792E-01
  196659
-3.930833207E-02
  196665
-1.210470133E-01
  196671
-1.884292515E-01
  196677
-1.173323700E-01
  196683
-2.918305947E-02
  196689
-7.453748578E-02
  196695
-9.896419781E-02
  196700
 5.763368009E-02
  196707
-1.495288195E-04
  196709
 1.115356274E-03
        2       0      57
  196611
-1.847709670E-02
  196617
-1.297997974E-03
  196619
           :
-2.976067064E-04
  196709
 2.228496429E-03
        3       0      57
  196611
-5.003520334E-02
  196617
-6.703697305E-02
  196619
           :
-2.439368145E-03
  196709
 1.811090503E-02
        4       0      57
  196611
-1.515793658E-01
  196617
-6.854211102E-02
  196619
           :
-1.419543836E-03
  196709
 1.052054613E-02
        5       0      57
  196611
 4.641623764E-02
  196617
 8.665747224E-03
  196619
           :
-1.271699994E-03
  196709
 9.484510803E-03
        6       1       1
 1.026752114E+00
```

For Non-Sparse and Binary Format (FORM = UNFORMAT)

Record 1			
Word ***Number***	Type	NCOL, NROW, NF, NTYPE, WORD1, WORD2 (4I8, A8)	
1	Integer	NCOL	Number of columns
2	Integer	NROW	Number of rows
3	Integer	NF	Form of matrix NF=2 General rectangular matrix NF=6 Symmetric matrix. Only the upper triangular (including diagonals) is inputted.
4	Integer	NTYPE	Type of matrix NTYPE=1 Real, single precision NTYPE=2 Real, double precision NTYPE=3 Complex, single precision NTYPE=4 Complex, double precision
5 and 6	Character	WORD1, WORD2	Character string up to 8 characters Two character string. Each has 4 characters. If no MNAME = ′c′ is specified, these characters are used as the name of the matrix.

Record 2			
Word ***Number***	Type	ICOL, IROW, NW, A(J) , J = IROW, IROW + NW / NC / ND - 1	
1	Integer	ICOL	Column number
2	Integer	IROW	Row position of first nonzero term
3	Integer	NW	Number of words in the column
NW	Real or Complex, Single or Double Precision	A	For: NTYPE = 1: NC = 1, ND = 1 and A is a real, single precision array NTYPE = 2: NC = 1, ND = 2 and A is a real, double precision array NTYPE = 3: NC = 2, ND = 1 and A is a complex, single precision array NTYPE = 4: NC = 2, ND = 2 and A is a complex, double precision array

Record 2 is repeated for each column.

At the end of the file, Record 2 with the last column number plus +1 and at least one dummy value in A must be included. Thus, the total number of Record 2 in the file is NCOL + 1.

Remarks:

These structural matrices required by the trim or transient response analysis include SMGH/AMGH, SMAA/AMAA, or MGG. Please see description of the ASE Case Control Command for the definition of these matrices.
'ASSIGN MATRIX=' Executive Control Command can be used to directly input the structural modal matrices and the generalized mass/stiffness matrices. The following table lists the names of the above matrices that should be specified in the MNAME entry.

Matrix Name (specified in the MNAME entry)	**Description**	**Remarks**
SPHI	Symmetric (or asymmetric) modal matrix containing NG rows and NMODE columns. Where NG is 6x number of structural grid points, and NMODE is the number of modes of structural finite element model which is imported by the **'ASSIGN FEM='** Executive Control Command.	The structural eigenvectors imported by the **'ASSIGN FEM='** Executive Control Command will be replaced by SPHI.
APHI	Same as SPHI but for the anti-symmetry structures.	
SMHH	Symmetric (or asymmetric) structural generalized mass matrix containing NMODE rows and NMODE columns.	The generalized mass matrix imported by the **'ASSIGN FEM='** Executive Control Command will be replaced by SMHH.
AMHH	Same as SMHH but for the anti-symmetric structure.	
SKHH	Same as SMHH but for the structural generalized matrix.	
AKHH	Same as SKHH but for the anti-symmetric structure.	
SMGH	[SMGH]=[MGG][SPHI] where MGG is the G-set mass matrix.	SMGH is used to compute the inertial loads in the symmetric/asymmetric static or dynamic aeroelastic analysis.
AMGH	[AMGH]=[MGG][APHI]	AMGH is used to compute the inertial loads in the anti-symmetric static or dynamic aeroelastic analysis.
MGG	The G-set mass matrix exported from the structural finite element analysis.	Use only if ASET = "NO" is specified in the **'ASSIGN FEM='** Executive Control Command.
SMAA	The A-set mass matrix exported from the symmetric/asymmetric structural finite element analysis.	Use only if ASET = "YES" is specified in the **'ASSIGN FEM='** Executive Control Command.

AMAA	The A-set mass matrix exported from the anti-symmetric structural finite element analysis.	Use only if ASET = “YES” is specified in the **‘ASSIGN FEM=’** Executive Control Command.

CEND

The End of the Executive Control Section

Description: Designates the end of the Executive Control Section.

Format:

CEND

Example:

```
CEND
```

Remarks:

CEND must exist at the end the Executive Control Section.

CPU

Number of Processors

Description: Defines the number of processors for parallel computation.

Format:

CPU N

Example:

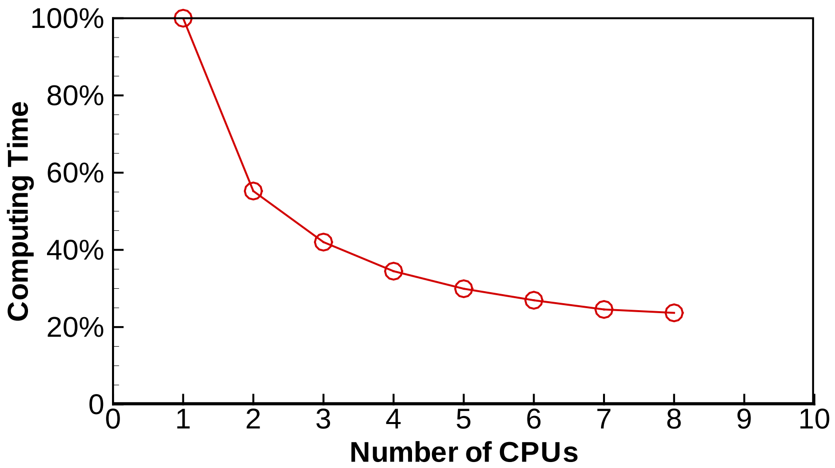
```
CPU 3
```

Describer	Meaning
N	Number of processors.

Remarks:

1. The **CPU** Executive Control Command is optional where N = 1 is the default value.

2. The following figure shows a typical savings of computational time v.s. number of processors.



DESSEN

Geometry Parameter Linking

Description: Activates the linking of a set of dependent geometry parameters.

Format:

DESSEN N

Example:

```
DESSEN 100
```

Describer	Meaning
N	Integer refers to a **DESSEN** bulk data card in the bulk data card section. (Default N=0)

Remarks:

1. The **DESSEN** Executive Control Command activates the linking of a set of geometry parameters by referring to a **DESSEN** bulk data card. If no **DESSEN** Executive Control Command is specified, then the **DESSEN** bulk data card is not activated.

DIAG

Diagnostic Output Options

Description: Request diagnostic output on special options.

Format:

DIAG $K_1, K_2, \ldots, K_i$

Example 1:

```
DIAG  1
```

Example 2:

```
DIAG  1, 3
```

Describer	Meaning
K_i	A list separated by commas of desired diagnostic.

Remarks:

1. The **DIAG** command is optional.

2. Multiple **DIAG** commands are allowed.

3. The following are the possible values for K_i and their corresponding actions.

K = 1	Turn on the dynamic memory allocation debugger.
K = 2	Print out the dynamic memory allocation history. This will generate massive output due to the large number of memory calls.
K = 3	Turn on the database file manager debugger.

MEMORY

Allocable Maximum Memory

Description: Defines the maximum memory in terms of megabytes (MB) that is allocable by ZEUS from the computers heap space memory.

Format:

MEMORY nMB

Example:

```
MEMORY 32MB
```

Remarks:

1. The **MEMORY** command is optional. If no **MEMORY** command is specified, the default value is 8000 megabytes (8000MB).

2. nMB represents an integer followed by the characters 'MB'.

3. ZEUS dynamically allocates memory within the computers heap space memory for matrix operations. For large matrices, ZEUS will occupy a large portion of the heap space for in-core matrix operations. This may degrade the performance of other jobs that are simultaneously running on the computer. To circumvent this problem, it is recommended that the **MEMORY** command be used to define the maximum allowable memory by ZEUS so that the out-of-core matrix operations are employed for large matrices. In this case, the rest of the computers heap space can be reserved for other jobs.

MPICPU **Number of Processors for MPI**

Description: Defines the number of processes to perform parallel computation using Message Passing Interface (MPI). For time-domain dynamic aeroelastic simulation, the computational mesh can be divided into several subdomains and then the time-domain Euler equation is solved on each subdomain using one MPI process. For frequency-domain linearized Euler solver, the MPI process is applied on each reduced frequency and mode pair (k-mode pair).

Format:

MPICPU N

Example:

```
MPICPU 256
```

Describer	Meaning
N	Number of processes to use.

Remarks:

1. For time-domain aeroelastic simulation modules **MLOADS**, **NANSI**, **NLFLTR**, and **ELOADS**, the computation mesh can be divided into several subdomains using the **BLKMPI** bulk data card. The Euler equation can be solved concurrently on each subdomain using the MPI parallel computation. Furthermore, within each subdomain, OpenMP can be applied using the CPU Executive Control Command to parallelize the time-accurate Euler computation.

2. For the frequency-domain linearized Euler solver, the computation to generate the unsteady aerodynamic forces (also known as Generalized Aerodynamic Forces, or GAFs) for each k-mode pair is independent of the other k-mode pairs. Therefore, MPI is used to parallelize the computation of each k-mode pair.

    a. First, the GAFs of the structure modes are computed. The number of processes is NK × NM, where NK is the number of reduced frequencies listed in the **MKAEROZ** bulk data card and NM is the number of modes used by the **FLUTTER** or **GENGAF** bulk data card (deleted modes by the MLIST entry in the **FLUTTER** or **GENGAF** bulk data cards are excluded from the analysis). Next, the GAFs of the control surface modes are computed (only active for the **GENGAF** bulk data card). The number of processes is NK × NC, where NC is the number of control surface kinematic modes referred to by the GENGAF bulk data card and defined by the **AESURFZ** and **AESLINK** bulk data cards. Finally, the GAFs of the gust mode are computed (only active for the **GENGAF** bulk data card). As there is only one gust mode, the number of MPI processes is NK.

b. The CPU Executive Control Command and the **MPICPU** Executive Control Command can coexist. In this case, the CPU Executive Control Command is first applied to solve the steady background flow solution prior to the computation by the linearized Euler solver. Then, the computation for each k-mode pair by any MPI process is parallelized by CPU number of OpenMP threads.

3. The Windows version of ZEUS is compiled against the Intel MPI library. The Intel MPI Runtime is available free of charge at https://software.intel.com/en-us/intel-mpi-library.

4. The Linux version of ZEUS is compiled against the MPICH library. For a tutorial of setting up your environment, see “Running an MPI Cluster within a LAN” at http://mpitutorial.com/tutorials/running-an-mpi-cluster-within-a-lan/
Note 1: Ensure that each hostname only maps to one IP address in the /etc/hosts file. If a hostname maps to multiple IP addresses, MPI will most likely throw errors when trying to run ZEUS if using hostnames in the machinefile.
Note 2: To enable MPI across multiple machines in a LAN, create a text file named “machinefile” in the same directory as the ZEUS input deck with a list of machine names (or IP addresses) and the number of cores allocated to each machine, and then call ZEUS with the “-mpi” switch:
```
zeus -mpi=machinefile <input deck name>
```
A sample machine file, with a total of 24 cores allocated among 4 machines corresponding to a **MPICPU** 24 command, is:
```
192.168.1.11:8
192.168.1.12:8
192.168.1.13:4
192.168.1.14:4
```
When a machine file is specified the zeus driver code calls the zeus_mpi.bin computation code with the system call “mpiexec -machinefile <machinefile> -n <MPICPU> zeus_mpi.bin”
When a machine file is not specified the zeus driver code calls the zeus_mpi.bin computation code with the system call “mpiexec -n <MPICPU> zeus_mpi.bin”
Note 3: Alternatively, if MPI is compiled with scheduler support (such as SLURM), the machinefile is not needed, as the scheduler will take care of node allocation. In this case ZEUS needs to be launched with the “-mpi” switch without specifying a machinefile:
```
zeus -mpi <input deck name>
```

5. A Linux system that employs a Portable Batch System (PBS) can be used to submit ZEUS jobs that make use of OpenMP and MPI for parallel processing. PBS job queues can be used to efficiently manage ZEUS jobs. When ZEUS jobs are submitted to the PBS job queue, the jobs wait in the queue until the requested computational resources are available. To submit a ZEUS job to a PBS queue users create PBS job scripts that are small text files containing information about what resources a job requires, including time, number of nodes, and memory. The script file also contains the commands needed to begin executing the desired computation. A sample PBS job script is shown as follows:

```
#PBS -S /bin/ksh
#PBS -N ZRUN049X3
#PBS -M myname@mycompany.com
#PBS -m abe
```

```
#PBS -q ME
#PBS -j oe
#PBS -o xxx.eo
#PBS -l select=4:ncpus=16:mpiprocs=1:mem=89gb
#PBS -l walltime=4:00:00
#
# Start Intel MPI
source $MODULESHOME/init/ksh
module purge
module load intel/mpi/64/4.1.3/049
module list
export PBS_JOBID_MODULES="${LOADEDMODULES//:/ }"
#
# START ZEUS MPI
NP=`wc -l < ${PBS_NODEFILE}`
NODES=`cat ${PBS_NODEFILE} | sort -n | uniq | wc -l`

mpdboot --verbose -n ${NODES} -r ssh -f ${PBS_NODEFILE}
mpdtrace -l
/home/user/zeus/zeus  -mpi=${PBS_NODEFILE}  xxx.inp  xxx.out
mpdallexit
```

Note that the commands for the PBS queuing system begin with #PBS

`#PBS -S /bin/ksh`

defines which type of shell script will be read

`#PBS -N ZRUN049X3`

identifies the job name

`#PBS -M myname@mycompany.com` and `#PBS -m abe`

inform PBS where to send message emails to the user. abe stands for send messages when the job aborts, begins, and ends

`#PBS -q ME`

identifies the queue name the job should be run in

`#PBS -o xxx.eo`

directs the standard output to be directed to the named file

`#PBS -l select=4:ncpus=16:mpiprocs=1:mem=89gb` and `#PBS -l walltime=4:00:00`

defines the PBS resource request (e.g., the sample job will require 4 hours, 16 processors per node and 4 nodes are requested with each launching one MPI process, etc.)

`/home/user/zeus/zeus  -mpi=${PBS_NODEFILE}  xxx.inp  xxx.out`

commands to execute ZEUS. "/home/user/zeus/zeus" is a script to setup runtime environment and invoke ZEUS itself.

The `# Start Intel MPI` section contains commands to start the Intel Message Passing Interface

The `# START ZEUS MPI` section contains the commands to launch the ZEUS job.

A sample ZEUS launching script "/home/user/zeus/zeus" is as follows:

```
#!/bin/sh
ulimit –s unlimited
export ZEUSEXE=/home/user/zeus/ZEUS
export ZLS_ZEUS=192.168.1.128
export PATH="$ZEUSEXE:$PATH"
zeus "$@"
```

The referenced ZEUSEXE and ZLS_ZEUS have to be modified based on your current setup, where ZEUSEXE is the installation directory of ZEUS.

The ZEUS driving program "zeus" will further invoke ZEUS MPI executable "zeus_mpi.bin"

The following example shows how to specify how many nodes to be used, how many cores on each node, how many MPI processes to launch on each node, and how many OpenMP threads for each MPI process:

```
#PBS –l select=32:ncpus=12:mem=23gb:mpiprocs=2:ompthreads=6
```

which means that 32 nodes are selected with 12 cores per node, and two MPI processes will be launched on each node with each MPI process spawning 6 OpenMP threads. The default value of "mpiprocs" is 1, while "ompthreads" default is equal to "ncpus". In the ZEUS input file, the Executive Control Command "CPU" should match ompthreads, and "MPICPU" should match the total number of MPI processes, i.e., 32 times 2 equal to 64 for this example. For the frequency-domain ZEUS analysis, the number of modes times the number of reduced frequencies should ideally be a multiple of the total number of MPI processes so that each MPI process will handle the same number of linearized Euler tasks.

6. ZEUS may require a larger stack size than is typically allowed by default. Hence it is recommended to call `'ulimit -s unlimited'` to raise the soft limit of the stack size in the shell before launching a ZEUS job. However, for MPI jobs that run across multiple machines, increasing the stack size limit in the shell will not increase limits for MPI threads spawned on a separate machine from the head node. In this case the default stack size of the machine must be increased. The method for doing so is distribution- specific, but the most common case is to modify `/etc/systemd/system.conf` and/or `etc/security/limits.conf.` To set a stack size soft limit of 256 MiB and an unlimited hard limit for all users, add

```
DefaultLimitSTACK=262144:infinity
```

to / `etc/systemd/system.conf` and

```
*    soft    stack    262144
*    hard    stack    infinity
```

to `/etc/security/limits.conf`.

See `` `man ulimit` ``, `` `man systemd-system.conf` ``, `` `man limits.conf` ``, and/or `` `man initscript` `` for more information.

SOLUTION

Alter the Solution Sequence

Description: Specifies a negative integer to stop the program after the execution of certain modules or to go directly to another module.

Format:

SOLUTION -n

Example:

```
SOL -1
```

Remarks:

1. The **SOLUTION** command is optional (default is 0).

2. A blank space must exist between the character string SOLUTION and the negative integer.

3. –n denotes a negative integer. Currently, only two options are available:

SOL -1 Stops the program execution after the aerodynamic geometry module is completed. This allows the user to verify the aerodynamic surface mesh and flowfield mesh before performing the unsteady aerodynamic computations.

SOL -2 Stops the program execution after the spline module is completed. This allows the user to verify the spline input before performing the unsteady aerodynamic computations.

SUBAC50 Shift the Center of Pressure

Description: Shifts the center of pressure from 25% chord to 50% chord on each panel for Mach number less than one.

Format:

SUBAC50

Example:

SUBAC50

Remarks:

1. The **SUBAC50** command is optional. If the **SUBAC50** command is not specified, the center of pressure on each panel is at 25% chord.

2. **SUBAC50** is active only for Mach number less than one. For Mach number greater than one, the center of pressure is always at 50% chord.

3. The **SUBAC50** command is applied to all modules.

$

Comment Statement

Description: Used to insert comments into the Executive Control Section.

Format:

$ followed by any characters up to column 80.

Example:

```
$   This is a test case.
```

Remarks:

1. $ must appear in the first column.

2. This command can be repeatedly used anywhere in the Executive Control Section.

3.2 CASE CONTROL SECTION

The Case Control Section allows the following Case Control Commands:

Command	Description	Remarks
BEGIN BULK	To end the Case Control Section and also to indicate the beginning of the Bulk Data Section.	Required
ECHO	Controls echo (printout) of the Bulk Data Section.	Optional
ELOADS	Invokes the transient ejection loads analysis.	Optional
FLUTTER	Invokes the flutter analysis discipline.	Optional
GENGAF	Generates the frequency-domain generalized aerodynamic forces.	Optional
GLOADS	Invokes the transient discrete gust loads analysis.	Optional
LABEL	Provides additional description of the subcase by a character string (up to 72 characters).	Optional
MLOADS	Invokes the transient maneuver loads analysis.	Optional
NANSI	Invokes the transient response analysis with nonlinear structural reduced order model.	Optional
NLFTLR	Invokes the nonlinear flutter analysis.	Optional
RETINAS	Generates the training data for aerodynamic reduced order model.	Optional
STABDRV	Generates aerodynamic stability derivatives	Optional
SUBCASE	Delimits and identifies a subcase section.	Required
SUBTITLE	Defines a subtitle of each subcase section by a character string (up to 72 characters).	Optional
TITLE	Describes the job by a character string (up to 72 characters).	Optional
TRIM	Invokes the static aeroelastic/trim analysis discipline.	Optional
$	Comment Statement.	Optional

All Case Control Commands can be written either in lower case or upper case.

The Case Control Section may contain many subcases. Each subcase is initiated by the command **SUBCASE**.

Within each subcase, only one discipline among **FLUTTER**, **GENGAF**, **TRIM**, **MLOADS**, **NANSI**, **RETINAS**, **ELOADS**, **GLOADS**, or **NLFLTR** can be selected.

TITLE and **ECHO** must appear before the subcase section.
SUBTITLE and **LABEL** must appear within the subcase section.

BEGIN BULK

The End of the Case Control Section

Description: To signify the end of the Case Control Section and the beginning of the Bulk Data Section.

Format:

BEGIN BULK

Example:

```
BEGIN BULK
```

Remarks:

BEGIN BULK must be located at the end of the Case Control Section.

ECHO

Controls Echo of the Bulk Data Section

Description: Controls the echo (printout) of the Bulk Data Section.

Format:

$$\text{ECHO} \quad = \quad \left\{ \begin{array}{c} \text{NONE} \\ \text{SORT} \\ \text{NOSORT} \end{array} \right\}$$

Example:

```
ECHO  =  NOSORT
```

Remarks:

1. **ECHO** = NONE no print.
   ECHO = SORT print out bulk data input cards in alphanumeric order.
   ECHO = NOSORT print out the unsorted bulk data input cards.

2. If no **ECHO** is specified, **ECHO** = NONE is used.

3. **ECHO** must appear before any **SUBCASE** Case Control Command.

4. No more than one **ECHO** is allowed.

5. The equal sign (' = ') is required.

ELOADS

Invokes the Transient Ejection Loads Analysis

Description: Invokes the transient ejection loads discipline by pointing to an identification number of the **ELOADS** bulk data card.

Format:

ELOADS = n

Example:

```
ELOADS  =  10
```

Remarks:

1. **ELOADS** Case Control Command must appear within a subcase section, i.e., between two **SUBCASE** Case Control Commands. The integer *n* is the identification number of the **ELOADS** bulk data card (Integer > 0). This **ELOADS** bulk data card must exist in the Bulk Data Section. **ELOADS** and n must be separated by an equal sign (' = ').

2. See Remarks 2, 3 and 4 of the **MLOADS** Case Control Command for other requirements of an **ELOADS** analysis.

FLUTTER — Invokes the Flutter Analysis Disciplines

Description: Invokes the frequency-domain flutter analysis disciplines using the G-method and k-method by referring to an identification number of the **FLUTTER** or **FLTFAST** bulk data card.

Format:

FLUTTER = n

Example:

```
FLUTTER  =  100
```

Remarks:

1. The **FLUTTER** Case Control Command must appear within a subcase section; i.e. between two SUBCASE Case Control Commands.

2. The integer *n* is the identification number of a **FLUTTER** or **FLTFAST** bulk data card (integer > 0). This **FLUTTER** bulk data card must exist in the Bulk data section.

3. **FLUTTER** and *n* must be separated by an equal sign (“=”).

GENGAF

Generates the Frequency Domain Generalized Aerodynamic Forces

Description: Invokes the **GENGAF** module to generate the frequency-domain generalized aerodynamic forces.

Format:

GENGAF = n

Example:

```
GENGAF  =  10
```

Remarks:

1. **GENGAF** Case Control Command must appear within a subcase section, i.e., between two SUBCASE Case Control Commands. The integer *n* is the identification number of the **GENGAF** bulk data card (Integer > 0). This **GENGAF** bulk data card must exist in the Bulk Data Section. **GENGAF** and *n* must be separated by an equal sign ("=").

GLOADS

Invokes the Transient Discrete Gust Loads Analysis

Description: Invokes the transient discrete gust loads discipline by pointing to an identification number of the **GLOADS** bulk data card.

Format:

GLOADS = n

Example:

```
GLOADS  =  10
```

Remarks:

1. **GLOADS** Case Control Command must appear within a subcase section, i.e., between two **SUBCASE** Case Control Commands. The integer *n* is the identification number of the **GLOADS** bulk data card (Integer > 0). This **GLOADS** bulk data card must exist in the Bulk Data Section. **GLOADS** and n must be separated by an equal sign ('=').

2. See Remarks 2, 3, and 4 of the **MLOADS** Case Control Command for other requirements of an **GLOADS** analysis.

LABEL

Provides Additional Description of a Subcase

Description: Provides additional description of a subcase by a character string up to 72 characters in length.

Format:

LABEL = ' A '

Example:

```
LABEL =  This is a test case.
```

Remarks:

1. The **LABEL** Case Control Command must appear within a subcase section.

2. ' A ' represents a character string up to 72 characters in length that allows for additional description of the subcase within which the **LABEL** Case Control Command is located.

3. Within each subcase section, only one **LABEL** Case Control Command is allowed.

4. If no **LABEL** exists in a subcase section, then the character string ' A ' is blank.

MLOADS

Invokes the Transient Maneuver Loads Analysis

Description: Invokes the transient maneuver loads discipline by pointing to an identification number of the **MLOADS** bulk data card.

Format:

MLOADS = n

Example:

```
MLOADS  =  10
```

Remarks:

1. **MLOADS** Case Control Command must appear within a subcase section, i.e., between two **SUBCASE** Case Control Commands. The integer n is the identification number of the **MLOADS** bulk data card (Integer > 0). This **MLOADS** bulk data card must exist in the Bulk Data Section. **MLOADS** and n must be separated by an equal sign (' = ').

2. When the forces at structural grid points are computed (see **MLDPRNT** bulk data card), a matrix called ′SKGH′ for symmetric (or asymmetric) boundary condition and a matrix called ′AKGH′ for anti-symmetric boundary condition must be input by the **'ASSIGN MATRIX='** Executive Control Command. These matrices are the product of the G-set stiffness matrix and the G-set modal matrix of the structural finite element model (G-set is defined as 6 x number of structural finite element grid points). The equations to obtain these matrices are shown as follows:

$$[SKGH] = [KGG][PHG]_s$$

$$[AKGH] = [KGG][PHG]_a$$

where

[KGG] is the stiffness matrix of the G-set d.o.f.

$[PHG]_s$ is the symmetric modal matrix of the G-set d.o.f.

and

$[PHG]_a$ is the anti-symmetric modal matrix of the G-set d.o.f.

[SKGH] and [AKGH] are used to compute the forces at the structural finite element grid points.

The following example shows the MSC.NASTRAN DMAP alter statements that generate these matrices by the NASTRAN/OUTPUT4 module.

```
ASSIGN OUTPUT4='demo1.kgh',UNIT=12,FORM=FORMATTED
SOL 103
COMPILE SEMODES SOUIN=MSCSOU LIST NOREF  $
ALTER 177 $
MATGEN EQEXINS/INTEXT/9//LUSETS $ GENERATE EXTERNAL SEQUENCE MATRIX
MPYAD KGG,PHG,/MKHINT $ MKHINT IS THE KGH IN INTERNAL SEQUENCE
MPYAD INTEXT,MKHINT,/KGH/1 $ TRANSFORM MKHINT TO EXTERNAL SEQUENCE
OUTPUT4 KGH//-1/12/2 $ OUTPUT KGH TO UNIT=12 IN demo1.mgh
ENDALTER
CEND
```

Once the file ′demo1.kgh′ is generated by NASTRAN, it can be directly input into ZEUS by the **‘ASSIGN MATRIX=**’ Executive Control Command.

- The format for symmetric or asymmetric boundary condition is:
  ASSIGN MATRIX = ′demo1.kgh′, MNAME = ′SKGH′, FORM = ′FORMAT′

- The format for anti-symmetric boundary condition is:
  ASSIGN MATRIX = ′demo1.kgh′, MNAME = ′AKGH′, FORM = ′FORMAT′

Note: The name of the matrix is defined as ′KGH′ in the NASTRAN DMAP alter statements. However, in the **‘ASSIGN MATRIX=’** Executive Control Command, it is replaced by MNAME = ′SKGH′ for the symmetric boundary condition and MNAME = ′AKGH′ for the anti-symmetric boundary condition.

3. If the matrices [SMGH] and/or [AMGH] are imported, importing the matrices [SKGH] and/or [AKGH] is no longer necessary. ZEUS will automatically compute the [SKGH] and/or [AKGH] matrices by the following equations:

$$\left[\mathrm{SKGH}\right] = [\omega_f^{\,2}]\left[\mathrm{SMGH}\right]$$

$$\left[\mathrm{AKGH}\right] = [\omega_f^{\,2}]\left[\mathrm{AMGH}\right]$$

where $[\omega_f^{\,2}]$ are the natural frequencies that are imported by the **‘ASSIGN FEM=’** Executive Control Command.

4. In fact, all SMGH, AMGH, SKGH, and AKGH can be obtained by importing a single G-set mass matrix (called MGG) using the **‘ASSIGN MATRIX=’** Executive Control Command. ZEUS will automatically perform the matrix multiplications to compute SMGH, AMGH, SKGH, and AKGH. Therefore, importing MGG into ZEUS is highly recommended. This can be achieved by the following Executive Control Command: **‘ASSIGN MATRIX=’** “demo1.mgg”, **MNAME** = “MGG”.

ASSIGN MATRIX = ′demo1.MGG′, MNAME = ′MGG′, FORM = ′FORMAT′

The MGG matrix can be obtained from NASTRAN using the following ALTER statements.

```
ASSIGN OUTPUT4='demo1.mgg',UNIT=12,FORM=FOAMAT
SOL 103
COMPILE SEMODES SOUIN=MSCSOU LIST NOREF  $
ALTER 'STRAIN ENERGY'
MATGEN EQEXINS/INTEXT/9//LUSETS $ GENERATE EXTERNAL SEQUENCE MATRIX
MPYAD INTEXT,MGG,/MGGT/1        $ TRANSFORM MGG TO EXTERNAL SEQUENCE
OUTPUT4 MGGT//-1/12/2           $ OUTPUT MGG TO UNIT=12 IN demo1.mgg
ENDALTER
CEND
```

5. If ASET = “YES” is specified in the **‘ASSIGN FEM=’** Executive Control Command, [MGG] should be replaced by [SMAA] for the symmetric/asymmetric analysis and [AMAA] for the anti-symmetric analysis, where [SMAA] and [AMAA] are the A-set mass matrices of the symmetric/asymmetric and anti-symmetric structural model, respectively.

NANSI

Invokes the Transient Response Analysis with Nonlinear Structural Reduced Order Model

Description: Invokes the transient response analysis with an externally imported nonlinear structural Reduced Order Model (ROM) for Nonlinear Aerodynamic and Nonlinear Structural Interaction (NANSI) analysis.

Format:

NANSI = n

Example:

```
NANSI =  10
```

Remarks:

1. **NANSI** Case Control Command must appear within a subcase section, i.e., between two SUBCASE Case Control Commands. The integer *n* is the identification number of the **NANSI** bulk data card (Integer > 0). This **NANSI** bulk data card must exist in the Bulk Data Section. **NANSI** and n must be separated by an equal sign ("=").

2. See Remarks 2, 3 and 4 of the **MLOADS** Case Control Command for other requirements of a **NANSI** analysis.

NLFLTR

Invokes the Nonlinear Flutter Analysis

Description: Invokes the nonlinear flutter analysis discipline by pointing to an identification number of the **NLFLTR** bulk data card.

Format:

NLFLTR = n

Example:

```
NLFLTR =  100
```

Remarks:

1. **NLFLTR** Case Control Command must appear within a subcase section, i.e., between two **SUBCASE** Case Control Commands. The integer *n* is the identification number of the **NLFLTR** bulk data card (Integer > 0). This **NLFLTR** bulk data card must exist in the Bulk Data Section. **NLFLTR** and *n* must be separated by an equal sign (‘=’)

2. To define the nonlinear structural matrices as a function of the nonlinear parameters, the user must import those structural matrices by the **‘ASSIGN MATRIX=’** Executive Control Commands. See the **NLSYSM** bulk data for the description of these structural matrices.

RETINAS

Training Data for Aerodynamic Reduced Order Model

Description: Invokes the **RETINAS** module to compute the training data for the generation of an aerodynamic Reduced Order Model (ROM)

Format:

RETINAS = n

Example:

```
RETINAS =  10
```

Remarks:

1. **RETINAS** Case Control Command must appear within a subcase section, i.e., between two **SUBCASE** Case Control Commands. The integer *n* is the identification number of the **RETINAS** bulk data card (Integer > 0). This **RETINAS** bulk data card must exist in the Bulk Data Section. **RETINAS** and n must be separated by an equal sign ("=").

STABDRV

Aerodynamic stability Derivatives

Description: Generates aerodynamic stability derivatives.

Format:

STABDRV = n

Example:

```
STABDRV =  2
```

Remarks:

1. ‘n’ is an integer that refers to a **STABDRV** bulk data card.

SUBCASE

Delimits and Identify a Subcase Section

Description: To start a subcase section and assign an identification number to the subcase.

Format:

SUBCASE = n

Example:

```
SUBCASE =  2
```

Remarks:

2. The Case Control Section can contain many subcase sections. Each subcase section must be started by a **SUBCASE** = n Case Control Command.

3. 'n' is an integer that assigns an identification number to the subcase section. Among all **SUBCASE** Case Control Commands, *n* must be unique.

3. Within each subcase section, only one discipline (e.g., **FLUTTER**, **ASE**, **TRIM, MLOADS**, **ELOADS**, or **GLOADS** Case Control Command) is allowed.

4. **SUBTITLE** and **LABEL** Case Control Commands must be located within each subcase section.

SUBTITLE

Defines a Subtitle of Each Subcase Section

Description: Defines a subtitle of each subcase section by a character string up to 72 characters in length.

Format:

SUBTITLE = ' A '

Example:

```
SUBTITLE =  Flutter Analysis at M = 0.8
```

Remarks:

1. The **SUBTITLE** Case Control Command must appear within a subcase section.

2. ' A ' represents a character string up to 72 characters in length that allows for additional description of the subcase section.

3. Within each subcase section, only one **SUBTITLE** Case Control Command is allowed.

4. If no **SUBTITLE** exists in a subcase section, then the character string ' A ' is blank.

TITLE

Title of the Job

Description: Provides the title of the job by a character string up to 72 characters in length.

Format:

TITLE = ' A '

Example:

```
TITLE =  ZEUS Analysis of a Demo Case
```

Remarks:

1. Only one **TITLE** Case Control Command is allowed in the entire Case Control Section. **TITLE** must appear before the **SUBCASE** Case Control Command.

2. ' A ' represents a character string up to 72 characters in length to provide the title of the job.

3. If no **TITLE** exists in a subcase section, then the character string ' A ' is blank.

TRIM

Invokes the Static Aeroelastic/Trim Analysis Discipline

Description: Invokes the static aeroelastic/trim analysis discipline by referring to an identification number of the **TRIM** bulk data card.

Format:

TRIM = n

Example:

```
TRIM = 103
```

Remarks:

1. The **TRIM** Case Control Command must appear within a subcase section, i.e., between two **SUBCASE** Case Control Commands.

2. The integer n is the identification number of the **TRIM** bulk data card (Integer > 0). This **TRIM** bulk data card must exist in the Bulk Data Section.

3. **TRIM** and n must be separated by an equal sign (' = ').

4. For a symmetric trim system (trim system involving only the longitudinal d.o.f.), the free vibration solution of the finite element model with symmetric boundary condition must be imported by the **'ASSIGN FEM='** Executive Control Command with BOUNDARY = 'SYM'. For an anti-symmetric trim system (trim system involving only the lateral d.o.f.), the free vibration solution of the finite element model with anti-symmetric boundary condition must be imported by the **'ASSIGN FEM='** Executive Control Command with BOUNDARY = 'ANTI'. For an asymmetric trim system (trim system involving both the longitudinal and the lateral d.o.f.), both free vibration solutions must be imported. However, for an asymmetric configuration (entry XZSYM = "NO" in the **AEROZ** bulk data card), only one **'ASSIGN FEM='** Executive Control Command with BOUNDARY = 'ASYM' is required.

5. Computing the distributed inertial loads resulting from the trim system requires a matrix called 'SMGH' for symmetric (or asymmetric) structural boundary condition and a matrix called 'AMGH' for anti-symmetric structural boundary condition to be inputted by the **'ASSIGN MATRIX='** Executive Control Command. For the definitions of 'SMGH' and 'AMGH'.

   Note: 'SMGH' or 'AMGH' is required only if the structural finite element contains rigid body d.o.f.; i.e. the SUPORT entry in the **'ASSIGN FEM='** Executive Control Command specifies a non-zero integer.

$ — Comment Statement

Description: Used to insert comments into the Case Control Section.

Format:

$ followed by any characters up to column 80.

Example:

```
$  The next command is FLUTTER
```

Remarks:

1. **$** must appear in the first column.

2. This command can be repeatedly used anywhere in the Case Control Section.

Chapter 4

ZEUS BULK DATA SECTIONS

The Bulk Data Section begins right after the **BEGIN BULK** Case Control Command and ends at a bulk data card **ENDDATA**. The Bulk Data Section contains bulk data cards that specify:

- The geometry of the aerodynamic model to generate the surface mesh.
- The parameters to control the density of the flowfield mesh.
- Spline for displacement and force transferal between the structural finite element grid points and aerodynamic boxes.
- The flight condition for aerodynamic force generation.
- Disciplines (FLUTTER, static aeroelastic/TRIM, MLOADS, ELOADS, GENGAF, GLOADS, or NLFLTR) to be analyzed.
- Other miscellaneous inputs.

4.1 FORMAT OF BULK DATA CARDS

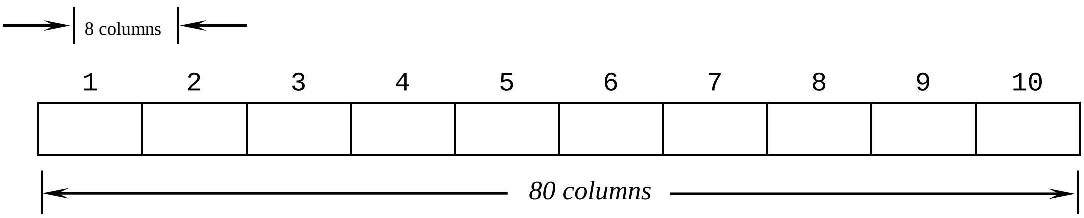
The format of the bulk data cards is identical to that in NASTRAN except for the so-called ‘Large Field Entry’ (i.e., 16 characters wide) which is not allowed (for definition of ‘Large Field Entry’, please see a NASTRAN User’s Manual).

The bulk data card contains ten fields per input data entry. The first field contains the character name of the bulk data card (**CAERO7**, **BODY7**, etc.). Fields two through nine contain data input information for the bulk data entry. The tenth field never contains data – it is reserved for a continuation card, if applicable.

Two types of format are allowed for each bulk data card; the fixed format and free format.

Fixed Format

Fixed format separates a bulk data card into ten equal fields of eight columns each.



A typical bulk data card is shown as follows:

1	2	3	4	5	6	7	8	9	10
PANLST1	SETID	MACROID	BOX1	BOX2					

name of
the Bulk Data Card

entries

not used

Field 10 is used
only for optional
continuation
information.

Example:

PANLST1	100	111	111	118					

The name of the bulk data card <u>must appear in the first field and start from the first column</u>. Three possible types of data can be specified for bulk data entries and are described as follows.

Integer	Numerical value with no decimal point.
Real	Numerical value with a decimal point.
Character	Can be any alphanumeric string.

Real numbers may be specified in various ways. The following examples are all acceptable:

```
3.14        3.14E+00    .314+01
.314+1      .314E+1     31.4-01
```

The above example shows that each bulk data card allows eight entries to be specified from fields two to field nine. If there are more than eight entries required for a bulk data card, the so-called "continuation label" is required in the tenth field and more than one input cards are needed. The additional input cards are called "continuation lines". A typical example of this kind of bulk data card is shown as follows:

AEFACT	SID	D1	D2	D3	D4	D5	D6	D7	CONT

"CONT" can be any alphanumeric string including blanks.

CONT	D8	D9	-etc-						

Example:

AEFACT	100	0.0	0.2	0.3	0.4	0.5	0.6	0.7	+A

+A	0.8	0.9							

There are several major differences between ZEUS and NASTRAN regarding the treatment of continuation lines. ZEUS has the following restrictions:

- The continuation lines must follow their associated bulk data card. No other bulk data cards can be inserted between continuation lines except a comment (**$**).
- If continuation label is blank, no other bulk data cards can be inserted between continuation lines including a comment (**$**).
- Duplicate continuation labels may be used. For example, the following bulk data cards with continuation lines are acceptable:

Example 1:

AEFACT	100	0.0	0.2	0.3	0.4	0.5	0.6	0.7	+A

+A	0.8	0.9	1.0	1.1	1.2	1.3	1.4	1.5	+A

+A	1.6								

Example 2:

AEFACT	100	0.0	0.2	0.3	0.4	0.5	0.6	0.7	+A

+A	0.8	0.9	1.0	1.1	1.2	1.3	1.4	1.5	+B

+B	1.6								

Free Format

In free format, the data entries must be separated by commas (separation by a blank is not allowed). The following shows the **AEFACT** bulk data card with one continuation line in free format:

Indicates an empty field; default value will be used.

```
AEFACT,  100,    ,  0.2,  0.3,  0.4,  0.5,  0.6,  0.7,  +A
+A,  0.8,  0.9
```

There are several rules for free format:

- Free format data must start in column 1.
- Each data entry (for all three types of data: integer, real and character) cannot exceed eight columns.
- To skip one entry, use two commas in succession (and so on).
- Fixed format and free format can be mixed. For example, the following is acceptable:

AEFACT	100	0.0	0.2	0.3	0.4	0.5	0.6	0.7	+A

```
+A,  0.8,  0.9
```

4.2 BULK DATA CARDS SUMMARY AND INTERRELATIONSHIPS

This section contains a summary of all the bulk data cards in the ZEUS system separated into logically related groups according to the ZEUS engineering modules.

4.2.1 AERODYNAMIC MODEL INPUT

The bulk data cards used to define the aerodynamic model are listed in the following table:

Name	Description	Remarks
ACOORD	Aerodynamic local coordinate system definition.	Optional
AEROZ	Basic aerodynamic reference parameters.	Required
AESURFZ	Aerodynamic control surface definition for static aeroelastic/trim analysis or transient maneuver loads analysis.	Optional
BLKLAY	BLOCK layer number for overset strategy.	Optional
BLKSEQ	BLOCK ID sequence for overset strategy.	Recommended for overset mesh
BLOCK	Generation of a rectangular block of mesh.	At least one **BLOCK, BLOCKT** or **BLOCK1** is Required.
BLOCKT	Generates a one-block of mesh to model a T-tail component.	At least one **BLOCK, BLOCKT** or **BLOCK1** is Required.
BLOCK1	Generation of a body-fitted mesh for the **BODY7** macroelement.	At least one **BLOCK, BLOCKT** or **BLOCK1** is Required.

BODY7	Aerodynamic body-like component geometry input.	Optional, but at least one **BODY7** or **CAERO7** is required.
CAERO7	Aerodynamic wing-like component geometry input.	Optional, but at least one **BODY7** or **CAERO7** is required.
FOILSEC	Defines an NACA-series type of airfoil section.	Optional
GAP	Adding more grid points or enhancing the quality of the mesh on the projected X-Y plane generated by the **BLOCK** bulk data card	Optional
GAPZ	Same as **GAP** except for the mesh on the projected Y-Z plane.	Optional
GAP1	Alternative form of **GAP**.	Optional
MESHPRM	Parameters for mesh generation.	Optional
MESHSAV	Save or retrieve the flowfield mesh.	Optional
PAFOIL7	Defines airfoil cross sections at the root and tip of a **CAERO7**.	Optional
PAFOIL8	Alternative form of the **PAFOIL7** bulk data card.	Optional
PBODY7	Aerodynamic body inlet aerodynamic box definition.	Optional
SEGMESH	Defines a mesh grid system for a body segment.	Optional
YZONEY	Defines a *y* location of a Y-zone.	Optional

The aerodynamic model input consists of two parts; the surface mesh input and the automated mesh generation. Three bulk data cards are used to define the surface mesh, namely **CAERO7** for lifting-surface-like components such as wings, tails, pylons, launchers, store fins, etc.; **BODY7** for the body-like components such as fuselage, engine stores, etc.; and **GAP/GAPZ/GAP1** for a fictitious surface to add more grid points or enhance the quality of the flowfield mesh. All **CAERO7**, **BODY7** and **GAP/GAPZ/GAP1** bulk data cards are referred to by the **BLOCK/BLOCKT** bulk data card that uses the surface mesh input to automatically generate a rectangular block of mesh.

The **BLOCK1** bulk data card can be also used to generate a body-fitted flowfield mesh for the modeling of a **BODY7** macroelement. If multiple **BLOCK/BLOCKT/BLOCK1** bulk data cards are used, the overset mesh scheme is automatically activated to handle the flow communication among blocks of mesh. To verify the quality of the mesh, it is recommended that the user use the **PLTAERO** bulk data card with entry MESH = “YES” to generate a graphic file for visual verification of the mesh. Control surfaces such as flaps, ailerons, and rudders are defined by the **AESURFZ**, **AESLINK**, **PZTMODE**, and **GRIDFRC** bulk data cards. Figure 4.1 presents a flow chart showing the interrelationship of the bulk data cards for the aerodynamic model input.

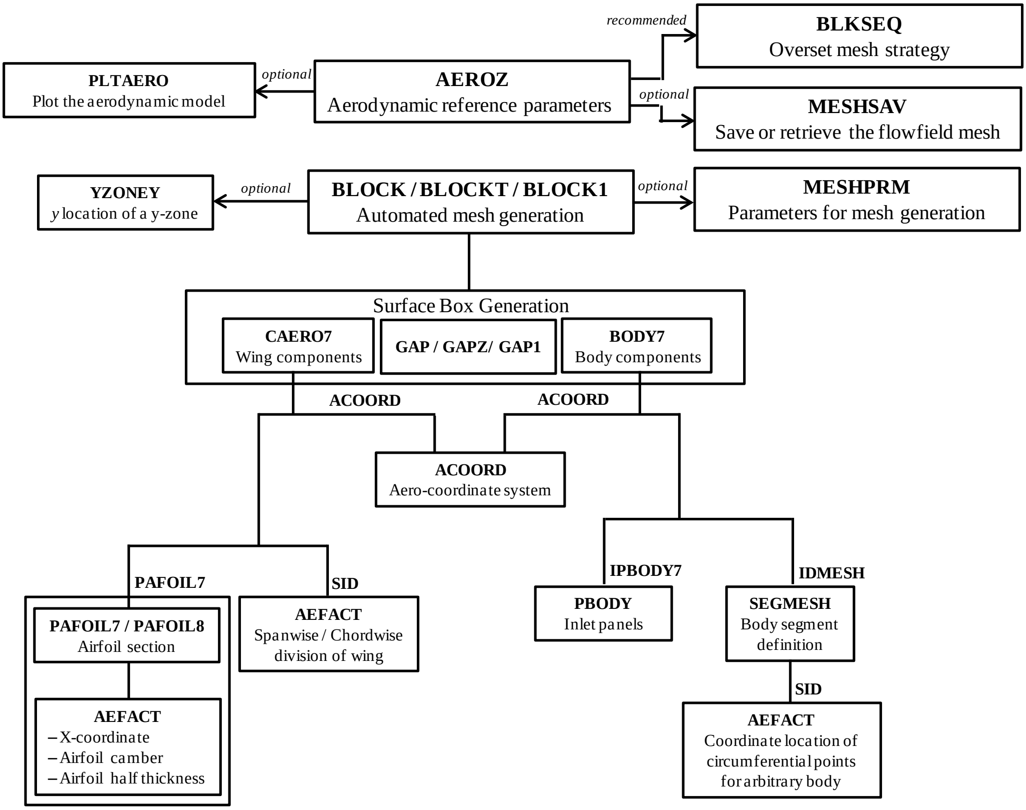


Figure 4.1 Bulk Data Interrelationship for Aerodynamic Geometry Input

4.2.2 ZONA'S DESIGN VARIABLE LINKING SCHEME (ZLINK)

ZONA's Design Variable Linking Scheme (ZLINK) module is included within the ZEUS software. This module is shared by many of ZONA's software (e.g., ZAERO and ZMORPH) and its bulk data input is, therefore, completely portable between the applications. This feature is very convenient, for example, when performing an optimization process where a specified set of design variables (e.g., those defining wing span, leading edge sweep angle, etc.) saved within a single file can be directly read in and used by all of these codes. Such an external file is included within each of these software bulk data input decks via the **INCLUDE** bulk data card. In this fashion, all of the standard input for these codes remains unchanged and only the single file containing the design variables is updated by the optimizer during each stage of the optimization process. Identical changes to the models of the different codes are then automatically handled by ZLINK within each of the applications.

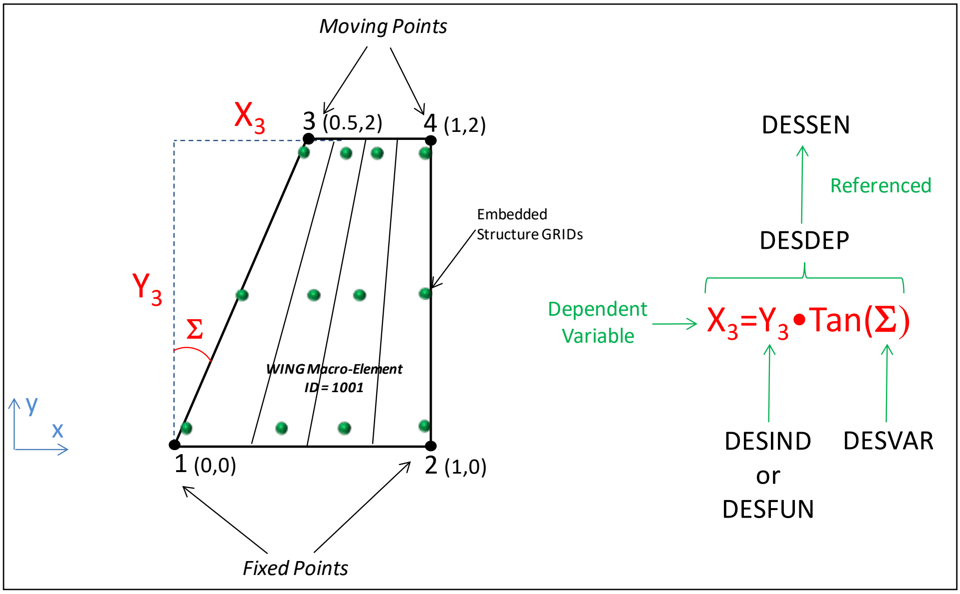
NOTE: <u>ZLINK is used to describe, and is completely responsible for, the geometrical shape change (i.e., morph) that takes place</u>. It allows for the input of design variables, independent and dependent variable definitions, and arbitrary general functions of virtually any complexity. ZLINK is very general and flexible in the sense that user input equations can be used to drive the morph. Based on final values computed for the dependent variables, the action of ZLINK is to internally modify existing bulk data input entries. ZEUS bulk data cards whose entries are modified via ZLINK to establish a geometric perturbation are: **CAERO7**, **BODY7**, **AEFACT**, **FOILSEC**, and **CORD2R**.

The five bulk data cards used to setup the ZLINK input are shown in the following table along with a description of each card. At a minimum, a single **DESDEP** and **DESSEN** bulk data card are required to execute a geometrical shape change within ZEUS.

Bulk Data Card	Description	Remarks
DESDEP	Defines a dependent geometric parameter.	*Required*
DESFUN	Defines a function to link the dependent geometric parameters with the independent geometric parameters and the shape design variables.	*Optional*
DESIND	Defines an independent geometric parameter.	*Optional*
DESSEN	Activates the shape design variables.	*Required*
DESVAR	Defines a shape design variable.	*Optional*

The following figure demonstrates how these bulk data cards can be used to define a simple shape change. Suppose we have a planar wing with a swept leading edge defined by a **CAERO7** macro-element with the points 1-4 located as shown and that we wish to change the leading edge sweep angle to any value defined by a design variable. For this case, points 1 and 2 remain fixed and points 3 and 4 move relative to the input sweep angle. First, we define a design variable, say SIGMA, and set its value to 35 degrees via the following **DESVAR** bulk data card with ID=10:

```
DESVAR   10      SIGMA   35.0
```



Next, we need to define the coordinate Y3 to be able to evaluate the equation shown in the figure; namely,

$$X3 = Y3 \times tan(\Sigma)$$

There are two ways Y3 can be defined; explicitly as a value (of 2.0) via the **DESFUN** bulk data card as:

```
DESFUN  20
        (       2.0                                             )
```

or by assigning an independent variable; such as the Y-value of point 3; where **CAERO7** with ID=1001 is the label of the wing macro-element:

```
DESIND  20      YTVAL   CAERO7  1001    YTL
```

A single **DESDEP** bulk data card is then used to alter the tip leading edge of the **CAERO7** macro-element X3 value:

```
DESDEP  10      SWEEP   CAERO7  1001    XTL     0
        1.0     20      TAN     0.01745 10
```

where XTL is the entry of the **CAERO7** bulk data card to be altered, 0 is the i-th index (as there is no repeating entry on the WING bulk data card), 20 is the ID of the **DESFUN** or **DESIND** bulk data card, and 10 is the ID of the **DESVAR** with LABEL=SIGMA bulk data card. The factor of 0.01745 is used to convert the degree input of SIGMA to radians.

Finally, referencing the **DESDEP** via a **DESSEN** card will trigger ZEUS to perform modification of the aerodynamic model defined by the **DESDEP** card(s). Note that changing the value of SIGMA on the **DESVAR** card and re-running ZEUS will generate a new aerodynamic model based on the newly updated **CAERO7** macro-element.

4.2.3 SPLINE INPUT (SPLINE MODULE)

The bulk data cards of the spline input define the interconnection between the aerodynamic model and the structural finite element model for displacement and force transferal. Specifically, the spline input generates a spline matrix that “attaches” every aerodynamic surface box to a set of structural finite element grid points. Here, the aerodynamic surface boxes represent the discretized aerodynamic model that is defined by the aerodynamic geometry input. The structural finite element grid points are imported through the external file specified in the **‘ASSIGN FEM=’** Executive Control Command.

The following table presents the bulk data cards for the spline input:

Name	Description	Remarks
ATTACH	Defines a rigid body connection between aerodynamic boxes and structural finite element grid points.	Optional
PANLST1	Defines a set of aerodynamic boxes (region defined by two aerodynamic box identification numbers).	Optional
PANLST2	Defines a set of aerodynamic boxes (region defined by individual aerodynamic box identification numbers).	Optional
PANLST3	Defines a set of aerodynamic boxes (region defined by the entry LABEL in the **CAERO7** or **BODY7** bulk data card).	Optional, but all aerodynamic box identification numbers must be uniquely and completely listed in **PANLST1**, **PANLST2** and/or **PANLST3**.
SET1	Defines a list of structural finite element grid points for spline.	Optional
SET2	Defines the aerodynamic macroelements in term of spanwise and chordwise points (zone) for spline.	Optional
SETADD	Defines a set of integers as a union integer set defined on the **SET1** bulk data cards.	Optional
SPLINE0	Imposes zero-displacement condition on aerodynamic boxes.	Optional
SPLINE1	Defines a surface spline method (Infinite Plate Spline method) for **CAERO7**.	Optional
SPLINE2	Defines a beam spline method for **CAERO7** / **BODY7**.	Optional
SPLINE3	Defines a 3-D spline (Thin Plate Spline method) for **CAERO7** / **BODY7**.	Optional
SPLINEF	Spline matrix for force mapping.	Optional
SPLINEM	Saves or retrieves the spline matrix.	Optional
SPLNDOF	Changes spline d.o.f. of structural model.	Optional

It should be noted that all identification numbers of the aerodynamic boxes must be uniquely and completely specified in the **PANLST1**, **PANLST2** and/or **PANLST3** bulk data cards. Violation of this condition results in fatal errors as following:

```
FATAL ERROR: AERODYNAMIC BOX WITH ID = ' xxxx ' IS NOT ATTACHED TO
FEM MODEL
```

This indicates that the aerodynamic box with identification number = ' xxxx ' is not specified in the **PANLST1, PANLST2** and/or **PANLST3** bulk data cards.

```
FATAL ERROR: AERODYNAMIC BOX WITH ID = ' xxxx ' = HAS BEEN SPLINED
MORE THAN ONCE
```

This indicates that the aerodynamic box with identification number = ' xxxx ' is repetitively specified in the **PANLST1**, **PANLST2** and/or **PANLST3** bulk data cards.

Figure 4.2 depicts the interrelationships of the bulk data cards for spline:

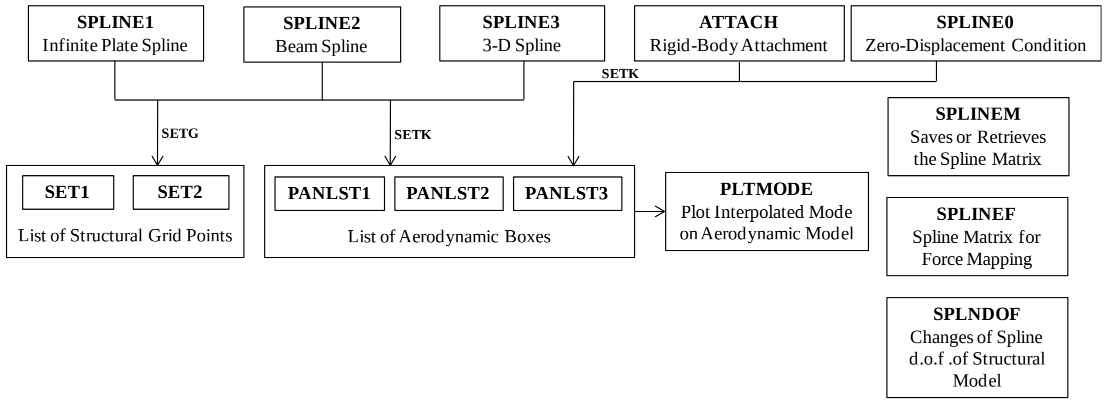


Figure 4.2 Bulk Data Interrelationship for Spline

4.2.4 INPUT FOR FLIGHT CONDITIONS

Input parameters for flight condition include Mach numbers, angle of attack, side slip angle, pitch rate, roll rate, and yaw rate. For viscous analysis, Reynolds numbers are also a required input. The bulk data cards for those input parameters are listed as follows.

Name	Description	Remarks
BLKMPI	Block Assignment for MPI Load Balance Strategy	Optional
FINDMPI	Automated Load Balancing for MPI	Optional
KEXPAN	K-expansion linearized unsteady aerodynamics at low reduced frequencies.	Optional
MKAEROZ	Defines Mach number and reduced frequencies.	Required
MKBLOCK	Assign different parameters for Euler solver in different clocks of mesh.	Optional
MKPARAL	Alter the default values of parameters used in the Linearized Euler-Solver module.	Optional
MKPARAM	Parameters for the Euler solver module.	Optional
TRIMFLT	Defines angle of attack, side slip angle, pitch rate, roll rate, and yaw rate of the steady mean flow.	Optional

VISCOUS	Defines parameter for viscous analysis.	Optional
STFLOW	Save or retrieve the steady flowfield solution.	Optional

Note that all analyses such as FLUTTER, TRM, MLOADS, GLOADS, ELOADS, GENGAF, and NLFLTR require the presence of the **MKAEROZ** bulk data card. The interrelationship of those bulk data cards is shown in Figure 4.3.

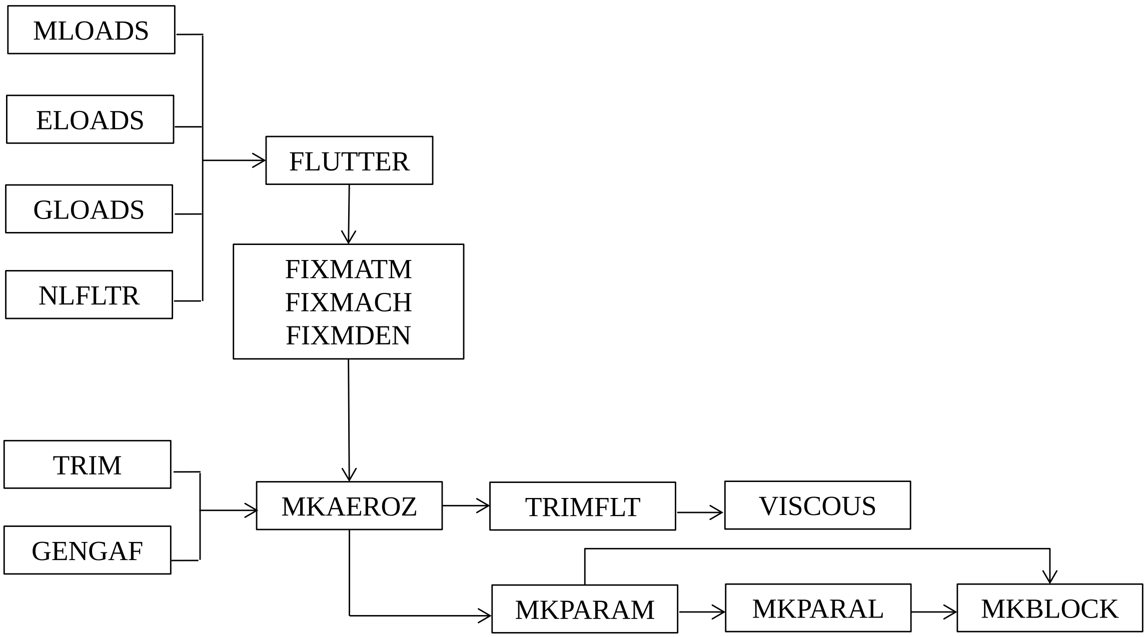


Figure 4.3 Bulk Data Interrelationship for Flight Condition Input

4.2.5 INPUT FOR FLUTTER ANALYSIS (FLUTTER MODULES)

The flutter analysis is "triggered" by a **FLUTTER** Case Control Command. The bulk data cards for flutter analysis are listed in the following table:

Name	Description	Remarks
ATMOS	Defines an atmosphere table for matched point flutter analysis.	Optional
CONM1	Defines a 6 × 6 matrix at a structural grid point for flutter analysis with mass perturbation.	Optional
CONM2	Defines a concentrated mass element at a structural grid point which can be used to modify the generalized mass matrix.	Optional
CONM2L or CONM2*	Large field input for **CONM2**.	Optional
CONMLST	Defines a list of **CONM1/CONM2** bulk data cards for mass	Optional

	perturbation.	
FIXMATM	Matched point flutter analysis at a fixed Mach number and an atmosphere table for various altitudes.	Optional
FIXMACH	Non-matched point flutter analysis at a fixed Mach number for various velocity and density pairs.	Optional
FIXMDEN	Non-matched point flutter analysis at a fixed Mach number and density pair for various velocities.	Optional
FLTFAST	Same as the **FLUTTER** bulk data card but using the composite sinusoidal excitation technique to generate the frequency-domain generalized aerodynamic forces.	Optional
FLTSEN	Flutter sensitivity analysis.	Optional
FLUTTER	Performs the frequency-domain flutter analysis by invoking the frequency-domain aerodynamic computation of the Euler-Solver module using the sinusoidal excitation technique.	Optional
FLUTTF	Defines a second order linear system *f(s)* such that: $f(s) = s^2 [M] + s[C] + [K]$ where s is the Laplace parameter.	Optional
TABDMP1	Defines modal damping as a tabular function of natural frequency.	Optional

The FLUTTER module first generates the frequency-domain generalized aerodynamic forces by invoking the frequency-domain unsteady aerodynamic computation of the Euler-Solver module. Prior to the frequency-domain unsteady aerodynamic computation, the Euler-Solver module always performs a steady aerodynamic computation on the rigid structure to obtain a converged steady aerodynamic solution from which the unsteady computation starts. The steady flow solution can be saved via the **STFLOW** bulk data card. Three methods are available to generate the frequency-domain generalized aerodynamic forces (GAF); (1) the linearized Euler solver, (2) sinusoidal excitation technique, and (3) the composite sinusoidal excitation technique. The linearized Euler solver solves the GAF directly in the frequency domain and is activated by the entry METHOD=3 in the **MKPARAM** bulk data card. The sinusoidal excitation technique is activated by the **FLUTTER** bulk data card (unless the linearized Euler solver is selected) that excites the structure at a given reduced frequency and transforms the time-domain response to frequency domain using Fourier transform. The composite sinusoidal excitation technique is activated by the **FLTFAST** bulk data card that excites the structure by a composite of multiple reduced frequencies input. The GAF can be saved by specifying the entry SAVE="SAVE" and retrieved back by SAVE="ACQUIRE" in the **MKAEROZ** bulk data card. Once the GAF is obtained, the FLUTTER module performs a flutter analysis using the *g-method* flutter solution technique to compute the damping and frequency of each structural mode. The flutter results are presented in terms of the v-g and v-f diagrams in which the flutter boundary is obtained when the damping curve crosses zero. The data of the v-g and v-f diagrams can be saved via the **PLTVG** bulk data card. Flutter mode animation data can be generated using the **PLTFLUT** bulk data card.

It should be noted that the generation of the frequency-domain generalized aerodynamic forces could be computationally time consuming. Another way to obtain the flutter boundary is to use the MLOADS module that performs the time-domain transient response analysis. The user can examine the damping of the transient responses at various flight conditions to determine the flutter boundary.

Various functions are incorporated in the flutter discipline. These functions include:

* Matched point flutter analysis by specifying **FIXMATM**. The matched point is performed based on the built-in standard atmosphere table or user-defined atmosphere table (by **ATMOS** bulk data card). The matched point flutter analysis is computed by the *g-method* only.
* Non-matched point flutter analysis by specifying **FIXMACH** or **FIXMDEN** using the *g-method*. For **FIXMDEN**, flutter analysis using the *K-method* and divergence speed analysis are also performed.
* Mass perturbation by specifying **CONMLST** to modify the mass matrix without returning to the structural finite element analysis.
* Omitting structural modes to reduce the size of the flutter equation.
* Introducing modal damping by **TABDMP1** bulk data card.
* Flutter mode tracking capability to identify the importance of each structural mode to the flutter mode.
* Flutter sensitivity analysis by defining the **DMIG/DMI** bulk data cards as the design variables.

The interrelationship of the bulk data cards for flutter analysis is depicted in Figure 4.4

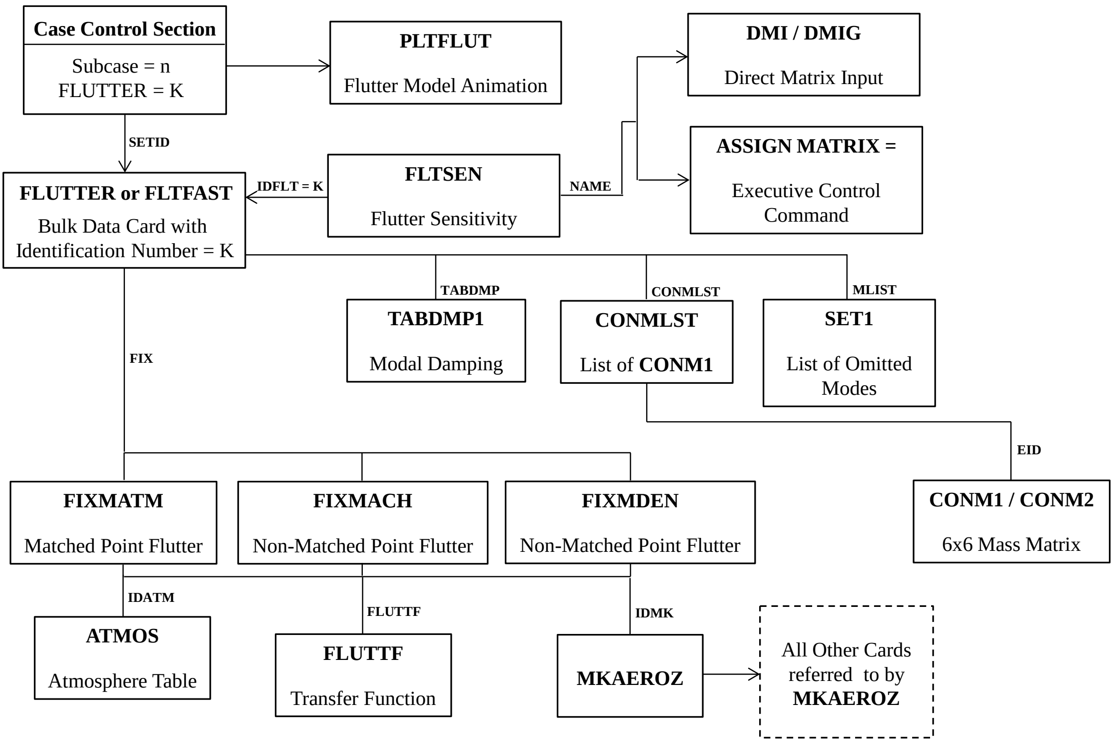


Figure 4.4 Bulk Data Interrelationship for Flutter Analysis

4.2.6 INPUT FOR STATIC AEROELASTIC/TRIM ANALYSIS (TRIM MODULE)

The function of the static aeroelastic/trim analysis is to solve the trim system and compute the flight loads. The solution of the trim system requires the balance of the inertial loads due to the accelerations of the trim degrees of freedom (d.o.f) ($\ddot{X}, \ddot{Y}, \ddot{Z}, \dot{p}, \dot{q}$, and $\dot{r}$; see Figure 4.5) and the aerodynamic loads generated by the trim variables (α, β, p, q, r, control surface deflections, etc.).

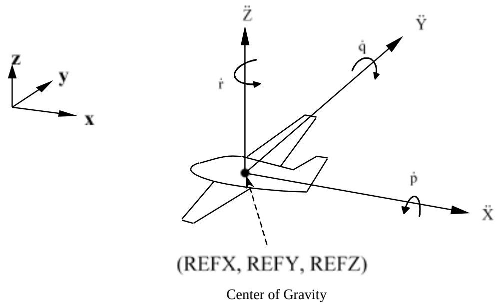


Figure 4.5 Definition of Trim Degrees of Freedom (d.o.f.)

The ZEUS TRIM module employs the ZAERO linear trim analysis module as the core solver to obtain a trim solution. The ZAERO linear trim analysis module uses the aerodynamic stability derivative of each trim variable to solve the trim equation for satisfying the trim condition. The user should refer to the ZAERO theoretical manual for the formulation of the ZAERO linear trim analysis In this linear trim analysis, all aerodynamic stability derivatives are assumed to be constant; i.e., independent of the mean flow condition. However, in the real flow, the aerodynamic stability derivatives are functions of the mean flow condition. Because ZEUS can capture this nonlinear effect, an iterative procedure is developed in the ZEUS TRIM module that iterates around the ZAERO linear trim module until a converged trim solution is achieved. Such an iterative procedure is depicted in Figure 4.6. An initial mean flow condition (α, β, *p, q ,r* and control surface deflection angles,.. etc.) is first specified by a **TRIMFLT** bulk data card. Based on these this mean flow condition, the Euler solver module first computes the mean flow solution in terms of the aerodynamic forces and moments using the pseudo time-domain aerodynamic computation then computes the aerodynamic stability derivatives from the mean flow solution for each trim variable. The number of the pseudo time step is 100 as a default value but that can be altered using the **MKPARAM** bulk data card with entry TRMSTEP

Two methods are available to compute the aerodynamic stability derivatives. The first method is the finite difference method that introduces a small perturbation due to the change of the trim variable to the mean flow condition then calculates the derivative using the difference between the perturbed and mean flow solutions. The pseudo time-domain aerodynamic computation is again employed for this perturbed flow computation except using the mean flow solution as the initial flow condition. To increase the computational efficiency, the number of pseudo time steps for this perturbed flow computation is reduced from TRMSTEP by a factor of 3 but that can be altered using a **PARAM** bulk data card with entry NAME = TRMREDUC. The second method is to use the linearized Euler solver that computes the derivatives by solving the linearized Euler equation. To invoke the linearized Euler solver, it requires to specify the entry METHOD=3 in the **MKPARAM** bulk data card and as an option, to add a **MKPARAL** bulk data card if a different set of parameters is to be used for the linearized Euler solver.

The mean flow solution and the aerodynamic stability derivatives of trim variables are used by the ZAERO linear trim module to compute the trim variable solutions. These trim variable solutions are then added to the initial mean flow condition to update the mean flow condition. Then the Euler solver module re-computes the mean flow solution and the aerodynamic stability derivatives for the next iteration. Based on the new mean flow solution and aerodynamic stability derivatives, the ZAERO linear trim module solves the trim solution again. If all trim variable solutions are smaller than 3% of their respective values at the first trim iteration (this 3% can be altered by a **PARAM** bulk data card with entry NAME="TRIMCNV"), a converged solution is achieved because the mean flow solution in this mean flow condition itself satisfies the trim condition. Otherwise, these trim variable solutions are added to the mean flow condition of the previous iteration for the next iteration. The maximum number of iterations that loops around the ZAERO linear trim module is 3, but that can be altered using a **PARAM** bulk data card with entry NAME = “MAXTRIM”.

The final trim solution can be saved on a file using the **MLDTRIM** bulk data card. This final trim solution can be retrieved back via the **MLDTRIM** bulk data card as the initial flow condition for transient response analysis such as MLOADS, GLOADS, ELOADS, and NLFLTR.

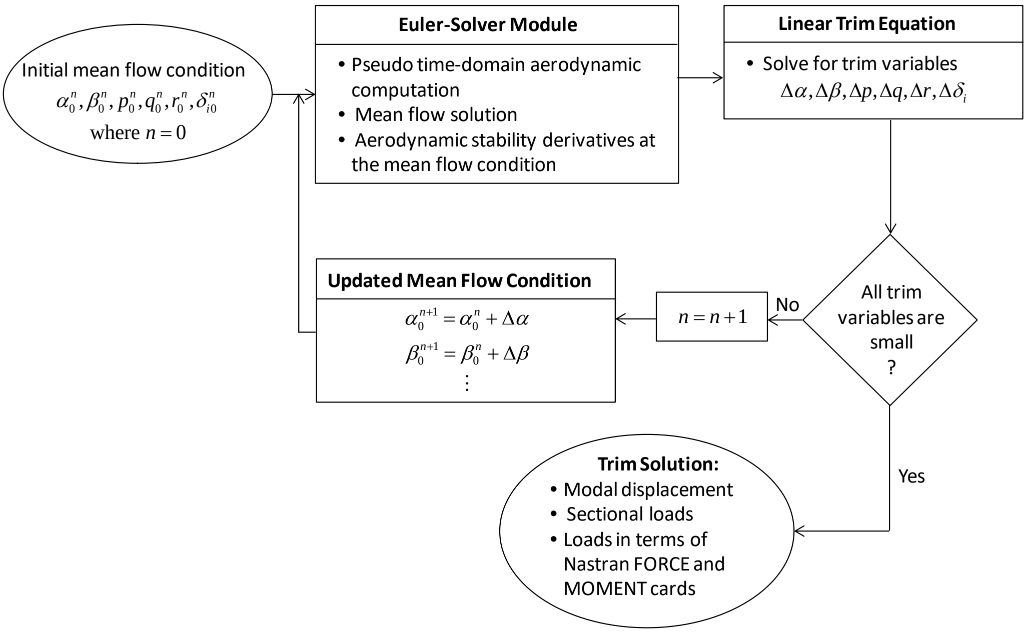


Figure 4.6 Iteration for the ZEUS Trim Analysis

There are several major differences between the ZAERO linear trim analysis and the NASTRAN trim analysis in solving the trim system:

- ZAERO linear trim analysis employs the modal approach to solve the trim system of the flexible aircraft whereas NASTRAN uses the direct method that includes all structural d.o.f. in the trim system. The modal approach assumes that the structural deformation $\{\boldsymbol{x}\}$ can be approximated as:

  $$\{x\} = [\text{PHG}]\,\{q\}$$

where [PHG] is the modal matrix containing the lower order modes of the structural finite element model, and $\{\boldsymbol{q}\}$ are the generalized coordinates.

Numerical experience shows that, for a complete aircraft structure, using the first fifty lower order modes for [PHG] is sufficient to achieve a converged solution. Thus, the modal approach reduces the size of the trim system from over thousands d.o.f. (for a complete aircraft structure, the number of d.o.f. in the structural finite element model can easily be in the thousands) down to as low as fifty. Thus, the modal approach offers a solution technique that is much more efficient than the direct method.

- NASTRAN is only capable of solving the determined trim system (the number of unknowns equal to the number of trim d.o.f.). In addition to the determined trim system, ZAERO linear trim analysis can also solve the over-determined trim system (i.e., where the number of unknowns is greater than the number of trim d.o.f.) by using a feasible direction technique that minimizes a user-defined objective function while satisfying a set of constraint functions. The objective and constraint functions can be specified in terms of the so-called "trim functions" that include induced drag, component loads, element stresses, lower and upper limits of the trim variables, etc.

The bulk data cards for static aeroelastic/trim discipline are listed in the following table:

Name	Description	Remarks
MLDTRIM	Exports the trim solution for a subsequent transient response analysis.	Optional
TRIM	Defines the flight condition, rigid body mass matrix, trim d.o.f. and the trim variables to perform static aeroelastic/trim analysis.	Required, if the **TRIM** Case Control Command is selected in the Case Control Section.
TRIMADD	Defines a trim function as a function of other trim functions.	Optional
TRIMCON	Defines a set of constraint functions to be satisfied for solving the over-determined trim system.	Required only for the over-determined trim system.
TRIMFNC	Defines a trim function whose value is depended on the trim variables and trim d.o.f.	Required only for the over-determined trim system.
TRIMLNK	Defines a set of coefficient and trim variable identification number pairs for trim variable linking.	Optional
TRIMOBJ	Defines an objective function to be minimized for solving the over-determined trim system.	Required only for the over-determined trim system.
TRIMVAR	Defines a trim variable for the static aeroelastic/trim analysis.	Required if a **TRIM** bulk data card is active.

The interrelationship between the bulk data cards for static aeroelastic/trim analysis is depicted in Figure 4.7. In addition to the above listed bulk data cards, the **PLTTRIM** bulk data card can be used to generate the plot files of the deformed aerodynamic model and steady pressure distributions. In addition, **PLTTRIM** can be used to generate a file that contains the flight loads in terms of NASTRAN **FORCE** and **MOMENT** bulk data cards at the structural finite element grid points. The user can insert this file back into the finite element model to perform a static analysis for the detailed stress calculations.

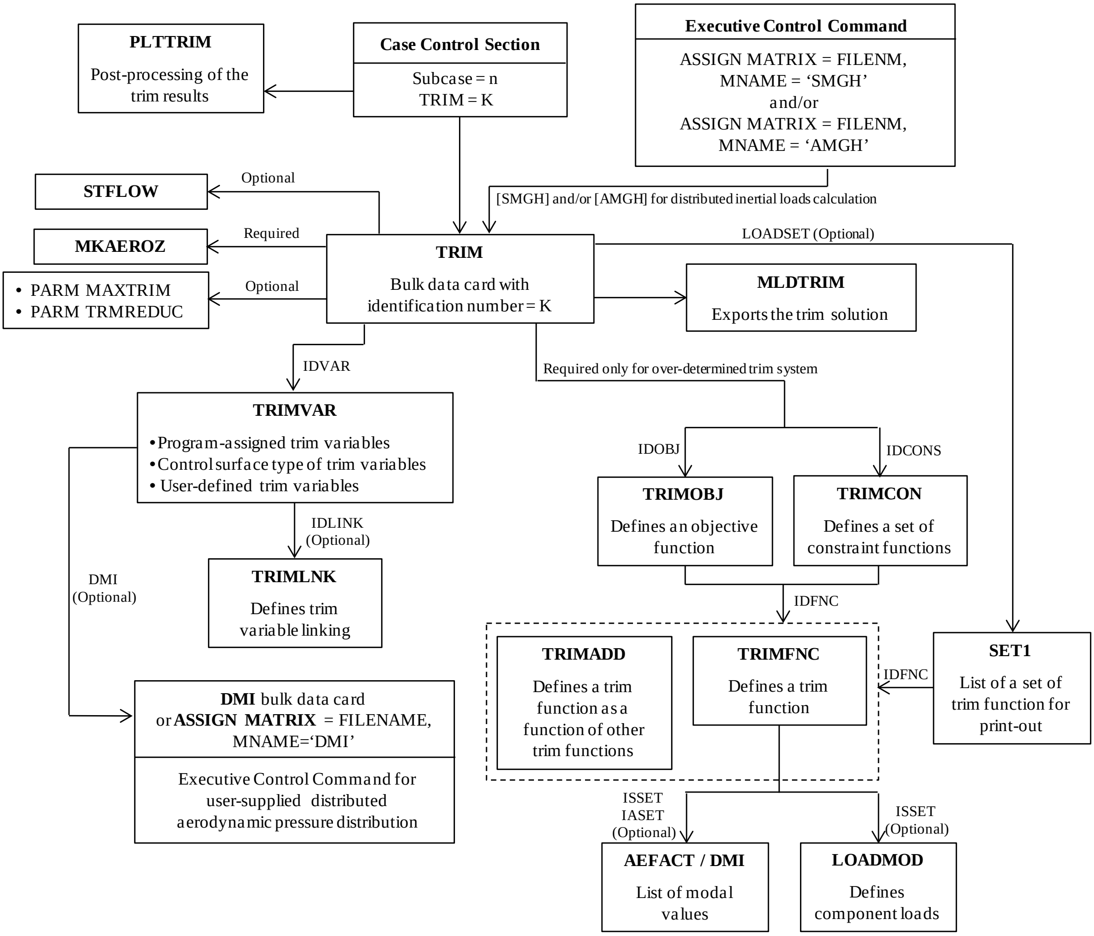


Figure 4.7 Bulk Data Interrelationship for Static Aeroelastic/Trim Analysis

In addition to the free vibration solutions, for computing the distributed inertial loads, the static aeroelastic/trim analysis also requires a matrix called [SMGH] for the symmetric trim system, a matrix called [AMGH] for the anti-symmetric trim system, and both for the asymmetric trim system that are imported by the **'ASSIGN MATRIX='** Executive Control Command. The distributed inertial loads exist only if the structural finite element model contains rigid body d.o.f. (a non-zero integer specified by the SUPORT entry of the **'ASSIGN FEM='** Executive Control Command). The equation for computing the inertial loads of a symmetric trim system reads:

$$\{\boldsymbol{F_I}\}_S = [\mathrm{MGG}]\{\ddot{\boldsymbol{X}}\}_S = [\mathrm{MGG}][\mathrm{PHG}]_S\{\ddot{\boldsymbol{u}}_{\boldsymbol{r}}\}_S = [\mathrm{SMGH}]\{\ddot{\boldsymbol{u}}_{\boldsymbol{r}}\}_S$$

where

$\{F_I\}$	is the distributed inertial loads.
[MGG]	is the mass matrix of the structural finite element model defined in the G-set d.o.f.
[PHG]	is the modal matrix of the free vibration solution that is imported by the **'ASSIGN FEM='** Executive Control Command.
$\{\ddot{u}_r\}$	represents the accelerations of the trim d.o.f.
$\{X\}$ = [PHG] $\{\ddot{u}_r\}$	is the acceleration vector that is approximated by the modal approach in terms of the product of [PHG] and $\{\ddot{u}_r\}$.
[SMGH] = [MGG] $[\mathrm{PHG}]_s$	and the subscript s denotes that the matrix/vector is for the symmetric structural modes.

Likewise, for the anti-symmetric trim system, it can be shown that the matrix [AMGH] is computed by:

$$[\mathrm{AMGH}] = [\mathrm{MGG}]\,[\mathrm{PHG}]_a$$

where $[\mathrm{PHG}]_a$ is the anti-symmetric modal matrix.

As an alternative, the [MGG] matrix can be directly imported into the program using the "ASSIGN MATRIX=' Executive Control command. In this case, the [SMGH] and [AMGH} matrices are not required because they will be automatically computed by the program.

4.2.7 INPUT FOR TRANSIENT MANEUVER LOADS ANALYSIS (MLOADS MODULE)

The MLOADS module computes the transient response of the aeroelastic system using the time-domain unsteady aerodynamic computation of the Euler-solver module. In general, the MOLADS module solves the following state-space equation:

$$\dot{X} = Ax + Bu$$

Where $X = \left\{\xi, \dot{\xi}\right\}^T$, ξ is the structural generalized coordinates

A involves the generalized stiffness, damping and mass matrices

B involves the inverse of the generalized mass matrix

and u is the generalized aerodynamic forces

At each time step, the state-space equation is first solved for ξ and $\dot{\xi}$. Then, the structural deformation is used to construct the unsteady boundary condition of the Euler equations. Generalized aerodynamic forces are then computed for the next time step using the Euler-solver module. The input of the MLOADS module is the time history of a pilot input command to simulate a maneuvering condition.

This feature allows the study of the structural loads due to aircraft maneuver. As an alternative, the initial generalized coordinates of the structural modes can be also specified to excite the aeroelastic system. From the transient response, the user can determine the stability boundary of the aeroelastic system by varying the flight conditions from low dynamic pressure to high dynamic pressure.

Prior to MLOADS analysis, a static aeroelastic analysis is performed first to obtain a mean flow solution from which the transient response analysis using the time-domain unsteady aerodynamic computation of the Euler-solver module starts. Note that the number of time steps of this static aeroelastic analysis is 100 as default, but that can be altered using a **MKPARAM** bulk data card with entry TRMSTEP. The flow solution of the static aeroelastic analysis can be saved/retrieved using the **STFLOW** bulk data card. As an alternative, the trim solution computed by a trim analysis can be retrieved by a **MLDTRIM** bulk data card as the initial flow condition for the time-domain MLOADS analysis. In this case, the static aeroelastic analysis will be skipped.

To output the time history of the forces at the structural finite element grid points, it is required to import a matrix called "SMGH" for the symmetric/asymmetric maneuver and a matrix called "AMGH" for the anti-symmetric maneuver through the **'ASSIGN MATRIX='** Executive Control Commands. SMGH and AMGH are computed by:

$$[\mathrm{SMGH}] = [\mathrm{MGG}][\mathrm{PHG}]_s, \qquad [\mathrm{AMGH}] = [\mathrm{MGG}][\mathrm{PHG}]_a$$

where:

[MGG] is the mass matrix of the G-set d.o.f. (G-set is defined as 6 × number of structural finite element grid points).

$[\mathrm{PHG}]_s$ and $[\mathrm{PHG}]_a$ are the symmetric and anti-symmetric modal matrixes of the G-set d.o.f., respectively.

Because [SMGH] or [AMGH] is usually not a part of the finite element standard output (except for NE/NASTRAN), therefore, to obtain them, it is required to modify the computational sequence of the finite element codes.

The following example shows the MSC.NASTRAN DMAP alter statements that generate matrices [SMGH] and [AMGH] by the NASTRAN/OUTPUT4 module.

```
ASSIGN OUTPUT4='filename' STATUS=UNKNOWN UNIT=12 FORM=FORMATTED
SOL 103
COMPILE SEMODES SOUIN=MSCSOU LIST NOREF  $
ALTER 'STRAIN ENERGY' $
MATGEN EQEXINS/INTEXT/9//LUSETS $ GENERATE EXTERNAL SEQUENCE MATRIX
MPYAD MGG,PHG,/MKHINT $ MKHINT IS THE MGH IN INTERNAL SEQUENCE
MPYAD INTEXT,MKHINT,/MGH $ TRANSFORM MKHINT TO EXTERNAL SEQUENCE
OUTPUT4 MGH//-1/12/2 $ OUTPUT MGH TO UNIT=12 IN filename
ENDALTER
CEND
```

Once the file 'filename' is generated by NASTRAN, it can be directly inputted into ZEUS by the **'ASSIGN MATRIX='** Executive Control Command. For example:

- For a symmetric/asymmetric boundary condition:
  ASSIGN MATRIX = ′ filename ′, MNAME = ′SMGH′, FORM = ′FORMAT′

- For an anti-symmetric boundary condition:
  ASSIGN MATRIX = ′ filename ′, MNAME = ′AMGH′, FORM = ′FORMAT′

Note that the name of the matrix is defined as ′MGH′ in the NASTRAN DMAP alter statements. However, in the **'ASSIGN MATRIX='** Executive Control Command, it is replaced by MNAME = ′SMGH′ for the symmetric/asymmetric boundary condition or asymmetric boundary condition and MNAME = ′AMGH′ for the anti-symmetric boundary condition.

As an alternative, the [MGG] matrix can be directly imported into the program using the "ASSIGN MATRIX=' Executive Control command. In this case, the [SMGH] and [AMGH} matrices are not required because they will be automatically computed by the program.
The bulk data cards for the transient maneuver loads analysis are listed in the following table.

Name	Description	Remarks
EXTINP	Defines the input of the control element to a control surface.	Optional
MLOADS	Defines the control system, aeroelastic system, airframe states, pilot input commands and time integration for the transient maneuver load analysis.	Required if **MLOADS** Case Control Command is selected in the Case Control Section.
MLDCOMD	Defines the time histories of the pilot input commands.	Required if the **MLOADS** bulk data card is activated.
MLDTIME	Defines the starting time, ending time, time step and output time for the transient analysis.	Required if the **MLOADS** bulk data card is activated.
MLDPRNT	Defines an ASCII file to store the time histories of parameter due to the transient maneuver loads.	Optional
MLDSTAT	Defines the airframe states and their initial values for dynamic load analysis.	Optional
RESTART	Save or Retrieve Entire Flow Solution of a Transient Analysis	Optional
TABLED1	Defines a tabular function for use in generating frequency-dependent or time-dependent table.	Required if the **MLOADS** bulk data card is activated.

In addition to the above listed bulk data cards, the **PLTTIME** bulk data card can be used to generate the plot files of the maneuver position of the oscillating aerodynamic model at each time step. Also, **PLTTIME** can be used to generate a file that contains the flight loads in terms of NASTRAN **FORCE** and **MOMENT** bulk data cards at the structural finite element grid point for all time steps (with different load set identification numbers for each time step). These are two options to compute the flight loads; the mode displacement method and the mode acceleration method. The user can insert this file back to the finite element model to perform a static analysis for detailed stress calculations by selecting a particular load set identification number.

The interrelationship among all the bulk data cards for the transient maneuver loads analysis is depicted in Figure 4.8.

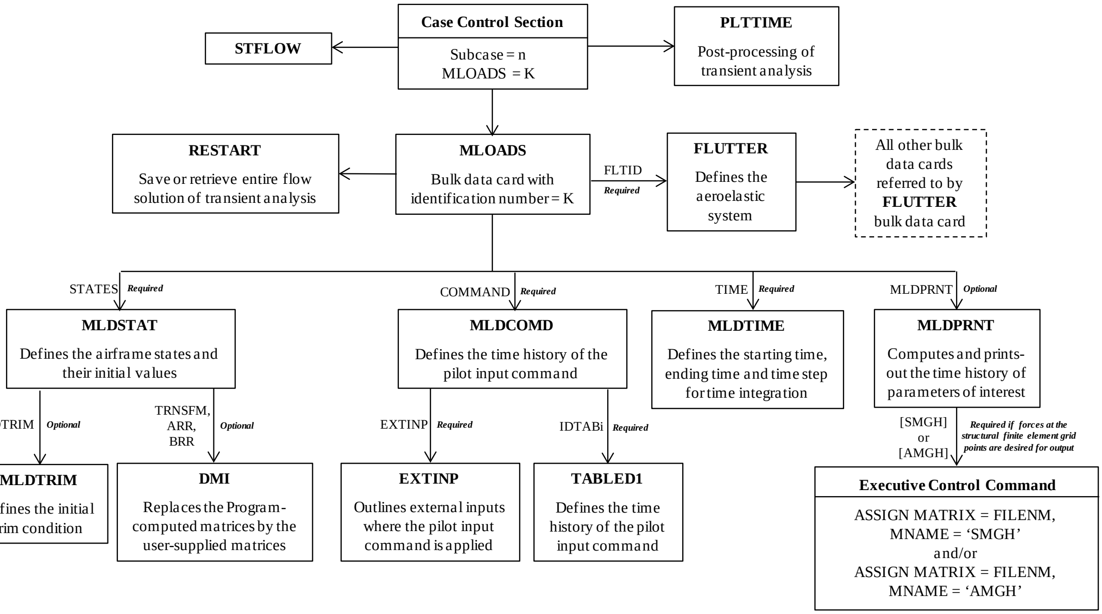


Figure 4.8 Bulk Data Interrelationship for Transient Maneuver Load Analysis

4.2.8 INPUT FOR TRANSIENT EJECTION LOADS ANALYSIS (ELOADS MODULE)

The function of the transient Ejection LOADS (ELOADS) analysis is to compute the transient response of the structures due to store ejections. It allows multiple store ejections (in a sequential scheduling) while the aircraft is maneuvering due to the pilot's input command. The computational procedure of the transient ejection loads analysis is very similar to that of the transient Maneuver LOADS (MLOADS) analysis. Therefore, the majority of the ELOADS inputs are identical to the MLOADS analysis except the additional input for the ejection forcing functions and the change of structural properties due to the separation of stores from the aircraft structures. Because of the structural properties variation after each store ejection, the system matrices of the state space equations is time-varying. These time-varying system matrices are discretized into several piecewise constant system matrices forming a set of state space equations. This discretization process is based on time intervals, which are defined by the starting time and ending time of the transient response computation as well as the ejection times of the stores.

In addition to the bulk data cards for the MLOADS analysis, two bulk data cards are required for the transient ejection loads analysis, which is presented in the following table.

Name	Description	Remarks
ELOADS	Defines the aeroelastic system, time integration, ejection forcing functions and changes of structural properties due to store separation for transient ejection loads analysis.	Required if **ELOADS** Case Control Command is selected in the Case Control Section.
DFORCE	Defines a dynamic forcing function and the structural grid points where these forces are applied.	Required

The interrelationship among all the bulk data cards for the transient ejection loads analysis is depicted in Figure 4.9.

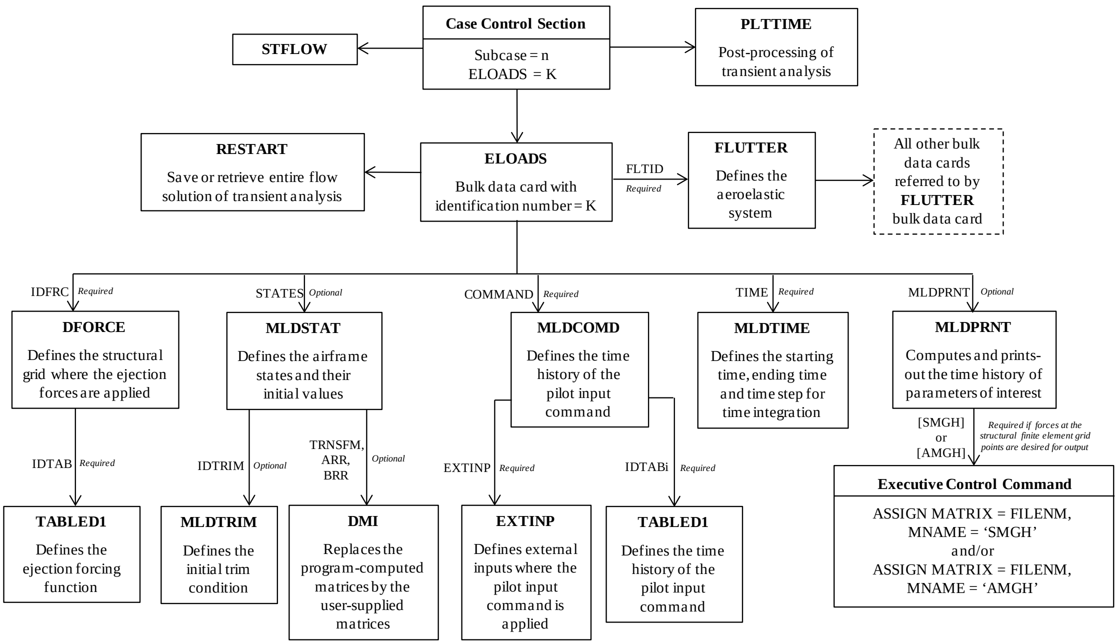


Figure 4.9 Bulk Data Interrelationship for Transient Ejection Load Analysis

Note that the **MLDCOMD** bulk data card is an optional input for the ELOADS analysis, but it is required for the MLOADS analysis.

Similar to the MLOADS analysis, prior to ELOADS analysis, a static aeroelastic analysis is performed first to obtain a mean flow solution from which the transient response analysis using the time-domain unsteady aerodynamic computation of the Euler-solver module starts. Note that the number of time steps of this static aeroelastic analysis is 100 as default, but that can be altered using a **MKPARAM** bulk data card with entry TRMSTEP. The flow solution of the static aeroelastic analysis can be saved/retrieved using the **STFLOW** bulk data card. As an alternative, the trim solution computed by a trim analysis can be retrieved by a **MLDTRIM** bulk data card as the initial flow condition for the time-domain ELOADS analysis. In this case, the static aeroelastic analysis will be skipped

The **PLTTIME** bulk data card can also be used to generate the plot file for the ELOADS analysis. Please see Section 4.2.6 for the description of the **PLTTIME** bulk data card. Also, the transient loads can be obtained by the mode displacement method or the mode acceleration method via a **MLDPRNT** bulk data card.

4.2.9 INPUT FOR DISCRETE GUST LOADS ANALYSIS (GLOADS MODULE)

The function of the GLOADS analysis is to compute the transient responses of the structures while the airframe is encountering a discrete gust field. The majority of the GLOADS inputs are identical to MLOADS analysis, except that the **MLDCOMD** bulk data card for MLOADS is replaced by the **DGUST** (discrete gust) bulk data card to specify a discrete gust profile. The aerodynamic force due to the discrete gust is computed using the discrete gust aerodynamic computation of the Euler-solver module.

In addition to the bulk data cards for the MLOADS analysis, two bulk data cards are required for the GLOADS analysis, which is presented in the following table.

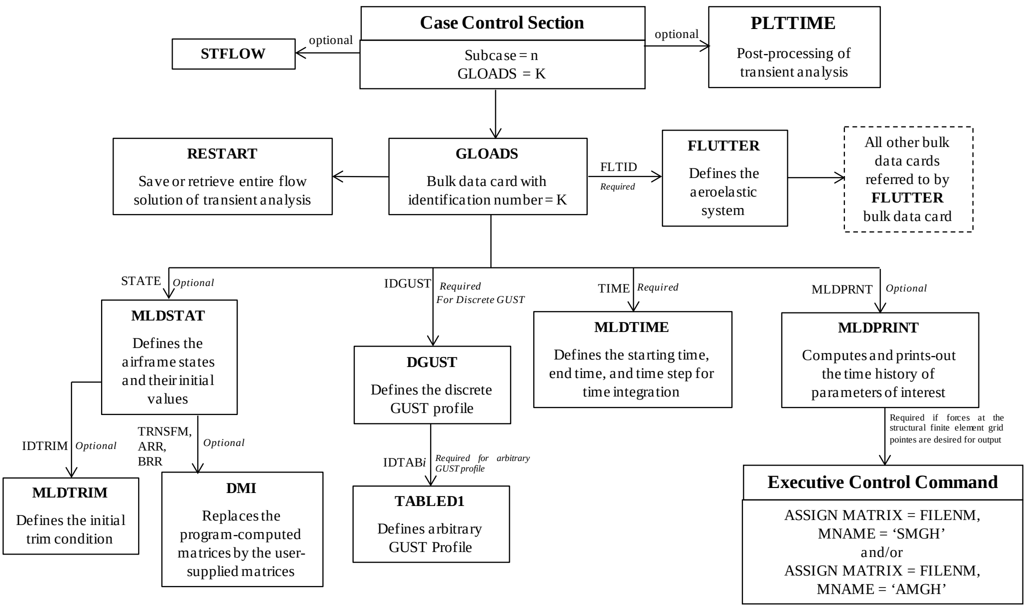


Figure 4.10 Bulk Data Interrelationship for Transient Discrete Gust Load Analysis

Name	Description	Remarks
GLOADS	Defines the control system, aeroelastic system, airframe states, discrete gust profile, and time integration for transient discrete gust loads analysis.	Required, if **GLOADS** Case Control Command is selected in the Case Control Section.
DGUST	Defines a discrete gust profile for the transient gust loads analysis.	Required for discrete gust analysis.

The interrelationship among all the bulk data cards for the transient discrete gust loads analysis is depicted in Figure 4.10.

Similar to the MLOADS analysis, prior to GLOADS analysis, a static aeroelastic analysis is performed first to obtain a mean flow solution from which the transient response analysis using the time-domain unsteady aerodynamic computation of the Euler-solver module starts. Note that the number of time steps of this static aeroelastic analysis is 100 as default but that can be altered using a **MKPARAM** bulk data card with entry TRMSTEP. The flow solution of the static aeroelastic analysis can be saved/retrieved using the **STFLOW** bulk data card. As an alternative, the trim solution computed by a trim analysis can be retrieved by a **MLDTRIM** bulk data card as the initial flow condition for the time-domain EGOADS analysis. In this case, the static aeroelastic analysis will be skipped.

The **PLTTIME** bulk data card can also be used to generate the plot file for the GLOADS analysis. Please see Section 4.2.6 for the description of the **PLTTIME** bulk data card. Also, the transient loads can be obtained by the mode displacement method or the mode acceleration method via a **MLDPRNT** bulk data card.

4.2.10 INPUT FOR NONLINEAR FLUTTER ANALYSIS (NLFLTR MODULE)

The NLFLTR module computes the transient response if an aeroelastic system that contains a concentrated structural nonlinearity such as free-play. To specify such a nonlinear aeroelastic system, the user must define the structural nonlinearity as a function of a nonlinear parameter using the **NLSYSM** bulk data card. Thus, the nonlinear parameters divide the nonlinear structural into multiple sub-linear domains. At each time step, the value of the nonlinear parameter is first calculated to determine its corresponding sub-linear structure and construct a linear structural state-space equation for the solution of the next time step. The time-domain unsteady aerodynamic computation of the Euler-solver module is employed to generate the aerodynamic forces at each time step. To excite the nonlinear aeroelastic system, the user can either specify a pilot input command or a discrete gust profile. The output of the NLFLTR is very similar to that of the MLOADS module.

The input bulk data cards for **NLFLTR** module is very similar to those of the MLOADS module except two additional bulk data cards presented in the following table.

NAME	Description	Remarks
NLFLTR	Define the data needed to perform a nonlinear flutter analysis.	Required if **NLFLTR** Case Control Command is selected in the Case Control Section.
NLSYSM	Define the matrices of the aeroelastic system for nonlinear open/closed loop flutter analysis.	Required

Similar to the MLOADS analysis, prior to the NLFLTR analysis, a static aeroelastic analysis is performed first to obtain a mean flow solution from which the transient response analysis using the time-domain unsteady aerodynamic computation of the Euler-solver module starts. Note that the number of time steps of this static aeroelastic analysis is 100 as default, but that can be altered using a **MKPARAM** bulk data card with entry TRMSTEP. The flow solution of the static aeroelastic analysis can be saved/retrieved using the **STFLOW** bulk data card. As an alternative, the trim solution computed by a trim analysis can be retrieved by a **MLDTRIM** bulk data card as the initial flow condition for the time-domain NLFLTR analysis. In this case, the static aeroelastic analysis will be skipped

The **PLTTIME** bulk data card can also be used to generate the plot file for the NLFLTR analysis. Please see Section 4.2.6 for the description of the **PLTTIME** bulk data card. Also, the transient loads can be obtained by the mode displacement method or the mode acceleration method via a **MLDPRNT** bulk data card.

The interrelationship among all bulk data cards for the nonlinear flutter analysis is depicted in Figure 4.11.

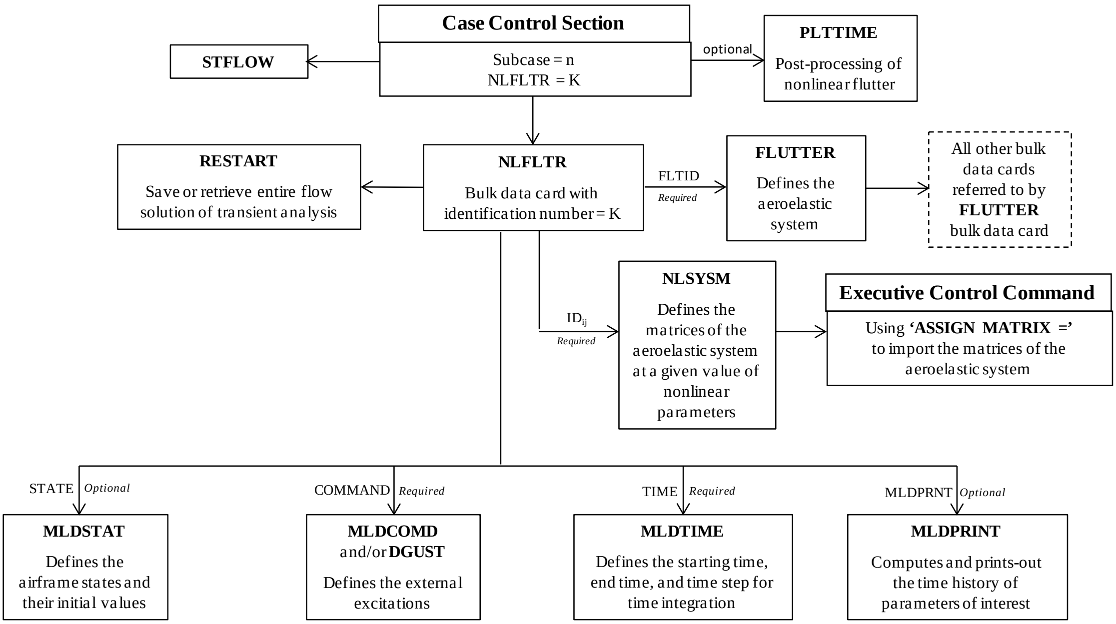


Figure 4.11 Bulk Data Interrelationship for Nonlinear Flutter Analysis

4.2.11 THE FREQUENCY-DOMAIN GENERALIZED AERODYNAMIC FORCE GENERATION (GENGAF MODULE)

The objective of the GENGAF module is to generate the frequency-domain Generalized Aerodynamic Forces (GAF) that can be imported back to the FLUTTER module or to other frequency-domain-based aeroelastic code such as ZEUS for rapid aeroelastic analysis. There are three sets of GAF, namely the GAF due to the structural modes (QHH), the GAF due to control surface oscillation (QHC), and the GAF due to sinusoidal gust excitation (QHG) defined as:

$$QHH\left(ik\right)=\phi^T f_a\left(\phi,\dot{\phi}\right)$$
$$QHC\left(ik\right)=\phi^T f_a\left(\phi_c,\dot{\phi}_c\right)$$
$$QHG\left(ik\right)=\phi^T f_a\left(w_g,\dot{w}_g\right)$$

where ϕ is the modal matrix whose columns contain the structural modes

ϕ_c is the control matrix whose columns contain the control surface modes

w_g represents the sinusoidal gust

and f_a is the aerodynamic forces and k is the reduced frequency defined as $k=\omega(REFC/2)/V$ where V is the free stream velocity, and REFC is the reference chord specified in the **AEROZ** bulk data card.

Similar to the FLUTTER module, two methods are available for computing the frequency-domain GAF; the linearized Euler solver that solves for the GAF matrices directly in the frequency domain and the sinusoidal excitation technique applied to the time domain full order Euler solver. The linearized Euler solver is activated by specifying the entry METHOD=3 in the **MKPARAM** bulk data card.

Unlike the FLUTTER module that computes the generalized aerodynamic forces on a rigid structure, prior to the frequency-domain unsteady aerodynamic computation, the GENGAF module performs a static aeroelastic analysis on the flexible structure to obtain the mean flow solution based on the mean flow condition specified in the **TRIMFLT** bulk data card. This mean flow solution on the flexible structure is used as the initial flow solution from which the frequency-domain unsteady aerodynamic computations are computed. Thus, the frequency-domain generalized aerodynamic forces computed by the GENGAF module can be treated as the perturbed unsteady aerodynamics about an aeroelastically deformed structure. The number of time steps of the static aeroelastic computation is 100, but that can be altered using the **MKPARAM** bulk data card with entry TRMSTEP.

Two bulk data cards are required for the GENGAF modules which are presented in the following table:

Name	Description	Remarks
GENGAF	Generation of the frequency-domain generalized aerodynamic forces.	Required if **GENGAF** Case Control Command is specified.

SURFLST	Defines a list of the control surface.	Optional

As an option after the GAF matrices are obtained, the GENGAF module can further perform a frequency-domain flutter analysis using the g-method provided that the identification number of the **GENGAF** bulk data card matches with that of a **FLUTTER** bulk data card.

The interrelationship among all bulk data cards for the GENGAF module is depicted in Figure 4.12.

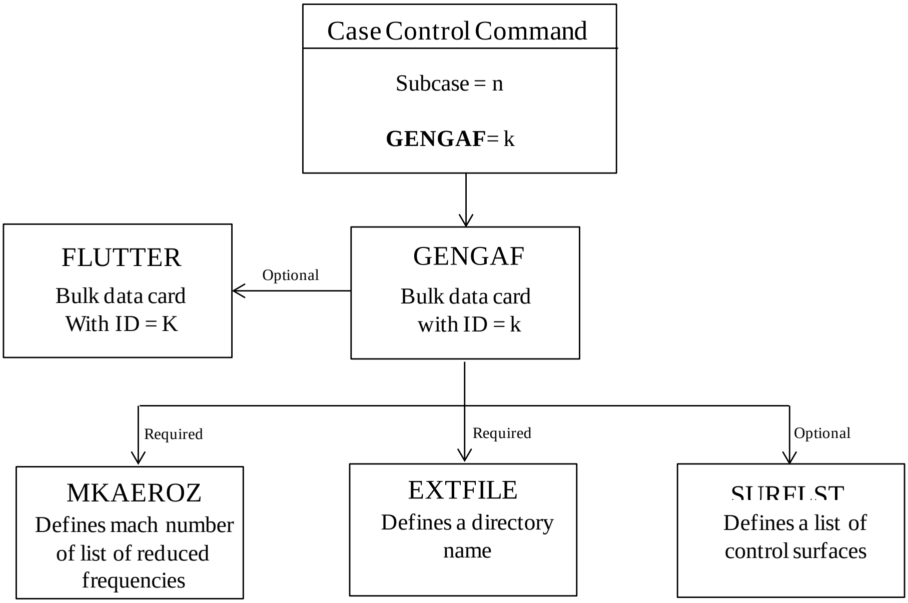


Figure 4.12 Bulk Data Interrelationship for Nonlinear Flutter Analysis

4.2.12 INPUT FOR PLOT FILE GENERATION

ZEUS does not provide graphic capability. Instead, ZEUS generates files that can be read by TECPLOT, FEMAP, PATRAN, or I-DEAS for post-processing. The bulk data cards shown in the following table can be specified to generate various output files.

Name	Description	Remarks
PLTAERO	Generates an ASCII text file for plotting the aerodynamic model.	Optional
PLTCP	Generates an ASCII text file for plotting the steady/unsteady pressure coefficients.	Optional
PLTFLUT	Generates an ASCII text file for plotting the flutter or ASE mode.	Optional

PLTMODE	Generates an ASCII text file for plotting the interpolated structural mode on the aerodynamic model.	Optional
PLTSLP	Generates an ASCII text file for plotting the slopes of the surface mesh on the aerodynamic model.	Optional
PLTSURF	ASCII text file generation for plotting the aerodynamic control surface.	Optional
PLTTIME	Generates an ASCII text file for the post-processing of the transient analysis.	Optional
PLTTRIM	Generates an ASCII text file for the post-processing of the static aeroelastic/trim analysis.	Optional
PLTVG	Generates an X-Y plot file (ASCII text) for plotting the flutter frequency and damping curves.	Optional

These bulk data cards are not referred to by other bulk data cards. Their appearance in the Bulk Data Section "triggers" the program to generate their associated output files.

4.2.13 MISCELLANEOUS INPUT

Name	Description	Remarks
$	Used to insert comments into the Bulk Data Section.	Optional
AEFACT	Specifies a list of real numbers.	Optional
AESLINK	Defines an additional aerodynamic control surface by linking a set of **AESURFZ / PZTMODE** bulk data cards.	Optional
ALTER	Perform matrix operations without modifying the program.	Optional
CORD2R	Defines a rectangular coordinate system.	Optional
ENDDATA	To signify the end of the Bulk Data Section.	Required
EXTFILE	Defines a character string as the name of an external file.	Optional
DMI	Header of direct matrix input.	Optional
DMIG	Direct Matrix Input at the structural finite element grid points.	Optional
DMIL	Defines the values of the matrix elements by 16-column fields.	Optional
DMIS	Defines the values of the matrix elements by 8-column fields.	Optional
DRELI	Loads input data directly into a relational entity.	Optional
FEMASET	Reduces the number of grid points of the structural finite element model from the G-set grids to the A-set grids.	Optional
FEMSAVE	Save the structural modal solution	Optional

FREQ	Frequency list.	Optional
FREQ1	Frequency list, Alternative form.	Optional
GRIDFRC	Defines a control force at a set of a structural finite element grid points.	Optional
INCLUDE	Inserts an external file into the Bulk Data Section.	Optional
LOADMOD	Defines a load mode of a set of structural grid points for computing component loads.	Optional
OMITMOD	Delete Structural Modes.	Optional
OUTPUT4	Exports a matrix data entity in the OUTPUT4 format to a data file.	Optional
PARAM	Alter the values of global parameters used in the computation.	Optional
PCHFILE	Imports a NASTRAN Punch output file that contains the modal values of element forces, stresses, strains, etc.	Optional
PZTMODE	Defines a structural deformation due to smart structural actuation for static aeroelastic/trim analysis or the transient response analysis.	Optional
SET1	Defines a list of identification numbers. If used for spline, it contains a list of identification numbers of structural finite element grid points.	Optional
STABDRV	Generates the aerodynamic stability derivatives	Must exist if **STABDRV** case control command is specified

4.3 BULK DATA DESCRIPTIONS

This section contains a complete description of each ZEUS bulk data card.

$ Comment Definition

Description: For user convenience in inserting commentary material into the Bulk Data Section. The **$** entry is otherwise ignored by the program. These entries will not appear in a sorted echo.

Format and Example:

1	2	3	4	5	6	7	8	9	10
$ Followed by any characters in columns 2-80									

$ THIS (*,,"$$)--/									

ACOORD

Aerodynamic Coordinate System

Description: Defines a local coordinate system for an aerodynamic component referenced by the **BLOCK**, **BODY7**, **CAERO7**, **GAP**, **GAP1**, and/or **GAPZ** bulk data cards.

Format and Example:

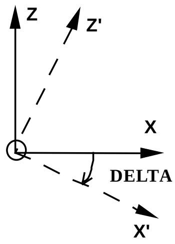
1	2	3	4	5	6	7	8	9	10
ACOORD	ID	XORIGN	YORIGN	ZORIGN	DELTA	THETA			
ACOORD	10	250.0	52.5	15.0	0.0	0.0			

Field	Contents
ID	Coordinate system identification number. (Integer > 0)
XORIGN YORIGN ZORIGN	*X*, *Y* and *Z* location of the component origin. (Real)
DELTA	Pitch angle in degrees measured from the X-Z axes of the basic coordinate system to the X'-Z' axes of the component coordinate system, positive in direction shown (see Remark 4 figure). This parameter will not physically rotate the model. Its effects are introduced in the boundary condition. Therefore, **DELTA** must be a small value. (Real) (See Remark 4)
THETA	Roll angle in degrees measured from the Y-Z axes of the basic coordinate system to the Y'-Z' axes of the component coordinate system, positive in direction shown (see Remark 4 Figures). Unlike **DELTA**, **THETA** will physically rotate the model. (Real)

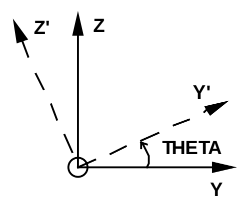
Remarks:

1. Coordinate system Identification Numbers (ID) on all **ACOORD** bulk data cards must be unique.
2. If **ACOORD** is referenced by a **BODY7** bulk data card, the X-axis of the coordinate system defines the centerline of the body.
3. All coordinate locations are with reference to the basic coordinate system. **ACOORD** defines a rectangular coordinate system whose X-axis must be parallel to the X-axis of the basic coordinate system.
4. Since most underwing stores have a small inclination angle to the free stream, **DELTA** can be used to provide a simpler means for defining this inclination.

Definition of Angle DELTA



Definition of Angle THETA



ACTU

Actuator Transfer Function

Description: Defines the actuator transfer function.

Format and Example:

1	2	3	4	5	6	7	8	9	10
ACTU	ID	A0	A1	A2					

ACTU	10	0.8	0.5	0.2					

Field	Contents
ID	Identification number. (Integer > 0) (See Remarks 1 and 2)
Ai	The numerator and denominator coefficients in the actuator transfer function. (Real)

$$\frac{\delta}{u_{ac}} = \frac{A_0}{s^3 + A_2 * s^2 + A_1 * s + A_0}$$

Remarks:

1. The actuators are referred to by an **AESURFZ**, **AESLINK** or **PZTMODE** bulk data card.

AEFACT

List of Real Numbers

Description: Used to specify lists of real numbers.

Format and Example:

1	2	3	4	5	6	7	8	9	10
AEFACT	SID	VALUE1	VALUE2	VALUE3	VALUE4	VALUE5	VALUE6	D7	CONT
CONT	VALUE8	VALUE9	-etc-						

AEFACT	97	.3	.7	1.0					

Field	Contents
SID	Set identification number. (Unique Integer > 0)
VALUEi	Number (Real)

Remarks:

1. Embedded blank fields are forbidden.

AEROZ

Model Physical Data

Description: Defines the basic aerodynamic reference parameters.

Format and Example:

1	2	3	4	5	6	7	8	9	10
AEROZ	ACSID	XZSYM	FLIP	FMMUNIT	FMLUNIT	REFC	REFB	REFS	CONT
CONT	REFX	REFY	REFZ						

AEROZ	1	YES	NO	SLIN	IN	400.0	300	12000.	+AEROZ
+AEROZ	10.	0.	0.						

Field	Contents
ACSID	Identification number of a **CORD2R** bulk data card defining a coordinate system where the x-axis is toward the pilot's face (from a pilot situated in the finite element model) and the y-axis is on the pilot's Right Hand Side (RHS). (Integer >0 or Blank)(See Remark 2)
XZSYM	Character string, either "YES" or "NO" or "H2F" XZSYM = "YES": Both the aerodynamic and structural models are the half-span models. This implies that the aerodynamics configuration is symmetric about its x-z plane which requires only a half-span model. This half span aerodynamic model must be on the Right Hand Side (RHS) of the pilot. The aerodynamics influence between the RHS and left-hand side (LHS) is automatically accounted for by the program. XZSYM = "NO": Both the aerodynamic and structural models are the full-span models. XZSYM = "H2F": The aerodynamic model is a full-span model, but the structural model is a half-span model. This half span aerodynamic model must be on the RHS of the pilot. In this case, the program will internally generate a set of full-span structural modes by combining the symmetric and anti-symmetric modes. Also, a set of mirror-imaged structural grid points are created internally from those grid points in the half-span structural model. (Character, Default = "YES") (See Remark 2)
FLIP	Character string, either "YES" or "NO"; For FLIP = YES, the structural model is on the Left Hand Side (LHS) of the pilot, but aerodynamic model is on the RHS. (Character, Default = "NO"). (See Remark 3)

FMMUNIT	**<u>Not used.</u>** Units of mass used in the structural finite element model. This parameter is automatically assigned by the program to be "LBF/" if English units are used or "N/" if metric units are used on the FMLUNIT entry. Note that if FMLUNIT is assigned to be "NONE", the program will automatically set FMMUNIT to "NONE".
FMLUNIT	Units of length used in the structural finite element model as well as all length dimensions involved in the aerodynamic model. Must be one of “IN”, “FT”, “M”, “MM”, “CM”, “KM”, or “NONE”. (Character, Default = “NONE”). (See Remark 4)
REFC	Reference chord length. Units must be in FMLUNIT. (Real ≥ 0, Default = 1.0) (See Remark 5)
REFB	Reference span length. Units must be in FMLUNIT. (Real ≥ 0, Default = 1.0) (See Remark 5)
REFS	Reference area. Units must be in FMLUNIT^2. (Real ≥ 0, Default = 1.0) Note that the reference area should account for the area on both the right hand and the left sides of the configuration even if only a RHS configuration is modeled, i.e. XZ SYM = “YES.” (See Remark 5)
REFX, REFY, REFZ	Location of aerodynamic moment center for computing aerodynamic force and moment coefficients due to rigid body motion. (Real) (See Remark 6)

<u>Remarks:</u>

1. <u>This card must exist</u>. Only one **AEROZ** is allowed.

   The program assumes that the flow is in the positive x-direction in the basic coordinate system and that the aerodynamic model is on the RHS of the x-z plane (i.e., positive y-direction). However, for the spline module that requires the perfect overlapping between the aerodynamic model and the structural, the structural model may be oriented in an arbitrary coordinate system. In this case, for the displacements and loads spline between the aerodynamic and structural models, the structural grid points will be transformed to the aerodynamic coordinate system according to ACSID.

   To verify the coordinate system specified by ACSID, it is recommended that the entry FEMGRID = “YES” in the **PLTAERO** be used to visualize the aerodynamic model and the transformed structure grids together.

2. For a half-span aerodynamic model (XZSYM= “YES”), two ‘**ASSIGN FEM**=’ Executive Control Commands, one with Boundary = “SYM” and the other with BOUNDARY = “ANTI”, are allowed to be specified in the input file.

   For both full span aerodynamic and structural models (XZSYM= “NO”), only one ‘**ASSIGN FEM**=’ Executive Control Command with Boundary = “ASYM” is allowed. Consequently, only asymmetric aeroelastic analysis is performed.

   For a full-span aerodynamic model and a half-span structural model XZSYM= “H2F”), both symmetric and anti-symmetric modes must be imported by two ‘**ASSIGN FEM**=’ Executive Control Commands. All the mirror-imaged structural grid points internally created by the program have the same identification numbers as those grid points in the half-span structural model but with a negative sign. These mirror-imaged grid points (with negative identification numbers) can be referred to by the **SPLINEi/ATTACH** bulk data card, the **LOADMOD** bulk data card, and the **AESURFZ** bulk data card for the left-hand-side aerodynamic model. Also, only asymmetric aeroelastic analysis and asymmetric control surface are allowed.

Furthermore, the option XZSYM = "H2F" requires the G-set mass matrix called MGG to be imported by the "**ASSIGN MATRIX =**" Executive Control Command. The MGG matrix can be obtained from NASTRAN using the following ALTER statements.

```
ASSIGN OUTPUT4='demo1.mgg',UNIT=12,FORM=FOAMAT
SOL 103
COMPILE SEMODES SOUIN=MSCSOU LIST NOREF  $
ALTER 'STRAIN ENERGY'
MATGEN EQEXINS/INTEXT/9//LUSETS $ GENERATE EXTERNAL SEQUENCE MATRIX
MPYAD INTEXT,MGG,/MGGT/1        $ TRANSFORM MGG TO EXTERNAL SEQUENCE
OUTPUT4 MGGT//-1/12/2           $ OUTPUT MGG TO UNIT=12 IN demo1.mgg
ENDALTER
CEND
```

3. It is possible that the structural model may be located on the left-hand side (i.e., negative y-axis) of the coordinate system **ACSID**. In this situation, the structural model can be flipped from the left to the right-hand side by specifying **FLIP**= "YES".

4. FMLUNIT is the length unit involved in the structural analysis. The unit of length of the aerodynamic model must also be in FMLUNIT. Thus, the units of length of structural and aerodynamic models must be the same. FMMUNIT, formerly required as input, is automatically set to be a consistent mass unit based on the input length unit. For example, if the length unit is meters, the mass unit will end up kilogram; if the length unit is inches, the mass unit will end up slinch, and so on. In other words, for any metric length unit input, a "N/" will automatically be applied for the mass unit and if English length unit is input, a "LBF/" will automatically be applied for the mass unit. This always ensures that consistent units are used.

5. The non-dimensional aerodynamic force and moment coefficients are defined as:

Lift Coefficient	$C_L = \dfrac{L}{q_\infty (REFS)}$	, L is the lift force
Drag Coefficient	$C_D = \dfrac{D}{q_\infty (REFS)}$	, D is the drag force
Pitch Moment Coefficient	$C_M = \dfrac{M}{q_\infty (REFS)(REFC)}$	, M is the pitch moment
Side Force Coefficient	$C_Y = \dfrac{Y}{q_\infty (REFS)}$	, Y is the side force
Roll Moment Coefficient	$C_\ell = \dfrac{\ell}{q_\infty (REFS)(REFB)}$	, ℓ is the roll moment
Yaw Moment Coefficient	$C_n = \dfrac{N}{q_\infty (REFS)(REFB)}$	, N is the yaw moment

Note that all forces and moments computed by the program account for those generated by both sides of the configuration, even if only a RHS configuration is modeled. Therefore, REFS should account for the area on both sides of the configuration.

6. All aerodynamic moment coefficients, as well as stability derivatives, are computed using REFX, REFY, and REFZ as the aerodynamic moment center.

AESLINK Aerodynamic Control Surface Linking

Description: Defines an additional aerodynamic control surface by linking a set of **AESURFZ** bulk data cards.

Format and Example:

1	2	3	4	5	6	7	8	9	10
AESLINK	LABEL	TYPE	ACTID						CONT
CONT	COEFF1	AESURF1	COEFF2	AESURF2	…	-etc-			

AESLINK	AES1	SYM	100						+A
+A	1.0	AES2	0.5	AES3	0.3	AES4			

Field	Contents
LABEL	Unique alphanumeric string of up to eight characters used to define an additional aerodynamic control surface. (Character) (See Remark 1)
TYPE	Type of boundary condition. (Character) (See Remark 2) SYM - symmetric ANTISYM - anti-symmetric ASYM - asymmetric
ACTID	Identification number of the **ACTU** bulk data card defining the transfer function of the actuator attached to this control surface. (Integer ≥ 0) (See Remark 3)
COEFF*i*	A list of coefficients to define the linear combination of a set of **AESURFZ** bulk data cards. (Real) (See Remark 4)
AESURF*i*	A list of LABEL entries defined in the **AESURFZ** bulk data cards. (Character)

Remarks:

1. **AESLINK** provides a means to handle more than one aerodynamic control surface that is driven by one actuator or one control input command. Among all **AESLINK**, **AESURFZ**, **PZTMODE**, and **GRIDFRC** no duplicated LABEL is allowed.

2. TYPE must match the TYPE entry defined in the **AESURFZ** bulk data cards that are specified in the AESURF*i* list.

3. ACTID is only used for transient response analysis.

4. The resulting aerodynamic forces/moments of **AESLINK** is:

$$\varphi_L = \sum_i Coeffi \; \varphi_i$$

where φ_L is the aerodynamic forces/moments of **AESLINK**.
φ_i is the aerodynamic forces/moments of the i^{th} **AESURFZ**.

AESURFZ

Control Surface Definition

Description: Specifies an aerodynamic control surface for static aeroelastic/trim analysis, or the transient response analysis.

Format and Example:

1	2	3	4	5	6	7	8	9	10
AESURFZ	LABEL	TYPE	CID	SETK	SETG	ACTID			

AESURFZ	RUDDER	ASYM	1	10	20	0			

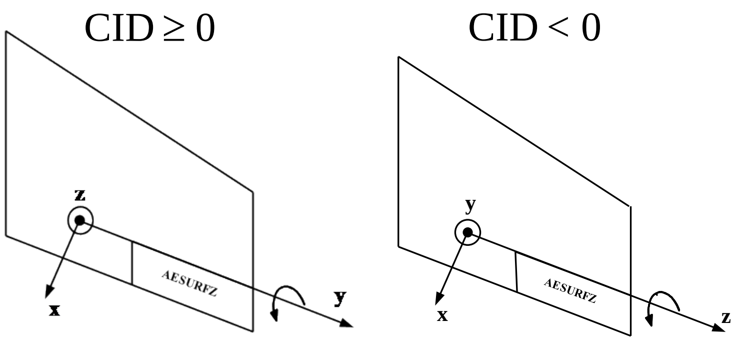
Field	Contents
LABEL	Unique alphanumeric string of up to eight characters used to identify the control surface. (Character) (See Remark 2)
TYPE	Type of surface (Character) SYM symmetric surface ANTISYM anti-symmetric surface ASYM asymmetric surface
CID	The absolute value of CID is the identification number of a rectangular coordinate system (**CORD2R** bulk data card). For CID ≥ 0; Y-axis of this coordinate system defines the hinge line of the control surface. For CID < 0, the Z-axis of the coordinate system defines the hinge line of the control surface. (Integer or Blank) (See Remark 3)
SETK	Identification number of **PANLST1**, **PANLST2** or **PANLST3** bulk data card used to identify the aerodynamic box ID's of the control surface. (Integer > 0) (See Remark 4)
SETG	Identification number of **SET1** or **SETADD** bulk data card used to identify the structural grid point ID's of the control surface. (Integer > 0 or Blank) (See Remark 5)
ACTID	Identification number of the **ACTU** bulk data card defining the transfer function of the actuator attached to this control surface. (Integer≥ 0) (See Remark 6)

Remarks:

1. **AESURFZ** is used for the **TRIM** and dynamic loads analysis. The user can activate the **PLTSURF** bulk data card to view the deflected control surface.

2. The **LABEL** is arbitrary, but all labels must be unique.

3. The y-axis or z-axis of the rectangular coordinate system should pass through the hinge line of the control surface. The rotation about the y-axis or z-axis by the right-hand rule defines the direction of the control surface deflection. For instance, the figure shown below indicates that the positive deflection of the control surface is deflecting downward.



If CID = 0, then the y-axis of the basic coordinates is used to define the hinge line location.

4. The aerodynamic box numbering schemes are illustrated in the **CAERO7** and **BODY7** bulk data cards.

5. SETG is used to compute the so-called "inertial coupling loads" due to the motion of the control surface which are required by the transient response analysis. All the structural finite element grid points located within the control surface should be included in the **SET1** or **SETADD** bulk data card list. Missing structural finite element grid point that has mass attached to it can lead to incorrect inertial coupling loads.

6. ACTID is used only for the MLOADS response analysis.

ALTER

Perform Matrix Operation

Description: Performs matrix operations without modifying the program.

Format and Example:

1	2	3	4	5	6	7	8	9	10
ALTER	STEP	MODULE	RESULT	OPERATOR	COEFFA	MATRIXA	SYMBOL	MATRIXB	

ALTER	3	FEM	SMHH	TRNS	1.0	SPHI	*	MGH	

Field	Contents
STEP	Index of operation sequence. (Integer > 0) (See Remark 1)
MODULE	Character either “FEM” or “SPLINE” to specify the module after which the matrix operations are performed. (Character) (See Remark 2)
RESULT	Character string defining the name of the resulting matrix from the matrix operation. (Character)
OPERATOR	Character string either “INV”, “TRNS”, “PRINT”, “GTOA”, “ATOG”, “COLGTOA”, “COLATOG”, “ROWGTOA”, “ROWATOG”, ”STOG”, “ROWSTOG”, “CLOSTOG”, “=” or blank that defines a matrix operation for the matrix “MATRIXA”. (Character, Default =”=”) (See Remark 3) Where, “INV” = Invert [(COEFFA)[MATRIXA]] “DEL” = delete matrix [MATRIXA]. “TRNS” = Transposed [(COEFFA)[MATRIXA]] “PRINT” = Print out the matrix [RESULT] “GTOA” = Reduce rows and columns of [MATRIXA] from G-set to A-set or remove the rows amd columns associated with the SPOINT/EPOINT. “ATOG” = Expand rows and columns of [MATRIXA] from A-set to G-set. The elements in the expanded submatrices are zero. “STOG” = Expand rows and columns of [MATRIXA] from S-set to G-set. The elements in the expanded submatrices are zero. “COLGTOA” = Reduce the columns of [MATRIXA] from G-set to A-set or remove the columns associated with the SPOINT/EPOINT. “COLATOG” = Expand the columns of [MATRIXA] from A-set to G-set “COLSTOG” = Expand the columns of [MATRIXA] from S-set to G-set “ROWGTOA” = Reduce the rows of [MATRIXA] from G-set to A-set

		or remove the rows associated with the SPOINT/EPOINT
"ROWATOG"	=	Expand the rows of [MATRIXA] from A-set to G-set
"ROWSTOG"	=	Expand the rows of [MATRIXA] from S-set to G-set
"="	=	Equal

Note: G-set is 6× (number of structural GRID+SPOINT points of the FEM model), where GRID and SPOINT are the NASTRAN **GRID** and **SPOINT** bulk data cards, respectively.

A-set is 6× (number of structural grid points defined by the DISP = *n* NASTRAN Executive Control Command, or the grid point defined by the **FEMASET** bulk data card).

See **'ASSIGN FEM='** Executive Control Command for the description of G-set and A-set.

S-set is is 6× (number of structural GRID points)+number of SPOINT points.

COEFFA — A real multiplication factor for matrix "MATRIXA". (Real, Default = 1.0)

MATRIXA — Character string that is the name of the matrix "MATRIXA". (Character) (See Remark 4)

SYMBOL — Character string either "+", "-", "*", "//" or blank where

"+"	represents addition
"-"	represents subtraction
"*"	represents multiplication
"//"	represents appending

(Character)

MATRIXB — Character string represents the name of the matrix "MATRIXB". Used only if SYMBOL is not blank. (Character)

<u>Remarks</u>:

1. The **ALTER** bulk data cards provide the means to perform certain matrix operations without modifying the program source code. These matrix operations are executed before the program invokes any disciplines (flutter, ASE, trim or dynamic loads analysis). Note that the **ALTER** bulk data card is not referred to by any other bulk data cards. Its existence in the Bulk Data Section "triggers" the program to perform the matrix operations. Multiple **ALTER** bulk data cards can be specified where the execution sequence of the matrix operation defined by each **ALTER** bulk data card is performed according to the ascending order of the entry STEP.

2. The execution of these **ALTER** bulk data cards are performed after the computation of the engineering module that is specified by the MODULE entry is completed where:

   MODULE = "FEM". The Matrices exist on the run-time database include those imported by **DMI** and **DMIG** bulk data cards, **'ASSIGN FEM='** and **'ASSIGN MATRIX='** Executive Control Commands.

   MODULE = "SPLINE". The execution of these **ALTER** bulk data card after the computation of the SPLINE module is completed. The matrixes exist on the run-time database include the SPLINE matrix (called UGTKG) and those of the control surface modes and **LOADMOD** (generated by the **LOADMOD** bulk data card).

3. The resulting matrix is computed based on the following equation:

   [RESULT] = [“OPERATR” [(COEFFA) [MATRIXA]]] “SYMBOL” [MATRIXB]

   For example,

   [SMHH] = [TRNS [(2.0) [SPHI]]] * [MGH]

4. The matrix [MATRIXA] (and [MATRIXB] if SYMBOL ≠ blank) must already exist on the run-time database. Note that if the matrix [RESULT] exists on the run-time database, it will be replaced by the resulting new matrix.

5. The following are examples of the applications using the **ALTER** bulk data cards to add mass to the generalized mass matrix such as:

   [SMHH] = [SMHH] + [SPHI]^T [DELTAM] [SPHI]

   where [SMHH] is the symmetric generalized mass matrix.
   [SPHI] is the symmetric modal matrix.
   Note that [SMHH] and [SPHI] are imported by the **‘ASSIGN FEM=’** Executive Control Command.
   [DALTAM] contains the mass in the G-set d.o.f. that is to be added into the generalized mass matrix. Note that DELTAM can be defined by the **DMIG** bulk data card.

   The following three **ALTER** bulk data cards can be used to perform the above task.

ALTER	1	FEM	TMP	TRNS	1.0	SPHI	*	DELTAM	

ALTER	2	FEM	TMP		1.0	TMP	*	SPHI	

ALTER	3	FEM	SMHH		1.0	SMHH	+	TMP	

ATMOS

User Defined Atmospheric Table

Description: Defines the altitude-speed of sound-density-temperature relationship as a tabular function for matched point flutter analysis.

Format and Example:

1	2	3	4	5	6	7	8	9	10
ATMOS	IDATM	AMMUNIT	AMLUNIT	AMTUNIT					CONT
CONT	ALT1	SOUND1	DEN1	TEMP1	ALT2	SOUND2	DEN2	TEMP2	CONT
CONT	ALTi	SOUNDi	DENi	TEMPi	-etc-				

ATMOS	10	SLUG	FT	F					+AT1
+AT1	-10000.	1300.	.0028	50.	0.	1100.	.0023	45.	+AT2
+AT2	10000.	1000.	.0020	40.					

Field	Contents
IDATM	Identification number. (Integer > 0)
AMMUNIT	Units of mass used in the table. Must be one of “SLIN”, “SLUG”, “LBM”, “G”, “KG”, “LBF/”, “N/”, or “NONE”. (Character, Default = “NONE”). (See Remark 2)
AMLUNIT	Units of length used in the table. Must be one of “IN”, “FT”, “M”, “MM”, “CM”, “KM”, or “NONE”. (Character, Default = “NONE”) (See Remark 3)
AMTUNIT	Units of temperature. Must be “F’, “C”, or “R”. (Character)
ALTi	Altitudes. Units must be in AMLUNIT. (Real) (See Remark 4)
SOUNDi	Speeds of sound. Units must be in AMLUNIT/sec. (Real > 0) (See Remark 4)
DENi	Densities. Units must be in AMMUNIT/AMLUNIT^3. (Rea l> 0) (See Remark 4)
TEMPi	Temperature. Units must be in AMTUNIT. (Real) (See Remark 5)

Remarks:

1. IDATM is referred by the **FIXMATM** bulk data card for matched point flutter analysis.

2. “SLUG” = slug, “SLIN” = slinch, “LBM” = pound mass, “G” = gram, and “KG” = kilogram. If AMMUNIT = “NONE” is specified, it is assumed that the mass units is the same as the FTMUNIT entry in the **FIXMATM** bulk data card. Note that 12 slugs = 1 slinch.

3. SLUG" = slug, "SLIN" = slinch (slinch = slug × 12.0), "LBM" = pound mass, "G" = gram, "KG" = kilogram, "LBF/" = $\left(\frac{\text{lbf}}{\text{AMLUNIT}}\right)\text{sec}^2$ and "N/" = $\left(\frac{\text{Newton}}{\text{AMLUNIT}}\right)\text{sec}^2$. If AMMUNIT = "NONE" is specified, it is assumed that the units of length is the same as the FTMUNIT entry in the **FIXMATM** bulk data card.

   "FT" = foot, "IN" = inch, "MM" = millimeter, "CM" = centimeter, "M" = meter, and "KM" = kilometer. If AMLUNIT = "NONE" is specified, it is assumed that the mass unit is the same as the FTLUNIT entry in the **FIXMATM** bulk data card.

   If AMMUNIT = "NONE", AMLUNIT must also be "NONE", and vice versa.

   Note that AMMUNIT = LBF/, or N/ is the options to specify the units of mass in terms of units of force and length (AMLUNIT). For instance, AMMUNIT = LBF/ and AMLUNIT = IN imply that the units of mass = $\left(\frac{\text{lbf}}{\text{inch}}\right)\text{sec}^2$ = slinch. It should be noted that there is a slash ("/") attached to the force unit.

   Length and mass units associated with altitudes, speeds of sound and densities will be converted from those of AMMUNIT and AMLUNIT to FTMUNIT and FTLUNIT, respectively.

4. Altitudes listed in the table must be in the <u>ascending</u> order, i.e., from low altitude to high altitude. The dynamic pressures associated with the speeds of sound and densities listed in the table must be in <u>descending</u> order.

5. Temperatures are not used in the flutter analysis. They are for reference purposes only.

ATTACH Aerodynamic Box-To-Grid Spline Attachment

Description: Defines aerodynamic box(es) to be attached to a reference structural grid for splining.

Format and Example:

1	2	3	4	5	6	7	8	9	10
ATTACH	EID	MODEL	SETK	REFGRID					

ATTACH	1	WING	10	3					

Field	Contents
EID	Element identification number. (Integer > 0) (See Remark 2)
MODEL	Not used.
SETK	Identification number of **PANLST1, PANLST2** or **PANLST3** bulk data card used to identify the aerodynamic box ID's. (Integer > 0)
REFGRID	Reference structural grid point identification number. (Integer > 0) (See Remark 3)

Remarks:

1. For an aerodynamic component not represented in the structural model, **ATTACH** is used to translate the displacements and loads between a structural grid point and the aerodynamic component.

   A typical example is an underwing store that is modeled structurally by a concentrated mass at a single structural grid point. In this case, the respective aerodynamic model of the underwing store will be splined to this single structural grid point by **ATTACH**. The resulting motion on the aerodynamic boxes will be a rigid body motion that follows the motion of this single structural grid point.

2. **EID** is used only for error messages.

3. The translational and rotational d.o.f. at the reference grid point defines a rigid body type of motion of the aerodynamic component.

BLKLAY

Layering of Blocks for Overset Strategy

Description: Defines the block layers for proper set-up of the overset mesh.

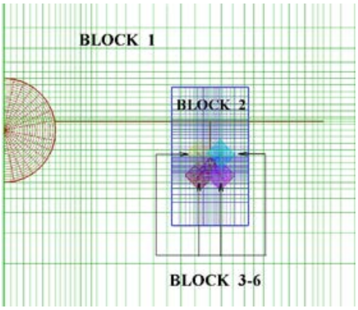
Format and Example:

1	2	3	4	5	6	7	8	9	10
BLKLAY	ID	G1	G2	G3	G4	G5	G6	G7	CONT
CONT	G8	-etc-							

BLKLAY	1	1	2	3	3	3	3	2	+BC
+BC	2	2							

Field	Contents
ID	Identification number. (Integer > 0) (See Remark 1)
G*i*	List of layer number for corresponding **BLOCK/BLOCKT** or **BLOCK1** listed in the **BLKSEQ** bulk data card. (Integer $\neq$ 0) (See Remark 2)

Remarks:



1. The **BLKLAY** is an optional input. ID should match with the ID of **BLKSEQ** bulk data card, and the **BLKLAY** bulk data card has to be used together with the **BLKSEQ** bulk data card.
2. The **BLKLAY** bulk data card provides more information to control the overset mesh strategy in addition to the **BLKSEQ** bulk data card. The **BLKLAY** bulk data card specifies the layer number for the corresponding **BLOCK/BLOCKT** or **BLOCK1** listed in the **BLKSEQ** bulk data card. If no **BLKLAY** bulk data card exists, the overset strategy is performed only according to the block sequence.

   With the BLKLAY bulk data card, the overset strategy will perform another additional checks for based on the absolute value of the layering number of each block. The blocks with the same layer number will not directly communicate with each other, a block can only be hole-cut by blocks with a larger layer number, and the far field boundary of a block can only receive donor cell flowfield information from blocks with a smaller layer number. For example, in the case of a fuselage-wing configuration with underwing store and fins shown in the figure above, the global block that contains the fuselage and main wing can should be given the lowest layer number 1. The mesh domain of the store block is bigger than those of the fin blocks, so the store block is layer 2, and all the fin blocks have layer number 3. An exception is applied to the hole-cutting rules for cells close to a surface (BODY7 or CAERO7): the cells adjacent to a surface in a block with a lower layer number can cut into a block with a higher layer number, causing those cells in the higher layer number block to become receiver cells. Additionally, cells in the outer boundary layer which are adjacent to a surface are prevented from becoming receiver cells. An additional restriction can be placed on block communication by specifying a negative layer number, which means that the given block can only communicate with blocks where the absolute value of the layer number is one less than or one greater than the absolute value of the given block's layer number. For example, specifying the fin block layer number as "-3" prevents it from communicating directly with the main block (whose layer number is 1).

.

BLKMPI Block Assignment for MPI Load Balance Strategy

Description: Assign different blocks to different MPI processes for load balancing of time-domain ZEUS analysis (MLOADS,GLOADS,NLFLTR and NANSI modules) with Message-Passing-Interface (MPI). The total number of MPI processes is specified **MPICPU** executive control command.

Format and Example:

1	2	3	4	5	6	7	8	9	10
BLKSEQ	ID	G1	G2	G3	G4	G5	G6	G7	CONT
CONT	G8	-etc-							

BLKSEQ	1	-6	7	8	9	10	-4	15	+BC
+BC	16	16							

Field	Contents
ID	Identification number. (Integer > 0) (See Remark 1)
G*i*	List the index of the MPI process or the number of MPI processes to handle the computation of each block of mesh. (Integer) (See Remark 2)

Remarks:

1. ID is not referred to by any other bulk data card and is only used for error output.
2. The **BLKMPI** bulk data card is used to specify one or more MPI processes for handling each BLOCK of mesh. Based on the sequence defined by **BLKSEQ** bulk data card, each block is assigned an integer. A negative integer means the corresponding block is handled by a number of MPI processes and that number is the absolute value of the integer. A positive integer specifies the index of the single MPI process to handle the corresponding block. The index of MPI process has to be in increasing order starting from 1 and up to the number of total MPI processes defined by the **MPICPU** executive command. The index of a MPI process denoted by any positive integer appearing right after a negative integer has to consider the fact that more than one MPI processes have been used exclusively for the previous block. If the same MPI process index has been assigned to more than one BLOCKs in sequence, that means the sequence of blocks are handled by the same MPI process. For any block assigned with multiple MPI processes such as N>1, ZEUS cuts the block into N sub-blocks along the Z direction with each sub-block having roughly the same number of grid cells in Z direction. Please note that if any block is assigned with multiple MPI processes, residual smoothing has to be turned off by setting the flag LVRSMOO in the **MKPARAM** bulk data card to be 0. The reason is that the block handled by more than one MPI processes will have residual smoothing applied to individual sub-blocks and that might cause convergence issue for the Euler solver.

BLKSEQ

Block ID Sequence for Overset Strategy

Description: Defines the block sequence for proper set-up of the overset mesh.

Format and Example:

1	2	3	4	5	6	7	8	9	10
BLKSEQ	ID	G1	G2	G3	G4	G5	G6	G7	CONT
CONT	G8	-etc-							

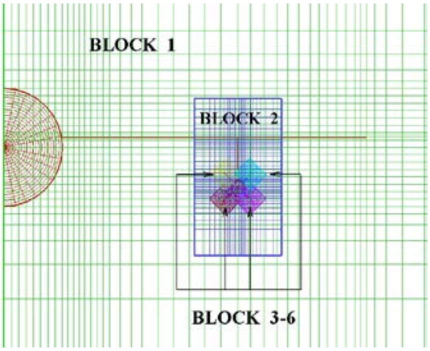
BLKSEQ	1	1	2	3	4	5	6	7	+BC
+BC	8	9							

Field	Contents
ID	Identification number. (Integer > 0) (See Remark 1)
G*i*	List of **BLOCK**, **BLOCKT** or **BLOCK1** identification numbers. (Integer > 0) (See Remark 2)

Remarks:

1. ID is not referred to by any other bulk data card and is only used for error output.

   For cases with multiple blocks (including **BLOCK/BLOCKT/BLOCK1** bulk data card), the overset strategy needs to know the sequence of the blocks. It is critical to put the bigger blocks ahead of the smaller blocks for the overset strategy to work properly. So, whenever there is an overlap between two blocks, the bigger block should be placed ahead of the smaller block. For example, in the case of a fuselage-wing configuration with underwing store and fins shown in the following figure, the global block that contains fuselage and main wing, obviously is the first in the sequence. The store block is bigger than fin blocks, so the store block follows. Since the fin blocks are of equal size, the relative position of the fin blocks doesn't matter. Therefore, the blocks are arranged as: 1) Global block, 2) Store block, 3) Four fin blocks.



2. The **BLKSEQ** bulk data card is used to specify this block sequence. If no **BLKSEQ** bulk data card exists, the block sequence is automatically performed according to the ascending order of the identification numbers of the **BLOCK/BLOCKT** bulk data cards followed by the ascending order of the identification numbers of the **BLOCK1** bulk data cards. Because the identification numbers

of **BLOCK/BLOCKT** and **BLOCK1** must be unique, the list of identification numbers in the **BLKSEQ** bulk data card should have no duplicate ones.

3. If there is no **BLOCK1** bulk data card, the **BLKSEQ** bulk data card is not needed if the user assigns the identification numbers of the **BLOCK/BLOCKT** bulk data card according to the proper sequence from bigger grid blocks to smaller grid blocks. However, if **BLOCK1** bulk data card exists, by default (see remark 2), the body-fitted grid blocks (defined by **BLOCK1**) is processed after the Cartesian grid blocks (defined by **BLOCK/BLOCKT**) in the overset mesh scheme. In this case when the body-fitted grid block is the bigger grid block, relatively to other smaller Cartesian grid blocks, this can lead to large error in the interpolation of the flow solutions in the overlapping region. To circumvent this problem, the user must list the proper sequence of the **BLOCK/BLOCKT/BLOCK1** identification numbers in the **BLKSEQ** bulk data card.

BLOCK

Automatically Generates a Rectangular Block of Mesh

Description: Defines the outer boundary of a rectangular block of mesh and selects a set of bodies, horizontal surfaces and vertical surfaces which can be fitted into such a block of mesh.

Format and Example:

1	2	3	4	5	6	7	8	9	10
BLOCK	IDBLK	IACORD	XLEAD	XTRAIL	YLEFT	YRIGHT	ZBOT	ZTOP	CONT
CONT	YL_1	YR_1	$ID1_1$	$ID2_1$	$ID3_1$	$ID4_1$	$ID5_1$	$ID6_1$	CONT
CONT	YL_2	YR_2	$ID1_2$	$ID2_2$	$ID3_2$	$ID4_2$	$ID5_2$	$ID6_2$	CONT
CONT	...	...	...	...	...	...	...	...	...

BLOCK	101	10	-500.0	1000.0	-700.0	700.0	-300.0	300.0	+B1
+B1	-100.0	100.0	300	701					+B2
+B2	100.0	200.	40	31	1001				+B3
+B3	200.0	340.0	3001						

Field	Contents
IDBLK	The absolute value of IDBLK is the identification number. Among all **BLOCK**, **BLOCKT** and **BLOCK1**, all identification numbers must be unique. Note that IDBLK can be a negative integer. In this case, the residual smoothing for this block of mesh is deactivated even if the entry LVRSMOO = 1in the **MKPARAM** bulk data card is specified. If any other bulk data card refers to this **BLOCK** bulk data card, it must refer to the absolute value of IDBLK. (Integer $\neq 0$) (See Remark 1)
IACORD	Identification number of an **ACOORD** bulk data card to define a local coordinate system. In this local coordinate system, a block of mesh is generated. (Integer ≥ 0) (See Remark 2)
XLEAD, XTRAIL	The upstream and downstream boundaries of the rectangular mesh, respectively. Note that XTRAIL > XLEAD. (Real) (See Remark 3)
YLEFT, YRIGHT	The left and right boundaries of the rectangular mesh, respectively. Note that YRIGHT > YLEFT. (Real) (See Remark 4)
ZBOT, ZTOP	The bottom and top boundaries of the rectangular mesh, respectively. Note that ZTOP > ZBOT. (Real)
YL_i, YR_i	The Y location of the left and right of a strip to define a spanwise zone, called the Y-

Zone, within which a set of CAERO7, BODY7, GAP, GAP1, and/or GAPZ macroelements are located. Note that $YR_i > YL_i$ and $YR_i = YL_{i+1}$.

As an alternative YL_i or YR_i can be an integer that is the identification number of a **YZONEY** bulk data card. In this **YZONEY** bulk data card, the Y location of the left/right of a strip is specified by the most left/right surface grid point of a CAERO7 or BODY7 Macroelement. (Real or Integer) (See Remark 5)

IDj_i Identification number of a **CAERO7**, **BODY7**, **GAP**, **GAP1**, or **GAPZ** bulk data card that is located within the ith Y-Zone.

Note that $ID1_i$ can be a character string begins with a character "$" then followed by an integer such as "$1000". In this case, the integer refers to the identification number of the **SET1** bulk data card to list the identification numbers of a set of **CAERO7**, **BODY7**, **GAP**, **GAP1**, or **GAPZ** bulk data cards. In this way, the number of **CAERO7**, **BODY7**, **GAP**, and **GAP1**, or **GAPZ** macroelements located in the ith Y-Zone is unlimited. (Integer ≥ 0 or Character) (See Remark 6)

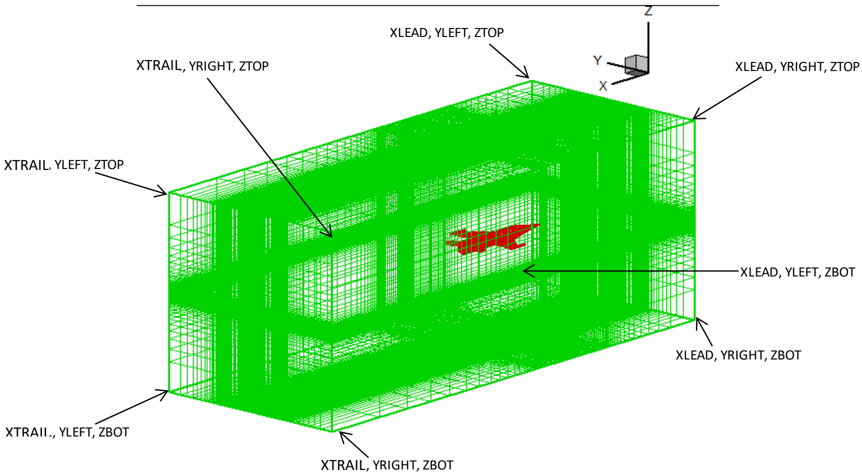
Remarks:

1. The purpose of the **BLOCK** bulk data card is to automatically generate a rectangular block of mesh. Within this rectangular block of mesh, a set of CAERO7, BODY7, GAP, GAP1, and/or GAPZ macroelements can be fitted into a "sheared" Cartesian Mesh. This block of mesh is automatically generated by growing the gridlines from the surface mesh defined by the CAERO7, BODY7, GAP, GAP1, and GAPZ to the outer boundary of the block. Note that this growth rate of the gridlines is specified in the **MESHPRM** bulk data card.

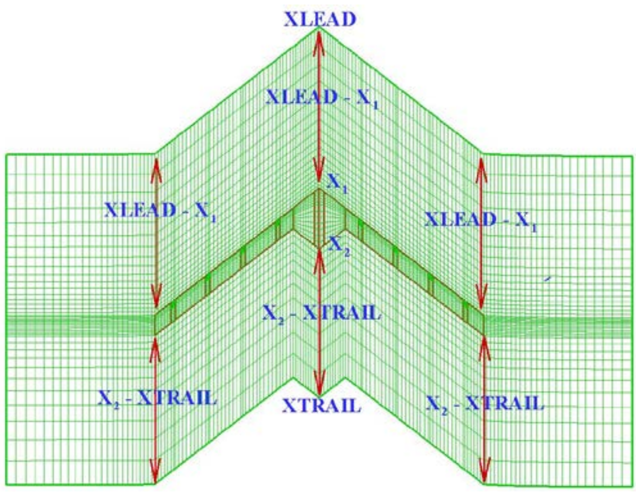
2. All surface meshes of the CAERO7, BODY7, GAP, GAP1, and GAPZ macroelements are transformed in the local coordinate system defined by the **ACOORD** bulk data card. After the block of mesh is generated by the automated mesh generation scheme, all grid points of the block of mesh are then transformed back to the global coordinate system. For a vertical lifting surface (90° dihedral angle) with non-zero sweep angle, it is recommended that it be included in a different **BLOCK** bulk data card with an **ACOORD** bulk data card whose X-Y plane being located in the plane of the vertical surface. On this local X-Y plane, the vertical surface becomes a horizontal surface so that a sheared mesh can be generated to accommodate the sweep angle of the vertical surface. This is because the automated mesh generation scheme can only shear the mesh on the X-Y plane and the Y-Z plane to accommodate the sweep angle and the dihedral angle, respectively, of a lifting surface. However, it cannot generate a sheared mesh on the X-Z plane for the swept vertical tail. If the user insists on defining a lifting surface with 90° dihedral angle (for instance, a vertical tail or pylon) to be located on a plane parallel to the X-Z plane, the sweep angle will be reset to zero by the program in order to fit into the mesh on the X-Z plane. Thus, the vertical surface is approximated as a rectangular surface without sweep angle.

3. XLEAD, XTRAIL, YRIGHT, YLEFT, ZBOT, and ZTOP are defined in the local coordinate system specified by the **ACOORD** bulk data card and they jointly define the outer boundary of the rectangular block of mesh. See the figure below.

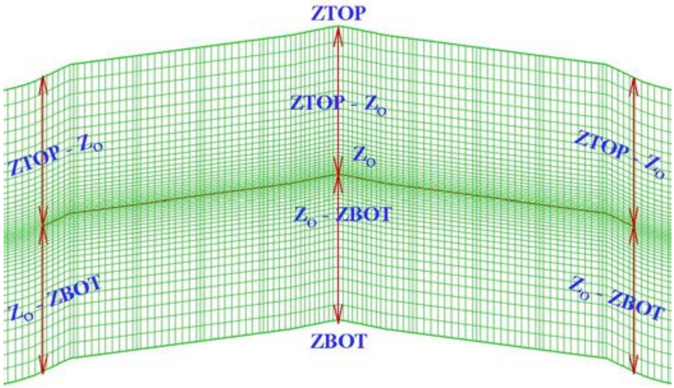
   It is suggested that, for the global block of mesh, the length from XLEAD to the nose of configuration should be greater than twice of the total length of the configuration. The length from the trailing edge of the configuration to XTRAIL should be greater than 5 times of the total length of the configuration. And (YRIGHT-YLEFT) should be greater than 2 times the span of the configuration. For the embedded block of mesh, these conditions can be relaxed.



As an option, if the entry XGROWTH in the **MESHPRM** bulk data card is negative, then the outer boundary projected on the x-y plane is not rectangular. Rather, it becomes a block of mesh shown in the following figure:



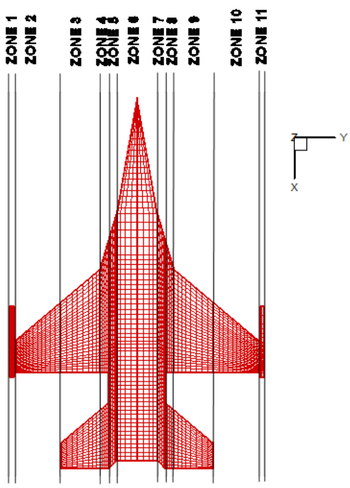
Likewise, if the entry ZGROWTH in the **MESHPRM** bulk data card is negative, then the outer boundary projected on the y-z plane becomes:



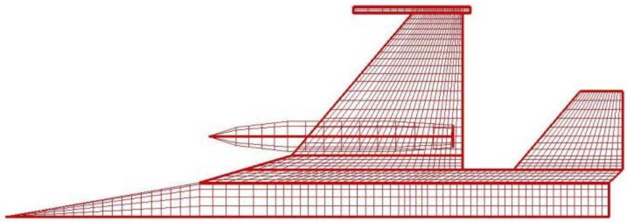
4. For all horizontal surfaces in the block mesh, their respective CAERO7 macroelements must have the consistent direction from root to tip (defined by the YRL and YTL entries, respectively, in the **CAERO7** bulk data card) with the direction from YLEFT to YRIGHT of the block of mesh. Thus, for a lifting surface located on the left-hand side, it is recommended that its y-location of root be specified in the entry YTL and y-location of tip in the entry YRL.

5. The automated mesh generation scheme for each block of mesh uses a Y-Zone technique which divides the selected components (CAERO7, BODY7, GAP, GAP1, and GAPZ) into several Y-Zones. To set up these Y-Zones, the users must first project the surface boundaries of all selected components of the **BLOCK** bulk data card onto the X-Y plane defined by the **ACOORD** bulk data card. Then the Y-Zone technique can automatically generate a block of mesh using a line-tracing method across all Y-Zones. Note that the Y-Zones must go from the left to the right. Also, note that the direction of left to right for the Y-Zones has to be consistent with the direction of root to tip in the definition of the **CAERO7** bulk data cards. Otherwise, the output of pressure on the upper surface of the wing will be displayed on the lower surface and vice-versa. For a whole aircraft configuration with tip launchers, the definition of Y-Zones is shown as follows.

- **ZONE 1: Left launcher**
- **ZONE 2: Output left wing**
- **ZONE 3: Inboard left wing and outboard left horizontal tail**
- **ZONA 4: Outboard left strake and inboard left horizontal tail**
- **ZONE 5: Inboard left stake**
- **ZONE 6: Fuselage**
- **ZONE 7: Inboard right strake**
- **ZONE 8: Outboard right strake and inboard right horizontal tail**
- **ZONE 9: Inboard right wing and outboard right horizontal tail**
- **ZONE 10: Outboard right wing**
- **ZONE 11: Right launcher**



6. The selection of all CAERO7 and BODY7 cannot be arbitrary; they must appear to be fitted into a rectangular mesh. For example, fuselage, wing, horizontal tail, and pylon (with no swept angle) can be fitted into a single block of rectangular mesh. Under wing store can be fitted into another block of rectangular mesh. The rule of this selection is to project all components in a block onto the X-Y plane. If no overlapping among components appears, those selected components can be fitted into a single block of mesh such as the whole aircraft configuration with tip launcher shown above. For a half aircraft with under wing store as the one shown below, there is an overlapping region between the wing and the store on the projected X-Y plane. For this case, two **BLOCK** bulk data cards must be used to model such a configuration; one for the half aircraft configuration without the under wing store and the other for the under wing store.



BLOCKT One Block of Mesh for T-Tail Model

Description: Defines the outer boundary of a rectangular block of mesh containing a T-tail component.

Format and Example:

1	2	3	4	5	6	7	8	9	10
BLOCKT	IDBLK	IACORD	XLEAD	XTRAIL	YLEFT	YRIGHT	ZBOT	ZTOP	CONT
CONT	HT	YCENTER	ZCENTER	TOL	IDELET	KEEPLE	KEEPTE		CONT
CONT	YL_1	YR_1	$ID1_1$	$ID2_1$	$ID3_1$	$ID4_1$	$ID5_1$	$ID6_1$	CONT
CONT				…					CONT
CONT	YL_{NY}	YR_{NY}	$ID1_{NY}$	$ID2_{NY}$	$ID3_{NY}$	$ID4_{NY}$	$ID5_{NY}$	$ID6_{NY}$	CONT
CONT	VT	ZVLOW	ZVTOP						CONT
CONT	ZB_1	ZT_1	$IDZ1_1$	$IDZ2_1$	$IDZ3_1$	$IDZ4_1$	$IDZ5_1$	$IDZ6_1$	CONT
CONT				…					CONT
CONT	ZB_{NZ}	ZT_{NZ}	$IDZ1_{NZ}$	$IDZ2_{NZ}$	$IDZ3_{NZ}$	$IDZ4_{NZ}$	$IDZ5_{NZ}$	$IDZ6_{NZ}$	

BLOCKT	100	3	-200.0	200.0	-100.0	100.0	-150.0	150.0	+B
+B	HT	0.0	1.0	0.07		2	1		+B
+B	-3.0	0.0	1000						+B
+B	0.0	3.0	2000						+B
+B	VT								+B
+B	-10.0	-5.0	3000						+B
+B	-5.0	-2.0	4000						+B
+B	-2.0	0.0	5000						

Field	Contents
IDBLK	The absolute value of IDBLK is the identification number. Among all **BLOCK**, **BLOCKT** and **BLOCK1**, all identification numbers must be unique. Note that IDBLK can be a negative integer. In this case, the residual smoothing for this block of mesh is deactivated even if the entry LVRSMOO = 1in the **MKPARAM** bulk data card is specified. If any other bulk data card refers to this **BLOCKT** bulk data card, it must refer to the absolute value of IDBLK. Note also that the **BLOCKT** bulk data card can only be used for full-span aerodynamic models (XZSYM=YES or H2F in the **AEROZ** bulk data card). (Integer $\neq 0$) (See Remark 1)
IACORD	Identification number of an **ACOORD** bulk data card to define a local coordinate system. In this local coordinate system, a block of mesh is generated. (Integer ≥ 0) (See Remark 2)
XLEAD, XTRAIL	The upstream and downstream boundaries of the rectangular mesh, respectively. Note that XTRAIL > XLEAD. (Real) (See Remark 3)
YLEFT, YRIGHT	The left and right boundaries of the rectangular mesh, respectively. Note that YRIGHT > YLEFT. (Real) (See Remark 4)
ZBOT, ZTOP	The bottom and top boundaries of the rectangular mesh, respectively. Note that ZTOP > ZBOT. (Real)
HT	Character string that must be "HT" to indicate that the following input is for horizontal tails/bodies. (Character)
YCENTER, ZCENTER	*y* and *z* location (in the local coordinates defined by the **ACOORD** bulk data card) where the horizontal tail (or body) intersects with the vertical tail. (Real) (See Remark 5)
TOL	A fraction of REFC (defined in the **AEROZ** bulk data card) to delete one of the chordwise Y-lines in the flowfield (not on the surface mesh) where two chordwise Y-lines are located within TOL × REFC. (Real > 0.0, Default = 0.05)
IDELET	Identification number of a **DELLINE** bulk data card to delete an unwanted gridline. (Integer ≥ 0)
KEEPLE	Number of gridlines ahead of the leading edge to be kept even if the mesh size is smaller than TOL × REFC (Integer ≥ 0).
KEEPTE	Same as KEEPLE except for those gridlines behind the trailing edge (Integer ≥ 0).
YL_i, YR_i	The Y location of the left and right of a strip to define a spanwise zone, called the Y-Zone, within which a set of CAERO7, BODY7, GAP, GAP1, and/or GAPZ macroelements are located. Note that $YR_i > YL_i$ and $YR_i = YL_{i+1}$. As an alternative YL_i or YR_i can be an integer that is the identification number of a **YZONEY** bulk data card. In this **YZONEY** bulk data card, the Y location of the left/right of a strip is specified by the most left/right surface grid point of a CAERO7 or BODY7 Macroelement.
IDj_i	Identification number of a **CAERO7**, **BODY7**, **GAP**, **GAP1**, or **GAPZ** bulk data card that is located within the ith Y-Zone. Note that $ID1_i$ can be a character string begins with a character "$" then followed by an

integer such as "$1000". In this case, the integer refers to the identification number of the **SET1** bulk data card to list the identification numbers of a set of **CAERO7**, **BODY7**, **GAP**, **GAP1**, or **GAPZ** bulk data cards. In this way, the number of **CAERO7**, **BODY7**, **GAP**, and **GAP1**, or **GAPZ** macroelements located in the ith Y-Zone is unlimited. (Integer ≥ 0 or Character) (See Remark 6)

VT — Character string that must be "VT" to indicate that the following input is for the vertical tails.

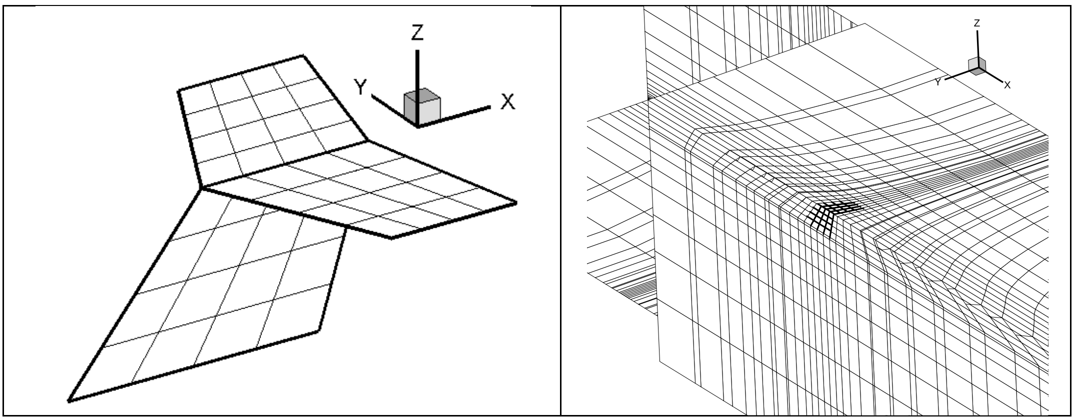
ZVLOW, ZVTOP — The lowest and highest z locations within which the gridline sizes along the z axis are determined by the spanwise cuts of the vertical surfaces. If ZVLOW and/or ZVTOP are blank, the lowest and highest points of the vertical tail are selected to be ZVLOW and ZVTOP, respectively. (Real or Blank) (See Remark 7)

ZB_i, ZT_i — Similar YL_i and YR_i except for the z locations of the Z-zone. The **BLOCKT** bulk data card first transforms the vertical tail to a horizontal plane. Then, on this horizontal plane, the ZBi and ZTi are defined so that the **YZONEY** bulk data card can be used. The root and tip definition of the vertical tail is based on the **CAERO7** bulk data card input, where the root of the **CAERO7** corresponds to the left-hand side, and the tip corresponds to the right-hand side.

$IDZJ_i$ — Same as IDj_i except refers to the identification number of a **CAERO7** bulk data card that models a vertical tail. Note that $IDZJ_i$ cannot refer to a **BODY7** bulk data card but can refer to the **GAP**/**GAP1** bulk data card. (Integer > 0 or character)

Remarks:

1. Modeling a T-tail configuration accurately using the **BLOCK** bulk data card requires the overset mesh; one **BLOCK** bulk data card models the horizontal tail and the other models the vertical tail. However, overset mesh sometimes may decrease the accuracy of the solution. Also, if both horizontal and vertical tails are included in one **BLOCK** bulk data card, in the computational mesh the vertical tail is approximated by an unwept rectangular surface which obviously decreases the accuracy of the solution. To circumvent this problem, the **BLOCKT** bulk data card can be used to generate one block of mesh to model the T-tail configuration. See figures below.

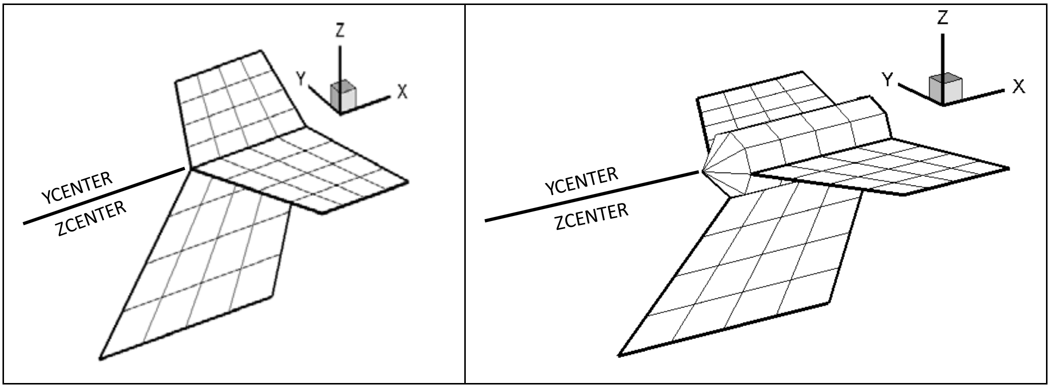


2. See remark 2 of the **BLOCK** bulk data card.

3. See remark 3 of the **BLOCK** bulk data card.

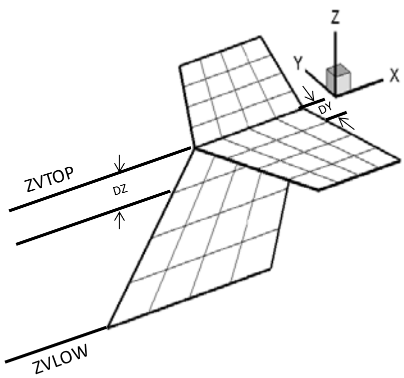
4. See remark 4 of the **BLOCK** bulk data card.

5. YCENTER and ZCENTER are defined in the local coordinates specified by the **ACOORD** bulk data card. They are the critical input based on which the automated mesh generation scheme generates a block of mesh. See figures below.



6. It is highly recommended that the chordwise divisions among all **CAERO7** and **BODY7** to model the horizontal tail and vertical tail be coherent. Otherwise, extra gridlines could be generated whose gridline sizes may not be smooth.

7. The gridline size along the z axis is critical to the solution accuracy of the horizontal tail because those gridlines are normal to the horizontal tail. The figure below shows that the gridline sizes (DZ) between ZVLOW and ZVTOP are determined by the spanwise divisions of the vertical tail. In order to have small gridline sizes normal to the horizontal tail, the spanwise divisions of the vertical tail should be refined. Similarly, the gridlines along the y-axis (DY) are determined by the spanwise divisions of the horizontal tail and should be refined because those gridlines are normal to the vertical tail.



BLOCK1 Automatically Generates a Body-Fitted Mesh

Description: Generates a body-fitted O-type mesh around a BODY7 macroelement.

Format and Example:

1	2	3	4	5	6	7	8	9	10
BLOCK1	IDBLK1	IDBODY	IACORD	XLEAD	XTRAIL	R	NR	AEFACT	CONT
CONT	XGROWTH	NSOURCE	W	TOL	ITMAX	RATIO	XWAKE	VISRAT	

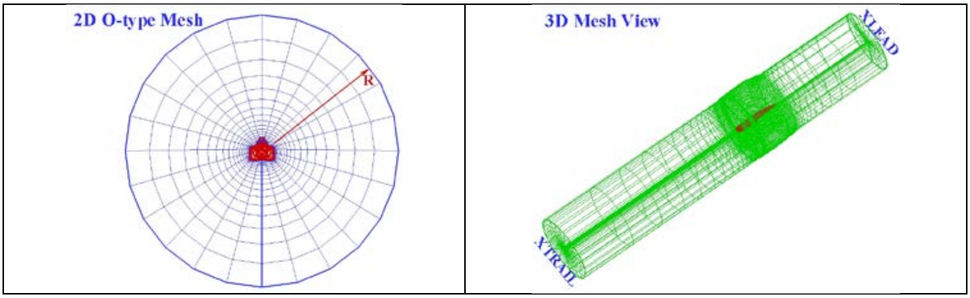
BLOCK1	2	4001	4001	-100.0	100.0	100.0	21	COS	+B1
+B1	1.3	1	1.66666	1.E-05	4000	500.0	1.5	1.2	

Field	Contents
IDBLK1	The absolute value of IDBLK is the identification number. Among all **BLOCK**, **BLOCKT** and **BLOCK1**, all identification numbers must be unique. Note that IDBLK can be a negative integer. In this case, the residual smoothing for this block of mesh is deactivated even if the entry LVRSMOO = 1in the **MKPARAM** bulk data card is specified. If any other bulk data card refers to this **BLOCK1** bulk data card, it must refer to the absolute value of IDBLK. (Integer $\neq$ 0) (See Remark 1)
IDBODY	Identification number of the body around which the body-fitted mesh is generated. (Integer $\geq$ 0)
IACORD	Identification number of an **ACOORD** bulk data card to define a local coordinate system where the block of mesh is located. If IACORD=0, the **ACOORD** bulk data card referred to by the **BODY7** bulk data card is used. (Integer $\geq$ 0)
XLEAD, XTRAIL	The upstream and downstream boundaries of the body-conforming mesh, respectively. Note that XTRAIL > XLEAD. (Real)
R	The distance between the body surface and the outer boundary of the O-type mesh in the radial direction. (Real > 0.0)
NR	The number of grid points along the radial direction. NR can be a negative integer. For this case, the negative sign triggers the program to generate NR/3 additional grid points along the radial direction inside of the body. The flow solution at those points inside of the body is always nullified by the Euler solver. This negative sign is recommended for the case where the mesh generated by the **BLOCK1** bulk data card is an embedded mesh in the other block of mesh. The nullified flow solution inside of the body can ensure zero-flow solution in the overlapping region inside of the body of the other block of mesh. (Integer > 2 or <-2)
AEFACT	Integer or character string. If AEFACT is an integer: For AEFACT = 0 or blank, the grid point distributions along each radial line are

	determined by the program. For AEFACT > 0, AEFACT is the identification number of an **AEFACT** bulk data card to list NR number of grid point distributions along each radial line in terms of the percentages of the total length of the radial line. There must be NR number of values listed in this **AEFACT** bulk data card which must start from 0.0 and end at 100. If AEFACT is a character string, AEFACT must be "COS". In this case, a cosine distribution of the grid points along each radial, line is automatically calculated by the program.(Integer or Character)
XGROWTH	Growth rate of the mesh along the x-axis. (Real > 1.0, Default = 1.3)
NSOURCE, W, TOL, ITMAX, RATIO	Parameters to control the 2-D mesh generation around each section of the body. (See Remark 2)
XWAKE	The length of the wake region behind the truncated-end body divided by the body diameter at the end of the body. (Real ≥ 0 .0, Default=1.0) (See Remark 3)
VISRAT	A factor to increase VIS2 and VIS4 parameters defined in **MKPARAM** bulk data card. VISRAT allows the user to increase the numerical artificial damping only for the current body-fitted mesh if a convergence problem occurs in this body-fitted block. Note that VIS2 and VIS4 remain the same for other blocks of mesh. (Real ≥ 1.0, Default=1.0)

Remarks:

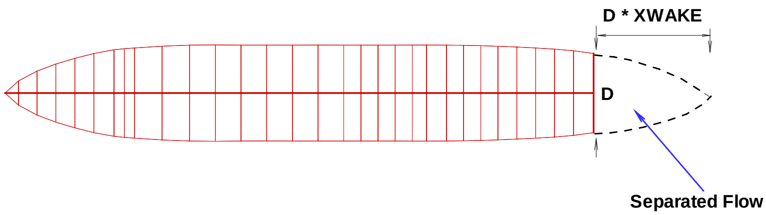
1. The purpose of the **BLOCK1** bulk data card is to automatically generate a body-fitted mesh around a body. First, a 2-D O-type of mesh around each body cross section is generated by solving a Laplace or Possion equation. A final 3-D mesh is obtained by smoothly connecting those 2-D meshes along the x-axis and extending beyond the leading edge and trailing edge of the body to XLEAD and XTRAIL, respectively. Shown below is a 2-D and 3-D view of such body-fitted mesh. Note that among all **BLOCK, BLOCKT** and **BLOCK1** bulk data cards, no duplicated identification numbers are allowed.



Unlike the **BLOCK/BLOCKT** bulk data card that can automatically generates a Cartesian mesh to contain multiple wings and body components, the O-type mesh generated by the **BLOCK1** bulk data card must contain only one BODY7 macroelement and cannot contain other body or wing. Therefore, for a wing-body configuration, if the body mesh is generated by the **BLOCK1** bulk data card, the overset mesh scheme must be used where the wing mesh (generated by a **BLOCK/BLOCKT** bulk data card) and the body mesh have sufficient overlapping region to allow the communication of the flow solutions between the wing mesh (generated by **BLOCK/BLOCKT** bulk data card) and body mesh (generate by the **BLOCK1** bulk data card).

2. NSOURCE is an integer and must be 0 or 1. If NSOURCE = 1, Possion equation is solved to generate the O-type of mesh (Default). Otherwise, Laplace equation is used. *W* is the relaxation number to solve the Possion equation (0.0 < REAL < 2.0). TOL is the convergence limit for Possion iteration (Real > 0.0, Default = 1.E-05), and ITMAX is the maximum iteration number for Possion iteration (Integer > 0, Default = 4000). RATIO is a threshold value to prevent the failure of 2-D mesh generation (Real > 0.0, Default = 500.0).

3. Behind a truncated-end body, separated flow may occur which is normally unsteady in nature. This unsteady separated flow may give convergence problem to the Euler solver. To circumvent this problem, it is assumed that the surface mesh of the body-fitted mesh be extended by the length of D × XWAKE, where *D* is the diameter at the end of the body, to represent the streamline outside the separated flow region. Thereby, the region inside the streamline is not a part of the computational domain. See the figure below.



BODY7 Aerodynamic Body Component

Description: Defines the surface mesh of an aerodynamic body macroelement for a body-like component.

Format and Example:

1	2	3	4	5	6	7	8	9	10
BODY7	BID	LABEL	IPBODY7	ACOORD	NSEG	IDMESH	FACTDY	FACTDZ	CONT
CONT	ZBOT	ZTOP							

BODY7	4	BODY	2	8	1	20	1.1	1.2	+B
+B	-10.	15.							

Field	Contents
BID	Identification number. (Integer > 0)
LABEL	An arbitrary character string (up to 8 characters) used to define the body. (Character)
IPBODY7	Identification number of **PBODY7** bulk data card (specifying inlet aerodynamic boxes) (Integer ≥ 0 or Blank, Default = 0) (See Remark 1)
ACOORD	Identification number of **ACOORD** bulk data card (specifying body center line location and orientation) (Integer ≥ 0 or Blank, Default = 0) (See Remark 2)
NSEG	Must be 1 or -1. For NSEG = 1, the grid lines projected on the Y-Z plane in the wing-body region is automatically determined by the circumferential points specified by the **SEGMESH** bulk data card. For NSEG = -1, deactivate the automatic grid-line generation on the projected Y-Z plane. (Integer) (See Remark 3)
IDMESH	Identification number of **SEGMESH** bulk data card (specifying body segments). (Integer > 0) (See Remark 4)
FACTDY	To control the number of spanwise X-lines on the X-Y plane for modeling the width of the rectangular box that embodies the BODY7 macroelement. Used only if the **BODY7** bulk data card is referred to by the **BLOCK/BLOCKT** bulk data card. (Real>0.0, Default=1.0) (See remark 5)
FACTDZ	To control the number of horizontal X-lines on X-Z plane for modeling the height of the rectangular box that embodies the BODY7 macroelement. Used only if the **BODY7** bulk data card is referred to by the **BLOCK/BLOCKT** bulk data card. (Real>0.0, Default=1.0)

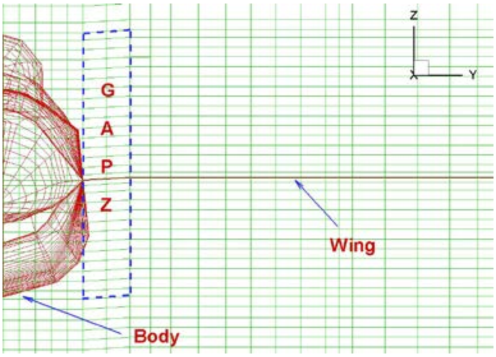
ZBOT	The Z location of the bottom of the rectangular box that embodies the actual surface of the body. (Real, Default = the lowest *Z* location among all surface grid points of the body) (See Remark 6)
ZTOP	The Z location of the top of the rectangular box that embodies the actual surface of the body. ZBOT and ZTOP are defined in the local coordinates specified by the **ACOORD** bulk data card referred to by the **BLOCK** bulk data card in which the body is modeled; not in the **ACOORD** bulk data card referred to by the **BODY7** bulk data card. (Real, Default=the highest *Z* location among all surface grid points of the body).

Remarks:

1. If **IPBODY7** is zero or blank, no **PBODY7** bulk data cards are needed.

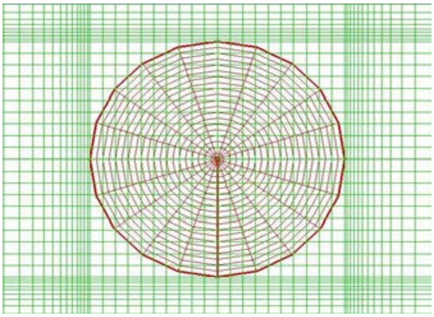
2. The X-axis specified by the **ACOORD** bulk data card defines the centerline of the body macroelement. If **ACOORD** is zero, the X-axis of the basic coordinate system is used.

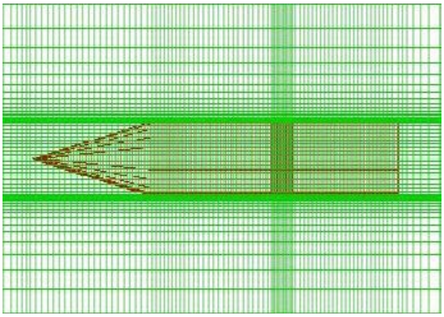
3. For a body-wing configuration shown in the figure below, the gridlines between the top and the bottom of the body is automatically determined by the circumferential points (defined by the **SEGMESH** bulk data card) if NSEG = 1. These gridlines may not have a good grid spacing for the wing. To control the grid spacing, the user can specify NSEG = -1 that deactivates the automatic grid line generation and use the **GAPZ** bulk data card to specify the desired grid lines.



4. **BODY7** generates a set of body boxes and grids. The identification numbers of these body boxes and grids are numbered sequentially beginning with BID. For instance, if BID=101 then the identification numbers of body aerodynamic boxes and grids are 101, 102, etc.

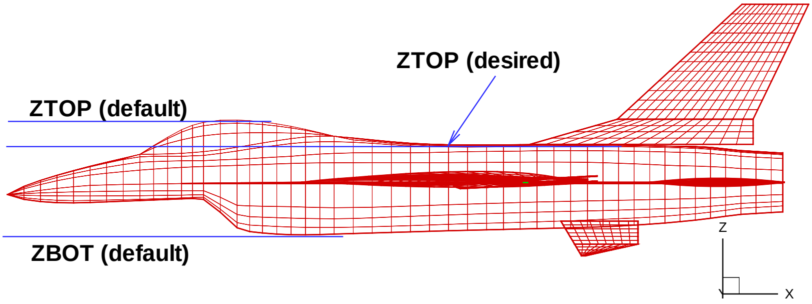
   It should be noted that the Cartesian mesh generated by the **BLOCK/BLOCKT** bulk data card approximates the body surface by a rectangular box (see figure below) then builds a rectangular mesh around this box. The boundary condition on the true body surface is mapped on the rectangular box using a slender body theory. After the flow solutions are solved on the rectangular box, they are mapped back to the true body surface. As an option, a body-fitted mesh to model the BODY7 macroelement can be generated using the **BLOCK1** bulk data card. In this case, the exact Euler boundary condition is applied on the body surface mesh to provide more accurate solution. The drawback of the **BLOCK1** bulk data card is that the body-fitted mesh cannot contain multiple body or wing components. For wing-body configuration, overset mesh scheme must be employed if the body is modeled by the **BLOCK1** bulk data card.





5. The number of the spanwise X-lines is computed according to (NRAD-1) × FACTDY+1, where NRAD is the number of circumferential points on the body that is specified in the NRAD entry of the **SEGMESH** bulk data card. Likewise, the number of horizontal X-lines is computed according to (NRAD-1) × FACTDZ+1.

6. For a fuselage with canopy, the default ZTOP is the highest point on the canopy. This ZTOP defines a rectangular box whose top surface could be too high and could intersect with the vertical tail. In this case, the lower portion of the vertical tail is immersed in the rectangular box in which no flow solution is computed. To circumvent this problem, it is recommended that ZTOP be used to define a lower top surface of the rectangular box. The users should be aware that the transpiration boundary condition of ZEUS involves only the slopes of the surface panel model and does not involve the exact location of the surface panel model. Therefore, the location of the rectangular box is not critical.



CAERO7 Aerodynamic Wing Component

Description: Defines the surface mesh of an aerodynamic wing macroelement of a wing-like component.

Format and Example:

1	2	3	4	5	6	7	8	9	10
CAERO7	WID	LABEL	ACOORD	NSPAN	NCHORD	LSPAN	ZTAIC	PAFOIL7	CONT
CONT	XRL	YRL	ZRL	RCH	LRCHD	ATTCHR			CONT
CONT	XTL	YTL	ZTL	TCH	LTCHD	ATTCHT			

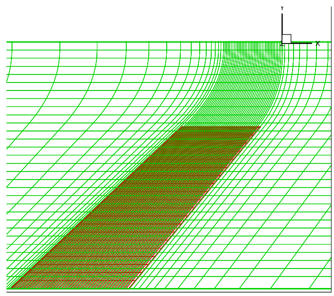
CAERO7	101	WING	8	5	4	20	0	0	+BC
+BC	0.0	0.0	0.0	1.0	10	4			+EF
+EF	0.0	1.0	0.0	1.0	11	0			

Field	Contents
WID	Identification number (Integer > 0) (See Remark 1)
LABEL	An arbitrary character string (up to 8 characters) used to define the wing (Character)
ACOORD	Identification number of **ACOORD** (specifying a local coordinate system and orientation) bulk data card. (Integer ≥ 0, Default = 0) (See Remark 2)
NSPAN	Number of spanwise divisions of wing component. (Integer ≥ 3, except for the 2D analysis).
NCHORD	Number of chordwise divisions of wing component (Integer ≥ 3) (See Remark 3). Note that NCHORD can be a negative integer. In this case, the chord length of the leading edge box is automatically reduced by half.
LSPAN	Identification number of **AEFACT** bulk data card is used to specify the spanwise divisions of the wing component in percentage of the wing span. The number of values listed in the **AEFACT** bulk data card must be NSPAN. If LSPAN = 0, then NSPAN evenly distributed spanwise divisions are used. (Integer ≥ 0) (See Remark 4)
ZTAIC	Not used.
PAFOIL7	Identification number of a **PAFOIL7** or **PAFOIL8** bulk data card to specify sectional airfoil coordinates. If PAFOIL7 = 0, it is assumed that the **CAERO7** wing component is a flat plate. (Integer ≥ 0) (See Remark 5)
XRL YRL ZRL	*X*, *Y* and *Z* location of the root chord leading edge. (Real)

RCH	Length of the root chord. (Real ≥ 0.0)
LRCHD	Identification number of **AEFACT** bulk data card used to specify the root chord divisions of the wing component in percentage of the root chord. The number of values listed in the **AEFACT** bulk data card must be NCHORD. If LRCHD = 0, then NCHORD evenly distributed chordwise divisions for the root is used. (Integer ≥ 0) (See Remark 4)
ATTCHR	Not used.
XTL YTL ZTL	*X*, *Y* and *Z* location of the tip chord leading edge. (Real)
TCH	Length of the tip chord. (Real ≥ 0.0)
LTCHD	Identification number of **AEFACT** bulk data card is used to specify the tip chord divisions of the wing component in percentage of the tip chord. The number of values listed in the **AEFACT** bulk data card must be NCHORD. If LTCHD = 0, then evenly distributed chordwise divisions for the tip is used. (Integer ≥ 0) (See Remark 4)
ATTCHT	Not used.

<u>Remarks:</u>

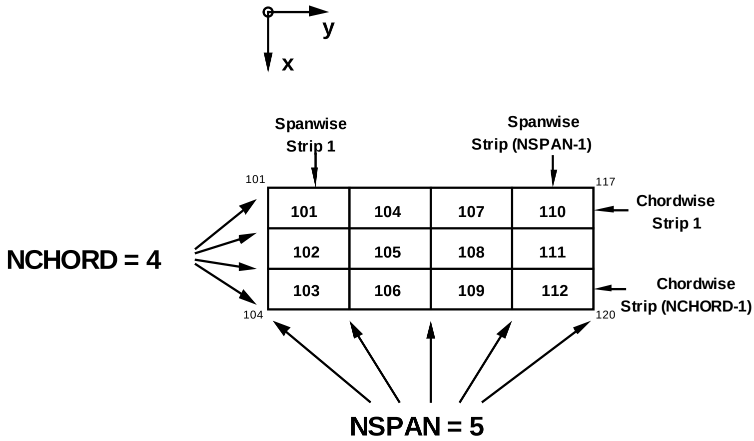
1. The identification numbers of the aerodynamic grids and boxes are numbered sequentially beginning with WID. Based on the surface mesh defined by CAERO7, the program can automatically extends the surface mesh to far field and generates a mesh surrounding the CAERO7 by referring to a **BLOCK/BLOCKT** bulk data card. See the figure on the right.



2. All coordinate locations defined above in XRL, YRL, ZRL, XTL, YTL, and ZTL are in the local wing coordinate system defined by the **ACOORD** bulk data card.

3. The number of spanwise and chordwise divisions of the wing component includes the end points; therefore, there will be NSPAN-1 spanwise strips, NCHORD-1 chordwise strips, NSPAN × NCHORD aerodynamic grids and (NSPAN-1) × (NCHORD-1) aerodynamic boxes generated by each **CAERO7** bulk data card. Among all aerodynamic grids and boxes (of the **CAERO7** and **BODY7** bulk data cards) no duplicate identification number is allowed. The following figure demonstrates the numbering scheme. In the example given below, a **CAERO7** has WID = 101, NSPAN = 5 and NCHORD = 4. There are (5-1) × (4-1) = 12 aerodynamic boxes and 5 × 4 = 20 aerodynamic grid points generated for this lifting surface. Wing boxes are numbered starting with the wing id of 101 and ending at 112. Wing aerodynamic grid points are numbered starting with the wing id of 101 and ending at 120. A duplicate identification number (i.e., aerodynamic box(es) and aerodynamic grid point(s)) would occur, for example, if another lifting surface were defined with a wing id of say 112, since there would be two aerodynamic boxes with id's of 112 and

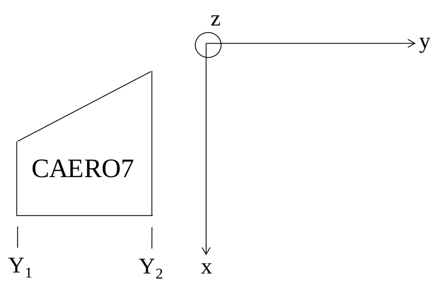
duplicate aerodynamic grids of 112, 113, etc. Therefore, for this case, the next closest Wing ID (WID) or Body ID (BID) that could be used is 121.



4. The values listed in these **AEFACT** bulk data cards referenced by LSPAN, LRCHD, and LTCHD must start with <u>0.0</u> and end with <u>100.0</u>.

5. The <u>upper surface</u> of a **CAERO7** is defined by the normal vector which is computed by the cross product of the vector along the chord (leading edge to trailing edge) to the vector along the span (root to tip), the upper/lower surface of the airfoil section specified by the **PAFOIL7** or **PAFOIL8** bulk data card must be consistent with the normal vector of the CAERO7 macroelement.

   The following figure shows a CAERO7 macroelement located in the negative Y-axis (the left-hand side of the pilot). If Y_2 is defined as the *y* location of the wing root (YRL) and Y_1 as the wing tip (YTL), then the normal vector of this CAERO7 is toward the negative Z-axis. Consequently, the camber of the airfoil section defined by **PAFOIL7** bulk data card must be reversed. To avoid this, it is highly recommended that YRL = Y_1 and YTL = Y_2 be specified so that the normal vector of this CAERO7 is toward the positive Z-axis.



CONM1

Concentrated Mass Element Connection (General Form)

Description: Defines a 6 × 6 symmetric mass matrix at a structural grid point of the structural model.

Format and Example:

1	2	3	4	5	6	7	8	9	10
CONM1	EID	G	CID	M11	M21	M22	M31	M32	CONT
CONT	M33	M41	M42	M43	M44	M51	M52	M53	CONT
CONT	M54	M55	M61	M62	M63	M64	M65	M66	

CONM1	2	22	2	2.9		6.3			+BC
+BC	4.8				28.6				+DE
+DE		28.6						28.6	

Field	Contents
EID	Unique identification number. (Integer > 0) (See Remark 1)
G	Structural grid point identification number. (Integer ≠ 0)
CID	If CID > 0, CID is the identification number of a coordinate system that is defined in the structural finite element model. (Integer ≥ 0, or Blank) (See Remark 2) If CID < 0, the absolute value of CID is the identification number of a **CORD2R** bulk data card. (Integer < 0)
Mij	Mass matrix values. (Real) (See Remark 3)

Remarks:

1. **CONM1** is referred to by a **CONMLST** bulk data card that can be used to modify the generalized mass matrix for a subcase. The definition of the **CONM1** bulk data card is identical to that of NASTRAN.

2. A coordinate system such as CORD1C, CORD1R, CORD1S, CORD2C, CORD2R, or CORD2S, with ID = CID must exist in the NASTRAN finite element model that is imported by the **'ASSIGN FEM='** Executive Control Command. For non-NASTRAN finite element model, CID must be zero.

3. The mass matrix is defined in the local coordinate system CID. If CID = 0, basic coordinate system is assumed.

CONM2

Concentrated Mass Element Connection (Rigid Body Form)

Description: Defines a concentrated mass at a structural grid point of the structural model.

Format and Example:

1	2	3	4	5	6	7	8	9	10
CONM2	EID	G	CID	M	X1	X2	X3		CONT
CONT	I11	I21	I22	I31	I32	I33			

CONM2	2	15	6	49.7					+C
+C	16.2		16.2			7.8			

Field	Contents
EID	Unique identification number. (Integer > 0) (See Remark 1)
G	Structural grid point identification number. (Integer ≠ 0)
CID	If CID > 0, CID is the identification number of a coordinate system that is defined in the structural finite element model. (Integer > 0, or Blank) (See Remark 2) If CID < -1, the absolute value of CID is the identification number of a **CORD2R** bulk data card. (Integer < -1).
M	Mass value. (Real)
X1,X2,X3	Offset distances from the grid point to the center of gravity of the mass in the coordinate system defined in field 4, unless CID = -1, in which case X1, X2, and X3 are the coordinates, not offsets, of the center of gravity of the mass in the basic coordinate system. (Real) (See Remark 3)
Iij	Mass moments of the inertia measured at the mass center of gravity in the coordinate system defined by field 4. If CID = -1, the basic coordinate system is implied. (Real) (See Remark 4)

Remarks:

1. **CONM2** is referred to by a **CONMLST** bulk data card that can be used to modify the generalized mass matrix for a subcase. The definition of the **CONM2** bulk data card is identical to that of NASTRAN.

2. For CID > 0, a coordinate system such as CORD1C, CORD1R, CORD1S, CORD2C, CORD2R, or CORD2S, with ID = CID must exist in the NASTRAN finite element model that is imported by

the **'ASSIGN FEM='** Executive Control Command. For non-NASTRAN finite element model, CID must be less than or equal to zero.

3. For CID > 0, X1, X2 and X3 and the mass matrix are defined in the local coordinate system with ID = CID.

4. The form of the inertia matrix about its center of gravity is taken as:

$$\begin{bmatrix} \boldsymbol{M} & & & & & \\ & \boldsymbol{M} & & & \textbf{\textit{symmetric}} & \\ & & \boldsymbol{M} & & & \\ & & & \boldsymbol{I11} & & \\ & & & -\boldsymbol{I21} & \boldsymbol{I22} & \\ & & & -\boldsymbol{I31} & -\boldsymbol{I32} & \boldsymbol{I33} \end{bmatrix}$$

where

$$\boldsymbol{M} = \int \rho \, dV$$

$$\boldsymbol{I11} = \int \rho \left(x_2^2 + x_3^2 \right) dV$$

$$\boldsymbol{I22} = \int \rho \left(x_1^2 + x_3^2 \right) dV$$

$$\boldsymbol{I33} = \int \rho \left(x_1^2 + x_2^2 \right) dV$$

$$\boldsymbol{I21} = \int \rho x_1 x_2 \, dV$$

$$\boldsymbol{I31} = \int \rho x_1 x_3 \, dV$$

$$\boldsymbol{I32} = \int \rho x_{\boldsymbol{2}} x_{\boldsymbol{3}} \, dV$$

and x1, x2, and x3 are components of the distance from the center of gravity in the coordinate system defined in field 4. The negative signs for the off-diagonal terms are supplied automatically.

CONM2L or CONM2* Large Field Input for CONM2

Description: Defines a **CONM2** bulk data card but with large field (16 columns) input.

Format and Example:

1	2 3	4 5	6 7	8 9	10
CONM2L	EID	G	CID	M	CONT
CONT	X1	X2	X3		CONT
CONT	I11	I21	I22	I31	CONT
CONT	I32	I33			

CONM2*	2	15	6	49.7	+C
+C					+C
+C	16.2		16.2		+C
+C		7.8			

Remarks:

1. The definitions of each entry in the **CONM2L/CONM2*** bulk data cards are identical to those of the **CONM2** bulk data card.

2. If CONM2* is specified, it will be automatically converted to a **CONM2L** bulk data card.

CONMLST

CONM1/CONM2/CONM2L Bulk Data Cards List Definition

Description: **CONMLST** defines a list of the identification numbers of the **CONM1/CONM2/CONM2L** bulk data cards. **CONMLST** is referred to by a **FLUTTER/FLTFAST** bulk data card.

Format and Example:

1	2	3	4	5	6	7	8	9	10
CONMLST	SETID	EIGEN	FILENM		WTMASS				CONT
CONT	$FACT_1$	$CONMID_1$	$FACT_2$	$CONMID_2$	…	-etc-			

CONMLST	100								+BC
+BC	1.0	10	-3.0	20	-0.1	25			

Field	Contents
SETID	Identification number. (Integer ≠ 0) (See Remark 1)
EIGEN	Character string either "YES" or "NO". For EIGEN = "YES", the eigenvectors of the modified structures are recomputed. (Character, Default = "NO") (See Remark 2)
FILENM	Character string to specify the name of an ASCII file that stores the eigenvectors associated with each natural frequency. If the first character of FILENM is a dollar sign "$", the rest of the characters must be integers. This integer is the identification number of an **EXTFILE** bulk data card where the filename is specified. This feature allows for filenames up to 56 characters to be input. (Character) (See Remark 3)
WTMASS	Factor to convert weight to mass. (Real ≠ 0.0, Default = 1.0) (See Remark 4)
$FACT_i$	Multiplier to the mass matrix that is specified by the **CONM1/CONM2/CONM2L** bulk data card with EID = $CONMID_i$. (Real)
CONMIDi	Identification number of a **CONM1/CONM2** bulk data card. (Integer > 0) (See Remark 4)

Remarks:

1. The absolute value of **SETID** is referred to by the CONMLST entry of the **FLUTTER/FLTFAST** bulk data card.

   Note: SETID can be a negative integer. This will (1) recompute the natural frequencies of the structures. These recomputed natural frequencies are used for sorting the velocity-damping-frequency list of the flutter results, and (2) compute the eigenvectors of the modified structure and redefine the generalized coordinates based on this new set of eigenvectors. Therefore, the resulting generalized mass and stiffness matrices are diagonal.

2. A fatal error can occur if the perturbed mass matrix is not positive definite.

3. The size of the eigenvectors is NM × NM, where NM is the number of modes imported by the **'ASSIGN FEM='** Executive Control Command. The format of the data written on the file "FILENM" is in the OUTPUT4 format (see description of the **'ASSIGN MATRIX='** Executive Control Command for the OUTPUT4 format). If FILENM is blank, no eigenvectors is computed. Also, FILENM is required only for EIGEN = "YES".

4. WTMASS is applied to the mass matrices of all **CONM1**, **CONM2,** and **CONM2L** that are listed in the **CONMLST** bulk data card. WTMASS must be used if the unit of the mass matrices defined in the **CONM1/CONM2/CONM2L** bulk data card is in weight. For instance, if the unit of the mass matrices is in lbs and the length unit of the structural model (as well as the aerodynamic model) is in inches, WTMASS = 1/386 = 0.00259 is required.

5. Each mass matrix list in its associated **CONM1/CONM2/CONM2L** bulk data card gives a perturbed generalized mass matrix. All these matrices are added into the unperturbed generalized mass matrix. The mass perturbation option provides a change in the mass distribution of the structure without returning to the structural finite element analysis.

CORD2R Rectangular Coordinate System Definition

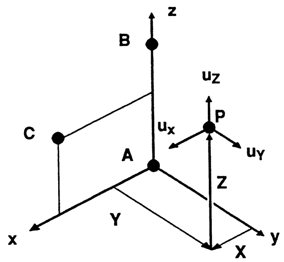
Description: Defines a rectangular coordinate system by reference to coordinates of three points. The first point defines the origin. The second point defines the direction of the z-axis. The third point defines a vector which, with the z-axis, defines the x-z plane. The reference coordinate system must be independently defined.

Format and Example:

1	2	3	4	5	6	7	8	9	10
CORD2R	CID	RID	A1	A2	A3	B1	B2	B3	CONT
CONT	C1	C2	C3						

CORD2R	3	17	-2.9	1.0	0.0	3.6	0.0	1.0	+BC
+BC	5.2	1.0	-2.9						

Field	Contents
CID	Coordinate system identification number. (Integer > 0)
RID	Not used.
Ai, Bi, Ci	Coordinates of three points in the basic coordinate. (Real)



Remarks:

1. The continuation entry must be present.

2. The three points (A1, A2, and A3), (B1, B2 and B3), and (C1, C2 and C3) must be unique and noncollinear. Noncollinearity is checked by the geometry processor.

3. Coordinate system identification numbers on **CORD2R** must all be unique.

4. The location of a grid point (*P* in the sketch) in this coordinate system is given by (*X*, *Y*, and *Z*).

5. The displacement coordinate directions at *P* are shown by (u_x, u_y, u_z).

CPFACT

Weighting Factor of Unsteady Aerodynamic Pressure

Description: Multiplies the computed unsteady aerodynamic pressures by a weighting factor.

Format and Example:

1	2	3	4	5	6	7	8	9	10
CPFACT	EID	IDMK	SYM	KINDEX	TYPE	LABEL	REAL	IMAG	CONT
CONT	PANLST1	PANLST2	PANLST3	…		-etc-			

CPFACT	100	90	SYM	ALL	FEM	ALL	1.0	0.0	+CP1
+CP1	301	701							

Field	Contents
EID	**CPFACT** identification number (Integer > 0) (See Remark 1)
IDMK	Identification number of **MKAEROZ** bulk data card whose generated unsteady aerodynamic pressure coefficients are multiplied by a weighting factor. (Integer > 0)
SYM	Symmetric condition of the unsteady aerodynamic pressures generated by the **MKAEROZ** bulk data card (Character). SYM = ‘SYM’ for symmetric condition SYM = ‘ANTI’ for antisymmetric condition SYM = ‘ASYM’ for asymmetric condition
KINDEX	Not used.
TYPE	Not used.
LABEL	Not used.
REAL	The weighting factor. (Real)
IMAG	Not used.
PANLSTi	List of identification numbers of the **PANLST1**, **PANLST2** and/or **PANLST3** bulk data cards. The unsteady aerodynamic pressures on the aerodynamic boxes that are listed in the **PANLSTi** bulk data cards are multiplied by the weighting factor. (Integer) (See Remark 2)

Remarks:

1. It is often that the users need to modify the unsteady aerodynamic pressures in a certain region of the aerodynamic model and consequently change the generalized aerodynamic forces for better aeroelastic predictions. For instance, the inviscid aerodynamic methods may overestimate the hinge moments of the control surface modes. To reduce the computed hinge moment, the users can apply a factor to the unsteady aerodynamic pressures of the aerodynamic boxes on the control surfaces.

2. EID is not referred to by any other bulk data card. The existence of each CPFACT in the Bulk Data Section "triggers" the multiplication procedure of the unsteady aerodynamic pressures by the weighting factors. EID is used for error message output only.

   The aerodynamic pressures on those aerodynamic boxes listed in the **PANLST*i*** bulk data card are multiplied by the weight factor REAL. Note that if the weight factor is applied to the lower surface of those aerodynamic boxes of a CAERO7 macroelement, a negative sign must be specified in PANLIST*i*. For instance, PANLSTi = -100 refers to a PANLSTi with ID = 100 whose listed aerodynamic boxes must belong to a CAERO7 macroelement. Because of the negative sign, the weighting factor REAL is multiplied to the pressure on the lower surface of those aerodynamic boxes.

DELLINE

Delete Gridlines

Description: Deletes gridlines in the mesh generated by a **BLOCK** bulk data card.

Format and Example:

1	2	3	4	5	6	7	8	9	10
DELLINE	ID								CONT
CONT	IJK_1	$INDEX_1$	IJK_2	$INDEX_2$	…	-etc-			

DELLINE	100								+D
+D	I	3	J	100	K	25			

Field	Contents
ID	Identification number (Integer > 0) (See Remark 1)
IJK_i	Character either “I:, “J”, or “K” (Character) (See Remark 2)
$INDEX_i$	Gridline index whose corresponding gridline is deleted from the mesh. (Integer > 0)

Remarks:

1. The **DELLINE** bulk data card is referred to by the **MESHPRM** bulk data card to delete undesired gridlines in the mesh generated by **BLOCK** bulk data card. This should be the last resort to get rid of redundant unwanted gridlines after modifying mesh generation parameters fails to do the job.

2. IJK_i = “I” indicates the spanwise X-line on the projected X-Y plane, IJK_i = “J” indicates the chordwise Y-line in the projected X-Y plane and IJK_i = “K” indicates the horizontal X-line on the projected X-Z plane which is to be deleted from the mesh.

DESDEP — Dependent Geometry Parameter

Description: Defines a dependent geometry parameter. This function is defined by the following equation where *DESDEP* represents the resulting value of the **DESDEP** bulk data card.

$$DESDEP = \sum_{i=1}^{n} A_i \cdot IDPRMA_i \cdot FUN_i \left(B_i \cdot IDPRMB_i \cdot \oplus_i \; C_i \cdot IDPRMC_i \right)$$

Format and Example:

1	2	3	4	5	6	7	8	9	10
DESDEP	IDDEP	LABEL	BULK	ID	ENTRY	ITH			CONT
CONT	A_1	$IDPRMA_1$	FUN_1	B_1	$IDPRMB_1$	$SYMBOL_1$	C_1	$IDPRMC_1$	CONT
CONT	A_2	$IDPRMA_2$	FUN_2	B_2	$IDPRMB_2$	$SYMBOL_2$	C_2	$IDPRMC_2$	CONT
CONT	-etc-	…	…	-etc-	…	…	…	-etc-	

DESDEP	100	SWEEP	WING	200	XTL	1			+D1
+D1	1.0	100	ATAN	-1.0	110	/	2.0	120	+D2
+D2	1.0	130	*	3.0	140				

Field	Contents
IDDEP	Identification number that is referred to by a **DESSEN** bulk data card. (Integer > 0) (See Remark 1)
LABEL	Any character string (up to 8 characters) with no embedded blanks to define the label of the dependent geometry parameter. (Character)
BULK	The name of a bulk data card that contains the dependent geometry input. (Character) (See Remark 2)
ID	Identification number of the bulk data card specified in the entry BULK. If the bulk data card referenced by BULK is identified by a label rather than an integer, then ID must be input as the label. (Integer or Character)
ENTRY	The name of the entry of the bulk data card specified in the entry BULK. The value of this entry is the dependent geometry parameter that is subjected to the change by the shape design variable. (Character)
ITH	For open-ended bulk data cards, there can be multiple entries of ENTRY (i.e., when the bulk data entry name is subscripted with an "i"). ITH indicates which "i-th"entry is selected as the dependent geometry parameter. For any ENTRY that does not have a subscript i, ITH should be set to 0 (Integer ≥ 0, Default = 1) (See Remark 3)
A_i	A real coefficient involved in the above equation (Real, Default = 1.0)
$IDPRMA_i$	Identification number of a **DESVAR**, **DESIND** or **DESFUN** bulk data card whose resulting value is plugged into the above equation. If $IDPRMA_i$=0, the resulting value is 1.0 (Integer ≥ 0)

FUN_i	Character string to define a function involved in the above equation. (Character) FUN_i = SIN — the function is Sine FUN_i = COS — the function is Cosine FUN_i = TAN — the function is Tangent FUN_i = ASIN — the function is Arcsine FUN_i = ACOS — the function is Arccosine FUN_i = ATAN — the function is Arctan FUN_i = SQRT — the function is a Square Root FUN_i = LOG — the function is a Natural Logarithm FUN_i = LOG10 — the function is a Logarithm of base 10 FUN_i = EXP — the function is an Exponential FUN_i = ABS — the function is the Absolute Value FUN_i = * — the function is Multiplication FUN_i = / — the function is Division FUN_i = UNIT — the resulting function is 1.0 and the remaining entries Bi, IDPRMB Ci, and IDPRMCi are not used.
B_i	A real coefficient shown in the above equation (Real, Default = 1.0)
$IDPRMB_i$	Same as IDPRMAi except the resulting value is plugged into its position shown in the above equation. (Integer ≥ 0)
$SYMBOL_i$	Character string representing the symbol $\oplus$ in the above equation. (Character, Default = +) $SYMBOL_i$ = + results in $(B_i \cdot IDPRMB_i) + (C_i \cdot IDPRMC_i)$ $SYMBOL_i$ = - results in $(B_i \cdot IDPRMB_i) - (C_i \cdot IDPRMC_i)$ $SYMBOL_i$ = * results in $(B_i \cdot IDPRMB_i) * (C_i \cdot IDPRMC_i)$ $SYMBOL_i$ = / results in $(B_i \cdot IDPRMB_i) / (C_i \cdot IDPRMC_i)$
C_i	A real coefficient shown in the above equation (Real)
$IDPRMC_i$	Same as $IDPRMA_i$ except its resulting value is plugged into its position shown in the above equation. (Integer ≥ 0)

Remarks:

1. The **DESDEP** bulk data card defines a dependent geometry parameter whose value depends on the value of the shape design variable. The **DESDEP** bulk data card refers to an ENTRY in a bulk data card that is associated with the geometry input of the box model. The value of this ENTRY is calculated by the equation shown above that is a function of the bulk data card(s) **DESFUN, DESIND** or **DESVAR**. Note that only those **DESDEP** bulk data cards referenced in the **DESSEN** bulk data card are active (i.e., those not referenced by any DESSEN have no effect).

2. The entries BULK, ID, ENTRY, and ITH jointly define a dependent geometry parameter whose value is subjected to the change by the shape design variable being specified (typically by a **DESVAR** bulk data card).

3. The following example shows what the ITH value would be for an open-ended bulk data card.

```
SEGMESH 2001     10      11
        1       10.0     0.0     15.0          to change any real value on this line, ITH = 1
        1       20.0     0.0     15.0          to change any real value on this line, ITH = 2
        1       30.0     0.0     15.0          to change any real value on this line, ITH = 3
        1       40.0     0.0     15.0          to change any real value on this line, ITH = 4
                                               and so on...
```

To change an entry for a non-open-ended bulk data card, ITH should be set to 0.

to change any of these example entries, ITH = 0

```
CORD2R  101              0.0     0.0     0.0     0.0     0.0     1.0
        0.0     1.0      1.0
```

DESFUN

Variable Linking Function

Description: Defines a function to link the dependent geometry parameter with the independent geometry parameters and the shape design variables.

Format and Example:

1	2	3	4	5	6	7	8	9	10
DESFUN	IDPRM	FUN							CONT
CONT	$OPEN_1$	A_1	$IDVARA_1$	B_1	$IDVARB_1$	C_1	$CLOSE_1$	E_1	CONT
CONT	$OPEN_2$	A_2	$IDVARA_2$	B_2	$IDVARB_2$	C_2	$CLOSE_2$	E_2	CONT
CONT				…	etc	…			

DESFUN	100	SIN							+D1
+D1	(	2.0	10	1.0	20	2.0			+D2
+D2	+	3.0	0		0		)	2.0	+D3
+D3	/(	1.0	10	2.0	0				+D4
+D4	-	1.0	0		30	3.0	)	3.0	

Field	Contents
IDPRM	Unique identification number that is referred to by a **DESDEP** bulk data card. Among all **DESFUN**, **DESIND,** and **DESVAR** bulk data cards, no duplicate identification number is allowed. (Integer > 0) (See Remark 1)
FUN	Character string to define a function. (Character, Default = blank) FUN = SIN — the function is Sine FUN = COS — the function is Cosine FUN = TAN — the function is Tangent FUN = ASIN — the function is Arcsine FUN = ACOS — the function is Arccosine FUN = ATAN — the function is Arctan FUN = SQRT — the function is a Square Root FUN = LOG — the function is a Natural Logarithm FUN = LOG10 — the function is a Logarithm of base 10 FUN = EXP — the function is an Exponential FUN = ABS — the function is the Absolute Value FUN = blank — no function is applied
$OPEN_i$	For $OPEN_1$, $OPEN_1$ must be one of "(", "((", "(((", "((((", "(((((", "((((((", or "(((((((". For $OPEN_i$ where i > 1, the first character in $OPEN_i$ must be either "+", "-", "*", or "/". The rest of the characters must be one of "(", "((", "(((", "((((", "(((((", "((((((", "(((((((" or blank. (Character)
A_i	A real coefficient. (Real, Default = 1.0)

$IDVARA_i$	Identification number of a **DESVAR**, **DESIND** or another **DESFUN** bulk data card whose resulting value is used to in the function. If $IDVARA_i = 0$, the value is assumed to be 1.0. (Integer ≥ 0) (See Remark 2)
B_i	A Real coefficient that is the exponent of $IDVARA_i$, i.e. $\left(\mathrm{IDVARA_i}\right)^{B_i}$. (Real, Default = 1.0)
$IDVARB_i$	Same as $IDVAR_i$ except defining the second value in the function. (Integer ≥ 0)
C_i	Same as B_i except for $\left(\mathrm{IDVARB_i}\right)^{C_i}$.
$CLOSE_i$	Character string either ")", "))", ")))", "))))", ")))))", "))))))", ")))))))" or blank. (Character)
E_i	A real coefficient that is the exponent of the function that is enclosed by a pair of open parenthesis ("(") and a closed parenthesis (")"). (Real, Default =1.0)

Remarks:

1. The **DESFUN** bulk data card defined an algebraic equation whose resulting value is a function of the shape design variables (defined by the **DESVAR** bulk data card) and the independent geometry parameters (defined by the **DESIND** bulk data card). The **DESFUN** bulk data card is referred to by a **DESDEP** bulk data card to link the dependent geometry parameter with the shape design variable and the independent geometry parameter. This algebraic equation is defined as:

$$F = FUN\left[\sum_{i=1}^{n}\left(OPEN_i\right)A_i\cdot\left(IDVARA_i\right)^{B_i}\cdot\left(IDVARB_i\right)^{C_i}\left(CLOSE_i\right)^{E_i}\right]$$

2. The **DESFUN** bulk data card can be used to define any algebraic equation as a function of $IDVARA_i$ and $IDVARB_i$.

   For example, let x be the value of a shape design variable being defined by the **DESVAR** bulk data card with ID = 10 and y be the value of an independent geometry parameter being defined by the **DESIND** bulk data card with ID = 20, if the algebraic equation is:

$$F = \frac{\left(\left(3.14x^2 + 1.4xy^3\right) + \sqrt{\left(x^2 + x + 1.0\right)^2 + \sqrt{1000.0 + 2y^2 + x}}\right)}{\left(10.0xy + \left(x + 10.\right)y + x^3\right)^{3.2}}$$

   the corresponding **DESFUN** bulk data card is:

DESFUN	1								+D1
+D1	((	3.14	10	2.0	0				+D2
+D2	+	1.4	10	1.0	20	3.0	)	1.0	+D3
+D3	+((	1.0	10	2.0	0			1.0	+D4
+D4	+	1.0	10	1.0	0			1.0	+D5
+D5	+	1.0	0	0.0	0	0.0	)	2.0	+D6
+D6	+(	1000.0	0	0.0	0	0.0			+D7
+D7	+	2.0	20	2.0	0	0.0			+D8
+D8	+	1.0	10	1.0			)	0.5	+D9

+D9	+	0.0	0				)	0.5	+D10
+D10	+						)	1.0	+D11
+D11	/(	10.0	10	1.0	20	1.0			+D12
+D12	+(	1.0	10	1.0					+D13
+D13	+	10.0	0	1.0			)		+D14
+D14	*	1.0	20						+D15
+D15	+	1.0	10	3.0			)	3.2	

DESIND — Independent Geometry Parameter

Description: Defines an independent geometry parameter whose value is used to calculate the value of a dependent geometry parameter.

Format and Example:

1	2	3	4	5	6	7	8	9	10
DESIND	IDGEO	LABEL	BULK	ID	ENTRY	ITH			

DESIND	100	LE	SEGMESH	10	X	50			

Field	Contents
IDGEO	Unique identification number that is referred to by a **DESFUN** or **DESDEP** bulk data card. Among all **DESIND**, **DEFUN** and **DESVAR** bulk data cards, no duplicate identification number is allowed. (Integer > 0) (See Remark 1)
LABEL	Any character string to define the label of the independent geometry parameter (Character)
BULK	The name of a bulk data card that is associated with the geometry input of the aerodynamic model. (Character) (See Remark 2)
ID	Identification number of the bulk data card specified in the entry BULK. If the bulk data card **BULK** is identified by label, ID is a character string that is used to refer to this bulk data card. (Integer or Character)
ENTRY	The name of the entry of the bulk data card specified by the entry BULK. The value of this entry is the independent geometry parameter that is involved in the equation defined in the **DESFUN** bulk data card. (Character)
ITH	For an open-ended bulk data card, there could be multiple entries of ENTRY; ITH indicates which entry is selected as the independent geometry parameter. (Integer ≥ 0, Default = 1)

Remarks:

1. The **DESIND** bulk data card defines an independent geometry parameter whose value is independent of the shape design variable and is only used to calculate the value of the dependent geometry parameter. The **DESIND** bulk data card refers to an ENTRY in a bulk data card that is associated with the geometry input of the aerodynamic model.

2. The entries BULK, ID, ENTRY, and ITH jointly define an independent geometry parameter whose value is used to calculate the value of a dependent geometry parameter via the **DESFUN** bulk data card.

DESSEN — Linking Dependent Geometry Parameters

Description: Links a set of dependent geometry parameters by shape design variables.

Format and Example:

1	2	3	4	5	6	7	8	9	10
DESSEN	IDSEN	FILENM							CONT
CONT	$IDDEP_1$	$IDDEP_2$	…	etc	…				

DESSEN	10								+D
+D	101	201	301						

Field	Contents
IDSEN	Identification number that is referred to by a **DESSEN** Executive Control Command. (Integer > 0) (See Remark 1)
FILENM	A filename that stores those bulk data cards whose input entries are changed by the **DESDEP** bulk data card. The user can replace those bulk data cards in the input deck by those stored in FILENM to obtain an updated model. (Character or Blank)
$IDDEP_i$	Identification number of a **DESDEP** bulk data card whose defined dependent geometry parameter is perturbed by the shape design variable. (Integer > 0) (See Remark 2)

Remarks:

1. Because only one **DESSEN** Executive Control Command is allowed to be specified in the Executive Control Section, one job only computes the sensitivity of one shape design variable.

2. Among all **DESDEP** bulk data cards in the input file, only those listed in the **DESSEN** bulk data card are activated.

DESVAR

Shape Design Variable

Description: Defines a shape design variable.

Format and Example:

1	2	3	4	5	6	7	8	9	10
DESVAR	IDVAR	LABEL	BASE						

DESVAR	10	SWEEP	30.0						

Field	Contents
IDVAR	Unique identification number. Among all **DESVAR, DESFUN,** and **DESIND** bulk data cards, no duplicate identification number are allowed. (Integer > 0) (See Remark 1)
LABEL	Any character string to define the label of the shape design variable. (Character)
BASE	Baseline value of the shape design variable. (Real) (See Remark 2)

Remarks:

1. The shape design variable may not be an input entry of those bulk data cards for defining the geometry of the aerodynamic model. It is a shape design variable defined by the user and is controlled by the optimizer. If the value of the shape design variable is updated by the optimizer, all dependent geometry parameters being defined by the **DESDEP** bulk data cards listed in the **DESSEN** bulk data card are changed accordingly.

2. BASE is used to compute the sensitivity of the aerodynamic surface mesh using the Complex Variable Differentiation (CVD) technique. Note that the BASE must be properly calculated for input so that all dependent geometry parameters that are a function of the shape design variable matches with their corresponding value specified in their associated bulk data card.

DFORCE

Dynamic Forcing Function Definition at a Structural Grid Point

Description: Defines a set of dynamic forcing functions at the structural grid points (or on the aerodynamic boxes) where these forces are applied.

Format and Example:

1	2	3	4	5	6	7	8	9	10
DFORCE	IDFRC	PANLST	FACTP	IDTABP					CONT
CONT	IDGRID1	COMP1	FACTOR1	IDTAB1	IDGRID2	COMP2	FACTOR2	IDTAB2	CONT
CONT	IDGRID3	COMP3	FACTOR3	IDTAB3	…	-etc-			

DFORCE	10								+D
+D	123	3	10000.0	100					

Field	Contents
SETID	Unique set identification number. (See Remark 1) (Integer > 0)
PANLST FACTP IDTAB	Not used.
IDGRIDi	Identification number of a structural finite element grid points that are imported by the **'ASSIGN FEM='** Executive Control Command. IDGRID can be a negative integer that is the identification of a mirror-image grid point due to the entry XZSYM="H2F" in the **AEROZ** bulk data card. (Integer)
COMPi	Component number along which the forcing function is applied. COMP must be either: 1, 2, 3, 4, 5, or 6 where 1, 2 and 3 represent the forces along *x*, *y* and *z* directions, respectively. 4, 5 and 6 represent the moment about the *x*, *y* and *z* directions, respectively. (Integer > 0) (See Remark 2)
FACTORi	Factor (Real)
IDTABi	Identification number of a **TABLED1** bulk data card specifying the forcing function that is applied at grid point IDGRIDi along the direction of COMPi.

Remarks:

1. If the **DFORCE** bulk data card is referred to by the **ELOADS** bulk data card for transient ejection loads analysis, the time history specified in the **TABLED1** bulk data card is relative to the EJTIMEi entry of the **ELOADS** bulk data card.

2. The direction of the force is defined in the output displacement coordinates of the grid in the structural finite element model, i.e., the local coordinate system for displacement of the structural grid.

DGUST Discrete Gust Profile

Description: Defines a discrete gust profile in the vertical direction for the transient discrete gust loads analysis.

Format and Example:

1	2	3	4	5	6	7	8	9	10
DGUST	SID	TYPE	LG	WGV	X0				

DGUST	10	OMCOS	1.0	0.1	-3.0				

Field	Contents
SID	Unique set identification number. (Integer > 0) (See Remark 1)
TYPE	Character string to specify the type of the gust profile. TYPE must be either "STEP", "PULSE", "SINE", "OMCOS", "RANDOM" or "TABLE".

TYPE = "STEP" Step function from τ = 0.0 to τ = ∞, representing a sharp-edged gust.

TYPE = "PULSE"
$$T(\tau) = \begin{cases} 0, & \text{for} \quad \tau < 0 \\ e^{-L_g \tau^2} & \text{for} \quad \tau \geq 0 \end{cases}$$

TYPE = "SINE"
$$T(\tau) = \begin{cases} 0, & \text{for } \tau < 0 \\ \sin\left(\dfrac{2\pi\tau}{L_g}\right), & \text{for } 0 \leq \tau \leq L_g \\ 0, & \text{for } \tau > L_g \end{cases}$$

TYPE = "OMCOS"

$$T(\tau) = \begin{cases} 0, & \text{for } \tau < 0 \\ \frac{1}{2}\left(1 - \cos\left(\frac{2\pi\tau}{L_g}\right)\right), & \text{for } 0 \le \tau \le L_g \\ 0, & \text{for } \tau > L_g \end{cases}$$

TYPE = "RANDOM"

$$T(\tau) = \begin{cases} 0, & \text{for } \tau < 0 \\ \text{random number between -1.0 and 1.0}, & \text{for } 0 \le \tau \le L_g \\ 0, & \text{for } \tau > L_g \end{cases}$$

TYPE = "TABLE" — Arbitrary gust profile where $\tau = t - \frac{x - x_0}{V}$ and L_g is defined in the LG entry. (Character) (See Remark 2)

LG — LG can be either a real number or an integer.

TYPE = "PULSE", "SINE" "OMCOS", or "RANDOM"	LG is a real number of L_g shown in the above equation.
TYPE = "TABLE"	LG is an integer that is the identification number of a **TABLED1** bulk data card to define the profile of the arbitrary gust. (Real or Integer)

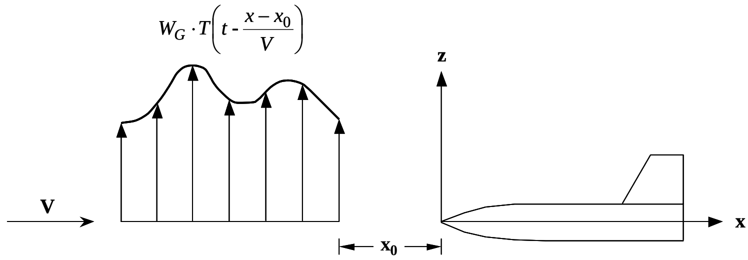
WGV — Gust velocity/forward velocity (WG/V) representing a scale factor to the gust profile. (Real $\neq$ 0.0)

X0 — Streamwise location relative to the origin of the aerodynamic model for the gust reference point. (Real)

Remarks:

1. **DGUST** is referred to by a **GLOADS** or **NLFLTR** bulk data card.

2. The resulting gust profile is show in the figure below:



where:

x is the streamwise locations of the control point of the aerodynamic model,

x_0 is the streamwise gust reference point,

V is the free stream velocity defined in the **FIXMACH**, **FIXMATM** or **FIXMDEN** bulk data card.

DMI

Header of Direct Matrix Input

Description: Defines the header information of **DMIS** or **DMIL** bulk data cards.

Format and Example:

1	2	3	4	5	6	7	8	9	10
DMI	NAME	ZERO	FORM	TIN	TOUT	LARGE	M	N	

DMI	BBB	0	2	1	1	DMIS	7	2	

Field	Contents
NAME	Name of the matrix. (Character) (See Remark 1)
ZERO	Must be an integer "0".
FORM	Form of matrix, as follows: (Integer) 2 = General rectangular matrix 6 = Symmetric matrix
TIN	Type of matrix being inputted, as follows: (Integer) 1 = Real, single precision (one field used/element) 2 = Real, double precision (one field used/element) 3 = Complex, single precision (two fields used/element) 4 = Complex, double precision (two fields used/element)
TOUT	Not used.
LARGE	Character string either = "DMIL" or "DMIS". (Character) (See Remark 2) LARGE = "DMIL", the element of the matrix is defined by the **DMIL** bulk data card. LARGE = "DMIS", the element of the matrix is defined by the **DMIS** bulk data card.
M	Number of rows in NAME. (Integer > 0)
N	Number of columns in NAME. (Integer > 0)

Remarks:

1. The name of the matrix cannot be the same as the name of any data entities existed on the runtime database.

2. **DMIL** bulk data card is the large field matrix input if high precision is required for defining the numerical values of the matrix elements. Otherwise, use the **DMIS** bulk data card.

DMIG

Direct Matrix Input at Structural Finite Element Grid Points

Description: Defines structure-related direct input matrices with terms located by specifying the identification numbers of the structural Finite Element (FEM) grid points and their component values.

Format and Example:

1	2	3	4	5	6	7	8	9	10
DMIG	NAME	PREC	FORM	FACTOR					CONT
CONT	GCOL	CCOL	GROW	CROW	x_{ij}		y_{ij}		CONT
CONT	GCOL	CCOL	GROW	CROW	x_{ij}		y_{ij}		

DMIG	MASS1	RSP	SYM						+A
+A	1001	4	2001	2	1.25E+05				+B
+B	1001	4	3001	3	2.7E+04		-etc-		

Field	Contents
NAME	Character string to define the name of the matrix. (Character) (See Remark 1)
PREC	The precision of the matrix. Any one of the following character string: RSP, RDP, CSP or CDP. (Character) (See Remark 2)
FORM	Character string either REC or SYM. (Character) (See Remark 3)
FACTOR	A factor multiplied to the matrix. Note that if FACTOR = 0.0, then GCOL, CCOl, GROW, and CROW entries can be any positive integer because those entries are not processed by the **DMIG** bulk data card. (Real, Default = 1.0)
GCOL	Identification number of a grid point in the structural finite element model for column index. (Integer > 0) (See Remark 4)
CCOL	Component number for GCOL. 1 ≤ CCOL ≤ 6. (Integer > 0)
GROW	Identification number of a grid point in the structural finite element model for row index. (Integer > 0) (See Remark 5)
CROW	Component number for GROW. 1 ≤ CROW ≤ 6. (Integer > 0)
x_{ij}, y_{ij}	Matrix terms. x_{ij} is real part for real or complex matrix. y_{ij} is the imaginary part for complex matrix. y_{ij} is not used for real matrix. Note that x_{ij} and y_{ij} occupy 2 fields for each input value. (Real) (See Remark 6)

Remarks:

1. **DMIG** creates a matrix with entity name = NAME. The size of matrix is *G-set* by *G-set* where *G-set* is 6 × (number of structural grid points).

2 RSP = Real Single Precision, RDP = Real Double Precision, CSP = Complex Single Precision and CDP = Complex Double Precision.

3. REC = Rectangular matrix. SYM = Symmetric matrix. Note that if FORM = SYM, only the upper triangular part of the matrix (including the diagonal) is allowed for input.

4. GCOL and CCOL define the column index. The column index can be calculated by $6 \times n$ + CCOL, where n is the number of structural grid points whose identification numbers are smaller than GCOL.

5. GROW and CROW define the row index of the matrix. The row index can be calculated by $6 \times n$ + CROW, where n is the number of structural grid points whose identification numbers are smaller than GROW.

6. The column index and row index can uniquely define the location of x_{ij} (and y_{ij} for complex matrix) in the matrix. All terms in the matrix that are not specified in the **DMIG** bulk data card will be zero. The mass unit and the length unit involved in the terms must be consistent with the FMMUNIT and FMLUNIT entries defined in the **AEROZ** bulk data card.

DMIL

Matrix Element Value Definition by Large Fields (16-Column Fields)

Description: Defines the values of matrix elements by 16-column fields. **DMIL** is referred to by **DMI** bulk data cards.

Format and Example:

1	2-3	4-5	6-7	8-9	10
DMIL	NAME	J	I1	A(I1, J)	CONT
CONT	…	-etc-	I2	A(I2, J)	CONT
CONT	…	-etc-			

DMIL	BBB	1	1	1.0	+D1
+D1	4.0	8.0	2.0	7	+D2
+D2	9.0				

Field	Contents
NAME	Name of the matrix. NAME must be the same as the entry NAME of the **DMI** bulk data card. (Character)
J	Column number of NAME. (Integer > 0)
I1, I2, etc.	Row number of NAME, which indicates the beginning of a group of nonzero elements in the column. (Integer > 0)
A(Ix, J)	Real part of element (see TIN of **DMI** bulk data card). (Real)
(Ix, J)	Imaginary part of element (see TIN of **DMI** bulk data card). (Real)

Remarks:

1. **DMIL** is referred by the **DMI** bulk data card with entry LARGE = ′**DMIL**′. The size and type of the matrix is defined in the **DMI** bulk data card.
2. The matrix elements is shown as follows:

$$[\ \mathrm{NAME}\ ] = \begin{bmatrix} \mathrm{A}(1,1) & \mathrm{A}(1,2) & \cdots & \mathrm{A}(1,\mathrm{N}) \\ \mathrm{A}(2,1) & \mathrm{A}(2,2) & \cdots & \mathrm{A}(2,\mathrm{N}) \\ \vdots & \vdots & & \vdots \\ \mathrm{A}(\mathrm{M},1) & \mathrm{A}(\mathrm{M},2) & \cdots & \mathrm{A}(\mathrm{M},\mathrm{N}) \end{bmatrix}$$

where *M* is the number of rows and *N* is the number of columns. *M* and *N* are defined in the **DMI** bulk data card.

3. For symmetric matrix, only the input of the upper triangular part (including the diagonals) is allowed, i.e., $I \le J$.

4. Only nonzero terms need to be entered. Therefore, I1, I2, etc. are the row locations of the first nonzero element in the J^{th} column.

5. Complex input must have both the real and imaginary parts entered if either part is nonzero; i.e., the zero component must be inputted explicitly.

   <u>Example of a Complex Matrix:</u>

DMIL	QQQ	1	1	4.0	+Q1
+Q1	2.0	5.0	0.0	4	+Q2
+Q2	6.0	6.0			
DMIL	QQQ	2	2	7.0	+Q3
+Q3	7.0	4	4.0	4.0	

$$[\mathrm{QQQ}] = \begin{bmatrix} 4.0 + 2.0\mathrm{i} & 0.0 + 0.0\mathrm{i} \\ 5.0 + 0.0\mathrm{i} & 7.0 + 7.0i \\ 0.0\ + 0.0\mathrm{i} & 0.0 + 0.0\mathrm{i} \\ 6.0 + 6.0\mathrm{i} & 4.0 + 4.0\mathrm{i} \end{bmatrix}$$

6. **DMIL** can be repeatedly specified for each column of the matrix. For columns that are not referred to by the **DMIL** bulk data card, null columns are assumed.

DMIS

Matrix Element Value Definition by 8-Column Fields

Description: Defines the values of the matrix elements by 8-column fields. **DMIS** is referred to by **DMI** bulk data cards.

Format and Example:

1	2	3	4	5	6	7	8	9	10
DMIS	NAME	J	I1	A(I1,J)	A(I1+1,J)	…	-etc-	I2	CONT
CONT	A(I2,J)	…	-etc-						

DMIS	BBB	1	1	1.0	4.0	8.0	2.0	7	+D1
+D1	9.0								

Field	Contents
NAME	Name of the matrix. NAME must be the same as the entry NAME of the **DMI** bulk data card. (Character)
J	Column number of NAME. (Integer > 0)
I1, I2, etc.	Row number of NAME, which indicates the beginning of a group of nonzero elements in the column. (Integer > 0)
A(Ix, J)	Real part of element (see TIN of **DMI** bulk data card). (Real)
B(Ix, J)	Imaginary part of element (see TIN of **DMI** bulk data card). (Real)

Remarks:

1. **DMIS** is referred to by the **DMI** bulk data card with entry LARGE = ′**DMIS**′. The size and type of the matrix is defined in the **DMI** bulk data card.

2. The locations of the matrix elements is shown as follows:

$$[\text{ NAME }] = \begin{bmatrix} A(1,1) & A(1,2) & \cdots & A(1,N) \\ A(2,1) & A(2,2) & \cdots & A(2,N) \\ \vdots & \vdots & & \vdots \\ A(M,1) & A(M,2) & \cdots & A(M,N) \end{bmatrix}$$

where M is the number of rows and N is the number of columns. M and N are defined in the **DMI** bulk data card.

3. For symmetric matrix, only the input of the upper triangular part (including the diagonals) is allowed, i.e., $I \leq J$.

4. Only nonzero terms need to be entered. Therefore, I1, I2, etc. are the row locations of the first nonzero element in the J^{th} column.

5. Complex input must have both the real and imaginary parts entered if either part is nonzero; i.e., the zero component must be inputted explicitly.

   <u>Example of a Complex Matrix:</u>

DMIS	QQQ	1	1	4.0	2.0	5.0	0.0	4	
	6.0	6.0							
DMIS	QQQ	2	2	7.0	7.0	4	4.0	4.0	

$$[QQQ] = \begin{bmatrix} 4.0+2.0i & 0.0+0.0i \\ 5.0+0.0i & 7.0+7.0i \\ 0.0+0.0i & 0.0+0.0i \\ 6.0+6.0i & 4.0+4.0i \end{bmatrix}$$

6. **DMIS** can be repeatedly specified for each column of the matrix. For columns that are not referred to by the **DMIS** bulk data card, null columns are assumed.

DRELI

Direct RELational Entity Input

Description: Loads input data directly to a relational entity.

Format and Example:

1	2	3	4	5	6	7	8	9	10
DRELI	NAME	ACTION	NENTRY	ATTR1	ATTR2	ATTR3	ATTR4	ATTR5	CONT
CONT	ATTR6	…	-etc-						

Example 1:

DRELI	CORD2R	REPROJ	2	CID	RID	A1	A2	A3	+D
+D	B1	B2	B3	C1	C2	C3			

Example 2:

DRELI	CORD2R	READD	1	3	17	-2.9	1.0	0.0	+C1
+C1	3.6	0.0	1.0	5.2	1.0	-2.9			

Field	Contents
NAME	The name of the relational entity. (Character) (See Remark 1)
ACTION	Action must be either ‘**REPROJ’** or ‘**READD’**. For ACTION = ‘REPROJ’, **DRELI** defines the attributes of the relational entity. (Character) (See Remark 2). For ACTION = ‘READD’, **DRELI** inputs the values of these attributes that are defined by ACTION = ‘REPROJ’. (Character) (See Remark 3)
NENTRY	For ACTION = ‘REPROJ’, NENTRY is the total number of entries of the relational entity. (Integer > 0) (See Remark 4) For ACTION = ‘READD’, NENTRY is the index of the entry. NENTRY must be between 1 and the total number of entries. (Integer > 0) (See Remark 5)
ATTRi	For ACTION = ‘REPROJ’, ATTRi is the name of the attributes. (Character) (See Remark 6) For ACTION = ‘READD’, ATTRi is the input data corresponding to its attributes. (Integer, Real or Character) (See Remark 7)

Remarks:

1. **DRELI** bulk data card provides a means to directly input the data to a relational entity. This relational entity must be one of the relational entities defined in the file 'RELATION.DAT' under the ZGEN directory.
2. ACTION = 'REPROJ' performs the projection for the relational entity. This is equivalent to the function of the **REPROJ** module.
3. ACTION = 'READD' loads the input data of an entry to the relational entity. This is equivalent to the function of **READD** module.
4. The number of **DRELI** bulk data cards with ACTION = 'READD' must be equal to the total number of entries.
5. The indices of the entry of **DRELI** bulk data cards with ACTION = 'READD' start from 1 and end at the total number of entries.

The name of the attribute must exist in the attribute list of the schema.

6. The type of the input data must match the type of the attribute that is defined in the file 'RELATION.DAT' under the ZGEN directory. The following example shows how to replace the input of the bulk data card **SET1** by **DRELI**:

By **SET1** input:

SET1	10	3	5	7					

By DRELI input:

DRELI	SET1	REPROJ	3	SETID	GRIDI				
DRELI	SET1	READD	1	10	3				
DRELI	SET1	READD	2	10	5				
DRELI	SET1	READD	3	10	7				

DYNSAVE

Save or Retrieve Data Entities Created by GENDYN Module

Description: Save or retrieve data entities created by the GENDYN module.

Format and Example:

1	2	3	4	5	6	7	8	9	10
DYNSAVE	SAVE	FILENM							
DYNSAVE	ACQUIRE	GENDYN.DAT							

Field	Contents
SAVE	Character string either "SAVE" or "AQUIRE". For SAVE = "SAVE", the data entities created by the GENDYN module are saved in an external file. For SAVE = "ACQUIRE", those data entities are retrieved. In this case, the **PLTMODE** bulk data card is not processed. (Character) (See Remark 1)
FILENM	Unformatted file name to store the data entities created by the GENDYN module. If the first character of FILENM starts with a dollar sign “$”, the rest of the characters must be integers. This integer is the identification number of an **EXTFILE** bulk data card where the filename is specified. This feature allows for filenames up to 56 characters to be input. (Character)

Remarks:

1. For a large structural model, the computational time of the GENDYN module could be long. The data entities created by the GENDYN module include the mode shape matrix with the rigid body modes (due to entry SUPORT in the "**ASSIGN FEM**" executive control command), the updated generalized mass and stiffness matrices, etc.

ELOADS

Transient Ejection Loads Analysis

Description: Defines the aeroelastic system, time integration, ejection forcing functions and changes of structural properties due to store separation for transient ejection loads analysis.

Format and Example:

1	2	3	4	5	6	7	8	9	10
ELOADS	SID	CONID	FLTID	RESTART	STATES	COMMAND	MLDTIME	MLDPRNT	CONT
CONT	$EJTIME_1$	$IDFRC_1$	$CONML_1$	PANLST1	GSET1	$DELM_1$	$DELC_1$	$DELK_1$	CONT
CONT	$EJTIME_2$	$IDFRC_2$	$CONML_2$	PANLST2	GSET2	$DELM_2$	$DELC_2$	$DELK_2$	CONT
CONT	…	-etc-							

ELOADS	100	10	20	25	30	100	1000	20	+E1
+E1	0.1	10	0	100	YES	$DELTAM_1$	$DELTAC_1$	$DELTAK_1$	+E2
+E2	0.3	30	1	0	NO		DELTAC2		

Field	Contents
SID	Unique set identification number. (Integer > 0) (See Remark 1)
CONID	Not used.
FLTID	Identification number of a **FLUTTER** bulk data card specifying the flight condition and the associated structural and aerodynamic matrices. (Integer > 0) (See Remark 2)
RESTART	Identification number of a **RESTART** bulk data card to save or retrieve the entire flow solution at the transient analysis. (Integer ≥ 0) (See Remark 3)
STATES	Identification number of a **MLDSTAT** bulk data card specifying the parameters of the airframe states. Note that the **MLDSTAT** bulk data card can refer to a **MLDTRIM** bulk data card to retrieve the trim solution (generated by a trim analysis) as the initial flow condition for the transient response analysis. (Integer ≥ 0)
COMMAND	Optional input; identification number of a **MLDCOMD** bulk data card specifying the parameters of the pilot's input commands for the maneuver. (Integer ≥ 0) (See Remark 4)
MLDTIME	Identification number of a **MLDTIME** bulk data card specifying the parameters of the time integration for solving the transient response problem. (Integer > 0)

MLDPRNT	Identification number of a **MLDPRNT** bulk data card specifying the time history of the parameters that are to be printed out. (Integer ≥ 0)
$EJTIME_i$	Ejection starting time. (Real) (See Remark 5)
$IDFRC_i$	Identification number of a **DFORCE** bulk data card specifying the ejection force and the structural grid point where the force is applied. Note that the time of the ejection force defined in the **DFORCE** bulk data card is relative to EJTIMEi. (Integer > 0) (See Remark 6)
$CONML_i$	Optional input; identification number of a **CONMLST** bulk data card specifying the change of the mass due to the separation of the store. (Integer ≥ 0) (See Remark 7)
$PANLST_i$	Identification number of a **SET1** bulk data card that lists a set of **PANLST1**, **PANLST2** or **PANLST3** bulk data card to specify a list of aerodynamic boxes of the store. (Integer ≥ 0) (See Remark 8)
$GSET_i$	Character string either "YES" or "NO". For GSET = "YES" The matrix defined by the DELM, DELC, and DELK entries are in the G-set degrees of freedom (G-set d.o.f. = 6 × number of structural finite element grid points). For GSET = "NO" These matrices are in the generalized coordinates. (Character) (See Remark 9)
DELM	Optional input; Character string that matches the matrix name of a **DMI** bulk data card, **DMIG** bulk data card, or **'ASSIGN MATRIX='** Executive Control Command. This matrix represents the change of the mass due to the separation of the store. If DELM is blank, no change in mass is assumed. (Character) (See Remark 10)
DELC	Same as DELM, but for the damping matrix. (Character)
DELK	Same as DELM, but for the stiffness matrix. (Character)

Remarks:

1. For the transient ejection load analysis, the **ELOADS** discipline must be selected in the Case Control Section with **ELOADS** = SID.

   Note:
   - Prior to the **ELOADS** analysis, a static aeroelastic analysis is performed first to obtain a mean flow solution from which the transient response analysis using the time-domain unsteady aerodynamic computation of the Euler-Solver module starts. Note that the number of time step of this static aeroelastic analysis is 100 as default, but that can be alerted using a **MKPARAM** bulk data card with entry TRMSTEP". The pressure distribution of such a static aeroelastic analysis can be visualized using the **PLTCP** bulk data card. The flow solution of the static aeroelastic analysis can be saved/retrieved using the **STFLOW** bulk data card. As an alternative, the trim solution computed by a trim analysis can be retrieved by a **MLDTRIM** bulk data card as the initial flow condition for the time-domain ELOADS analysis. In this case, the static aeroelastic analysis will be skipped

   - If the time history of component loads or forces at structural finite element grid points is desired for output, the "SMGH" (for symmetric/asymmetric structural boundary condition) or the

"MGG" (the G-set mass matrix) must be imported either by the **'ASSIGN MATRIX='** Executive Control Command or the **DMI** bulk data card.

2. Note, unlike the flutter analysis where all flight conditions listed in the **FIXMACH**, **FIXMATM** or **FIXMDEN** bulk data card are included in the analysis, the transient ejection load analysis only computes one flight condition per analysis. Thus, only the first velocity and density pair (for **FIXMACH**), the first altitude (for **FIXMATM**), or the first velocity (for **FIXMDEN**) listed in their respective bulk data cards is included in the analysis. The rest of the values in the list are ignored.

3. The **RESTART** bulk data card allows the user to save or retrieve the entire flow solution at the last time step of the transient analysis to a file. In the event of an abnormal termination during the transient analysis, or if the user wishes to extend the physical computational time from a previous transient analysis, the entire flow solution at the last time stop saved on the file can be retrieved for a continuous transient analysis.

4. The COMMAND entry can simulate the condition of store ejection while the aircraft is maneuvering due to the pilot input command.

5. $EJTIME_i$ must be in the ascending order. Multiple $EJTIME_i$ entries allow multiple store ejections in a sequential order.

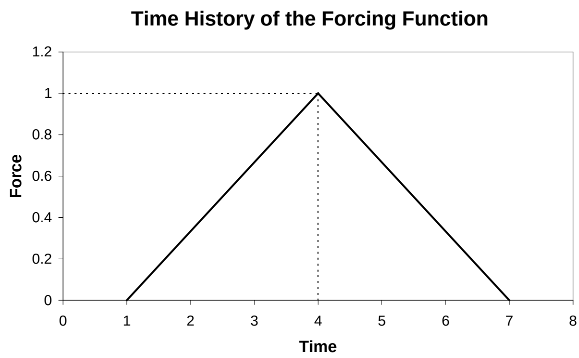
   Note:

   The first ejection time ($EJTIME_1$) must be greater than the starting time (TSTART entry of the **MLDTIME** bulk data card) of the time integration computation

6. The **DFORCE** bulk data card refers to a **TABLED1** bulk data card to specify the time history of the ejection forcing function. The time history of the forcing function is relative to $EJTIME_i$ (entry of **ELOADS** bulk data card). Therefore, the **TABLED1** bulk data card (example below):

TABLED1	10								+T
+T	0.0	0.0	3.0	1.0	6.0	0.0			

for $EJTIME_i$ = 1.0 will give the following forcing function:



Note: The units of the force must be consistent with the FTMUNIY and FTLUNIT entries of the **FIXMACH**, **FIXMATM**, or **FIXMDEN** bulk data card.

7. The change of the mass (ΔM) due to the separation of the store is relative to the mass of the previous ejection time ($EJTIME_i$ entry). For instance, the resulting mass of the structures after $EJTIME_n$ is:

$$M_n = M_{n-1} - \Delta M_n$$

where:

M_n	is the mass of the structures after $EJTIME_n$
M_{n-1}	is the mass of the structures before $EJTIME_n$ (or after $EJTIME_{n-1}$)
ΔM_n	is the change of the mass due to the separation of the store at $EJTIME_n$

Thus, for an initial mass of the structures (M_0), the mass after $EJTIME_n$ is:

$$M_n = M_0 - \sum_{i=1}^{n} \Delta M_i$$

Note: The changes in damping and stiffness of the structures are in the same definitions as the changes in the mass.

8. PANLST lists the aerodynamic boxes of the store whose contributions to the Generalized Aerodynamic Forces (GAF) are removed to simulate the reduction of aerodynamic forces due to the separation of the store. Note that the aerodynamic boxes on the lower surface of a CAERO7 macroelement can be selected by adding a negative sign to its corresponding identification number listed in the **SET1** bulk data card.

9. For GSET = "YES", the program will convert the DELM matrix to the generalized coordinates by:

$$\left[\Delta \bar{M}_i\right] = \left[\varphi\right]^T \left[\Delta M_i\right] \left[\varphi\right]$$

where:

$\left[\Delta \bar{M}_i\right]$	is the change of the generalized mass matrix due to the separation of the mass after ejection time $EJTIME_i$
$[\varphi]$	is the modal matrix imported by the **'ASSIGN FEM='** Executive Control Command

The above equation is also applied to the DELC and DELK matrices.

10. $DELM_1$ defines the change of mass of the structures (in additional to $CONML_i$) by the direct matrix input. $DELC_i$ and $DELK_i$ provide the options to present the change of the damping and stiffness of the structures respectively.

Note: The units of $DELM_i$, $DELC_i$ and $DELK_i$ must be consistent with the FMMUNIT and FMLUNIT of the **AEROZ** bulk data card.

ENDDATA — To End the Bulk Data Section

Description: Designates the end of the Bulk Data Section.

Format:

ENDDATA

Remark:

ENDDATA must exist at the end of the Bulk Data Section.

EXTFILE

External File

Description: Defines a character string as the name of an external file.

Format and Example:

1	2	3	4	5	6	7	8	9	10
EXTFILE	ID	FILENM							

EXTFILE	100	/ZEUS/TestCases/flutter/case1/ext.dat	

Field	Contents
ID	Unique identification number. (Integer > 0) (See Remark 1)
FILENM	This feature allows for filenames up to 56 characters (7 fields are available, 8 characters per field) with no embedded blank to be input. Note that unlike all other bulk data cards where any characters are converted to upper case, these characters will not be converted to upper case. This feature is important for the UNIX system because it is case sensitive.

Remarks:

1. The **EXTFILE** bulk data card is referred to by other bulk data cards that require external file for input or output. Whenever an external file name is needed in a bulk data card for input or output, rather than directly specifying a character string for the file name, the user can specify a character string started with a dollar sign “$” and followed by an integer; for instance $101. This integer is used to refer to the identification number of the **EXTFILE** bulk data card where the actual file name is specified by FILENM.

2. **EXTFILE** can be used to enforce the reading of file names in LOWER CASE if needed. File name case sensitivity can be an issue for the UNIX operating systems. In this situation, **EXTFILE** can be used to circumvent this problem.

EXTINP External Input

Description: Defines the input of the control element to a control surface.

Format and Example:

1	2	3	4	5	6	7	8	9	10
EXTINP	ID	TYPE	ITFID	CI	LABEL				

EXTINP	100		400	1	PILOT				

Field	Contents
ID	Identification number. (Integer > 0) (See Remark 1)
TYPE	Not used.
ITFID	Identification number of an **ACTU** bulk data card that is referred to by an **AESURFZ** or **AESLINK** bulk data card.
CI	Not used.
LABEL	Any alphanumeric string up to eight characters to describe this external input.

Remarks:

1. EXTINP is referred to by a **MLDCOMD** bulk data card.

FEMASET — Reduces the Structural Finite Element Model

Description: Reduces the number of grid points of the structural finite element model from the G-set grids to the A-set grids.

Format and Example:

1	2	3	4	5	6	7	8	9	10
FEMASET	C	GRID1	GRID2	GRID3		-etc-			

FEMASET		101	103	109	1000				

Alternate Form:

1	2	3	4	5	6	7	8	9	10
FEMASET		GRID1	THRU	GRID2					

Field	Contents
C	Not used.
GRIDi	Identification number of a structural grid point in the structural finite element model that is imported by the **'ASSIGN FEM='** Executive Control Command. (Integer > 0)

Remarks:

1. **FEMASET** is an optional input. Its existence "triggers" the program to remove all the structural grid points from the finite element model that are not a part of the GRIDi list.

2. Once the **FEMASET** bulk data card is used, the structural grid points that are referred to by the **SPLINEi**, **ATTACH**, **LOADMOD**, **AESURFZ**,…, etc. bulk data cards must be a part of the GRIDi list. For trim or dynamic loads analysis, all grid points in the finite element model that have mass attached should be included in the GRIDi list.

3. Multiple **FEMASET** bulk data cards can coexist in the Bulk Data Section. The following two **FEMASET** bulk data cards give the GRIDi as 101, 102, 103, 1000, and 1001:

FEMASET		101	THRU	103					
FEMASET		1000	1001						

FEMSAVE Save the Structural Modal Solution

Description: Saves the structural modal solution that is imported by the **'ASSIGN FEM='** executive control command.

Format and Example:

1	2	3	4	5	6	7	8	9	10
FEMSAVE	BOUND	FILENM							
FEMSAVE	ANTI	FEMSOL.DAT							

Field	Contents
BOUND	Character string either 'SYM', 'ANTI' or 'ASYM' that matches with the entry BOUNDARY of the **'ASSIGN FEM='** executive control command. (Character)
FILENM	The name of the file that stores the data entities generated by the **'ASSIGN FEM='** executive control command. If the first character of FILENM starts with a dollar sign "$", the rest of the characters must be integers. This integer is the identification number of an **EXTFILE** bulk data card where the filename is specified. This feature allows for filenames up to 56 characters to be input. (Character ≠ Blank) (See Remark 1)

Remarks:

1. For a large size of structural finite element model, extracting the structural grids, eigenvalues, and eigenvectors from the output file of the finite element solver could be time-consuming. Time can be saved on future runs by using the **FEMSAVE** bulk data card to save the structural modal solution, then specifying the entry FORM = "ACQUIRE" in the **'ASSIGN FEM='** executive control command to retrieve the modal solution.

FINDMPI — Automated Load Balancing for MPI

Description: Automatically generate the **BLKMPI** bulk data card input to achieve the best load balancing for time-domain ZEUS analysis (MLOADS,GLOADS,NLFLTR and NANSI modules) using MPI.

Format and Example:

1	2	3	4	5	6	7	8	9	10
FINDMPI	BLOCKID	NMPI	MINZ						

FINDMPI	4	2	3						

Field	Contents
BLOCKID	The ID of the reference block whose grid cell number will be used as the reference to determine the number of MPI processes applied to other blocks. If BLOCKID=0, then the block with the minimum number of grid cells is selected as the reference block. (Integer ≥ 0,default=0) (See Remark 1)
NMPI	The number of MPI processes to be applied to the reference block so that this reference block is divided into NMPI number of sub-blocks. (Integer > 0,default=1) (See Remark 2)
MINZ	The minimum number of grid cells in Z direction for any sub-block when cutting a big block into sub-blocks with each sub-block handled by one MPI process. (Integer ≥ 2, Default = 2) (See Remark 2)

Remarks:

1. For time-domain analysis by ZEUS using MPI, a **BLKMPI** bulk data card input is required to assign different blocks to different MPI processes. One can apply his/her own load balancing strategy to set up **BLKMPI** card input manually. **FINDMPI** card provides an alternative way to let ZEUS determine the **BLKMPI** card input based on a given reference block and two parameters NMPI and MINZ.

2. ZEUS uses the number of grid cells of the reference block divided by NMPI as the unit number of grid cells denoted by NCELL0. The reference block will be assigned with NMPI number of MPI processes, while the number of MPI processes assigned to any other block is the round-up integer of that block's number of grid cells divided by NCELL0. All blocks will be assigned with at least one MPI process. For any block assigned with more than one MPI processes such as N>1, ZEUS cuts the block into N sub-blocks along the Z direction with each sub-block having roughly the same number of grid cells in Z direction. MINZ is the minimum number of grid cells along the Z direction for sub-blocks. For any block, if the number of MPI processes determined earlier is too large resulting in any of its sub-blocks with less than MINZ number of grid cells in Z direction, the

number of MPI processes for this block has to be reduced until all its sub-blocks have at least MINZ number of grid cells in Z direction.

3. If ZEUS reads a **FINDMPI** card from the input file, it will determine the **BLKMPI** bulk data card input, print out this **BLKMPI** bulk data card in the output file, then terminate this job. An example of this **BLKMPI** bulk data card is shown as follows:

```
MPICPU        19
BLKMPI         1     -16      -1      -1      -1

                                        ****************************************
                                        ***                                  ***
                                        ***  Z E U S    T E R M I N A T E D ***
                                        ***                                  ***
                                        ***         N O R M A L L Y          ***
                                        ***                                  ***
                                        ***     10:33:52    03/03/2017       ***
                                        ***                                  ***
                                        ****************************************
```

The user then can cut the **BLKMPI** bulk data card input from the output file and insert it into ZEUS input file as well as insert the **MPICPU** executive control command into the executive control command section of the input file. Please remember to remove or comment out the **FINDMPI** card before you launch ZEUS MPI job with the inserted **BLKMPI** bulk data card setup.

FIXMACH

Non-Matched Point Flutter Analysis Using the *g*-Method

Description: Non-matched point flutter analysis at a FIXed MACH number with varying velocity and density pairs. Referred to by the **FLUTTER** bulk data card.

Format and Example:

1	2	3	4	5	6	7	8	9	10
FIXMACH	SETID	IDMK		FTMUNIT	FTLUNIT	VREF	FLUTTF	PRINT	CONT
CONT	V1	RHO1	V2	RHO2	…	…	-etc-	-etc-	

FIXMACH	100	10		slug	ft	1.0		3	+FIX1
+FIX1	900.	.002	1000.	.00238	1200.	.0030			

Field	Contents
SETID	Unique set of identification number among all **FIXMATM**, **FIXMACH**, and **FIXMDEN** bulk data cards. (Integer > 0) (See Remark 1)
IDMK	Identification number of an **MKAEROZ** bulk data card that defines the Mach number and its associated unsteady aerodynamics to be used in the flutter analysis. (Integer > 0) (See Remark 2)
FTMUNIT	Units of mass used in the flutter analysis. Must be one of "SLIN", "SLUG", "LBM", "G", "KG", "LBF/" or "N/". (Character). (See Remark 3)
FTLUNIT	Units of length used in the flutter analysis. Must be one of "IN", "FT", "M", "MM", "CM", or "KM". (Character). (See Remark 4)
VREF	Reference velocity. Used to normalize the output velocity. (Real, Default = 1.0) (See Remark 5)
FLUTTF	Identification number of a **FLUTTF** bulk data card that defines a second order linear system. (Integer ≥ 0) (See Remark 6)
PRINT	Print-out options (Integer): IABS(PRINT) = 0, or 1 prints out damping and frequency versus velocity diagrams. IABS(PRINT) = 2, prints out damping and frequency versus velocity/VREF diagrams. IABS(PRINT) = 3, prints out damping and frequency versus dynamic pressure diagrams. IABS(PRINT) = 4, prints out damping and frequency versus density diagrams. For PRINT < 0, prints out generalized mass, generalized stiffness, and generalized aerodynamic forces matrices.

Vi, RHOi	List of velocity and density pairs. The units of velocity must be in FTLUNIT/SEC and density must be in FTMUNIT/FTLUNIT^3. (Real > 0.0). (See Remark 7)

Remarks:

1. **FIXMACH** is referred to by the **FLUTTER** bulk data card that specifies the entry FIX = SETID. The g-Method is used for the flutter analysis.

2. The Mach number and its associated unsteady aerodynamics are those of the **MKAEROZ** bulk data card with identification number = IDMK. The g-Method computes the flutter frequencies and damping of structural modes at the given velocity and density pairs. No atmosphere table is required.

3. SLUG" = slug, "SLIN" = slinch (slinch = slug × 12.0), "LBM" = pound mass, "G" = gram, "KG" = kilogram, "LBF/" = $\left(\dfrac{\text{lbf}}{\text{FTLUNIT}}\right)\sec^2$ and "N/" = $\left(\dfrac{\text{Newton}}{\text{FTLUNIT}}\right)\sec^2$. Note that FTMUNIT cannot be "NONE'.

4. "FT" = foot, "IN" = inch, "MM" = millimeter, "CM" = centimeter, "M" = meter, and "KM" = kilometer. Note that FTLUNIT cannot be "NONE'.

   Note that FTMUNIT = LBF/, or N/ is the options to specify the units of mass in terms of units of force and length (FTLUNIT). For instance, FTMUNIT = LBF/ and FTLUNIT = IN imply that the units of mass = $\left(\dfrac{\text{lbf}}{\text{inch}}\right)\sec^2$ = slinch. It should be noted that there is a slash ("/") attached to the force unit.

   All units of length and mass involved in the flutter analysis, as well as the flutter result output, are in FTLUNIT and FTMUNIT, respectively. Thus, all units of length and mass specified in the **AEROZ** and **ATMOS** bulk data cards are converted to these of the flutter analysis.

5. Output velocities are V/VREF. A typical example is to convert velocity from ft/sec to knots by specifying VREF=1.68.

6. Optional input. The mass, damping and stiffness matrices defined in **FLUTTF** bulk data card are added to the flutter (or ASE) equations. Thus, **FLUTTF** can be used to define a second-order linear system representing a control system or to modify the mass, damping or stiffness of the finite element model.

7. The dynamic pressures associated with the velocity and density list must be in <u>ascending</u> order.

FIXMATM

Matched Point Flutter Analysis Using the g-Method

Description: Matched point flutter analysis at a FIXed Mach number (M) and a given ATMosphere table with varying altitudes. Referred to by the **FLUTTER** bulk data card.

Format and Example:

1	2	3	4	5	6	7	8	9	10
FIXMATM	SETID	IDMK	IDATM	FTMUNIT	FTLUNIT	VREF	FLUTTF	PRINT	CONT
CONT	ALT1	ALT2	…	ALTi	-etc-				

FIXMATM	100	10	12	slug	ft	1.0		-1	+FIX1
+FIX1	-10000.	0.	10000.	20000.	30000.				

Field	Contents
SETID	Unique set identification number among all **FIXMATM**, **FIXMACH**, and **FIXMDEN** bulk data cards. (Integer > 0) (See Remark 1)
IDMK	Identification number of a **MKAEROZ** bulk data card that defines the Mach number and its associated unsteady aerodynamics used in the flutter analysis. (Integer>0) (See Remark 2)
IDATM	Identification number of an **ATMOS** bulk data card to define an atmosphere table. (Integer ≥ 0) (See Remark 3)
FTMUNIT	Units of mass used in the flutter analysis. Must be one of “SLIN”, “SLUG”, “LBM”, “G”, “KG”, “LBF/” or “N/”. (Character) (See Remark 4)
FTLUNIT	Units of length used in the flutter analysis. Must be one of “IN”, “FT”, “M”, “MM”, “CM”, or “KM”. (Character) (See Remark 5)
VREF	Reference velocity. Used to normalize the output velocity. (Real, Default=1.0) (See Remark 6)
FLUTTF	Identification number of a **FLUTTF** bulk data card that defines a second order linear system. (Integer ≥ 0) (See Remark 7)
PRINT	Print-out options (Integer): IABS(PRINT) = 0, or 1, prints out damping and frequency versus velocity diagrams. IABS(PRINT) = 2, prints out damping and frequency versus velocity/VREF diagrams. IABS(PRINT) = 3, prints out damping and frequency versus dynamic pressure diagrams. IABS(PRINT) = 4, prints out damping and frequency versus density diagrams. IABS(PRINT) = 5, prints out damping and frequency versus altitudes diagrams.

	For PRINT < 0, prints out generalized mass, generalized stiffness, and generalized aerodynamic forces matrices.
ALTi	List of altitudes. Units must be in FTLUNIT. (Real) (See Remark 8)

Remarks:

1. **FIXMATM** is referred to by a **FLUTTER** bulk data card that specifies the entry FIX = SETID. The g-Method is used for flutter analysis.

2. **FIXMATM** can be used to simulate a typical flutter flight test condition where the aircraft flies at a fixed Mach number but with varying altitudes. The speed and density at each altitude (ALTi) and the fixed Mach number are provided by the atmosphere table referred to by the IDATM entry.

3. If IDATM = 0, then the standard atmosphere table is used. This standard atmosphere table is built into the program. The altitudes built in the standard atmosphere table ranges from -100 kft to 260 kft.

4. SLUG" = slug, "SLIN" = slinch (slinch = slug × 12.0), "LBM" = pound mass, "G" = gram, "KG" = kilogram, "LBF/" = $\left(\dfrac{\text{lbf}}{\text{FTLUNIT}}\right)\text{sec}^2$ and "N/" = $\left(\dfrac{\text{Newton}}{\text{FTLUNIT}}\right)\text{sec}^2$. Note that FTMUNIT cannot be "NONE'.

5. "FT" = foot, "IN" = inch, "MM" = millimeter, "CM" = centimeter, "M" = meter, and "KM" = kilometer. Note that FTLUNIT cannot be "NONE'.

   Note that FTMUNIT = LBF/, or N/ is the options to specify the units of mass in terms of units of force and length (FTLUNIT). For instance, FTMUNIT = LBF/ and FTLUNIT = IN imply that the units of mass = $\left(\dfrac{\text{lbf}}{\text{inch}}\right)\text{sec}^2$ = slinch. It should be noted that there is a slash ("/") attached to the force unit.

   All units of length and mass involved in the flutter analysis, as well as flutter result output, are in FTLUNIT and FTMUNIT, respectively. Thus, all units of length and mass specified in the **AEROZ** and **ATMOS** bulk data cards are converted to these of the flutter analysis.

6. Output velocities are V/VREF. A typical example is to convert velocity from ft/sec to knots by specifying VREF=1.68. Active only for flutter analysis.

7. Optional input. The mass, damping and stiffness matrices defined in **FLUTTF** bulk data card are added to the flutter (or ASE) equations. Thus, **FLUTTF** can be used to define a second-order linear system representing a control system or to modify the mass, damping or stiffness of the finite element model.

8. The list of altitudes must be in <u>ascending</u> order, i.e., from low altitude to high altitude. However, the output flutter results are listed from high altitude to low altitude.

FIXMDEN

Non-Matched Point Flutter Analysis Using Both the *g*-Method and the k-Method

Description: Non-matched point flutter analysis at a FIXed Mach number and DENsity with varying velocities. Referred to by the **FLUTTER** bulk data card.

Format and Example:

1	2	3	4	5	6	7	8	9	10
FIXMDEN	SETID	IDMK	DEN	FTMUNIT	FTLUNIT	VREF	FLUTTF	PRINT	CONT
CONT	V1	V2	…	Vi	…	…	-etc-		

FIXMDEN	100	10	.00238	slug	ft	1.0		3	+FIX1
+FIX1	900.	950.	1000.	1100.	1200.	2000.			

Field	Contents
SETID	Unique set identification number among all **FIXMATM**, **FIXMACH**, and **FIXMDEN** bulk data cards. (Integer > 0) (See Remark 1)
IDMK	Identification number of an **MKAEROZ** bulk data card that defines the Mach number and its associated unsteady aerodynamics used in the flutter analysis. (Integer>0) (See Remark 2)
DEN	Density. Units must be in FTMUNIT/FTLUNIT^3. (Real > 0.) (See Remark 2.)
FTMUNIT	Units of mass used in the flutter analysis. Must be one of “SLIN”, “SLUG”, “LBM”, “G”, “KG”, “LBF/” or “N/”. (Character) (See Remark 3)
FTLUNIT	Units of length used in the flutter analysis. Must be one of “IN”, “FT”, “M”, “MM”, “CM”, or “KM”. (Character) (See Remark 4)
VREF	Reference velocity. Used to normalize the output velocity. (Real, Default=1.0) (See Remark 5)
FLUTTF	Identification number of a **FLUTTF** bulk data card that defines a second order linear system. (Integer ≥ 0) (See Remark 6)
PRINT	Print-out options (Integer): IABS(PRINT) = 0, or 1, prints out damping and frequency versus velocity diagrams. IABS(PRINT) = 2, prints out damping and frequency versus velocity/VREF diagrams. IABS(PRINT) = 3, prints out damping and frequency versus dynamic pressure diagrams. For PRINT < 0, prints out generalized mass, generalized stiffness, and generalized aerodynamic forces matrices.

Vi List of velocities. The units of velocity must be in FTLUNIT/sec. (Real > 0.0). (See Remark 7)

Remarks:

1. **FIXMDEN** is referred to by the **FLUTTER** bulk data card that specifies the entry FIX = SETID. The flutter analysis is performed using both the g-Method and the K-Method to provide two sets of solutions for comparison. Furthermore, the divergence speeds are computed using the generalized aerodynamic forces at zero reduced frequency. All flutter and divergence speeds computed by the K-Method and the divergence speed analysis, respectively, that are lower than the highest velocity in velocity list (Vi) are printed out.

2. The Mach number and its associated unsteady aerodynamics are those of the **MKAEROZ** bulk data card with identification number = IDMK. The g-Method and the K-Method compute the flutter frequencies and damping of structural modes at the given density and list of velocities. No atmosphere table is required.

3. SLUG" = slug, "SLIN" = slinch (slinch = slug × 12.0), "LBM" = pound mass, "G" = gram, "KG" = kilogram, "LBF/" = $\left(\frac{\text{lbf}}{\text{FTLUNIT}}\right)\text{sec}^2$ and "N/" = $\left(\frac{\text{Newton}}{\text{FTLUNIT}}\right)\text{sec}^2$. Note that FTMUNIT cannot be "NONE'.

4. "FT" = foot, "IN" = inch, "MM" = millimeter, "CM" = centimeter, "M" = meter, and "KM" = kilometer. Note that FTLUNIT cannot be "NONE'.

   Note that FTMUNIT = LBF/, or N/ is the options to specify the units of mass in terms of the units of force and length (FTLUNIT). For instance, FTMUNIT = LBF/ and FTLUNIT = IN imply that the units of mass = $\left(\frac{\text{lbf}}{\text{inch}}\right)\text{sec}^2$ = slinch. It should be noted that there is a slash ("/") attached to the force unit.

   All units of length and mass involved in the flutter analysis, as well as flutter result output, are in FTLUNIT and FTMUNIT, respectively. Thus, all units of length and mass specified in the **AEROZ** and **ATMOS** bulk data cards are converted to these of the flutter analysis.

5. Output velocities are V/VREF. A typical example is to convert the velocity from ft/sec to knots by specifying VREF = 1.68.

6. Optional input. The mass, damping and stiffness matrices defined in the **FLUTTF** bulk data card are added to the flutter equations. Thus, **FLUTTF** can be used to define a second-order linear system representing a control system or to modify the mass, damping or stiffness of the finite element model.

7. The velocities in the list must be in <u>ascending</u> order.

FLTFAST

Composite Sinusoidal Excitation For Frequency-Domain Flutter Analysis

Description: Frequency-domain flutter analysis using a composite sinusoidal (COMSINE) excitation technique to generate the generalized aerodynamic forces.

Format and Example:

1	2	3	4	5	6	7	8	9	10
FLTFAST	SETID	SYM	FIX	NMODE	TABDMP	MLIST	CONMLST	NKSTEP	

FLTFAST	100	SYM	1	0	30	100	0	50	

Field	Contents
SETID	Unique set identification number. (Integer > 0) (See Remark 1)
SYM	Character string up to 8 characters with no embedded blanks. The first 4 characters can be either "SYMM" (or "SYM"), or "ASYM" that defines the boundary condition of the structural finite element model as well as the unsteady aerodynamics, where: SYMM Symmetric boundary condition ASYM Asymmetric boundary condition (See Remark 2) The last 4 characters are used to specify the interpolation scheme for the generalized aerodynamic matrices. They can be either blank for a cubic spline, "L" for a linear interpolation (such as SYM= "SYMML",= or "ASYML"), "P" for a second-order-polynomial interpolation (such as SYM= "SYMMP" or "ASYMP"), or an integer for a hybrid cubic spline and linear interpolation scheme (such as SYM= "SYMM1", "SYMM2", etc.) (Default = SYMML, See Remark 3)
FIX	Identification number of a **FIXMATM**, **FIXMACH**, or **FIXMDEN** bulk data card. (Integer > 0) (See Remark 4)
NMODE	Number of structural modes used in the flutter analysis. (Integer ≥ 0) (See Remark 5)
TABDMP	Identification number of a **TABDMP1** bulk data card specifying modal damping as a function of natural frequency. (Integer ≥ 0) (See Remark 6)
MLIST	Identification number of a **SET1** or **SETADD** bulk data card specifying a list of normal modes to be omitted from the flutter analysis (Integer ≥ 0) (See Remark 7)
CONMLST	Identification number of a **CONMLST** bulk data card specifying a list of identification numbers of the **CONM1** bulk data cards for mass perturbation. (Integer ≥ 0) (See Remark 8)
NKSTEP	Number of reduced frequency steps for the reduced-frequency-sweep technique of the g-method flutter solution used only for the FLUTTER discipline. (Integer ≥ 0, Default = 25) (See Remark 9)

Remarks:

1. All entries of the **FLTFAST** bulk data card are identical to the **FLUTTER** bulk data card except the default of NKSTEP = 10.

2. Similar to the **FLUTTER** bulk data card, the **FLTFAST** bulk data card is also referred to by the **FLUTTER** Case Control Command.

3. The composite sinusoidal excitation (COMSINE) technique defines the input excitation by a series of decay sine functions such as

$$f(t) = \sum_{i=1}^{NK} e^{-ak_iT} \sin(k_iT)$$

where NK is the number of reduced frequencies listed in the **MKAEROZ** bulk data card
k_i is the ith reduced frequency
a is the damping ratio with default a value of 0.25. The user can specify a **PARAM** bulk data card with entry NAME = "DAMPING" to alter this default value, then obtains the frequency-domain generalized aerodynamic forces by the Fourier transform and T = $t/\left(\frac{REFC}{2}\right)$. Where REFC is the reference chord defined in the **AEROZ** bulk data card.

It can be seen that the COMSINE technique excites the aeroelastic system with all of the reduced frequencies in a single time-domain computation as opposed to the sinusoidal excitation used by the **FLUTTER** bulk data card which performs the time-domain computation for each reduced frequency. Therefore, the COMSINE technique is much more computational efficient than the sinusoidal excitation technique.

The time step (DT) is automatically determined by the program using the following equation:

$$DT = \frac{4\pi}{\left(REFC/2\right) \cdot NSTEP \cdot k_{\min}}$$

where REFC is the reference chord defined in the **AEROZ** bulk data card, NSTEP is the number of time steps for the unsteady aerodynamic computation using COMSINE. The default value is 3001. This default value can be changed using a **PARAM** bulk data card with entry NAME = "FLTSTEP". K_{min} is the smallest non-zero reduced frequency listed in the **MKAEROZ** bulk data card.

Prior to the COMSINE technique being applied, a steady aerodynamic computation is performed first to ensure a converged steady solution before the unsteady computation. The number of pseudo time steps for this steady aerodynamic computation is 100 but that can be altered using a bulk data card, **MKPARAM** with entry TRMSTEP.

FLUTTER — Flutter Analysis Bulk Data Card

Description: Performs the frequency-domain flutter or defines the structural parameters for the transient response analysis.

Format and Example:

1	2	3	4	5	6	7	8	9	10
FLUTTER	SETID	SYM	FIX	NMODE	TABDMP	MLIST	CONMLST	NKSTEP	
FLUTTER	100	SYMM3	1	0	30	100	0	50	

Field	Contents
SETID	Unique set identification number. (Integer > 0) (See Remark 1)
SYM	Character string up to 8 characters with no embedded blanks. The first 4 characters can be either "SYMM" (or "SYM"), "ANTI" or "ASYM" that defines the boundary condition of the structural finite element model as well as the unsteady aerodynamics, where: SYMM Symmetric boundary condition. ANTI Anti-symmetric boundary condition. ASYM Asymmetric boundary condition .(See Remark 2) The last 4 characters are used to specify the interpolation scheme for the generalized aerodynamic matrices. They can be either blank for a cubic spline, "L" for a linear interpolation (such as SYM= "SYMML",= or "ASYML"), "P" for a second-order-polynomial interpolation (such as SYM= "SYMMP" or "ASYMP"), or an integer for a hybrid cubic spline and linear interpolation scheme (such as SYM= "SYMM1", "SYMM2", etc.) (Default = SYMML, See Remark 3)
FIX	Identification number of a **FIXMATM**, **FIXMACH**, or **FIXMDEN** bulk data card. (Integer > 0) (See Remark 4)
NMODE	Number of structural modes used in the flutter analysis. (Integer ≥ 0) (See Remark 5)
TABDMP	Identification number of a **TABDMP1** bulk data card specifying modal damping as a function of natural frequency. (Integer ≥ 0) (See Remark 6)
MLIST	Identification number of a **SET1** or **SETADD** bulk data card specifying a list of normal modes to be omitted from the flutter analysis (Integer ≥ 0) (See Remark 7)
CONMLST	Identification number of a **CONMLST** bulk data card specifying a list of identification numbers of the **CONM1** bulk data cards for mass perturbation. (Integer ≥ 0) (See Remark 8)

NKSTEP Number of reduced frequency steps for the reduced-frequency-sweep technique of the g-method flutter solution used only for the FLUTTER discipline. (Integer ≥ 0, Default = 25) (See Remark 9)

Remarks:

1. The **FLUTTER** bulk data card is referred to by a **FLUTTER** Case Control Command for the frequency-domain flutter analysis or by a **MLOADS**, **ELOADS**, **GLOADS**, or **NLFLTR** bulk data card for the transient response analysis.

   If the linearized Euler solver is used to generate the frequency-domain Generalized Aerodynamic Force (GAF) matrix by specifying the entry METHOD = 3 in the **MKPARAM** bulk data card, the pseudo times steps are specified by the entry TRMSTEP in the **MKPARAL** bulk data card. Otherwise, a sinusoidal excitation technique is used to transfer the time-domain unsteady aerodynamics to the frequency-domain using the frequency-domain unsteady aerodynamic computation of the Euler-Solver module. This sinusoidal excitation technique is performed on each reduced frequency listed in the **MKAEROZ** bulk data card and for each structural mode. Thus, the NK reduced frequencies and NMODE structural modes; the total number of time-domain computations is NK × NMODE. This is a very slow procedure but yields very accurate frequency-domain unsteady aerodynamics.

   The GAF matrix can be saved by specifying the entry SAVE="SAVE" in the **MKAEROZ** bulk data card. The GAF matrix can be also retrieved by specifying the entry SAVE="ACQUIRE" to skip the GAF computation through the Euler solver module. In this case, the computational time will be very small.

   Note that the number of cycles of the sinusoidal excitation can be altered using the **PARAM** bulk data card with NAME = “N2CYC”.

   For a given reduced frequency *k*, the total time ($T_{\max}$) of the time-domain computation is two-cycles of sinusoidal excitation at a given structural mode, thus $T_{\max} = 2\left(\dfrac{2\pi \cdot REFC/2}{k}\right)$.

   The time step is then computed by $\dfrac{T_{\max}}{NSTEP}$, where *NSTEP* = 201 as a default value. The user can use a **MKPARAM** bulk data card with entry FLTSTEP to alter the default value of NSTEP.

   After the time-domain aerodynamic forces are obtained, ZEUS performs a Fourier series analysis to transfer the time-domain solution to the frequency domain. This frequency-domain aerodynamic force is called generalized aerodynamic forces and is used by the g-method flutter solution technique for flutter solutions.

   It should be noted that prior to the time-domain unsteady computation, a steady aerodynamic analysis is always performed first. This is to ensure a converged steady solution so that the unsteady solution becomes a small perturbation about the steady mean flow solution. The number of the pseudo time-step is 100 as a default value that can be changed using a **MKPARAM** bulk data card with entry TRMSTEP. This steady flow solution can be saved or retrieved using a **STFLOW** bulk data card. The pressure distribution of such a steady aerodynamic analysis can be visualized using the **PLTCP** bulk data card.

2. The corresponding structural modal data must be matched with the entry BOUNDARY of the ′**ASSIGN FEM =**′ command in the Executive Control Section.

3. The g-method reduced-frequency-swept technique requires the interpolation of the generalized aerodynamic matrix from those computed at the reduced frequency list defined in the **MKAEROZ** bulk data card. The flutter solution computed by the g-method may depend on the interpolation scheme especially if the unsteady aerodynamic computations at high reduced frequencies are not converged.

   Since the cubic spline (SYM= “SYMM” , “ANTI”, or “ASYM”) provides a continuous slope of the generalized aerodynamic matrices with respect to the reduced frequencies, it normally gives accurate aerodynamic damping solution because the g-method shows that the aerodynamic damping is related to the slope of the generalized aerodynamics. However, if non-converged generalized aerodynamics at high frequencies occurs, they may contaminate the interpolated results at low reduced frequencies due to the cubic spline scheme.

   The above problem can be completely removed by using the linear interpolation (SYM= “SYMML” , “ANTIL’ or “ASYML”). However, this gives discontinuous slopes of the damping solution.

   The second-order-polynomial interpolation scheme (SYM= “SYMMP”, “ANTIP”, or “ASYMP”) selects every other three consecutive points and establishes a second order polynomial for interpolation, thereby avoids the impact of high-frequency aerodynamics on the interpolated low-frequency results if they are not within three consecutive points. However, the slopes of the interpolated results computed by the second-order-polynomial interpolation scheme are only piecewisely continuous.

   The hybrid cubic spline and linear interpolation scheme (such as SYM= “SYMMn”, “ANTIn”, or “ASYMn”, where n is an integer) divides the reduced frequencies into a low-frequency region and a high-frequency region where the high-frequency region contains the last n+1 reduced frequencies in the reduced frequency list. The cubic spline is employed for the low-frequency region and the linear interpolation for the high-frequency region. In so doing, the damping solution is accurate at low reduced frequencies, and the dependence of low-frequency results on high-frequency aerodynamics can be avoided.

4. A **FIXMATM**, **FIXMACH**, or **FIXMDEN** bulk data card is used to define the altitudes, Mach numbers, densities, or velocities for matched or non-matched point flutter analysis.

5. If NMODE = 0, all structural modes are used for flutter analysis.

6. If TABDMP = 0, zero modal damping is assumed.

7. If MLIST is blank or zero, the number of modes used in the flutter analysis is NMODE.

8. If CONMLST = 0, no mass perturbation is performed. The CONMLST option provides the capability to change the mass of the structural finite element model without re-running the finite element analysis. A typical example of a CONMLST application is the fuel variation with respect to a baseline fuel weight (the finite element model corresponds to the baseline fuel weight which is imported by the **'ASSIGN FEM='** Executive Control Command). Another application of the CONMLST option is the fictitious mass method for accurate transient dynamic loads analysis (see Section 5.10.2 for the description of the fictitious mass method).

9. The reduced frequency step (ΔK) is defined by $\Delta K = K_{\max} / NKSTEP$, where K_{max} is the maximum reduced frequency specified in the **MKAEROZ** bulk data card. The reduced-frequency-swept technique increases the reduced frequency incrementally by ΔK to search for the flutter solution. Normally, NKSTEP = 25 is sufficient to provide accurate flutter solution. However, if some peculiar aerodynamic lags occur, it is recommended that NKSTEP be increased to verify the flutter results.

FLUTTF

Second Order Transfer Function

Description: Defines a second order linear system *f(s)* such that:

$$f(s) = s^2 [\boldsymbol{M}] + s[\boldsymbol{C}] + [\boldsymbol{K}]$$

where *s* is the Laplace parameter.

Format and Example:

1	2	3	4	5	6	7	8	9	10
FLUTTF	SETID	GSET	C1	MASS	C2	DAMPING	C3	STIFF	

FLUTTF	101	YES	-1.0	M2GG	0.0		-1.0	K2GG	

Field	Contents
SETID	Unique set identification number. (Integer ≠ 0) (See Remark 1)
GSET	Character string either "YES" or "NO". (Default = "YES") (See Remark 2) For GSET = "YES", the matrices MASS, DAMPING and STIFF are defined in the G-set degrees of freedom (G-set d.o.f. = 6 × number of structural finite element grid points). For GSET = "NO", the matrices MASS, DAMPING, and STIFF are defined in the generalized coordinates.
C1, C2, C3	Multipliers to the MASS, DAMPING and STIFF matrices, respectively. (Real, Default = 1.0)
MASS, DAMPING, STIFF	Character strings specifying the name of the MASS, DAMPING, and STIFF matrices, respectively, that are imported by the **DMI** bulk data card, **DMIG** bulk data card or the **'ASSIGN MATRIX='** Executive Control Command. (Character or Blank) (See Remark 3)

Remarks:

1. The absolute value of SETID is referred to by the **FIXMATM**, **FIXMACH** or **FIXMDEN** bulk data card. The transfer function *f(s)* is added to the flutter (or ASE) equation. Therefore, the total generalized mass, damping and stiffness matrices in the flutter (or ASE) equations are those imported from the **'ASSIGN FEM='** Executive Control Command plus those defined by the **FLUTTF** bulk data card, respectively.

   Note: SETID can be a negative integer. This will (1) recompute the natural frequencies of the structures. These recomputed natural frequencies are used for sorting the velocity-damping-frequency list of flutter results, and (2) compute the eigenvectors of the modified structure and redefine the generalized coordinates based on this new set of eigenvectors.

Therefore, the resulting generalized mass and stiffness matrices are diagonal. A fatal error can occur if the perturbed mass matrix is not positive definite.

2. For GSET = "YES", the program will convert the MASS, DAMPING and STIFF matrices to the generalized coordinates by:

$$[\boldsymbol{M}] = [\varphi]^T \text{ [MASS] } [\varphi]$$

$$[\boldsymbol{C}] = [\varphi]^T \text{ [DAMPING] } [\varphi]$$

$$[\boldsymbol{K}] = [\varphi]^T \text{ [STIFF] } [\varphi]$$

where $[\varphi]$ is the modal matrix imported by the **'ASSIGN FEM='** Executive Control Command.

For GSET = "NO", MASS, DAMPING, and STIFF matrices must be square matrices and have their sizes being equal to number of modes imported by the **'ASSIGN FEM='** Executive Control Command.

It should be noted that the mass and length units involved in the transfer function must be consistent with those defined in the **AEROZ** bulk data card.

3. MASS, DAMPING or STIFF can be blank. This implies that the corresponding matrix is a null matrix.

FOILSEC

NACA Airfoil Section

Description: Defines an NACA-series type of airfoil section.

Format and Example:

1	2	3	4	5	6	7	8	9	10
FOILSEC	ID								CONT
CONT	$COEFF_1$	$TYPE_1$	$PROFILE_1$	$FIRST_1$	$SECOND_1$	$THIRD_1$	$FOURTH_1$	$FIFTH_1$	CONT
CONT	$COEFF_2$	$TYPE_2$	$PROFILE_2$	$FIRST_2$	$SECOND_2$	$THIRD_2$	$FOURTH_2$	$FIFTH_2$	CONT
CONT			…	etc	…				

FOILSEC	100								+F1
+F1	0.1	NACA	5M	2	4	0	10	34	+F2
+F2	1.0	USER	LINEAR	10	20	30			

Field	Contents
ID	Identification number that is referred to by a **PAFOIL8** bulk data card. (Integer > 0) (See Remark 1)
$COEFF_i$	A coefficient to multiply to the airfoil section. (Real ≥ 0.0) (See Remark 2)
$ITYPE_i$	Character string either "NACA" or "USER". For TYPE="NACA", airfoil is an NACA-series type of airfoil section For TYPE="USER", airfoil is a user-defined airfoil section (Character, Default="NACA")
$PROFILE_i$	Character string For TYPE="NACA" PROFILE can be one of followings: "4", "4M", "5", "5M", "16", "63", "63A", "64", "64A", "65", "65A", "66", or "67" (Character, Default = "4") For TYPE="USER" If PROFILE="LINEAR", use linear interpolation to interpolate the airfoil section from the user-defined airfoil thickness distribution. Otherwise, use cubic spline for interpolation. (Character, Default = "CUBIC")

$FIRST_i$ $SECOND_i$ $THIRD_i$ $FOURTH_i$ $FIFTH_i$	For TYPE="NACA"

PROFILE="4", NACA 4-digit airfoil

$FIRST_i$	the maximum camber in percent chord
$SECOND_i$	the maximum camber location in tenths of chord
$THIRD_i$	the airfoil thickness in percent chord

$FOURTH_i$ and $FIFTH_i$ are not used

Example: NACA 2412

PROFILE="4M", NACA 4-digit modified airfoil

$FIRST_i$	the maximum camber in percent chord
$SECOND_i$	the maximum camber location in tenths of chord
$THIRD_i$	the airfoil thickness in percent chord
$FOURTH_i$	the first and second digit of the appended number (the first digit = 0, 3, 6, or 9, which indicates leading edge radius index number; the second digit indicates the location of maximum thickness in tenths of chord)

and $FIFTH_i$ is not used

Example: NACA 2412-04

PROFILE="5", NACA 5-digit airfoil

$FIRST_i$	the design lift coefficient, $C_l = FIRST_i \times 3/20$
$SECOND_i$	the location of maximum camber in tenths of chord (= $SECOND_i/20$)
$THIRD_i$	=0: a non-reflexed, =1: a reflexed camberline
$FOURTH_i$	= the airfoil thickness in percent chord

$FIFTH_i$ is not used

Example: NACA 24012

PROFILE="5M", NACA 5-digit modified airfoil

$FIRST_i$	the design lift coefficient, $C_l = FIRST_i \times 3/20$
$SECOND_i$	the location of maximum camber in tenths of chord (= $SECOND_i/20$)
$THIRD_i$	=0: a non-reflexed, =1: a reflexed camberline
$FOURTH_i$	the airfoil thickness in percent chord
$FIFTH_i$	the first and second digit of the appended number (the first digit = 0, 3, 6, or 9, which indicates leading edge radius index number; the second digit indicates the location of maximum thickness in tenths of chord)

Example: NACA 24012-34

PROFILE="16", NACA 16-series airfoil

$FIRST_i$	the design lift coefficient in tenths
$SECOND_i$	the airfoil thickness in percent chord

$THIRD_i$, $FOURTH_i$, and $FIFTH_i$ are not used

Example: NACA 16-212

PROFILE="63", "63A", "64", "64A", "65", "65A", "66", and "67"

NACA 6-series airfoil

$FIRST_i$	the design lift coefficient in tenths
$SECOND_i$	the airfoil thickness in percent chord

$THIRD_i$, $FOURTH_i$, and $FIFTH_i$ are not used

Example: NACA 63-412, NACA 65A-310

For TYPE="USER"

FIRST$_i$ — is the identification number of an **AEFACT** bulk data card used to specify the x- coordinate locations, in percentage of the chord length, where the thickness and camber are specified. The first value listed in the **AEFACT** bulk data card must be 0.0 an the last value must be 100.0

SECOND$_i$ — is the identification number of **AEFACT** bulk data card used to specify the half thickness of the airfoil in percentage of the chord length. (Integer > 0)

THIRD$_i$ — is the identification number of an **AEFACT** bulk data card used to specify the camber of the airfoil in percentage of the chord length.

FOURTH$_i$ and FIFTH$_i$ are not used

(Integer > 0)

Remarks:

1. The **FOILSEC** bulk data card is referred to by a **PAFOIL8** bulk data card to define an NACA-series type of airfoil section.

2. The resulting airfoil shape is the superposition of all airfoil sections multiplied by COEFF$_i$. Thus, the resulting airfoil shape is:

$$F(x) = \sum_{i=1}^{n} COEFF_i \times f_i(x)$$

where

$F(x)$ is the resulting airfoil shape as a function of the chord (x)

and $f_i(x)$ is the ith airfoil section

FREQ — Frequency List

Description: Defines a set of frequencies to be used in the solution of frequency response problems.

Format and Example:

1	2	3	4	5	6	7	8	9	10
FREQ	SETID	F1	F2	F3	F4	F5	F6	F7	
	F8	F9	F10	-etc-					

FREQ	30	0.01	0.2	0.4	0.7	1.2	3.1	3.4	
	10.0	11.0							

Field	Contents
SETID	Unique identification number (Integer > 0) (See Remark 1)
Fi	Frequency value in units of Hz. (Real ≥ 0.0)

Remarks:

1. No duplicated identification number among **FREQ** and **FREQ1** bulk data cards is allowed.

FREQ1 Frequency List

Description: Defines a set of frequencies to be used in the solution of frequency response problems by specification of a starting frequency increment, and the number of increments desired.

Format and Example:

1	2	3	4	5	6	7	8	9	10
FREQ1	SETID	F1	DF	NDF					

FREQ1	6	2.9	0.5	13					

Field	Contents
SETID	Unique set identification number (Integer > 0) (See Remark 1)
F1	First frequency in set. (Real ≥ 0.0) (See Remark 2)
DF	Frequency increment. (Real > 0)
NDF	Number of frequency increments. (Integer > 0; Default = 1) (See Remark 3)

Remarks:

1. No duplicated identification number among **FREQ** and **FREQ1** bulk data cards is allowed.

2. The unites for F1 and DF are Hz.

3. The frequencies defined by this entry are given by

$$f_i = \text{F1} + \text{DF} \times (i - 1)$$

   where i = 1 to (NDF + 1).

GAP

Fictitious Surface on The X-Y Plane

Description: Defines the fictitious surface to add more grid points or to enhance the quality of the mesh on the projected X-Y plane.

Format and Example:

1	2	3	4	5	6	7	8	9	10
GAP	EID	LABEL	ACOORD	NSPAN	NCHORD	LSPAN			CONT
CONT	XRL	YRL	ZRL	RCH	LRCHD				CONT
CONT	XTL	YTL	ZTL	TCH	LTCHD				CONT

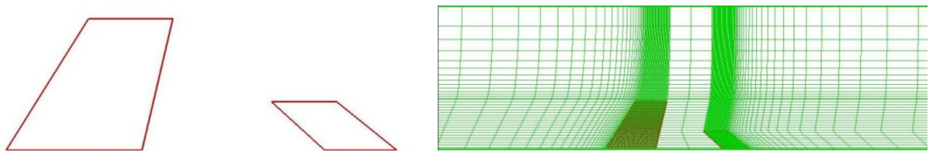
GAP	100		10		21				+G
+G	1.0	0.0	10.0	20.	1				+G
+G	3.0	3.0	10.0	10.					

Field	Contents
EID	Identification Number (Integer > 0) (See Remark 1)
LABEL	Not used.
ACOORD	Identification number of **ACOORD** (specifying a local coordinate system and orientation) bulk data card (Integer ≥ 0, Default =0) (See Remark 2)
NSPAN	Number of spanwise divisions (Integer ≥ 2, Default = 2)
NCHORD	Number of chordwise divisions (Integer ≥ 2, Default = 2) (See Remark 3)
LSPAN	Identification number of **AEFACT** bulk data card used to specify the spanwise divisions of the gap. The number of values listed in AEFACT must be NSPAN. If LSPAN = 0, NSPAN number of evenly distributed spanwise divisions is used. If LSPAN > 0, the values listed in the **AEFACT** bulk data card is the spanwise divisions in terms of percentage of the span. If LSPAN < 0, the values listed in the **AEFACT** bulk data card are the true y locations defined in the local coordinate system defined by the ACOORD entry of the **GAP** bulk data card. (Integer, Default = 0) (See Remark 4)
XRL YRL ZRL	X, Y, and Z location of the root chord leading edge. Note that XRL must be located at the trailing edge of the forward wing. (Real)
RCH	Length of the root chord. (Real ≥ 0.0)

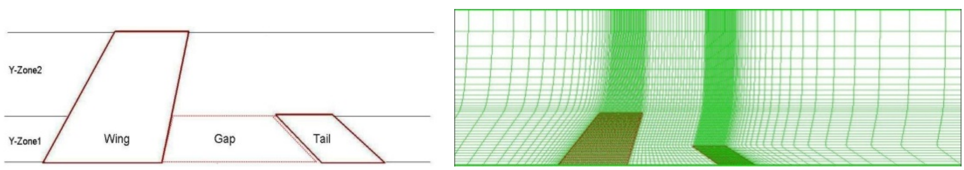
LRCHD	Identification number of **AEFACT** bulk data card used to specify the root chord divisions of the gap component. The number of values listed in AEFACT must be NCHORD. If LRCHD = 0, NCHORD number of evenly distributed chordwise divisions at the root is used. If LRCHD > 0, the values listed in the **AEFACT** bulk data card is the chordwise divisions in terms of percentage of the root chord. If LRCHD < 0, the values listed in the **AEFACT** bulk data card are the true x locations defined in the local coordinate system defined by the ACOORD entry of the **GAP** bulk data card. (Integer, Default = 0) (See Remark 4)
XTL YTL ZTL	Same as XRL, YRL, and ZRL, respectively, except for the tip chord leading edge. (Real)
TCH	Same as RCH except for the tip chord. (Real ≥ 0.0)
LTCHD	Same as LRCHD except for the tip chord.

Remarks:

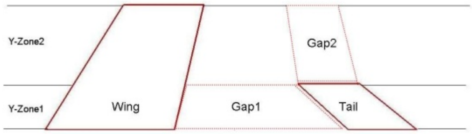
1. The **GAP** bulk data card is referred to by a **BLOCK/BLOCKT** bulk data card. The purpose of the **GAP** bulk data card is to add more gridlines to the mesh on the projected X-Y plane generated by the **BLOCK** bulk data card. For instance, for a wing-horizontal tail configuration shown in the figure below, if no **GAP** bulk data card is used, there remains a gap between the wing and the tail.

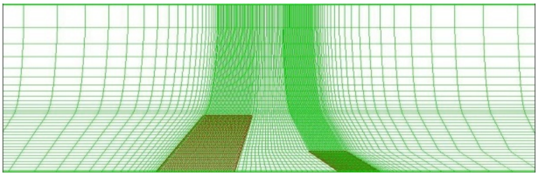


To add more grid points between the wing and the tail, it is recommended to specify a **GAP** bulk data card as shown in the figure below. The resultant grid now does not have any problems as the earlier grid.

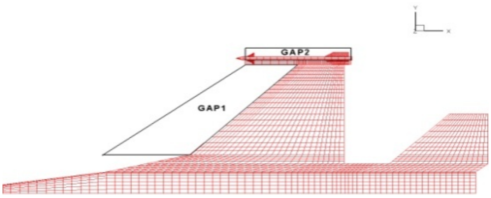


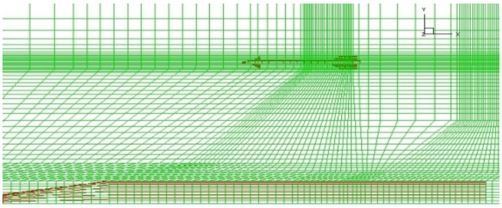
Another use of the **GAP** bulk data card is to enhance the quality of the mesh. For the configuration shown above, the slope of the grid lines outside the tip of the tail is determined by the sweep angle of the wing trailing edge (and not the horizontal tail leading edge) if no **GAP** bulk data card is used. Because the tail has a forward swept angle and the wing has a backward swept angle, the grid lines of the tail have an abrupt change in slope when they leave the tail tip. This may cause large error in the flow solver computation. To make those slopes smoother at the tail tip, one can specify a **GAP** bulk data card attached to the tail tip as the one shown in the following figure. The sweep angle of the **GAP** macroelement should be slightly less than or equal to that of the tail. In so doing, the slope of those grid lines outside the tail tip are determined by the swept angle of the **GAP** macroelement, not by the wing trailing edge. The resulting grid is shown in the figure below.





If GAP is only used to guide the chordwise X gridlines generation, the entries of NSPAN and LSPAN can be left BLANK, and the default values of 2 and 0 will be in effect. **GAP** can also be used to add spanwise Y gridlines by specifying spanwise divisions using NSPAN and LSPAN. For the configuration shown below, the fuselage, wing and tip missile launcher are modeled in the main block, and the tip missile with canards and fins are modeled in other blocks. As the tip missile is relatively small compared to the main wing, the mesh of main block might be too coarse in the region where the tip missile is located. In order to make the meshes of the main block and tip missile block comparable, we can use GAPs to add spanwise Y gridlines as needed. An alternative way is to make more spanwise cuts for the main wing and tip missile launcher, and use much smaller Y growth rate in the **MESHPRM** bulk data card for the main block. However, to use GAPs is much more convenient and has precise control of where to add gridlines. Please note that for the GAP ahead of the main wing leading edge, one only need to specify those spanwise cuts that correspond to the Y gridlines to be added, and there is no need to repeat the spanwise cuts of the main wing.





2. All coordinate locations defined above in XRL, YRL, ZRL, XTL, YTL, and ZTL are in the local coordinate system defined by the **ACOORD** bulk data card. The projected X-Y plane is located in the local coordinate system defined by the **BLOCK/BLOCKT** bulk data card with the entry IACORD.

3. For GAP used to add X gridlines between the wing and tail as for the case discussed in Remark 1, the first gridline generated by the **GAP** bulk data card is actually the trailing edge of the forward wing and the last gridline actually the leading edge of the rear wing. Thus, the number of the gridlines to be added to the mesh on the projected X-Y plane is NCHORD-2.

4. For LSPAN or LRCHD > 0, the values listed in the **AEFACT** bulk data cards must start with <u>0.0</u> and end with <u>100.0</u>.

GAP1

Fictitious Surface on The X-Y Plane

Description: Alternative form of the **GAP** bulk data card to define the fictitious surface for adding more grid points or for enhancing the quality of the rectangular mesh of a block.

Format and Example:

1	2	3	4	5	6	7	8	9	10
GAP1	EID	CAEROL	CAEROT	NCHORD	YZONEL	AEFACTL	YZONER	AEFACTR	CONT
CONT	NSPAN	LSPAN							

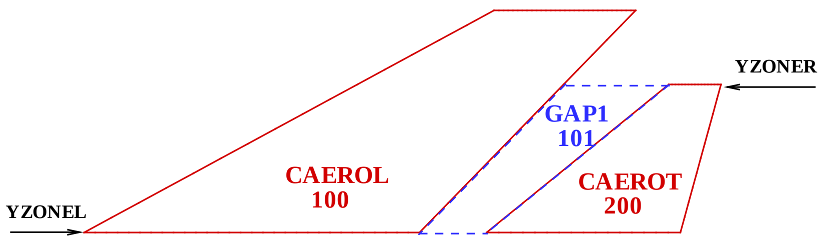
GAP1	101	100	200	25	0.0		10	20	
	5	30							

Field	Contents
EID	Identification number. The functionality of the **GAP1** bulk data card is the same as the **GAP** bulk data card. (Integer > 0)
CAEROL	Identification number of a **CAERO7** bulk data card whose trailing edge defines the line of the leading edge of the fictitious surface. (Integer > 0) (See Remark 1)
CAEROT	Identification number of a **CAERO7** bulk data card whose leading edge defines the line of the trailing edge of the fictitious surface. (Integer > 0)
NCHORD	Number of chordwise divisions (Integer ≥ 2, Default = 2)
YZONEL	The Y location of the inboard edge of the fictitious surface defined in the global coordinates. As an alternative, YZONEL can be an integer that is the identification number of a **YZONEY** bulk data card. In this **YZONEY** bulk data card, the Y location of the inboard edge is specified by the most left/right surface grid point of a CAERO7 or BODY7 Macroelement. (Real or Integer)
AEFACTL	Identification number of the **AEFACT** bulk data card used to specify the inboard chord divisions of the fictitious surface. The number of values listed in AEFACT must be NCHORD. If AEFACTL = 0, NCHORD number of evenly distributed chordwise divisions at the inboard edge is used. If AEFACTL > 0, the values listed in the **AEFACT** bulk data card is the chordwise divisions in terms of percentage of the root chord. If AEFACTL < 0, the values listed in the **AEFACT** bulk data card are the true x locations defined in the global coordinates. (Integer, Default = 0) (See Remark 2)
YZONER	Same as YZONEL except for the outboard edge of the fictitious surface. (Real or Integer)

AEFACTR	Same as AEFACTL except for the outboard edge of the fictitious surface. (Integer, Default = 0).
NSPAN	Number of spanwise divisions (Integer ≥ 2, Default = 2)
LSPAN	Identification number of **AEFACT** bulk data card used to specify the spanwise divisions of the gap. The number of values listed in AEFACT must be NSPAN. If LSPAN = 0, NSPAN number of evenly distributed spanwise divisions is used. If LSPAN > 0, the values listed in the **AEFACT** bulk data card is the spanwise divisions in terms of percentage of the span. If LSPAN < 0, the values listed in the **AEFACT** bulk data card are the true y locations defined in the global coordinates. (Integer, Default = 0) (See Remark 2)

Remarks:

1. The trailing edge and the leading edge of the two **CAERO7** bulk data cards with identification numbers being CAEROL and CAEROT, respectively, as well as YZONEL and YZONER jointly define the boundary of the fictitious surface as shown in the following figure:



2. For AEFACTL or LSPAN > 0, the values listed in the **AEFACT** cards must start with 0.0 and end with 100.0.

GAPZ

Fictitious Surface on The Y-Z Plane

Description: Defines the fictitious surface to add more grid points or to enhance the quality of the mesh on the projected Y-Z plane.

Format and Example:

1	2	3	4	5	6	7	8	9	10
GAPZ	EID	LABEL	ACOORD	NSPAN	NZGRID				CONT
CONT	XRL	YRL	ZRL	RZGAP	LRZGAP				CONT
CONT	XTL	YTL	ZTL	TZGAP	LTZGAP				CONT

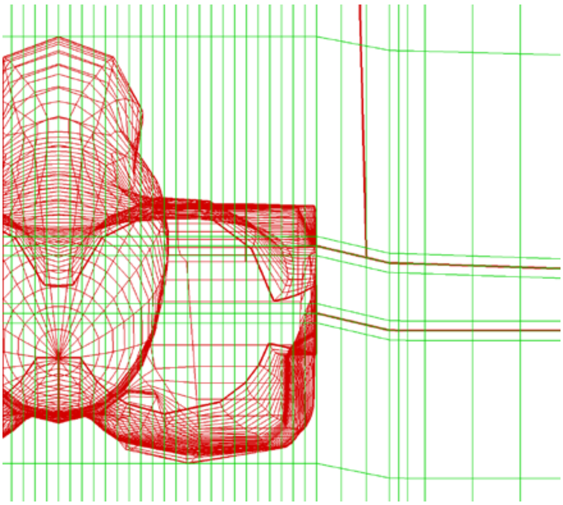
GAPZ	100		30		21				+G
+G	1.0	0.0	30.0	20.0	1				+G
+G	1.0	10.0	-10.0	10.0					

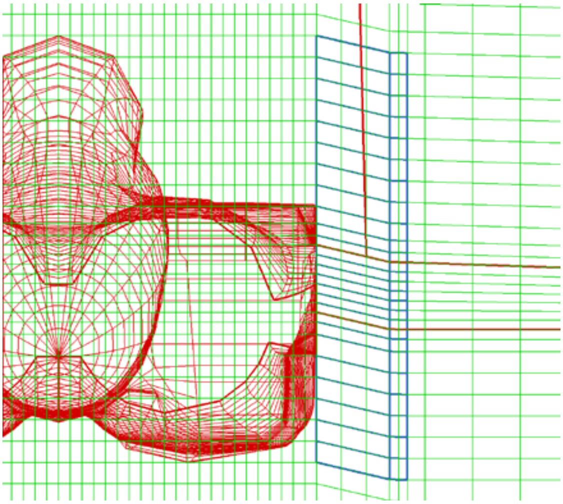
Field	Contents
EID	Identification number. (Integer > 0) (See Remark 1)
LABEL	Not used.
ACOORD	Identification number of an **ACOORD** (specifying a local coordinate system and orientation) bulk data card. (Integer ≥ 0, Default = 0) (See Remark 2)
NSPAN	Not used.
NZGRID	Number of gridlines from the lower to the upper surface of the **GAPZ** on the projected Y-Z plane. (Integer ≥ 2, Default = 2)
XRL YRL ZRL	X, Y, and Z location of the root of the lower surface of the **GAPZ** on the projected Y-Z plane. (Real)
RZGAP	Gap between the lower and upper surfaces of the **GAPZ** along the z direction at the root. Note that the upper root surface of the **GAPZ** is defined by ZRL + RZGAP. (Real > 0.0)
LRZGAP	Identification number of the **AEFACT** bulk data card used to specify NZGRID number of gridline locations between the lower and upper surfaces on the projected Y-Z plane. If LRZGAP = 0, an even distribution of the gridlines is used. If LRZGAP > 0, the values listed in the **AEFACT** bulk data card are in percentage of RZGAP. If LRZGAP < 0, the values listed in the **AEFACT** bulk data card are the z locations

	defined in the local coordinate system defined by the ACOORD entry of the **GAPZ** bulk data card. (Integer) (See Remark 3)
XTL YTL ZTL	Same as XRL, YRL, and ZRL, respectively, except for the tip edge. (Real)
TZGAP	Same as RZGAP except for the tip gap distance. (Real > 0.0)
LTZGAP	Same as LRZGAP except for the tip gap. (Integer)

Remarks:

1. The **GAPZ** bulk data card is referred to by a **BLOCK/BLOCKT** bulk data card. The purpose of the **GAPZ** bulk data card is to add more gridlines to the mesh on the projected Y-Z plane generated by the **BLOCK/BLOCKT** bulk data card. The figure below shows a fuselage, wing, and horizontal tail configuration where there is a gap along the z direction between the horizontal tail and the wing. If no **GAPZ** is specified (and automatic grid generation for the **BODY7** element is disabled), the program will automatically add a set of grid lines above and below the **CAERO7** elements in the projected Y-Z plane, as shown in the figure on the left. Because these gridlines may not be sufficient for an accurate solution, more gridlines can be added using the **GAPZ** bulk data card, as shown in the figure on the right (two **GAPZ** elements are defined in the figure). When using a **GAPZ** bulk data card, automatic grid generation for the **BODY7** element must be disabled by setting the NSEG field to -1.





2. All coordinate locations such as XRL, YRL, ZRL, XTL, YTL, and ZTL are in the local coordinates defined by the **ACOORD** bulk data card. The projected Y-Z plane is located in the local coordinate system defined by the **BLOCK/BLOCKT** bulk data card with entry IACORD. The THETA angle defined by the **ACOORD** bulk data cards referenced by the **GAPZ** and **BLOCK/BLOCKT** bulk data cards must be the same so that the z-axis of the **GAPZ** local coordinate system is aligned with the z-axis of the **BLOCK/BLOCKT** local coordinate system.

3. For LRZGAP > 0, the values listed in the **AEFACT** bulk data card must start with 0.0 and end with 100.0. For LRZGAP < 0, the z values listed in the **AEFACT** bulk data card are mapped to the local coordinate system defined by the **BLOCK/BLOCKT** bulk data card with entry IACORD. Therefore, it is recommended that the **GAPZ** bulk data card refer to the same **ACOORD** bulk data card.

GENGAF — Frequency-Domain Unsteady Aerodynamics

Description: Generates the frequency-domain Generalized Aerodynamic Force (GAF) matrices on an aeroelastically deformed structure and export these matrices to a directory.

Format and Example:

1	2	3	4	5	6	7	8	9	10
GENGAF	SID	EXTFILE	IDMK	SYM	MLIST	IDSURF	W	X0	CONT
CONT	QINF								

GENGAF	100	10	30	ASYM	20	30	0.0		+G
+G	3.2								

Field	Contents
SID	Identification Number that is referred to by a **GENGAF** Case Control Command. If SID<0, the computation of the generalized aerodynamic forces due to structural modes will be skipped. (Integer) Note that if the identification number of a **FLUTTER** bulk data card matches with SID, based on the generated GAF matrices, the GENGAF will continue to perform a frequency-domain flutter analysis using the g-method, (See Remark 1)
EXTFILE	Identification number of an **EXTFILE** bulk data card on which the name of the directory to store the computed GAF matrices is specified. (Integer > 0) (See Remark 2)
IDMK	Identification number of a **MKAEROZ** bulk data card defines the Mach number reduced frequencies and the mean flow condition for which the unsteady aerodynamics are computed. (Integer > 0) (See Remark 3)
SYM	Character string either “ASYM”, “SYMM”, “ASYMS”, or “SYMMS”. For SYM = “ASYM” or “SYMM”, the modal solution imported by the **‘ASSIGN FEM=’** Executive Control Command with BOUNDARY = “ASYM”, “SYMM”, respectively, is used to generate GAF using the composite sinusoidal (COMSINE) excitation technique. For SYMS = “ASYMS” or “SYMMS”; the GAF is generated using the sinusoidal excitation technique. However, if the entry METHOD=3 is specified in the **MKPARAM** bulk data card, the linearized Euler solver is used to generate the GAF matrices.(Character, Default = “SYMMS”) (See Remark 4)
MLIST	Identification number of a **SET1** or **SETADD** bulk data card to list a set of modal indices of the mode shapes. Those modes shapes are excluded from the computation of the generalized aerodynamic force matrices. However, the resulting GAF matrices still have size of NMODE, where NMODE is the number of modes imported by the “ASSIGN FEM=” Executive Control Command, except the rows/columns of those

excluded modes are null. (Integer ≥ 0) (See Remark 5)

IDSURF — Identification number of a **SURFLST** bulk data card to list a set of LABEL's of the **AESURFZ**/**AESLINK** bulk data card. The GAF of those control surface modes are computed. However, the resulting GAF matrices still have columns being equal to all **AESURFZ** bulk data cards specified in the bulk data section except the columns associated with those excluded control surfaces are null. (Integer >0 or Blank)

W — If the linearized Euler solver is used (METHOD=3 in the **MKPARAM** bulk data card), W is the gust induced angle of attack defined as (gust velocity)/(free stream velocity). Otherwise, W is a parameter to define the width of the traveling impulse functions for the generation of the GAF due to a sinusoidal gust. If W = 0.0, GAF due to gust is not computed and is replaced by a null column. (Real ≥ 0.0, Default = 0.0) (See Remark 6)

X0 — The streamwise gust reference point. (Real)

QINF — If QINF is a real number, QINF is a dynamic pressure at which the static aeroelastic analysis is performed. The length unit of QINF is based on that defined the **AEROZ** bulk data card. If QINF is an integer, QINF is the identification number of a **MLDSTAT** bulk data card to retrieve the trim solution computed by a trim analysis through a **MLDTRIM** bulk data card.(Real ≥ 0.0 or integer > 0, Default = 0.0) (See Remark 7)

Remarks:

1. There are three types of GAF that can be generated by the **GENGAF** bulk data card, namely the GAF due to structural modes (QHH), the GAF due to control surface modes (QHC) specified by the **SURFLST** bulk data card, and the GAF due to sinusoidal gust (QHG). The user can import these GAF matrices back to the FLUTTER module of ZEUS or into ZAERO for the frequency-domain aeroelastic/aeroservoelastic analysis.
   The matrices QHH, QHC, and QHG are functions of the reduced frequency (*k*) and calculated by:

$$QHH\left(ik\right)=\phi^{T}AJH(ik),\quad QHC\left(ik\right)=\phi^{T}AJC(ik),\quad QHG\left(t\right)=\phi^{T}AJG(ik)$$

   where ϕ is the modal matrix whose columns contain the structural modes
   ϕ_c is the control matrix whose columns contain the control surface modes
   k is the reduced frequency
   and $AJH\left(ik\right),\ AJC\left(ik\right),\ AJG\left(ik\right)$ are the J*-set aerodynamic forces on surface boxes due to structural modes, control surface modes, and sinusoidal gust, respectively.

   Note that to set the entry SID as a negative integer can avoid the generation of QHH. The generation of QHC and/or QHG can be avoided by specifying IDSURF = "NONE" or blank and W = 0.0, respectively.

2. All GAF and J*-set aerodynamic forces are stored under the directory whose name is specified in the **EXTFILE** bulk data card. Those matrices are in the OUTPUT4 format with the file names shown as follows:

   AJHS*iijj*.DAT: such as AJHS0101.DAT, AJHS0102.DAT, …etc.

AJCS*iijj*.DAT: such as AJCS0101.DAT, AJCS0102.DAT, …etc.
AJGS*iijj*.DAT: such as AJGS0101.DAT, AJGS0102.DAT, …etc.
QHHS*iijj*.DAT: such as QHHS0101.DAT, QHHS0102.DAT, …etc.
QHCS*iijj*.DAT: such as QHCS0101.DAT, QHCS0102.DAT, …etc.
QHGS*iijj*.DAT: such as QHGS0101.DAT, QHGS0102.DAT, …etc.
AJ0S0101.DAT

In addition, QLHS*iijj*, QLCS*iijj*, and QLGS*iijj* for the section load matrices specified by all the **LOADMOD** bulk data cards due to aerodynamic forces of structural modes, control surface modes, and sinusoidal gust, respectively, are also stored, where *ii* is a two-digit integer representing the index of the **MKAEROZ** bulk data card. For instance, ii = "01" refers to the **MKAEROZ** bulk data card whose IDMK entry is the smallest integer among all **MKAEROZ** bulk data cards in the input file and *jj* is a two-digit integer representing the index of the reduced frequencies listed in the **MKAEROZ** bulk data card. For instance, *jj* = "01" refers to the zero frequency, and *jj* = "02" refers to the lowest non-zero reduced frequency. The J-set aerodynamic force matrices compatible with ZAERO, where the Cp on the upper and lower surfaces of CAERO7 elements is combined to a delta Cp, are saved to AJHS*iijj*_ZAERO, AJCS*iijj*_ZAERO.DAT, AJGS*iijj*_ZAERO.DAT, and AJ0S0101_ZAERO.DAT.
Note that if there are NK numbers of reduced frequencies listed in the **MKAEROZ** bulk data card, there are NK numbers of files of each QHH, QHC, and QHG generated under the directory.

The J*-set Aerodynamic force matrices, once computed, can be imported back into the program by the following "ASSIGN MATRIX =" executive control commands:

ASSIGN MATRIX= AJHS0101.DAT, MNAME=AJHS0101
ASSIGN MATRIX= AJCS0101.DAT, MNAME=AJCS0101
ASSIGN MATRIX= AJGS0101.DAT, MNAME=AJGS0101
ASSIGN MATRIX= AJHS0102.DAT, MNAME=AJHS0102
ASSIGN MATRIX= AJCS0102.DAT, MNAME=AJCS0102
ASSIGN MATRIX= AJGS0102.DAT, MNAME=AJGS0102
... etc...

When those J*-set aerodynamic force matrices are imported into the program, the QHHS*iijj*, QHCS*iijj*, QHGS*iijj*, QLHS*iijj*, QLCS*iijj*, and QLGS*iijj* matrices will be computed from those J*-set aerodynamic force matrices directly so that the computational time is significantly reduced. Please see remark 3 of **MKAEROZ** bulk data card.

3. The frequency-domain unsteady aerodynamic computation of the Euler-solver module is used to compute the GAF matrices. However, if the entry METHOD=3 is specified in the **MKPARAM** bulk data card, the linearized Euler solver is used. Prior to the frequency-domain unsteady aerodynamic computation, a static aeroelastic analysis on the flexible aircraft is first performed to obtain the mean flow solution based on the mean flow condition defined in the **TRIMFLT** bulk data card. The mean flow solution on the aeroelastic deformed structure is used as the initial flow solution from which the frequency-domain unsteady aerodynamic computation starts. The number of time steps of the steady aerodynamic computation is 100 but that can be altered using the **MKPARAM** bulk data card with entry TRMSTEP. This flow solution of the static aeroelastic analysis can be saved or retrieved via a **STFLOW** bulk data card.

4. Please refer to the **FLUTTER** and **FLTFAST** bulk data cards for the description of the linearized Euler solver, the sinusoidal excitation, and composite sinusoidal excitation technique, respectively.

5. If there are NM number of modes being imported by the "ASSIGN FEM=" Executive Control Command, the resulting GAF matrices always have the size of NM. If the entry MLIST is used to remove modes, the columns/rows associated with those removed modes in the GAF matrices contain all zeros.

6. For the linearized Euler solver to compute the aerodynamic response due to gust, the following equation is added to the Euler boundary condition

   $$f(t) = We^{-\omega\left(t - \frac{X - X_o}{V}\right)}$$

   where V is the freestream velocity and is not required for input. W is the gust induced angle of attack specified in the entry W.
   and X_o is the entry X0 and ω is the frequency for the frequency-domain linearized Euler computation.

   For the full Euler solver, the time-domain traveling impulse function is defined as

   $$f(t) = e^{-w\left(t - \frac{X - X_o}{V}\right)^z}$$

   where W is the entry W.

   Because the definition of QHG is the aerodynamic response due to a frequency-domain sinusoidal gust whose time-domain counterpart is a traveling delta function. The traveling impulse function is an approximation of the traveling delta function if W is sufficiently large.

   However, very large W could cause numerical problem. Therefore, appropriate W for accurate QHG generation is required.

7. Unlike the FLUTTER module that computes the frequency-domain generalized aerodynamic forces on the rigid aircraft, the GENGAF computed the frequency-domain generalized aerodynamic forces on the flexible aircraft. Thus, the frequency-domain generalized aerodynamic forces computed by the GENGAF module can be treated as the unsteady perturbed aerodynamics on a statically deformed aircraft. This static deformation is computed by performing a static aeroelastic analysis with the dynamic pressure being specified by the entry QINF or by a trim analysis whose trim solution is retrieved by a **MLDTRIM** bulk data card. This implies that QINF=0.0 gives GAF on the rigid aircraft.

GLOADS

Gust Loads Analysis

Description: Defines the aeroelastic system, airframe states, discrete gust profile and the time integration for transient-discrete gust load analysis.

Format and Example:

1	2	3	4	5	6	7	8	9	10
GLOADS	SID	CONID	FLTID	RESTART	STATES	IDGUST	TIME	MLDPRNT	

GLOADS	100	10	20	25	30	100	1000	20	

Field	Contents
SID	Unique set identification number. (Integer > 0) (See Remark 1)
CONID	Not used.
FLTID	Identification number of a **FLUTTER** bulk data card specifying the flight condition and the associated structural and aerodynamic matrices. (Integer > 0) (See Remark 2)
RESTART	Identification number of a **RESTART** bulk data card to save or retrieve the entire flow solution at the transient analysis. (Integer ≥ 0) (See Remark 3)
STATES	Identification number of an **MLDSTAT** bulk data card specifying parameters for rigid-body conversion to airframe states. Note that the **MLDSTAT** bulk data card can refer to a **MLDTRIM** bulk data card to retrieve the trim solution (generated by a trim analysis) as the initial flow condition for the transient response analysis (Integer ≥ 0) (See Remark 4)
IDGUST	Identification number of a **DGUST** bulk data card specifying the discrete gust profile. In addition to the **DGUST** bulk data card, as an option, IDGUST can also refer to an **MLDCOMD** bulk data card for pilot input command. In this case, both cards will have the same ID and the total disturbance is the superposition of the discrete gust and the pilot input command. (Integer > 0)
TIME	Identification number of an **MLDTIME** bulk data card specifying the parameters of the time integration for solving the transient response problem. (Integer > 0)
MLDPRNT	Identification number of a **MLDPRNT** bulk data card specifying the time history of the parameters that are to be printed out. (Integer ≥ 0)

Remarks:

1. For gust loads analysis, the **GLOADS** discipline must be selected in the Case Control Section with **GLOADS** = SID.

Note:

- Prior to the GLOADS analysis, a static aeroelastic analysis is always performed first to obtain a mean flow solution from which the transient response analysis using time-domain unsteady aerodynamic computation of the Euler-solver module starts. Note that the number of time step of this static aeroelastic analysis is 100 as default, but that can be altered using a **MKPARAM** bulk data card with entry TRMSTEP. The pressure distribution of such a static aeroelastic analysis can be visualized using the **PLTCP** bulk data card. The flow solution of the static aeroelastic analysis can be saved/retrieved using the **STFLOW** bulk data card. As an alternative, the trim solution computed by a trim analysis can be retrieved by a **MLDTRIM** bulk data card as the initial flow condition for the time-domain GLOADS analysis. In this case, the static aeroelastic analysis will be skipped

- .If the time history of component loads or forces at structural finite element grid points is desired for output, the "SMGH" (for symmetric/asymmetric structural boundary condition) or the "MGG" (the G-set mass matrix) must be imported either by the **'ASSIGN MATRIX='** Executive Control Command or the **DMI** bulk data card.

2. All mass and length units involved in the subsequent bulk data cards referred to by **GLOADS** must be consistent with FTMUNIT and FTLUNIT, respectively. FTMUNIT and FTLUNIT are specified in the **FIXMACH**, **FIXMATM** or **FIXMDEN** bulk data card.

   Note:

   Unlike the flutter analysis where all flight conditions listed in the **FIXMACH**, **FIXMATM** or **FIXMDEN** bulk data card are included in the analysis, the transient discrete gust load analysis only computes one flight condition per analysis. Thus, only the first velocity and density pair (for **FIXMACH**), the first altitude (for **FIXMATM**), or the first velocity (for **FIXMDEN**) listed in their respective bulk data cards is included in the analysis. The rest of the values in the list are ignored.

3. The **RESTART** bulk data card allows the user to save or retrieve the entire flow solution at the last time step of the transient analysis to a file. In the event of an abnormal termination during the transient analysis, or if the user wishes to extend the physical computational time from a previous transient analysis, the entire flow solution at the last time stop saved on the file can be retrieved for a continuous transient analysis.

4. For STATES = 0, the rigid-body states of the state-space equations are defined by the structural rigid-body modes.

GRIDFRC

Direct Forces at FEM Grid Points

Description: Defines a control force at a set of structural finite element grid points.

Format and Example:

1	2	3	4	5	6	7	8	9	10
GRIDFRC	LABEL	TYPE	SISOID	GFORCE					CONT
CONT	IDGRID1	COMP1	FACTOR1	REMARK1	IDGRID2	COMP2	FACTOR2	REMARK2	CONT
CONT		…	-etc-	…					

GRIDFRC	GFORCE	SYM							+G
+G	97	3	3.0	FORCE3	-etc-				

Field	Contents
LABEL	Unique alphanumeric string up to 8 characters used to identify the control surface. (Character) (See Remark 1)
TYPE	Type of force. (Character) SYM Symmetric force ANTISYM Anti-symmetric force ASYM Asymmetric force
SISOID	Not used.
GFORCE	Character string referring to the name of a matrix that is imported by a **DMI** bulk data card or **'ASSIGN MATRIX='** Executive Control Command. This matrix contains NGSET rows and one column of force at all structural d.o.f. where NGSET = 6 × number of structural grid points. (Character or Blank)
IDGRIDi	Identification number of a structural finite element grid points that is imported from the **'ASSIGN FEM='** Executive Control Command. (Integer > 0)
COMPi	Component number either 1, 2, 3, 4, 5, or 6 representing the d.o.f. of the control force. 1, 2 and 3 represent the forces along the *x*, *y* and *z* directions, respectively. 4, 5 and 6 represent the moments about the *x*, *y* and *z* directions, respectively. (Integer > 0) (See Remark 2)
FACTORi	Multiplication factor. (Real, See Remark 3)
REMARKi	Any character string with no embedded blanks to describe the control force.

Remarks:

1. **GRIDFRC** can be selected as a control force for the TRIM or transient analysis.

2. The d.o.f. of the force or moment are defined in the output displacement coordinates of the grid in the structural finite element model (i.e., the local coordinate system for displacements of the structural finite element grid).

3. The units of forces and moments are FMMUNIT × (FMLUNIT/sec^2) and FMMUNIT × (FMLUNIT^2/sec^2), respectively, where FMMUNIT and FMLUNIT are defined in the **AEROZ** bulk data card.

INCLUDE

Insert an External File into the Bulk Data Section

Description: Inserts an external file into the Bulk Data Section. The **INCLUDE** statement may appear anywhere within the Bulk Data Section of the input deck.

Format and Example:

INCLUDE 'filename'

The following INCLUDE statement is used to obtain the Bulk Data from another file called External.dat:

```
BEGIN BULK
.
.    INCLUDE 'External.dat'
.
.
ENDDATA
```

Field	Contents
filename	Physical filename of the external file to be inserted. The user must supply the name according to installation or machine requirements. It is recommended that the filename be enclosed by a single right-hand quotation marks.

Remarks:

1. The **INCLUDE** statement does not allow continuations. The total length of the statement must be 72 characters or less.

KEXPAN

K-expansion method for Linearized unsteady aerodynamics at low reduced frequencies

Description: Activates the K-expansion method to compute the frequency domain linearized unsteady aerodynamics for use at low reduced frequencies.

Format and Example:

1	2	3	4	5	6	7	8	9	10
KEXPAN	ID	KLOW	NORDER	CFLKLOW	LVRSMOO	SAVKLOW	FILENM		CONT
CONT	NSTEP0	NSTEP1	NSTEP2	NSTEP3	NSTEP4	NSTEP5	NSTEP6	NSTEP7	CONT
CONT	NSTEP8	NSTEP9	NSTEP10	NSTEP11	NSTEP12	NSTEP13	NSTEP14	NSTEP15	

KEXPAN	200	0.02	11	3.5		SAVE	KLOW.DAT		+K
+K	300	350	400	450	500	550	600	650	+K
+K	700	750	800	850					

Field	Contents
ID	Identification number referred to by the **MKPARAL** bulk data card to activate the K-expansion method for computing the frequency domain unsteady aerodynamics force at low reduced frequencies. (Integer ≥ 0) (See Remark 1)
KLOW	A low reduced frequency below (or equal) which all the unsteady aerodynamic force at those reduced frequencies listed in the **MKAEROZ** bulk data cards are computed by the K-expansion method. (Real > 0.0, default = 0.03) (See Remark 2)
NORDER	The order of the K-expansion Euler equations. NORDER must be an odd integer and less than 15. ($3 \leq \text{Integer} < 15$, default $= 7$) (See Remark 3)
CFLKLOW	Courant-Fredeřichs-Lewy (CFL) number for the K-expansion method. If CFKLOW = 0.0, the entry CFL is the **MKPARAL** bulk data card is used. (Real ≥ 0.0, default = 0.0)

LVRSMOO	Residual smoothing for the K-expansion method LVRSMOO = 0 Turn off the residual smoothing option LVRSMOO = 1 Turn on the residual smoothing option LVRSMOO = −1 The entry LVRSMOO in the **MKPARAL** bulk data card is used (Integer, default = −1)
SAVKLOW	Character string either "SAVE" or "AQUIRE" to save the K-expansion solution to file "FILENM" or to retrieve the K-expansion solution from the "FILENM". (Characters or Blank) (See Remark 4)
FILENM	File name to specify the file name on which the K-expansion solution is saved or retrieved. If the first character of FILENM starts with a dollar sign "$", the rest of the characters must be integers. This integer is the identification number of an **EXTFILE** bulk data card where the filename is specified. This feature allows for filenames up to 72 characters to be input. (Character or Blank) (See Remark 5)
NSTEP$_i$	Number of iterations to solve the K-expansion equations. (Integer > 0) (See Remark 6)

Remarks:

1. The linearized Euler solver (activated by the entry METHOD = 3 in the **MKAEROZ** bulk data card) may encounter convergence problem in solving the linearize Euler equation at low reduced frequencies. The K-expansion method can circumvent the problem by solving a series of K-expansion equations. The linearized Euler equation reads:

$$ik\frac{Q(ik)}{L} + \frac{\partial(A_i Q(ik))}{\partial X_i} = 0$$

where k is the reduced frequency

L is the reference length $(L = \frac{REFC}{2})$

A_i is the steady background flow solved by the steady Euler solver.

The K-expansion method applies the Taylor expansion to the linearized Euler equation such as

$$Q(ik) = \sum_n (ik)^n \frac{Q_n}{L^n}$$, n = 0, 1, 2, ... NORDER

Substituting the above equations (called the K-expansion equations) into the linearized Euler equation leads to the following NORDER equations:

Order	K-expansion equation
$(ik)^0$	$\frac{\partial(A_i Q_0)}{\partial X_i} = 0$
(ik)	$ik\frac{Q_0}{L} + \frac{\partial(A_i Q_1)}{\partial X_i} = 0$

$(ik)^2$	$(ik)^2 \frac{Q_1}{L^2} + \frac{\partial(A_i Q_2)}{\partial X_i} = 0$
...	...
$(ik)^{NORDER}$	$(ik)^{NORDER} \frac{Q_{NORDER-1}}{L^{NORDER}} + \frac{\partial(A_i Q_{NORDER})}{\partial X_i} = 0$

Note that all Q_n are real numbers and can be solved by pseudo time marching scheme with $NSTEP_i$ iterations.

2. The K-expansion solution converges only if KLOW is small. Once the convergence of the K-expansion solution at reduced frequency = KLOW is verified, the frequency domain unsteady aerodynamics solutions at those reduced frequencies listed in the **MKAEROZ** bulk data card that is less than or equal to KLOW are solved by plugging the Q_n solutions into the K-expansion equations with little additional computational cost. For instance, the reduced frequencies listed in the **MKAEROZ** bulk data card are 0.0, 0.01, 0.02, 0.03, 0.1, and 0.5 and the KLOW = 0.03, the frequency-domain unsteady aerodynamic forces at reduced frequencies = 0.0, 0.01, 0.02, and 0.03 are provided by the K-expansion method. Those at reduced frequencies = 0.1 and 0.5 are solved by the linearized Euler solver.

3. Theoretically, a larger NORDER leads to a more accurate solution as long as KLOW is small. However, numerical experience shows that solving Q_n at large n may introduce numerical noise into the solution. Therefore, a larger NORDER may not give more accurate results. The best combination of KLOW and NORDER may depend on the aerodynamic mesh and structural modes.

4. It is recommended that the user should first specify a large NORDER and save those Q_n, n=0, 1, 2....NORDER solutions on file "FILENM", then specify many small reduced frequencies such as 0.0, 0.001, 0.002, ...KLOW in the **MKAEROZ** bulk data card (do not specify any reduced frequencies higher than KLOW). The K-expansion will print out the ratio of $(ik)^{NORDER} Q_{NORDER} \,/\, \sum_{n}^{NORDER} (ik)^n \frac{Q_n}{L^n}$ to show the convergence of the solution at each small reduced frequencies listed in the **MKAEROZ** bulk data card. If the ratio is large, the user must acquire the solution from the file "FILENM" and reduce NORDER. Since the Q_n solutions are independent of the reduced frequency, the solution of Q(ik) at any frequency can be computed rapidly by retrieving the Q_n from file "FILENM". This retrieving Q_n solution procedure allows the user to rapidly determine the best combination of KLOW and NORDER. If NORDER is a positive integer, the convergence ratios of the diagonal terms in the generalized aerodynamic force matrix will be printed out. If NORDER is a negative integer, the convergence ratios of the whole generalized aerodynamic force matrix will be printed out.

5. The number of Q_n solutions saved on file "FILENM" is the product of NORDER and the number of modes. However, if the iteration for solving the Q_n solution at n=M fails to converge, NORDER saved on the file "FILENM" is rest to be M – 1.

6. Numerical experience shows that solving Q_n at large n requires more iteration number. Therfore, it is recommended that $NSTEP_{i+1} > NSTEP_i$ by at least 30 iterations.

LOADMOD **Load Mode Generator**

Description: Defines the load mode of a set of structural grid points for computing component loads.

Format and Example:

1	2	3	4	5	6	7	8	9	10
LOADMOD	LID	LABEL	CP	SETK	SETG				

LOADMOD	10	XSHEAR	1	1					

Field	Contents
LID	**LOADMOD** identification number. (Integer > 0) (See Remark 1)
LABEL	Type of loads defined by the load mode. (Character) Must be one of the following: XSHEAR Shear force along X-axis of the coordinate system CP. YSHEAR Shear force along Y-axis of the coordinate system CP. ZSHEAR Shear force along Z-axis of the coordinate system CP. XMOMENT Bending moment about X-axis of the coordinate system CP. YMOMENT Bending moment about Y-axis of the coordinate system CP. ZMOMENT Bending moment about Z-axis of the coordinate system CP.
CP	Identification number of a rectangular coordinate system. (**CORD2R** bulk data card) (Integer ≥ 0) (See Remark 2)
SETK	Identification number of a **PANLST1**, **PANLST2** or **PANLST3** bulk data card used to define the aerodynamic box ids for computing aerodynamic forces. (Integer ≥ 0)
SETG	The absolute value of SETG refers to the identification number of **SET1** or **SETADD** bulk data card used to define the structural grid points for computing initial forces. (See Remark 3)

Remarks:

1. The **LOADMOD** bulk data card can be used to compute the loads (including aerodynamic loads and inertial loads) of a component, for instance the wing or an under-wing store.

2. If CP = 0, the basic coordinate system is used.

3. All structural grid points associated with the component should be included in the **SET1** or **SETADD** bulk data card. Missing structural grid points that have attached mass can lead to correct inertial loads. SETG can be 0 so that no initial force is considered. Also, SETG can be a negative integer. In this case, the sign of all structure grid ids listed in **SET1** or **SETADD** bulk data cards is reversed.

MESHPRM

Parameters for Mesh Generation

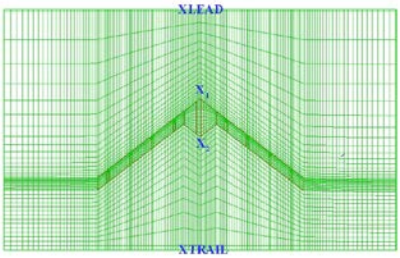
Description: Defines parameters to control the mesh size generated by the **BLOCK** bulk data card.

Format and Example:

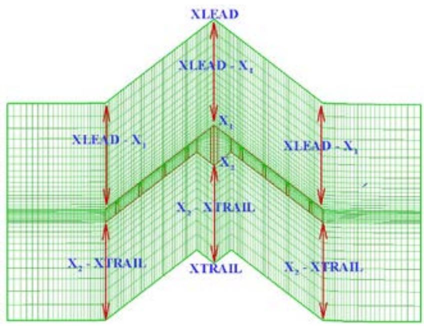
1	2	3	4	5	6	7	8	9	10
MESHPRM	ID	IDBLK	IDELET	DYFACT	DZFACT	XGROWTH	YGROWTH	ZGROWTH	CONT
CONT	XTOL	YTOL	ZTOL	ADDLINE	LSHEARL	LSHEARR			

MESHPRM	100	200		0.02	0.01	1.3	1.4	1.4	+M
+M				1	1	1			

Field	Contents
ID	Identification number. (Integer > 0) (See Remark 1)
IDBLK	Identification number of a **BLOCK/BLOCKT** bulk data card whose generated mesh size is controlled by the parameters defined in the **MESHPRM** bulk data card. (Integer > 0) (See Remark 2)
IDELET	Identification number of a **DELLINE** bulk data card to delete undesired gridlines. (Integer ≥ 0)
DYFACT	The spanwise mesh size near the surface of a vertical lifting surface or the left-hand side and right-hand side of a body is determined by DYFACT × REFC, where REFC is the reference chord defined in the **AEROZ** bulk data card. (Real > 0.0, Default = 0.01) (See Remark 3)
DZFACT	The vertical mesh size near the surface of the horizontal lifting surface or the top and bottom of a body is determined by DZFACT × REFC, whose REFC is the reference chord defined in the **AEROZ** bulk data card. (Real > 0.0, Default = 0.01) (See Remark 4)
XGROWTH	The absolute value of XGROWTH is the growth rate of the mesh along the x-axis. The sign of XGROWTH is also used to control the shape of the outer boundary projected on the x-y plane of the mesh. If XGROWTH is positive, the outer boundary on the x-y plane is rectangular and as the one shown in the following figure:

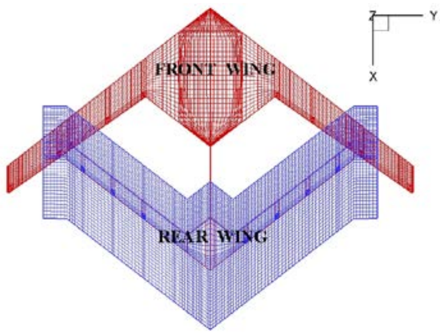


If XGROWTH is negative, the projected outer boundary on the x-y plane becomes:



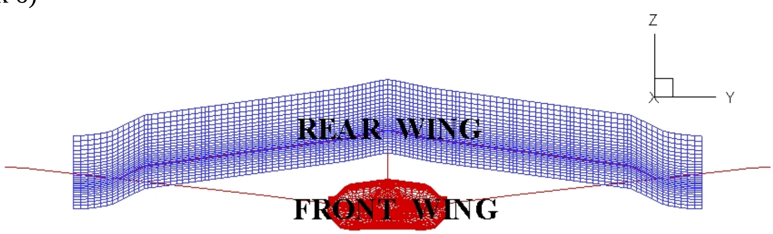
The purpose of this option is to provide the user more control of the domain of the mesh. For instance, to model a joined-wing as shown in the following figure using two-block overset mesh where the mesh of the front wing is modeled as the global mesh, it is recommended to minimize the domain of the mesh of the rear wing so that the region of the front wing surface mesh penetrated into rear wing mesh can be minimized.

This is because the cell size of the smaller block that is embedded in the global mesh near the outer boundary is normally much larger than the cell size of the surface mesh of the front wing. This could undermine the accuracy of the interpolation of the flow solution in the overlapping region due to the incompatible cell sizes. (Real > 1.0 or Real < -1.0, Default = 1.3) (See Remark 5)



YGROWTH Growth rate of the mesh along the y-axis. (Real > 1.0, Default = 1.3) (See Remark 6)

ZGROWTH The absolute value of ZGROWTH is the growth rate of the mesh along the z-axis. Similar to XGROWTH, the sign of ZGROWTH is used to control the shape of the outer boundary of the mesh projected on the y-z plane. For the join-wing example shown above, the outer boundary of the smaller mesh on the y-z plane can be the one shown as follows if ZGROWTH < 0.0. (Real > 1.0, Real < -1.0,Default=1.3) (See Remark 6)



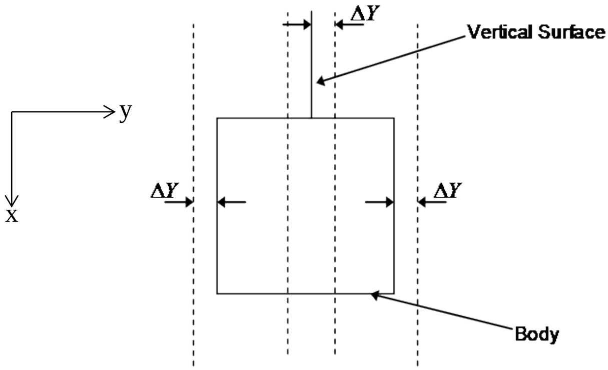
XTOL	XTOL × DXMIN serves as the small tolerance for deleting redundant gridlines in X direction, where DXMIN is the smallest distance between chordwise cuts of **CAERO7**/**BODY7** macroelements. (Real > 0.0, Default = 0.25)
YTOL	YTOL × DYMIN serves as the small tolerance for deleting redundant gridlines in Y direction, where DYMIN is the smallest distance between spanwise cuts of **CAERO7**/**BODY7** macroelements. (Real > 0.0, Default = 0.25)
ZTOL	ZTOL × DZMIN serves as the small tolerance for deleting redundant gridlines in Z direction, where DZMIN is the smallest distance between Z direction surface cuts of **CAERO7**/**BODY7** macroelements. (Real > 0.0, Default = 0.25)
ADDLINE	A flag to control the grid spacing of the chordwise Y-line ahead of the wing leading edge and behind the wing trailing edge. (Integer ≥ 0, Default = 0) (See Remark 7)
LSHEARL	If LSHEARL=1, the chordwise Y-lines emanating from the left hand side wing tip to the left hand side far field are sheared according to the leading and trailing edge sweep angle. If LSHEARL=0, those lines are perpendicular to the left hand side wing tip. For a lifting surface with sweptback leading edge and swept forward trailing edge, the chordwise spacing between sheared chordwise Y-lines could be very small when they are approaching far field. For this case, LSHEARL=0 is recommended to avoid convergence issues in the flow solver caused by those small spacing. (Integer, Default = 1)
LSHEARR	Same as LSHEARL except for the right hand side wing tip. (Integer, Default = 1)

Remarks:

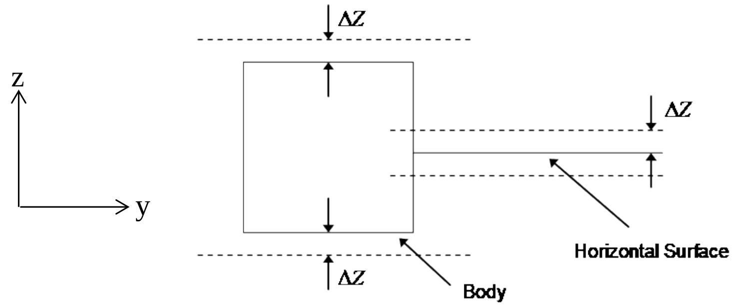
1. ID is used only for error message output.

2. If IDBLK matches the identification number of a **BLOCK** bulk data card, the parameters defined by the **MESHPRM** bulk data card are used to control the mesh size generated by this **BLOCK** bulk data card. Otherwise, the default values are used.

3. The following figure shows a vertical lifting surface and a body projected on the X-Y plane in the local coordinates defined by the **BLOCK** bulk data card (body is always approximated by a rectangular box if it is referred to by a **BLOCK** bulk data card). The mesh size (ΔY) near the surface is determined by DYFACT × REFC. The determination of the mesh size near the body surface is actually more complicated than presented here, and the details can be found in Section 5.4.2 of Chapter 5.



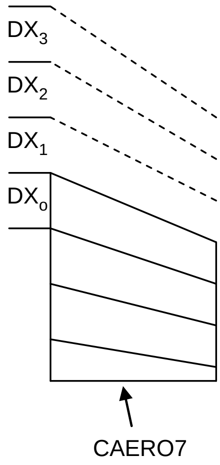
4. The following figure shows a horizontal lifting surface and a body projected on the Y-Z plane. The mesh size (ΔZ) near the surface is determined by DZFACT × REFC.



5. XGROWTH is used to compute the mesh size along X-axis by

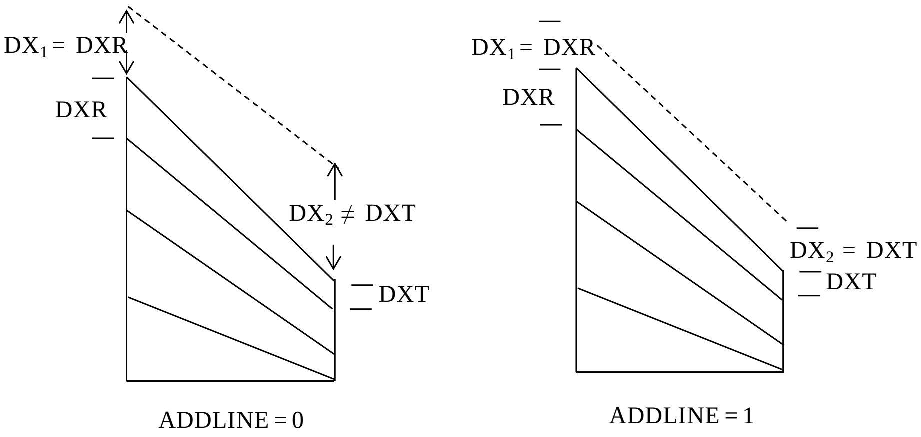
$$DX_{i+1} = DX_i * XGROWTH$$

In the following figure, DX_1 is the mesh size immediately ahead of the leading edge of a wing. Note that DX_1 is equal to DX_o that is the chordwise surface mesh size at the wing leading edge. Then $DX_2 = DX_1 \times XGROWTH$, $DX_3 = DX_1 \times (XGROWTH)^2$ and so on.

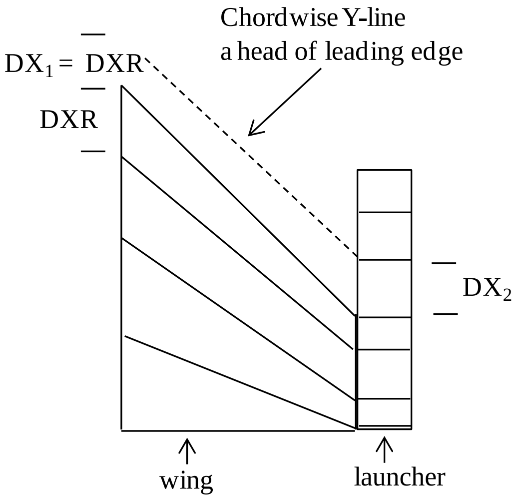


6. Similar to XGROWTH, YGROWTH and ZGROWTH are used to compute the mesh size along the y-axis and z-axis, respectively. Note the starting mesh sizes, DY1 and DZ1 are determined by DYFACT × REFC and DZFACT × REFC, respectively, described in Remarks 3 and 4.

7. The following figures show the chordwise Y-lines immediately ahead of the wing leading edge whose grid spacing can be control by ADDLINE. For ADDLINE = 0, $DX_2 \neq DXT$ may occur. For ADDLINE = 1, $DX_2 = DXT$ is ensured.



Note that ADDLINE is active only for the clean wing configuration. The following shows a wing with tip launcher configuration. In this case, DX2 is controlled by the chordwise divisions of the surface mesh of the launcher at the wing-launcher junction.


Chordwise Y-line
a head of leading edge
$DX_1 = \overline{DXR}$
$\overline{DXR}$
DX_2
wing
launcher

MESHSAV — Save or Retrieve Mesh

Description: Save or retrieve the flowfield mesh.

Format and Example:

1	2	3	4	5	6	7	8	9	10
MESHSAV	SAVE	FILENM							
MESHSAV	ACQU	MESH.DAT							

Field	Contents
SAVE	Character string either "SAVE" or "ACQUIRE". For SAVE = "SAVE", save the flowfield mesh. For SAVE = "ACQUIRE", retrieve the flowfield mesh. (Character) (See Remark 1)
FILENM	File name to specify the file name on which the flowfield mesh is saved or retrieved. If the first character; FILENM starts with a dollar sign "$", the rest of the characters must be integers. This integer is the identification number of an **EXTFILE** bulk data card where the filename is specified. This feature allows for filenames up to 56 characters to be input. (Character)

Remarks:

1. The **MESHSAV** bulk data card is not referred to by any other bulk data card. Its existence "triggers" the program to save/retrieve the flowfield mesh. For complex configuration, the generation of the flowfield mesh can be computational intensive. The user can save the flowfield mesh in a cold start run and retrieve the flowfield mesh in a restart run to save computational time.

MKAEROZ

Defines the Flight Condition

Description: Defines a Mach number, mean flow conditions, and a list of reduced frequencies.

Format and Example:

1	2	3	4	5	6	7	8	9	10
MKAEROZ	IDMK	MACH	IDPARAM	IDFLT	SAVE	FILENM		PRINT	CONT
CONT	FREQ1	FREQ2	FREQ3	FREQ4	-etc-				

MKAEROZ	100	0.9	3	2	SAVE	AERODATA		-3	+BC
+BC	0.001	0.1	0.3	-2.0					

Field	Contents
IDMK	Unique identification number. (Integer > 0)
MACH	Mach number. (Real ≥ 0.0)
IDPARAM	Identification number of **MKPARAM** bulk data card to define the parameters used in the Euler solver module. If IDPARAM = 0, the default values described in the **MKPARAM** bulk data card are used. (Integer ≥ 0, Default = 0)
IDFLT	Identification number of **TRIMFLT** bulk data card to define the mean flow conditions. (Integer ≥ 0, Default = 0)

SAVE Save the Generalized Aerodynamic Forces (GAF) data generated by the current **MKAEROZ** bulk data card to file "FILENM" or retrieve it from "FILENM". (Characters or Blank)

SAVE= "SAVE" saves the GAF data.(See Remark 1)
SAVE= "ACQUIRE" retrieves an existing file containing GAF data. (See Remark 2)
SAVE = "ADD" retrieves an existing file containing GAF data and adds new reduced frequencies to it.
SAVE = blank do not save nor retrieve the GAF data.

SAVE="PHIOLD" PHIOLD refers to the name of a direct matrix input either by the "ASSIGN MATRIX=" executive control command or **DMI** bulk data card that contains the mode shape matrix from which the saved GAF matrices were generated. (See remark 3.)

If SAVE is a character string that is not "SAVE", "ACQUIRE", "ADD" nor blank, the character string is the name of a direct matrix input that contains the mode shape matrix. This mode shape matrix was used to compute the saved GAF. Based on the saved GAF and its corresponding mode shape matrix, a new GAF is computed by a least square approach. (see Remark 3)

Note that this entry is used only when the **MKAEROZ** bulk data card is referred to by the **FLUTTER**, **FLTFAST**, **FLPRAM**, or **GENGAF** bulk data card by which the frequency-domain GAF is required.

FILENM File name to specify the file name on which the GAF data is saved or retrieved. If the first character of FILENM is a dollar sign "$", the rest of the characters must be integers. This integer is the identification number of an **EXTFILE** bulk data card where the filename is specified. This feature allows for filenames up to 56 characters to be input. (Character or Blank).

PRINT Print flag. (Integer, Default = 0)
PRINT = 0 No print.
|PRINT| ≥ 1 Prints out the steady aerodynamic pressure coefficients.
|PRINT| ≥ 2 Prints out the unsteady aerodynamic pressure coefficients.

FREQ*i* The absolute value of FREQ*i* represents the reduced frequency k which is defined as:

$$k = \frac{\omega\left(\frac{\text{REFC}}{2}\right)}{V_\infty}$$

where REFC is the reference chord length defined in the **AEROZ** bulk data card. Note that FREQ*i* is used only for the frequency-domain flutter analysis, i.e. if the **MKAEROZ** bulk data card is referred to by a **FLUTTER**, **FLTFAST**, or **GENGAF** bulk data card. In addition, FREQ*i* can be a negative number. In this case, the number of iterations for the linearized Euler solver is doubled to ensure the convergence of the unsteady aerodynamic solution for this particular reduced frequency.(Real) (see Remark 4)

Remarks:

1. GAF is created only if the **MKAEROZ** bulk data card is referred to by the **FLUTTER**, **FLTFAST**, or **GENGAF** bulk data card. For the transient response analysis, SAVE is not used because there is no GAF created.

   Note that the entry MLIST in the **FLUTTER**, **FLTFAST**, and **GENGAF** can be used to remove modes from the GAF computation. Because the computational time for generating GAF is proportional to the number of modes, reducing modes can save computational time. However, if there are NM number of modes being imported by the "ASSIGN FEM=" Executive Control Command, the resulting saved GAF matrices always have the size of NM. If the entry MLIST is used to remove modes, the columns/rows associated with those removed modes in the GAF matrices contain all zeros.

2. For frequency-domain flutter analysis, acquiring the saved GAF matrices skips the running of the Euler solver module for the GAF generation. Therefore, the flutter analysis becomes very fast.

3. If SAVE refers to the name of a mode shape matrix, the saved GAF must be generated by the GENGAF module. The GENGAF module computes the GAF based on a set of mode shape matrix defined herein as $\left[\phi_{old}\right]$ and outputs the following matrices; $QHHS_{iijj}$ such as QHHS0101.DAT, QHHS0102.DAT…etc and $AJHS_{iijj}$ such as AJHS0101.DAT, AJHS0102.DAT…etc. where $QHHS_{iijj}$ is the GAF matrix of the j^{th} reduced frequency and $AJHS_{iijj}$ is the unsteady pressure distribution on each surface panel due to each mode in $\left[\phi_{old}\right]$. Note that during the computation of the saved GAF, $\left[\phi_{old}\right]$ can be output by an **OUTPUT4** bulk data card such as OUTPUT4 SPHI PHIOLD.DAT.

   If the user wishes to perform a frequency- domain aeroelastic analysis but based on a modified structural mode shape matrix defined herein as $\left[\phi_{new}\right]$ and without recomputing the GAF, the user must import the GAF for the current analysis by the following Executive Control Command,

   ASSIGN MATRIX = ‘QHHS0101.DAT, MNAME = QHHS0101
   ⋮
   ASSIGN MATRIX = ‘QHHS010*n*.DAT, MNAME = QHHS010*n*

   ASSIGN MATRIX = ‘AJHS0101.DAT, MNAME = AJHS0101
   ⋮
   ASSIGN MATRIX = ‘AJHS010*n*.DAT, MNAME = AJHS010*n*

   where n is the number of reduced frequencies

   and import $\left[\phi_{old}\right]$ by

   ASSIGN MATRIX = ‘PHIOLD.DAT’, MNAME = PHIOLD

   For the current analysis, if SAVE = ‘PHIOLD’, the program computes the new GAF by the following least square approach:

$$\left[QHHS_{iijj}\right]=\left[\phi_{new}\right]^T\left[SKJ\right]^T\left[AJHS_{iijj}\right]\left[\left[\phi_{old}\right]^T\left[\phi_{old}\right]\right]^{-1}\left[\phi_{old}\right]^T\left[\phi_{new}\right]$$

where $\left[SKJ\right]$ is a integration matrix to convert pressure to force on each surface panel.

4. The GAF matrices can be either generated by the Euler solver using the Sinusoidal or Composite Sinusoidal technique or by the linearized Euler solver (by setting the entry METHOD=3 in the **MKPARAM** bulk data card). It is recommended that small reduced frequencies (for example, k<0.05) be avoided in the list of reduced frequencies. This is because small reduced frequency corresponds to a long period in the time domain which requires many time steps to achieve a converged solution. However, at k=0, the Euler equation becomes a steady equation which does not have such a convergence issue. Therefore, a zero reduced frequency should be always included in the list of reduced frequencies.

MKBLOCK — Assign Different Parameters for Euler Solver In Different Blocks of Mesh

Description: Assign different parameters (CFL, VIS2, and/or VIS4) to solve the Euler equation for different blocks of mesh.

Format and Example:

1	2	3	4	5	6	7	8	9	10
MKBLOCK	IDMKBLK	GLOBAL							CONT
CONT	$IDBLK_1$	$VALUE_1$	$IDBLK_2$	$VALUE_2$	...	etc	...		etc

MKBLOCK	100	7.0							+M
+M	2001	6.5	2100	4.5	3100	3.5			

Field	Contents
IDMKBLK	Identification number that is referred to by the entries CFL, VIS2 and/or VIS4 of the **MKPARAM** and **MKPARAL** bulk data cards (Integer > 0)(See Remark 1)
GLOBAL	A global value assigned to those blocks of mesh that are not listed in $IDBLK_i$ (Real ≥ 0.0)
$IDBLK_i$	Identification number of a **BLOCK/BLOCKT/BLOCK1** bulk data card for which the CFL/VIS2/VIS4 value for solving the Euler equation is assigned by the entry $VALUE_i$ (Integer > 0) (See Remark 2)
$VALUE_i$	The value of CFL/VIS2/VIS4 used for the block of mesh with ID = $IDBLK_i$ (Real > 0.0)

Remarks:

1. The **MKBLOCK** bulk data card allows the different CFL/VIS2/VIS4 values that are referred to by the **MKPARAM** and **MKPARAL** bulk data card being used by different blocks of mesh. Please see the description of the **MKPARAM** and **MKPARAL** bulk data cards.

2. If the block of mesh being generated by the **BLOCK/BLOCKT/BLOCK1** bulk data card is referred to by $IDBLK_i$, the CFL/VIS2/VIS4 value is assigned by the entry $VALUE_i$. Otherwise, the value is assigned by the entry GLOBAL

MKPARAL

Parameters for the Linearized Euler-Solver Module

Description: Alter the default values of parameters used in the Linearized Euler-Solver module.

Format and Example:

1	2	3	4	5	6	7	8	9	10
MKPARAL	IDPARAM	KEXPAN	TRMSTEP	FLTSTEP	NEWTN	NCYC	LVRSMOO	PRNTCOV	CONT
CONT	CFL	GAMMA	VIS2	VIS4	EPSC	TVDCOEF			

MKPARAL	100	200	100		4	2	1	2	+BC
+BC	7.5				0.001				

Field	Contents
IDPARAM	Unique identification number. **MKPARAL** is referred to by the **MKAEROZ** bulk data card. The **MKPARAL** bulk data card can be activated only if the entry METHOD = 3 is specified in the **MKPARAM** bulk data card. (Integer > 0) (See Remark 1)
KEXPAN	Identification number of a **KEXPAN** bulk data card to activate the K-expansion method for resolving the inaccurate unsteady aerodynamic problem computed by the linearized Euler solver at low reduced frequencies (Integer ≥ 0) (See Remark 2).
TRMSTEP	Number of time steps to compute the Linearized Euler solution. (Integer ≥ 0, Default = 100) (See Remark 3)
FLTSTEP	Not used.
NEWTN	The number of Newton sub-iterations per time step. (Integer, Default = 4) (See Remark 3)
NCYC	The number of Euler cycles per Newton sub-iteration. (Integer, Default = 2) (See Remark 3)
LVRSMOO	Residual smoothing for the Euler solver. (Integer, Default = 1) (See Remark 4) LVRSMOO = 0 Turn off the residual smoothing option. LVRSMOO = 1 Turn on the residual smoothing option.
PRNTCOV	CFD convergence history print flag. (Integer, Default = 2) PRNTCOV = 0 No print. PRNTCOV = 1 Print out the convergence history of the last EULER cycle of the last sub-iteration. PRNTCOV = 2 Print out the full convergence history.
CFL	Courant-Friedrichs-Lewy (CFL) number for the EULER solver. (Real>0.0, or Integer > 0 Default = 7.0) (See Remark 4)
GAMMA	Specific heat ratio of the fluid. (Real, Default = 1.4) .

VIS2	Second order artificial dissipation parameter for the EULER solver. (Real > 0.0, or Integer > 0, Default = 1.0)
VIS4	Fourth order artificial dissipation parameter for the EULER solver. (Real > 0.0, or Integer > 0, Default = 1.0).
EPSC	Convergence control for the linearized Euler run. (Real>0.0, Default = 0.0001) (See Remark 5)
TVDCOEF	A parameter involved in the TVD form of pressure sensor of Jameson-Schmidt-Turkel (JST) scheme. It can be used to smooth the undesired zigzag pressure distribution that could occur on lower surface of the wing at high angle of attack flow condition (0.0 ≤ Real < 1.0, Default = 0.0)

<u>Remarks:</u>

1. **MKPARAL** bulk data card provides an option for the Linearized Euler solver to use a different set of flow solver parameters other than those used in the full Euler run. The Linearized Euler solution is based on the steady flow solution obtained from a full Euler run, and thus IDPARAM is referred to by both **MKAEROZ** and **MKPARAM** bulk data card if Linearized Euler option is turned on by setting METHOD=3 in the **MKPARAM** bulk data card (see the chart below). **MKPARAL** bulk data card has almost the same format as **MKPARAM** bulk data card except that it has 6 additional parameters to possibly define a smaller computational domain. There are also some parameters originated from **MKPARAM** but not used in **MKPARAL** bulk data card, which means the same values of those parameters specified in **MKPARAM** bulk data card will be applied for the Linearized Euler run. If there is no **MKPARAL** bulk data card with the proper identification number existed, then the values of all the parameters specified in **MKPARAM** bulk data card will be in use for the Linearized Euler.

2. To ensure the convergence of the linearized Euler solution for each reduced frequency and each mode, the user must check the convergence history computed by the pseudo time marching scheme. If convergence is not achieved, the entry TRMSTEP must be increased. However, for some modes at low reduced frequencies, convergence cannot be achieved even if TRMSTEP is large. This indicates that the linearized Euler solver cannot give accurate unsteady aerodynamic solution at this reduced frequency that normally is low. This convergence problem can be circumvented by activating the K-expansion method.

3. There is no physical time step involved in Linearized Euler runs, and thus TRMSTEP is actually not necessary and can be simply set to 1. If there is only one block of mesh, then the value of NEWTN doesn't matter either, which means only the final product of TRMSTEP × NEWTN × NCYC counts. If overset mesh is involved, the value of NEWTN matters as overset interpolation is performed at the end of each Newton sub-iteration.

4. Definitions of LVRSMOO and CFL are the same as in **MKPARAM** bulk data card.

If CFL is an integer, this integer is the identification number of the **MKBLOCK** bulk data card that allows the different CFL numbers being used by the different blocks of mesh.

5. At the end of each time step, for each BLOCK, the ratio of the averaged residual over that at the last iteration of the first time step is compared to EPSC. If the ratios are less than EPSC for all the blocks, the convergence criteria are met, and the computation will stop even before TRMSTEP number of time steps are finished. The time history of the LOG10 of the residual ratio for each block will be plotted.

MKPARAM — Parameters for the Euler-Solver Module

Description: Alter the default values of parameters used in the Euler-Solver module.

Format and Example:

1	2	3	4	5	6	7	8	9	10
MKPARAM	IDPARAM	METHOD	TRMSTEP	FLTSTEP	NEWTN	NCYC	LVRSMOO	PRNTCOV	CONT
CONT	CFL	GAMMA	VIS2	VIS4		TVDCOEF			

MKPARAM	100	0	100		4	2	1	2	+BC
+BC	7.5	1.4	1.2	1.0					

Field	Contents
IDPARAM	Unique identification number. **MKPARAM** is referred to by the **MKAEROZ** bulk data card. (Integer > 0)
METHOD	Flow solver augmentation flag. (Integer, Default = 0) METHOD = 0 Normal run without any special options. METHOD = 1 Turn on preconditioning for low MACH number cases. If the Mach number specified in the **MKAEROZ** bulk data card is less than 0.2, turning on preconditioning is highly recommended. METHOD = 3 Turn on the linearized Euler-solver option for frequency-domain flutter analysis, or trim analysis. In this case, the **MKPARAL** bulk data card can be used to alter the default values of the parameter used in the linearized Euler solver. (See Remark 1)
TRMSTEP	Number of time steps to compute the steady, static aeroelastic solution or linearized Euler solution. Note that the number of time steps for transient response analysis is defined in the **MLDTIME** bulk data card. (Integer > 0, Default = 100)
FLTSTEP	Number of time steps to compute the frequency-domain generalized aerodynamic forces using the sinsouidal excitation technique (used by the **FLUTTER**) or composite sinusoidal excitation technique (used by the **FLTFASTR**) for flutter analysis. For the linearized Euler solver, FLTSTEP is not used. (Integer > 0, Default = 201)
NEWTN	The number of Newton sub-iterations per time step. NEWTN is a important parameter for the flow convergence of overset mesh and boundary layer coupling. (Integer, Default = 4) (See Remark 2)
NCYC	The number of Euler cycles per Newton sub-iteration. (Integer, Default = 2) (See Remark 3)
LVRSMOO	Residual smoothing for the Euler solver. (Integer, Default = 1) (See Remark 4) LVRSMOO = 0 Turn off the residual smoothing option. LVRSMOO = 1 Turn on the residual smoothing option.

PRNTCOV	CFD convergence history print flag. (Integer, Default = 2) PRNTCOV = 0 No print. PRNTCOV = 1 Print out the convergence history of the last EULER cycle of the last sub-iteration. PRNTCOV = 2 Print out the full convergence history.
CFL	Courant-Friedrichs-Lewy (CFL) number for the EULER solver. (Real > 0.0, or Integer > 0, Default = 7.0) (See Remark 5)
GAMMA	Specific heat ratio of the fluid. (Real, Default = 1.4)
VIS2	Second order artificial dissipation parameter for the EULER solver. (Real > 0.0, or Integer > 0, Default = 1.0) (See Remark 6)
VIS4	Fourth order artificial dissipation parameter for the EULER solver. (Real > 0.0, or Integer > 0, Default = 1.0)
TVDCOEF	A parameter involved in the TVD form of pressure sensor of Jameson-Schmidt-Turkel (JST) scheme. It can be used to smooth the undesired zigzag pressure distribution that could occur on lower surface of the wing at high angle of attack flow condition (0.0 ≤ Real < 1.0, Default = 0.0) (See Remark 7)

Remarks

1. The linearuized Euler solver solves the frequency-domain linearized Euler equation that is derived from the full-order Euler equation:

$$\frac{\partial Q}{\partial t}+\frac{\partial H_1}{\partial \xi}+\frac{\partial H_2}{\partial \eta}+\frac{\partial H_3}{\partial \varsigma}=0$$

where Q is the vector of conservative fluid variables and H_1, H_2, and H_3 are the convective fluxes in three curvilinear coordinate directions.
Assuming the structural oscillating amplitude to be very small, the total unsteady flow can be decomposed into a steady part and an unsteady small perturbation part.

$$Q=\bar{Q}+\tilde{Q}(t),\quad H_1=\bar{H}_1+\tilde{H}_1(t),\quad H_2=\bar{H}_2+\tilde{H}_2(t),\quad H_3=\bar{H}_3+\tilde{H}_3(t)$$

This leads to two equations:

(1) The steady Euler Equation: $\dfrac{\partial \bar{H}_1}{\partial \xi}+\dfrac{\partial \bar{H}_2}{\partial \eta}+\dfrac{\partial \bar{H}_3}{\partial \varsigma}=0$

(2) The linearized Euler equation: $\dfrac{\partial\left(\tilde{Q}(t)\right)}{\partial t}+\dfrac{\partial\left(\bar{A}_\xi\tilde{Q}(t)\right)}{\partial \xi}+\dfrac{\partial\left(\bar{A}_\eta\tilde{Q}(t)\right)}{\partial \eta}+\dfrac{\partial\left(\bar{A}_\varsigma\tilde{Q}(t)\right)}{\partial \varsigma}=0$

Applying Fourier transform to the linearized Euler equation yeilds the frequency-domain linearized Euler equation:

$$(i\omega)\tilde{Q}(i\omega)+\frac{\partial\left(\bar{A}_\xi\tilde{Q}(i\omega)\right)}{\partial \xi}+\frac{\partial\left(\bar{A}_\eta\tilde{Q}(i\omega)\right)}{\partial \eta}+\frac{\partial\left(\bar{A}_\varsigma\tilde{Q}(i\omega)\right)}{\partial \varsigma}=0$$

where $\bar{A}_\xi$, $\bar{A}_\eta$ and $\bar{A}_\varsigma$ are the mean flow convective flux Jacobians (called the steady background flow) and can be provided by solving the steady Euler equation whose steady flow solution can be saved using the **STFLOW** bulk data card.

The linearized Euler solver solves the frequency-domain unsteady aerodynamics for each reduced frequency-mode (k-mode) pair at a time using the psedo time-domain computation scheme, and the computation of each k-mode pair is independent of other k-mode pairs. This is an ideal case to use Message Passing Interface (MPI) for parallelization. Such a parallelized computation for the linearized Euler solver can be activated by specifying the **MPICPU** executive control command. However, the **MPICPU** executive control command is only available for the MPI version of ZEUS.

Because of the linearized unsteady aerodynamic forces, for a half-span aerodynamic model the linearized Euler solver can impose symmetric or anti-symmetric boundary condition on the x-z plane to generate unsteady aerodynamic forces due to symmetric or anti-symmetric modes, respectively. This can be achieved by matching the entry BOUNDARY in the "**ASSIGN FEM**=" executive control command with the the entry SYM in the **FLUTTER** bulk data card. Both entries can be SYMM for symmetric modes, ANTI for anti-symmetric modes, and ASYM for a full-span aerodynamic model and a full-span structural model.

2. For cases using the overset mesh, different blocks will communicate with each other at the end of each Newton sub-iteration, and the boundary-layer coupling is also performed per Newton sub-iteration. So for cases using overset grid and/or boundary-layer coupling, the parameter NEWTN is very critical for achieving converged solutions. At least four Newton sub-iterations per time step are recommended for such cases.

3. Within each time step, the Euler-solver module performs pseudo-time marching by Runge-Kutta scheme for each block of mesh. Each 5-stage Runge-Kutta cycle is called an Euler cycle (NCYC). Therefore, the number of (NEWTN × NCYC) Euler cycles are performed at every time step. It should be noted that for a model with only one block and without boundary layer coupling, the convergence of the flow solution within each time step depends only on the product of NEWTN and NCYC. For instance, NEWTN = 2 and NCYC = 4 is equivalent to NEWTN = 1 and NCYC = 8 in terms of flow solution.

4. An implicit variable-coefficient residual smoothing scheme is incorporated in the Euler-solver module that can extend the stability range of the Euler solver scheme so that a CFL number as large as 7.0 could be used. The residual smoothing option is always recommended unless the surface mesh has triangular panels in which case ZEUS will turn off residual smoothing and reduce the CFL number accordingly inside the code. Without residual smoothing, the largest CFL number attainable is generally less than 4.0, and thus ZEUS automatically reduce CFL number to 4.0 if residual smoothing is found to be turned off and CFL number is larger than 4.0.

5. In general, larger CFL number results in faster solution convergence. Therefore, larger CFL number is always desired except for some extreme cases such as high supersonic and /or high angle of attack flow conditions where large CFL number could lead to solution divergence.
   If CFL is an integer, this integer is the identification number of the **MKBLOCK** bulk data card that allows the different CFL numbers being used by the different blocks of mesh.

6. Larger VIS2/VIS4 adds more artificial dissipation to the Euler solver scheme that provides more stability to the solver. However, this could smear out the discontinuity of the flow solution across the shock if too much artificial dissipation is added. The default values of VIS2 and VIS4 are the optimal values and work the best for most cases except for extreme conditions such as high supersonic flow at high angle of attack where solution diverges and gradually increase of VIS2/VIS4 is recommended for a remedy try.

If VIS2/VIS4 is an integer, this integer is the identification number of the **MKBLOCK** bulk data card that allows the different VIS2/VIS4 numbers being used by the different blocks of mesh.

7. The pressure sensor switch function used in the JST scheme is defined as:

$$v_j = \left| \frac{p_{j-1} - 2p_j + p_{j+1}}{p_{j-1} + 2p_j + p_{j+1}} \right|$$

Using the TVD concept, an alternative for the pressure switch can be:

$$v_j = \frac{\left| p_{j+1} - 2p_j + p_{j-1} \right|}{(1-\omega)\rho + \omega\rho_{TVD}}, \ \rho_{TVD} = \left| p_{j+1} - p_j \right| + \left| p_j - p_{j-1} \right|, \ \rho = p_{j+1} + 2p_j + p_{j-1}$$

where ω is called TVDCOEF, and if ω =0.0, the original JST pressure switch is recovered.

Note: that using a large ω could potentially smear out the pressure jump across the shock. For cases involving transonic shock, TVDCOEF < 0.8 is recommended.

MLDCOMD

Pilot Input Commands for Transient Maneuver Load Analysis

Description: Defines the time history of pilot input commands.

Format and Example:

1	2	3	4	5	6	7	8	9	10
MLDCOMD	SETID								CONT
CONT	EXTINP1	IDTAB1	EXTINP2	IDTAB2	…	-etc-			

MLDCOMD	10								CONT
CONT	101	10	102	20					

Field	Contents
SETID	Unique identification number. (Integer > 0) (See Remark 1)
EXTINPi	Identification number of a **EXTINP** bulk data card where the pilot input command is applied. (Integer > 0)
IDTABi	Identification number of a **TABLED1** bulk data card to specify the time history of the pilot input commands. (Integer > 0)

Remarks:

1. The **MLDCOMD** bulk data card is referred to by the COMMAND entry of the **MLOADS or NLFLTR** bulk data card.

MLDPRNT

Computes and Prints out the Parameters of Interest in the Transient Response Analysis

Description: Defines an ASCII file to store the time histories or Power Spectral Density (PSD) of parameters due to the transient response loads. This ASCII file can be used to generate x-y plots. The damping and frequency of the transient response can be estimated using the ZARMA code.

Format and Example:

1	2	3	4	5	6	7	8	9	10
MLDPRNT	IDPRNT	FILENM		FORM	PSD	TSPNT	TEPNT	SOF	CONT
CONT	LABEL1	IKEY1	LABEL2	IKEY2	…	-etc-			

MLDPRNT	10	HA144MLD.PLT		TABLE				YES	+M1
+M1	STATE	X	MODALX	1	GRIDXT3	100			

Field	Contents
IDPRNT	Identification number (Integer > 0) (See Remark 1)
FILENM	The name of the ASCII file. This file name is always written in uppercase. If the input characters are entered in lowercase, the program converts them to uppercase. Note that FILENM can be blank. In this case, the output data is printed in the standard output file. If the first character of FILENM is a dollar sign "$", the rest of the characters must be integers. This integer is the identification number of an **EXTFILE** bulk data card where the filename is specified. This feature allows for filenames up to 56 characters to be input. (Character)
FORM	FORM = "TABLE" for general x-y tabular output FORM = "IDEAS" for generating an I-DEAS universal file FORM = "FEMAP" for generating a FEMAP neutral file FORM = "ESA" for generating a PEGSUS readable file Note: FORM is not used if FILENM is blank. (Character, Default = "TABLE")
PSD	Character string, either "TIME" or "PSD". PSD = "TIME" for time domain response PSD = "PSD" for power spectral density vs. frequency (Character, Default = TIME) (See Remark 2)
TSPNT	Starting time for printout. (Real or Blank) (Default = TSTART in the **MLDTIME** bulk data card)
TEPNT	Ending time for printout. (Real or Blank) (Default = TEND in the **MLDTIME** bulk data card)
SOF	Character strings either "NO" or "YES". For SOF = "NO", the forces/moments at the structural grid points (LABELi = FORCET1, FORCET2, etc…) and the

	component loads (LABELi = LOADMOD) are computed using the Mode Displacement (MD) method. For SOF = "YES", the mode acceleration method (also called the Summation Of Forces) is used. (Character, Default = "YES") (See Remark 3)
LABELi, IKEYi	LABELi is a character string and IKEYi can be either character string or integer (depending on LABELi). LABELi and IKEYi jointly define the parameter due to the transient maneuver loads. (See Remark 4)

Remarks:

1. The **MLDPRNT** bulk data card is referred to by various transient response analyses to output the time histories of the parameter of interest.

2. If PSD = "PSD", the frequency range is from 0.0 to f_{max} where $f_{max} = \frac{1}{2\,\Delta T}$ Hz and $\Delta T = DT * OUTDT$. DT and OUTDT are defined in the **MLDTIME** bulk data card.

   Note : Multiple **MLDPRNT** bulk data cards with the same IDPRNT entry can co-exist in the bulk data input. The following example gives two output ASCII files AAA.DAT and BBB.DAT. AAA.DAT contains the time domain response of the airframe state "ALPHA" and modal displacement of mode 3 in the x-y tabular format, whereas BBB.DAT contains the frequency domain PSD of the airframe state "ALPHA" and modal displacement of mode 3 in the I-DEAS universal file.

MLDPRNT	100	AAA.DAT		TABLE	TIME			YES	+M
+M	STATE	ALPHA	MODALX	3					

MLDPRNT	100	BBB.DAT		IDEAS	PSD			NO	+M
+M	STATE	ALPHA	MODALX	3					

3. In general, the equation of motion of the transient response problem reads:

$$[K]\{x\} = -[M]\{\ddot{x}\} - [C]\{\dot{x}\} + F_a(t) + F_e(t)$$

   where [K], [M], and [C] are the structural stiffness, mass and damping matrices, respectively.

   $F_a(t)$ is the aerodynamic forces including aerodynamic forces due to structural deformation, control surface deflections commanded by the control system, and the trim forces.

   $F_e(t)$ is the external excitation forces including ejection forces, pilot input command, gust, and gravity. The mode displacement method computes the forces at structural grid points {F} simply by:

$$\{F\} = [K]\{x\}$$

and the mode acceleration method by the summation of all force terms (Summation Of Forces) on the right-hand side of the equations of motion:

$$\{F\} = -[M]\{\ddot{x}\} + F_a(t) + F_e(t)$$

It is generally believed that the mode acceleration method is more accurate than the mode displacement method. However, the drawback of the mode acceleration method is that the forces due to structural damping $[C]\{\dot{x}\}$ cannot easily be included because the damping matrix [C] (defined in the G-set d.o.f.) is normally not available.

4. LABELi must match one of the character strings listed in the following table.

Label	**Description**	**Units**
"STATE"	For the airframe state. IKEY is a character string that must match one of the airframe states listed in the **MLDSTAT** bulk data card.	See the definition of units of airframe states described in the **MLDSTAT** bulk data card.
"DSTATE"	For *d(airframe state)/dt* that is the rate of the airframe state. IKEY is a character string that must match one of the airframe states listed in the **MLDSTAT** bulk data card.	Units of the airframe state per second.
"MODALX"	For the generalized modal coordinates of the structural modes. IKEY is an integer that is the index of the modes (including the rigid body modes). Note that the structural modes are those defined in the NMODE and MLIST entries of the **FLUTTER** bulk data card.	None
"MODALV"	Same as "MODALX", but for *d(MODALX)/dt*.	Per second
"MODALG"	Same as "MODALX", but for $d^2(MODALX)/dt^2$.	Per second^2
"MODALXVG"	To store the modal displacement, velocity, and acceleration in the OUTPUT4 format on an external file. IKEY is an integer that is the identification number of an **EXTFILE** bulk data card, which defines the name of the external file. The matrix stored on this external file in the OUTPUT4 format contains the following data: $\left[\{\xi\}_1 \{\dot{\xi}\}_1 \{\ddot{\xi}\}_1 \{\xi\}_2 \{\dot{\xi}\}_2 \{\ddot{\xi}\}_2 \cdots \{\xi\}_N \{\dot{\xi}\}_N \{\ddot{\xi}\}_N\right]$ where $\{\xi\}$ is the modal displacement solution of the transient response analysis which has length of number of modes.	N/A
"APPEND"	Same as "MODLAXVG" but appends the matrix to those generated by the previous subcases (if any). IKEY is an integer that is the identification number of an **EXTFILE** bulk data card, which defines the name of the external file. If the matrices computed by all previous subcases are already stored in this external file, these matrices are not overwritten. Instead, the	N/A

	matrix of the current case is to be appended to the existing matrices.	
"GRIDXT1"	For the displacement along the x direction at a structural finite element grid point. IKEY is an integer that is the identification of the grid point.	The length units defined in the FMLUNIT entry of the **AEROZ** bulk data card.
"GRIDXT2"	Same as "GRIDXT1", but along the y direction.	FMLUNIT
"GRIDXT3"	Same as "GRIDXT1", but along the z direction.	FMLUNIT
"GRIDXR1"	Same as "GRIDXT1", but for the rotation degree of freedom about the x direction.	Rad
"GRIDXR2"	Same as "GRIDXR1", but about the y direction.	Rad
"GRIDXR3"	Same as "GRIDXR1", but about the z direction.	Rad
"GRIDVT1"	Same as "GRIDXT1", but for *d(GRIDXT1)/dt.*	FMLUNIT/sec
"GRIDVT2"	Same as "GRIDVT1", but along the y direction.	FMLUNIT/sec
"GRIDVT3"	Same as "GRIDVT1", but along the z direction.	FMLUNIT/sec
"GRIDVR1"	Same as "GRIDVT1", but for the rotation degree of freedom about the x direction.	Rad/sec
"GRIDVR2"	Same as "GRIDVT1", but for the rotation degree of freedom about the y direction.	Rad/sec
"GRIDVR3"	Same as "GRIDVT1", but for the rotation degree of freedom about the z direction.	Rad/sec
"GRIDGT1"	Same as "GRIDXT1", but for $d^2(GRIDXT1)/dt^2$.	FMLUNIT/sec^2
"GRIDGT2"	Same as "GRIDGT1", but along the y direction.	FMLUNIT/sec^2
"GRIDGT3"	Same as "GRIDGT1", but along the z direction.	FMLUNIT/sec^2
"GRIDGR1"	Same as "GRIDGT1", but for the rotation degree of freedom about the x direction.	Rad/sec^2
"GRIDGR2"	Same as "GRIDGR1", but about the y direction.	Rad/sec^2
"GRIDGR3"	Same as "GRIDGR1", but about the z direction.	Rad/sec^2
"FORCET1"	For the force along the x direction at a structural finite element grid point. IKEY is an integer that is the identification number of the structural grid point. Note that if "FORCET1" is activated, the matrix "SKGH" for symmetric/asymmetric maneuver or "AKGH" for anti-symmetric maneuver must be imported either by the **'ASSGN MATRIX='** Executive Control Command or the **DMI** bulk data card. Otherwise, a fatal error occurs.	$\frac{(\text{FMMUNIT} \cdot \text{FMLUNIT})}{\text{sec}^2}$
"FORCET2"	Same as "FORCET1", but along the y direction.	Same as "FORCET1"
"FORCET3"	Same as "FORCET1", but along the z direction.	Same as "FORCET1"
"FORCER1"	Same as "FORCET1", but for the moment about the x direction.	$\frac{(\text{FMMUNIT} \cdot \text{FMLUNIT}^2)}{\text{sec}^2}$
"FORCER2"	Same as "FORCER1", but about the y direction.	Same as "FORCER1"
"FORCER3"	Same as "FORCER1", but about the z direction.	Same as "FORCER1"
"LOADMOD"	For the component loads that are defined by the **LOADMOD** bulk data card. IKEY is an integer representing the identification number of the **LOADMOD** bulk data card. Note that that matrix "SKGH" for symmetric/asymmetric maneuver or "AKGH" for anti-symmetric maneuver must be imported. Otherwise, a fatal error occurs.	Mass and length units defined by FMMUNIT and FMLUNIT, respectively.
"DMI"	For a parameter whose modal values are imported by the **DMI** bulk data card. IKEY is a character string that must match the NAME entry of a **DMI** bulk data card. Number of rows defined	N/A

	in DMI must be the same as the number of the structural modes that are imposed by the **'ASSIGN FEM='** Executive Control Command. These modal values can be stresses, grid point forces, strains, etc. and can be obtained from the free vibration FEM analysis.	
"PCHFILE"	For the structural parameters defined by a **PCHFILE** bulk data card. IKEY is an integer representing the identification number of a **PCHFILE** bulk data card that imports a NASTRAN punch output file containing the modal values of element forces, stresses, or strains. The time history of all structural parameters listed in the ELLST$_i$ and FIELD$_i$ entries in the **PCHFILE** bulk data card are printed out. Note that for output, the LABEL and IKEY entries of the **MLDPRNT** bulk data card are replaced by the LABEL$_i$ and ELLST$_i$ entries of the **PCHFILE** bulk data card, respectively.	N/A

5. A Windows plotting application called ZARMA.exe has been developed to allow for automated plotting of an MLDPRNT output file. This program, along with its User's Manual, can be found in the ZEUS installation directory under the \miscel\ZARMA folder. In addition to plotting the response output, ZARMA can be used to provide the frequency and damping content of a selected response output using the Auto-Regressive Moving Average (ARMA) methodology. Note that the FORM must be set to TABLE (the default) in order to generate the appropriate output file format that is readable by ZARMA.

MLDSTAT

Airframe States

Description: Defines the airframe states and their initial values for dynamic load analysis.

Format and Example:

1	2	3	4	5	6	7	8	9	10
MLDSTAT	IDSTAT	IDTRIM	TRNSFM	ARR	BRR	DXTOX	FILENM		CONT
CONT	STATE1	INITIAL1	STATE2	INITIAL2	…	-etc-			

MLDSTAT	10	386				YES	ABMATRIX.DAT		+M1
+M1	X	0.0	ALPHA	0.01	BETA	0.02			

Field	Contents
IDSTAT	Identification number (Integer > 0) (See Remark 1)
IDTRIM	Identification number of a **MLDTRIM** bulk data card to retrieved the trim solution generated by a trim analysis as the initial flow solution for the transient response analysis.(integer>0 or blank) (see remark 2)
TRNSFM	Not used
ARR, BRR	Not used.
DXTOX	Not used
FILENM	Not used.
STATEi	Optional input. Character string that must match the following characters. For symmetric maneuver, STATEi must be either "X", "H", "U", "W", "ALPHA", "THETA", or "Q". For anti-symmetric maneuver, STATEi must be either "Y", "BETA", "V", "PHI", "PSI", "P", or "R". For asymmetric maneuver, STATEi can be either of that for symmetric or anti-symmetric maneuver. (Character) (See Remark 3) Note that $STATE_i$ can be "NONE". This implies that no transformation of airframe states from the rigid body states is performed.
INITIALi	Initial value of STATEi. (Real) (See Remark 4)

Remarks:

1. **MLDSTAT** is referred to by the **MLOADS**, **ELOADS**, **GLOADS**, or **NLFLTR** bulk data card.
2. If the trim solution is retrieved by the **MLDTRIM** bulk data card, the static aeroelastic analysis prior to the transient response analysis will be skipped. In this case, the initial solution of the transient response analysis starts with the trim solution.
3. The type of maneuver is defined by the SYM entry of the **FLUTTER** bulk data card.

SYM = 'SYM' For symmetric maneuver
SYM = 'ASYM' For asymmetric maneuver

- The airframe states of a symmetric maneuver include:

States	Description	Units
"X"	Perturbed forward position relative to the aerodynamic moment center (positive forward).	FTLUNIT defined in the **FIXMACH**, **FIXMATM** or **FIXMDEN** bulk data card.
"H"	Perturbed altitude relative to REFX, REFY and REFZ (positive upward).	FTLUNIT
"THETA"	Perturbed Euler pitch angle (θ) about REFX, REFY and REFZ.	Rad
"U"	Perturbed forward velocity relative to REFX, REFY and REFZ (positive forward).	FTLUNIT/sec
"W"	Perturbed upward velocity relative to REFX, REFY and REFZ (positive downward).	FTLUNIT/sec
"ALPHA"	Perturbed angle of attack (α).	Rad
"Q"	Perturbed pitch rate (q) about to REFX, REFY and REFZ.	Rad/sec

*REFX, REFY and REFZ are defined in the **AEROZ** bulk data card*

Note : "W" and "ALPHA" cannot be both selected.

- The airframe states of an anti-symmetric maneuver include:

States	Description	Units
"Y"	Perturbed lateral position relative to REFX, REFY and REFZ (positive toward the pilot's right hand side).	FTLUNIT
"PHI"	Perturbed Euler roll angle (ϕ) about REFX, REFY and REFZ.	Rad
"PSI"	Perturbed Euler azimuth angle (ψ) about REFX, REFY and REFZ.	Rad

"V"	Perturbed lateral velocity (*v*) relative to REFX, REFY, and REFZ (positive toward the pilot's right-hand side).	FTLUNIT/sec
"BETA"	Perturbed slide slip angle (β).	Rad
"P"	Perturbed roll rate (p) about REFX, REFY, and REFZ.	Rad/sec
"R"	Perturbed yaw rate (r) about REFX, REFY, and REFZ.	Rad/sec

*REFX, REFY, and REFZ are defined in the **AEROZ** bulk data card*

Note : "V" and "BETA" cannot be both selected.

- The airframe states of the asymmetric maneuver include those of the symmetric maneuver plus those of the anti-symmetric maneuver.

4. Specifying $\alpha_{initial} \neq 0$ and $\theta_{initial} = 0$ gives an impulsive increase of the α condition at the initial time which can be used to simulate the Wagner's function.

MLDTIME

Transient Time Step

Description: Defines the starting time, ending time, time step, and output time for a transient analysis.

Format and Example:

1	2	3	4	5	6	7	8	9	10
MLDTIME	IDTIME	TSTART	TEND	DT	OUTDT	PRINT	METHOD	MODEZRO	CONT
CONT	$MODE_1$	$X0_1$	$V0_1$	$G0_1$	$MODE_2$	$X0_2$	$V0_2$	$G0_2$	CONT
CONT	...	-etc-	...						

MLDTIME	1	0.0	2.0	0.01	10	-1			
TAB1	2	0.01	0.003						

Field	Contents
IDTIME	Unique identification number. (Integer > 0) (See Remark 1)
TSTART	Starting time. (Real)
TEND	Ending time; TEND > TSTART (Real)
DT	Time Step (Real > 0.0)
OUTDT	Skip factor for output. Every OUTDT time step will be saved in the output. (Integer > 0, Default = 1)
PRINT	Print flag for output PRINT < 0, No output of numerical values of the time history. PRINT = 0, Output numerical values of the time history on the standard output file.
METHOD	Flag to select the time integration methods for solving the state-space equation. (Integer ≥ 0, Default = 0) (See Remark 2)
MODEZRO	Identification number of a **SET1** bulk data card to list a set of mode indices. The generalized coordinates of these modes during the transient analysis are forced to be zero. (Integer ≥0) (See Remark 3)
MODE*i*	The index of the structural mode of which the initial value of its generalized coordinates is specified. (Integer) (See Remark 4)
$X0_i$	The initial displacement of the MODE*i* mode.
$V0_i$	The initial velocity of the MODE*i* mode.
$G0_i$	Not used.

Remarks:

1. The **MLDTIME** bulk data card is required by a transient response analysis (MLOADS, ELOADS, NLFLTR,…etc). Prior to the transient response analysis, a static aeroelastic analysis is always performed first to obtain a mean flow solution from which the transient response analysis (the time-domain unsteady aerodynamic computation) starts. Note that the number of time steps of this static aeroelastic analysis is 100 as default, but that can be altered by a **MKPARAM** bulk data card with TRMSTEP.

2. For METHOD = 0, the discrete state-space equation is solved by

$$X_n = AX_{n-1} + \frac{1}{2}B\left(3U_n - U_{n-1}\right)$$

where A and B involve the structural generalized stiffness, mass, and damping matrices

X is the states
U is the generalized aerodynamic forces

and n is the index of time step

For METHOD = 1 , the discrete state-space equation is solved by

$$X_n = AX_{n-1} + BU_n$$

3. If rigid body modes are included in the transient analysis, these rigid body modes may diverge because of the lack of flight control system in the transient analysis to stabilize the rigid body modes. Thus, a small numerical noise may cause the rigid body modes to diverge. Note that this option is different from deleting the rigid body modes using MLIST entry in the **FLUTTER** bulk data card that removes the rigid body mode effects completely from the transient analysis. This option still keeps the influence of the rigid body modes to the elastic modes for the transient analysis.

4. $X0_i$ and $V0_i$ give the initial condition to excite the aeroelastic system for the transient responses.

MLDTRIM

Initial Trim Condition for Transient Response Analysis

Description: Exports the trim solutions from a trim analysis or imports the trim solutions to a transient response analysis.

Format and Example:

1	2	3	4	5	6	7	8	9	10
MLDTRIM	IDTIME	ISTFLW	FILTRM		FILDEF		FILLOD		
MLDTRIM	100	300	TRIMSOL.DAT		DEFORM.DAT		LOADS.DAT		

Field	Contents
IDTRIM	Unique identification number. (Integer > 0) (See Remark 1)
ISTFLW	Unique identification number of an **EXTFILE** bulk data card to specify a file name where the flow field solution at the trim condition is stored. (Integer > 0)
FILTRM	Filename of an ASCII file that stores the trim solutions in the OUTPUT4 format. (Character, cannot be Blank)
FILDEF	Filename of an ASCII file that stores the deformation at the trim condition in the OUTPUT4 format. (Character, cannot be Blank)
FILLOD	Filename of an ASCII file that stores the inertial loads at the trim condition in the OUTPUT4 format. (Character, cannot be Blank)
	If the first character of FILTRM, FILDEF or FILLOD is a dollar sign “$”, the rest of the characters must be integers. This integer is the identification number of an **EXTFILE** bulk data card where the filename is specified. This feature allows for filenames up to 56 characters to be input.

Remarks:

1. **MLDTRIM** bulk data card can be referred to either a trim analysis (the **TRIM** bulk data card) or a transient response (the **MLOADS**, **ELOADS**, **GLOADS**, or **NLFTR** bulk data cards)

   For trim analysis:

   If the IDTRIM entry matches with the TRIMID entry of a **TRIM** bulk data card, the trim results of this trim analysis are exported to those files.

   For transient response analysis:

   The IDTRIM entry is referred to by IDTRIM entry of a **MLDSTAT** bulk data card. The data stored on those files are used at the initial trim condition for a transient response analysis.

MLOADS — Transient Maneuver Load Analysis

Description: Defines the aeroelastic system, airframe states, pilot input commands and time integration for transient maneuver load analysis.

Format and Example:

1	2	3	4	5	6	7	8	9	10
MLOADS	SID	CONID	FLTID	RESTART	STATES	COMMAND	TIME	MLDPRNT	
MLOADS	100	10	20	25	30	100	1000	20	

Field	Contents
SID	Unique set identification number. (Integer > 0) (See Remark 1)
CONID	Not used.
FLTID	Identification number of a **FLUTTER** bulk data card specifying the flight condition and the associated structural matrices. (Integer > 0) (See Remark 2)
RESTART	Identification number of a R**ESTART** bulk data card to save or retrieve the entire flow solution of the transient analysis. (Integer ≥ 0) (See Remark 3)
STATES	Identification number of a **MLDSTAT** bulk data card specifying the parameters of the airframe states. Note that the **MLDSTAT** bulk data card can refer to a **MLDTRIM** bulk data card to retrieve the trim solution (generated by a trim analysis) as the initial flow condition for the transient response analysis (Integer ≥ 0)
COMMAND	Identification number of a **MLDCOMD** bulk data card specifying the parameters of the pilot's input commands for the maneuver. (Integer ≥ 0)
TIME	Identification number of a **MLDTIME** bulk data card specifying the parameters of the time integration for solving the transient response problem. (Integer > 0)
MLDPRNT	Identification number of a **MLDPRNT** bulk data card specifying the time history of parameters that are to be printed out. (Integer ≥ 0)

Remarks:

1. For the transient maneuver load analysis, the **MLOADS** discipline must be selected in the Case Control Section with MLOADS = SID. The MLOADS module can be used to obtain the flutter boundary of the aeroelastic system by performing the MLOADS analysis at various flight conditions. By calculating the damping of the transient response at each flight condition, the user can determine the flight condition where the damping becomes zero, i.e., the flutter boundary.

Note:

– Prior to the MLOADS analysis, a static aeroelastic analysis is always performed first to obtain a mean flow solution from which the transient response analysis using the time-domain unsteady aerodynamic computation of the Euler-solver module starts. Note that the number of time steps of this static aeroelastic analysis is 100 as default but that can be altered using a **MKPARAM** bulk data card with entry TRMSTEP. The pressure distribution of such a static aeroelastic analysis can be visualized using the **PLTCP** bulk data card. The flow solution of the static aeroelastic analysis can be saved/retrieved using the **STFLOW** bulk data card. As an alternative, the trim solution computed by a trim analysis can be retrieved by a **MLDTRIM** bulk data card as the initial flow condition for the time-domain GLOADS analysis. In this case, the static aeroelastic analysis will be skipped

– If the time history of component loads or forces at structural finite element grid points is desired for output, the "SMGH" (for symmetric/asymmetric structural boundary condition) or the "MGG" (the G-set mass matrix) must be imported either by the **'ASSIGN MATRIX='** Executive Control Command or the **DMI** bulk data card.

2. Only the **FIXMACH**, **FIXMATM,** and **FIXMDEN** bulk data cards can be referred by the **FLUTTER** bulk data card for transient maneuver load analysis. All mass and length units involved in the subsequent bulk data cards referred to by **MLOADS** must be consistent with FTMUNIT and FTLUNIT, respectively. FTMUNIT and FTLUNIT are specified in the **FIXMACH**, **FIXMATM** or **FIXMDEN** bulk data card.

Note:

Unlike the flutter analysis where all flight conditions listed in the **FIXMACH**, **FIXMATM** or **FIXMDEN** bulk data card are included in the analysis, the transient maneuver load analysis only computes one flight condition per analysis. Thus, only the first velocity and density pair (for **FIXMACH**), the first altitude (for **FIXMATM**), or the first velocity (for **FIXMDEN**) listed in their respective bulk data cards is included in the analysis. The rest of the values in the list are ignored.

8. The **RESTART** bulk data card allows the user to save or retrieve the entire flow solution at the last time step of the transient analysis to a file. In the event of an abnormal termination during the transient analysis, or if the user wishes to extend the physical computational time from a previous transient analysis, the entire flow solution at the last time stop saved on the file can be retrieved for a continuous transient analysis.

MODPHI

Modify the Modal Matrix

Description: Modifies the modal matrix of the structural finite element model that is imported from the **'ASSIGN MATRIX='** Executive Control Command by a matrix of direct matrix input.

Format and Example:

1	2	3	4	5	6	7	8	9	10
MODPHI	SYM	DMI	IDMS	NDMI	IPHIS				
MODPHI	ANTI	ABCDE	1	10	1				

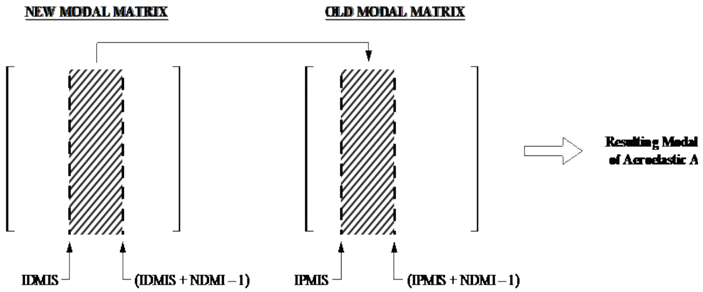
Field	Contents
SYM	Character string either "SYM", "ANTI" or "ASYM" to specify the type of the boundary condition of the structural modes that are imported by the **'ASSIGN FEM='** Executive Control Command. These modes are defined as '*the old modal matrix*' which is to be modified. (Character, see Remark 1)
DMI	Character string that matches the name of a direct matrix input that is imported by the **DMI** bulk data card or the **'ASSIGN MATRIX='** Executive Control Command. This matrix is defined as '*the new modal matrix*'. (Character, see Remark 2)
IDMIS	The starting column of the new modal matrix. (Integer > 0)
NDMI	The number of columns extracted from the new modal matrix (starting from the IDMISth column) which are used to replace the old modal matrix. (Integer > 0, see Remark 3)
IPHIS	The starting column of the old modal matrix from which the NDMI number of columns are replaced by the new modal matrix. (Integer > 0)

Remarks:

1. **MODPHI** is not referred to by other bulk data cards. The existence of **MODPHI** in the bulk data input "triggers" the program to perform the modal matrix modification. The SYM entry must match the SYM entry of the **'ASSIGN FEM='** Executive Control Command whose imported modal matrix is to be modified.

2. The number of rows of '*the new modal matrix*' must be the same as that of '*the old modal matrix*' which is the G-set degrees of freedom (6 × number of the structural grid points). However, the number of columns (modes) of these two matrices does not have to be the same.

3. The figure to the right shows the operation of the **MODPHI** bulk data card.



The generalized mass and stiffness matrices (imported by the **'ASSIGN FEM='** Executive Control Command with entry SYM being equal to the entry SYM of **MODPHI** bulk data card) are also modified by the resulting modal matrix. The generalized aerodynamic force matrices are also computed based on the resulting modal matrix. Therefore, the resulting modal matrix redefines the generalized modal coordinates of the aeroelastic system.

Note : Multiple **MODPHI** bulk data cards can be specified (even with a same SYM entry). These **MODPHI** can modify different columns of the modal matrix. For instance, the following two **MODPHI** bulk data cards use two new modal matrices (DMI1 and DMI2) to modify the old modal matrix.

MODPHI	SYM	DMI1	1	3	1				

MODPHI	SYM	DMI2	1	3	4				

NANSI

Nonlinear Structural ROM

Description: Performs transient response analysis with an externally imported nonlinear structural Reduced Order Model (ROM) for Nonlinear Aerodynamic and Nonlinear Structural Interactions (NANSI) analysis.

Format and Example:

1	2	3	4	5	6	7	8	9	10
NANSI	SID	CONID	FLTID	METHOD	RESTART	COMMAND	TIME	MLDPRNT	CONT
CONT	FACK2	GENSTF2	FACK3	GENSTF3	FACC2	GENDMP2	FACC3	GENDMP3	

NANSI	100		10	1	20	20	30	40	+N
+N	0.1	GENK2	2.0	$400			0.3	GENC3	

Field	Contents
SID	Identification number that is referred to by an **NANSI** Case Control Command. Note that for NANSI analysis, the generalized mass matrix imported by the "ASSIGN FEM=" executive control command must be a unit matrix. (Integer > 0) (See Remark 1)
CONID	Not used.
FLTID	Identification number of a **FLUTTER** bulk data card to define the flight condition. (Integer > 0)
METHOD	Method to solve the nonlinear equations of motion METHOD = 0 or 1, use the fixed point method METHOD = 2, use the Newton-Raphson scheme. METHOD = 3, use the fixed point method first. If the convergence fails, the method reverts to the Newton-Raphson scheme. (Integer ≥ 0, Default = 2)
RESTART	Identification number refers to a **RESTART** bulk data card. (Integer ≥ 0)
COMMAND	Identification number of a **MLDCOMD** bulk data card for specifying the parameters of the pilot’s input commands to excite the aeroelastic system. (Integer ≥ 0)
TIME	Identification number of a **MLDTIME** bulk data card to specify the parameters of the time integration. (Integer > 0)
MLDPRNT	Identification number of a **MLDPRNT** bulk data card to specify the time history of parameters for output. (Integer ≥ 0)
FACK2	A factor applied to the GENSTF2 matrix. (Real)
GENSTF2	Character string to specify a matrix that contains the quadratic nonlinear generalized stiffness matrix. Note that GENSTF2 can be imported by an "ASSIGN MATRIX="

	executive control command or a **DMI** bulk data card with size of NM×NM×NM, where NM is the number of modes being imported by the "ASSIGN FEM=" executive control command. (Character or blank)
FACK3	A factor applied to the GENSTF3 matrix. (Real)
GENSTF3	Same as GENSTF2 except for the cubic nonlinear generalized stiffness matrix with size of NM×NM×NM×NM. (Character or blank)
FACC2	A factor applied to the GENDMP2 matrix. (Real)
GENDMP2	Same as GENSTF2 except for the quadratic nonlinear generalized damping matrix with size of NM×NM×NM. (Character or blank)
FACC3	A factor applied to the GENDMP3 matrix. (Real)
GENDMP3	Same as GENDMP2 except for the cubic nonlinear generalized damping matrix with size of NM×NM×NM×NM. (Character or blank)

Remarks:

1. NANSI solves the following equations of motion by a Newmark-β scheme:

$$M_{ij}\ddot{q}_j + (D_{ij}^{(1)} + D_{ijl}^{(2)} q_l + D_{ijlp}^{(3)} q_l q_p)\dot{q}_j + (K_{ij}^{(1)} + K_{ijl}^{(2)} q_l + K_{ijlp}^{(3)} q_l q_p) q_j = F_i$$

   where M_{ij} are the generalized masses, $K_{ij}^{(1)}$, $K_{ijl}^{(2)}$, $K_{ijlp}^{(3)}$ are stiffness coefficients associated with the linear, quadratic, and cubic term, respectively. $D_{ij}^{(1)}, D_{ijl}^{(2)}$ and $D_{ijlp}^{(3)}$ are the damping. coefficients associated with the linear, quadratic, and cubic term, respectively. Note that $D_{ij}^{(1)}$ is specified by the **TABDMP1** bulk data card and M_{ij} and $K_{ij}^{(1)}$ are imported by the "ASSIGN FEM=" executive control command.

   and q_j is the generalized coordinates, F_i is the generalized aerodynamic forces.

   Because the structural matrices are defined in the modal space, not the physical domain, the size of the matrices is small, thereby defining as a reduced order model of the nonlinear structure.

2. If the first character of GENDMP2, GENDMP3, GENSTF2, or GENSTF3 is a dollar sign “$”, then the rest of the character string must be an integers that is the identification number of an **EXTFILE** bulk data card. In this **EXTFILE** bulk data card, the name of the external file where the data of the respective matrix is stored and specified.

NLFLTR Nonlinear Flutter Analysis

Description: Defines data needed to perform a nonlinear flutter analysis.

Format and Example:

1	2	3	4	5	6	7	8	9	10
NLFLTR	SID	CONID	FLTID	RESTART	STATES	COMMAND	MLDTIME	MLDPRNT	CONT
CONT	NA	LABELA1	IKEYA1	CA1	LABELA2	IKEYA2	CA2	INITA	CONT
CONT	NB	LABELB1	IKEYB1	CB1	LABELB2	IKEYB2	CB2	INITB	CONT
CONT	INTERP	VB1	VB2	VB3	VB4	VB5	VB6	VB7	CONT
CONT	VA1	ID_{11}	ID_{21}	ID_{31}	ID_{41}	ID_{51}	ID_{61}	ID_{71}	CONT
CONT	VA2	ID_{12}	ID_{22}	ID_{32}	ID_{42}	ID_{52}	ID_{62}	ID_{72}	CONT
CONT			...	-etc-	...				CONT
CONT	VA_{NA}	ID_{1NA}	ID_{2NA}	ID_{3NA}	ID_{4NA}	ID_{5NA}	ID_{6NA}	ID_{7NA}	

NLFLTR	10		11	12	10	1	3	4	+N1
+N1	3	GRIDXT3	90	1.0	GRIDXT3	91	-1.0	2.0	+N2
+N2	4	GRIDGR3	100	1.0	GRIDGR3	101	-3.0	NONE	+N3
+N3	LINEAR	-100.0	0.0	0.0	100.0				+N4
+N4	-20.0	-10	-11	12	13				+N5
+N5	0.0	101	111	121	131				+N6
+N6	40.0	20	30	-40	50				

Field	Contents
SID	Unique set identification number. (Integer > 0) (See Remark 1)
CONID	Not used.
FLTID	Identification number of a **FLUTTER** bulk data card specifying the flight condition and the associated structural matrices. (Integer > 0) (See Remark 2)
RESTART	Identification number of a **RESTART** bulk data card to save or retrieve the entire flow solution at the transient analysis. (Integer ≥ 0) (See Remark 3).
STATES	Identification number of a **MLDSTAT** bulk data card specifying the parameters of the

	airframe states. Note that the **MLDSTAT** bulk data card can refer to a **MLDTRIM** bulk data card to retrieve the trim solution (generated by a trim analysis) as the initial flow condition for the transient response analysis (Integer ≥ 0)
COMMAND	Identification number of **MLDCOMD** and/or **DGUST** bulk data cards that provide the external disturbance to excite the structures. (Integer > 0) (See Remark 4)
TIME	Identification number of a **MLDTIME** bulk data card specifying the parameters of the time integration for solving the transient response problem. (Integer > 0)
MLDPRNT	Identification number of a **MLDPRNT** bulk data card specifying the time history of the parameters that are to be printed out. (Integer ≥ 0)
NA	The number of values of the first nonlinear parameter at which the discrete nonlinear aeroelastic systems are specified. (Integer ≥ 0) (Note that the maximum NA is 99) (See Remark 5)
LABELA1, IKEYA1 LABELA2 IKEYA2	LABELA1 and LABELA2 are two character strings and IKEYA1 and IKEYA2 can be either character strings or integers (depending on LABELA1 and LABELA2). LABELA1, LABELA2, IKEYA1, and IKEYA2 jointly define the type of the first nonlinear parameter. (See Remark 6)
CA1,CA2	Real coefficients shown in the following equation: $V_a = CA1 \, . \, F_{a1} + CA2 \, . \, F_{a2}$ Where V_a is the value of the first nonlinear parameter. F_{a1} and F_{a2} are computed based on $LABELA_i$ and $IKEYA_i$ respectively. (Real) (See Remark 6)
INITA	INITA can be either a character string, integer or real number to specify the initial value of the first nonlinear parameter For character string, INITA must be "NONE". This "triggers" the program to automatically compute the initial value of the first nonlinear parameter by an iterative procedure For real number, INITA is the initial value of the first nonlinear parameter specified by the user. For integer, INITA is the identification number of a **TABLED1** bulk data card where the time history of the first nonlinear parameter (nonlinear parameter scheduling) is specified by the user. Thus, the time history of the first nonlinear parameter is not computed by the program. Rather, it is fixed according to the nonlinear parameter scheduling. In this case, LABELA1 and LABELA2 can be blank and IKEY1, IKEY2, CA1, and CA2 are not used. (Character, Real or Integer, Default = "NONE")
NB	Same as NA but for the second nonlinear parameter. Note that the maximum NB is 7. (Integer ≥ 0)
LABELB1, IKEYB1 LABELB2, IKEYB2	Same as LABELA1, IKEYA1, and LABELA2, IKEYA2 respectively but for the second nonlinear parameter.
CB1, CB2	Same as CA1, and CA2, respectively, but for the second nonlinear parameter. (See Remark 8)

INITB	Same as INITA but for the second nonlinear parameter (Character, Real, or Integer, Default = “NONE”)
INTERP	Character string either “LINEAR” or “CUBIC” specifying the method for the interpolation of the system matrices. For INTERP= “LINEAR”, linear interpolation is used. Otherwise, cubic spline is used. (Character, Default = “LINEAR”) (See Remark 8)
VB_i, i=1, NB	The value of the second nonlinear parameter at which the discrete nonlinear aeroelastic systems are defined. (Real) (See Remark 9)
VA_j, j=1, NA	The value of the first nonlinear parameter at which the discrete nonlinear aeroelastic systems are defined. (Real)
ID_{ij}, i=1, NB, j=1, NA	The absolute value of ID_{ij} is the identification number of a **NLSYSM** bulk data card to define the aeroelastic system at VB_i and VA_j. (Integer) Note that ID_{ij} can be a negative integer. This triggers the program to print out the generalized aerodynamic force, mass, damping and stiffness matrices of the aeroelastic system defined by the **NLSYSM** bulk data card. (See Remark 10)

Remarks:

1. For the nonlinear flutter analysis, the NLFLTR discipline must be selected in the Case Control Section with **NLFLTR=SID**.

   NOTE:
   - Prior to the NLFLTR analysis, a static aeroelastic analysis is always performed first to obtain a mean flow solution from which the transient response analysis using the time-domain unsteady aerodynamic computation of the Euler-solver module starts. Note that the number of time steps of this static aeroelastic analysis is 100 as default, but that can be altered using a **MKPARAM** bulk data card with entry TRMSTEP. The pressure distribution of such a static aeroelastic analysis can be visualized using the **PLTCP** bulk data card. The flow solution of the static aeroelastic analysis can be saved/retrieved using the **STFLOW** bulk data card. As an alternative, the trim solution computed by a trim analysis can be retrieved by a **MLDTRIM** bulk data card as the initial flow condition for the time-domain NLFLTR analysis. In this case, the static aeroelastic analysis will be skipped
   - .If the time history of component loads or forces at structural finite element grid points is desired for output, the “SMGH” (for symmetric/asymmetric structural boundary condition) or the "MGG" (the G-set mass matrix) must be imported either by the **‘ASSIGN MATRIX=’** Executive Control Command or the **DMI** bulk data card.

2. All mass and length units involved in the subsequent bulk data cards referred to cards can be referred by the **FLUTTER** bulk data card for transient maneuver load by **NLFLTR** must be consistent with FTMUNIT and FTLUNIT, respectively. FTMUNIT and FTLUNIT are specified in the **FIXMACH**, **FIXMATM** or **FIXMDEN** bulk data card.

   NOTE:
   Unlike the flutter analysis where all flight conditions listed in the **FIXMACH, FIXMATM** or **FIXMDEN** bulk data card are included in the analysis, the nonlinear flutter analysis only

computes one flight condition per analysis. Thus, only the first velocity and density pair (for **FIXMACH**). The first altitude (for **FIXMATM**), or the first velocity (for **FIXMDEN**) listed in their respective bulk data cards is included in the analysis. The rest of the values in the list are ignored.

3. The **RESTART** bulk data card allows the user to save or retrieve the entire flow solution at the last time step of the transient analysis to a file. In the event of an abnormal termination during the transient analysis, or if the user wishes to extend the physical computational time from a previous transient analysis, the entire flow solution at the last time stop saved on the file can be retrieved for a continuous transient analysis.

4. Because of the time domain nonlinear flutter analysis, some external disturbances are required to excite the structures. At least one **MLDCOMD** or **DGUST** bulk data card must be referred to by the **NLFLTR** bulk data card to provide such an external disturbance. Note that if both **MLDCOMD** and **DGUST** bulk data cards have the same identification number specified by the COMMAND entry, the external disturbance is the superposition of the forces generated by the pilot's input command and the discrete gust profile.

5. The NLFLTR analysis assumes that the nonlinearity of the aeroelastic system is a function of two parameters called the "nonlinear parameters" shown as follows:

$$\mathrm{M}(V_a,V_b)\ddot{\xi} + \mathrm{C}(V_a,V_b)\dot{\xi} + \mathrm{K}(V_a,V_b)\xi - q_\infty \mathrm{Q}(V_a,V_b)\xi = 0$$

where **M**, **C**, **K,** and **Q** are the generalized mass, damping, stiffness, and aerodynamic force matrices, respectively. q_∞ and φ are the dynamic pressure and the generalized coordinates, respectively. V_a and V_b are the values of the two nonlinear parameters where V_a is defined as the "first nonlinear parameter" and V_b as the "second nonlinear parameter".

To define the above nonlinear aeroelastic system, the user must specify the **M**, **C**, **K**, and **Q** matrices at a set of discrete values of V_a and V_b. The number of V_a is defined by the entry NA, and the number of V_b is defined by the entry NB.

NOTE:

If NA≤1, the program assumes that the nonlinear aeroelastic system is only a function of V_b. Consequently, all entries in the **NLFLTR** bulk data card associated with the first nonlinear parameter such as LABELA_i, IKEYA_i, CA_i, etc. are inactive. Likewise, if NB≤1, all entries associated with the second nonlinear parameter are inactive. The maximum number of NB is 7, and the maximum number of NA is 99. At most two nonlinear parameters are allowed to define the nonlinear aeroelastic system.

6. LABELA1, LABELA2, LABELB1, and LABELB2 must match one of the character strings listed in the following table.

LABELA$_i$ /LABELB$_i$	Description	Units
"GRIDXT1"	For the displacement along the x direction at a structural finite element grid point. IKEYA$_i$ / IKEYB$_i$ is an integer that is the identification of the grid point.	The length units defined in the FMLUNIT entry of the **AEROZ** bulk data card.
"GRIDXT2"	Same as "GRIDXT1", but along the y direction.	FMLUNIT
"GRIDXT3"	Same as "GRIDXT1", but along the z direction.	FMLUNIT
"GRIDXR1"	Same as "GRIDXT1", but for the rotation d.o.f. about the x direction.	Rad
"GRIDXR2"	Same as "GRIDXR1", but about the y direction.	Rad
"GRIDXR3"	Same as "GRIDXR1", but about the z-direction.	Rad
"GRIDVT1"	Same as "GRIDXT1", but for *d(GRIDXT1)/dt*.	FMLUNIT/sec
"GRIDVT2"	Same as "GRIDVT1", but along the y direction.	FMLUNIT/sec
"GRIDVT3"	Same as "GRIDVT1", but along the z direction.	FMLUNIT/sec
"GRIDVR1"	Same as "GRIDVT1", but for the rotation d.o.f. about the x direction.	Rad/sec
"GRIDVR2"	Same as "GRIDVT1", but for the rotation d.o.f. about the y direction.	Rad/sec
"GRIDVR3"	Same as "GRIDVT1", but for the rotation d.o.f. about the z direction.	Rad/sec
"GRIDGT1"	Same as "GRIDXT1", but for $d^2(GRIDXT1)/dt^2$	FMLUNIT/sec^2
"GRIDGT2"	Same as "GRIDGT1", but along the y direction.	FMLUNIT/sec^2
"GRIDGT3"	Same as "GRIDGT1", but along the z direction.	FMLUNIT/sec^2
"GRIDGR1"	Same as "GRIDGT1", but for the rotation d.o.f. about the x direction.	Rad/sec^2
"GRIDGR2"	Same as "GRIDGR1", but about the y direction.	Rad/sec^2
"GRIDGR3"	Same as "GRIDGR1", but about the z direction.	Rad/sec^2
"MODALX"	For the generalized modal coordinates of the structural modes. IKEY is an integer that is the index of the modes (including the rigid body modes). Note that the structural modes are those defined in the NMODE and MLIST entries of the **FLUTTER** bulk data card.	None
"MODALV"	Same as "MODALX", but for *d(MODALX)/dt*.	Per second
"MODALG"	Same as "MODALX", but for $d^2(MODALX)/dt^2$.	Per second^2
"FORCET1"	For the force along the x direction at a structural finite element grid point. IKEYA$_i$/IKEYB$_i$ is an integer that is the identification number of the structural grid point. Note that if "FORCET1" is activated, the matrix "SKGH" for symmetric/asymmetric maneuver or "AKGH" for anti-	$\frac{(\text{FMMUNIT} \cdot \text{FMLUNIT})}{\text{sec}^2}$

	symmetric maneuver must be imported either by the **'ASSGN MATRIX='** Executive Control Command or the **DMI** bulk data card. Otherwise, a fatal error occurs.	
"FORCET2"	Same as "FORCET1", but along the y direction.	Same as "FORCET1"
"FORCET3"	Same as "FORCET1", but along the z direction.	Same as "FORCET1"
"FORCER1"	Same as "FORCET1", but for the moment about the x direction.	$\frac{(\text{FMMUNIT} \cdot \text{FMLUNIT}^2)}{\text{sec}^2}$
"FORCER2"	Same as "FORCER1", but about the y direction.	Same as "FORCER1"
"FORCER3"	Same as "FORCER1", but about the z direction.	Same as "FORCER1"
"LOADMOD"	For the component loads that are defined by the **LOADMOD** bulk data card. $IKEYA_i/IKEYB_i$ is an integer representing the identification number of the **LOADMOD** bulk data card. Note that that matrix "SKGH" for symmetric/asymmetric maneuver or "AKGH" for anti-symmetric maneuver must be imported. Otherwise, a fatal error occurs.	Mass and length units defined by FMMUNIT and FMLUNIT, respectively.
"DMI"	For a parameter whose modal values are imported by the **DMI** bulk data card. IKEY is a character string that must match the NAME entry of a **DMI** bulk data card. Number of rows defined in DMI must be the same as the number of the structural modes that are imported by the **'ASSIGN FEM='** Executive Control Command. These modal values can be stresses, grid point forces, strains, etc. and can be obtained from the free vibration FEM analysis.	N/A

7. The values of the first and the second nonlinear parameters are computed by the following equations.

$$V_a = CA1 \,.\, F_{a1} + CA2 \,.\, F_{a2}$$
$$V_b = CB1 \,.\, F_{b1} + CB2 \,.\, F_{b2}$$

where F_{ai} and F_{bi} are the values defined by $LABELA_i$ and $LABELB_i$, respectively.

8. At each time step during the time integration of the nonlinear flutter analysis, the program first computes V_a and V_b and then obtains $\mathbf{M}(V_a, V_b)$, $\mathbf{C}(V_a, V_b)$, $\mathbf{K}(V_a, V_b)$, and $\mathbf{Q}(V_a, V_b)$ by the interpolation based on the current values of V_a and V_b. The method of interpolation is defined by the entry INTERP.

9. VB_i (as well as VA_i) must be in the ascending order. If INTERP = "LINEAR', VB_i=VB_{i+1} is allowed. If INTERP = "CUBIC", VB_{i+1} must be greater than VB_i.

10. ID_{ij}= 0 is allowed. This implies that the aeroelastic system at VB_i and VA_j is the same as the baseline structure that is imported by the **'ASSIGN FEM='** Executive Control Command.

NLSYSM Nonlinear Aeroelastic Matrices

Description: Defines the matrices of the aeroelastic system for nonlinear open-loop/closed-loop flutter analysis.

Format and Example:

1	2	3	4	5	6	7	8	9	10
NLSYSM	SID	NEWMODE	PHI	DELTQ	COEFFQ	GENQ	IDCON	CONFRC	CONT
CONT	DELTMAT	GSETMAT	COEFFM	MASS	COEFFC	DAMPING	COEFFK	STIFF	

1	2	3	4	5	6	7	8	9	10
NLSYSM	10	YES	NEWPHI						+N1
+N1	YES	NO	0.1	NEWM			1.0	NEWK	

Field	Contents
SID	Unique identification number. (Integer > 0) (See Remark 1)
NEWMODE	Character string either “YES” or “NO”. For NEWMODE = “YES”, the generalized aerodynamic force matrix defined in the GENQ entry is computed based on a new modal matrix defined in the PHI entry. For NEWMODE=”NO” GENQ is computed based on the modal matrix of the baseline structure that is imported by the **‘ASSIGN FEM=’** Executive Control Command. (Character, Default = “YES”) (See Remark 2)
PHI	Character string that matches the name of a direct matrix input by either a **‘ASSIGN MATRIX=’** Executive Control Command or a **DMI** bulk data card. This direct matrix input contains the modal data used to compute the generalized aerodynamic force matrix GENQ, generalized mass, damping and stiffness matrices (MASS, DAMPING, and STIFF, respectively) Note: PHI is used only if NEWMODE = “YES”. (Character) (See Remark 3)
DELTQ	Not used.
COEFFQ	Not used.
GENQ	Not used.
IDCON	Not used.
CONFRC	Character string that matches a direct matrix input by either an **‘ASSIGN MATRIX=’** Executive Control Command or a **DMI** bulk data card. This direct matrix input contains NMODE rows and one column representing a constant force vector on the right hand side of the equations of motion, where NMODE is the number of modes

	imported by the **'ASSIGN FEM='** Executive Control Command. If CONFRC is blank, zero force is assumed. (Character or Blank) (See Remark 4)
DELTMAT	Character string either "YES" or "NO". For DELTMAT = "YES", the matrix defined on the MASS, DAMPING, and STIFF entries represent an incremental change of these of the baseline structure. For DELTMAT= "NO", these matrices are directly used as the mass, damping and stiffness matrices respectively. (Character, Default = "NO")
GSETMAT	Character string either "YES" or "NO" for GSETMAT= "YES", the matrices defined in the MASS, DAMPING, and STIFF are in the G-set degrees of freedom (G-set d.o.f. = 6 × number of structural finite element grid points). For GSETMAT= "NO", these matrices are defined in the generalized coordinates whose size equals the number of modes imported by the **'ASSIGN FEM='** Executive Control Command. (Character)
COEFFM	Multiplication factor to the matrix defined by the MASS entry. (Real, Default = 1.0)
MASS	Character string that matches the name of a direct matrix input by either a **'ASSIGN MATRIX='** Executive Control Command, **DMI** bulk data card, or a **DMIG** bulk data card. If GSETMAT= "YES", the matrix contains the mass matrix defined in the G-set d.o.f. If GSETMAT= "NO", the matrix contains the generalized mass matrix whose size is equal to the number of modes imported by the **'ASSIGN FEM='** Executive Control Command. Note that if MASS is blank or "NULL", a null matrix is used. If MASS = "BASE", the generalized mass matrix of the baseline structure is used. (Character, Default = "NULL") (See Remark 5)
COEFFC	Same as COEFFM but for the damping matrix. (Real, Default = 1.0)
DAMPING	Same as MASS but for the damping matrix. (Character)
COEFFK	Same as COEFFM but for the stiffness matrix. (Real, Default = 1.0)
STIFF	Same as MASS but for the stiffness matrix. (Character)

Remarks:

1. **NLSYSM** is referred to by the **NLFLTR** bulk data card to define the nonlinear aeroelastic system matrices at a given value of nonlinear parameter.

2. For NEWMODE = "YES", the program computes a transformation matrix using the matrix defined in the PHI entry and the modal matrix of the baseline structure (imported by the **'ASSIGN FEM='** Executive Control Command). This transformation matrix is used to transform the generalized coordinates of PHI to the generalized coordinates of the baseline structure. Therefore, the generalized coordinates of the nonlinear flutter analysis are always based on those of the baseline structure.

3. The size of PHI must be the same as the modal matrix that is imported by the **'ASSIGN FEM='** Executive Control Command but excluding the omitted modes defined in the MLIST entry of the **FLUTTER** bulk data card.

4. These constant forces are constant generalized force vectors that are independent of the solution of the generalized coordinates. The constant forces are required in a nonlinear freeplay response

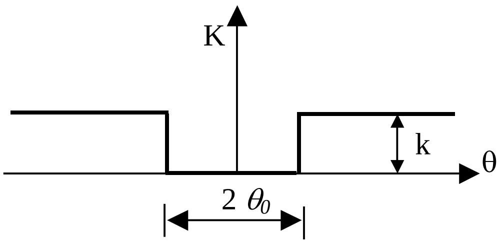
analysis to provide a correct torque vs. deflection angle relationship (see figure below). The equation of motion for a structure with freeplay can be written as:

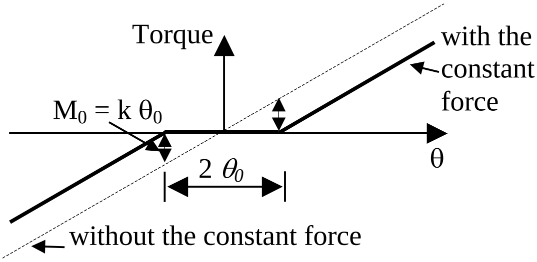
$$\left[\bar{M}\right]\left\{\ddot{\xi}\right\}+\left[\bar{K}_f\right]\left\{\xi\right\}=F_a\left(t\right)+\phi^T\left[K_f\right]\left\{\theta_o\right\} \qquad \theta>\theta_o$$

$$\left[\bar{M}\right]\left\{\ddot{\xi}\right\}+\left[\bar{K}_0\right]\left\{\xi\right\}=F_a\left(t\right) \qquad \theta_o\geq\theta\geq-\theta_o$$

$$\left[\bar{M}\right]\left\{\ddot{\xi}\right\}+\left[\bar{K}_f\right]\left\{\xi\right\}=F_a\left(t\right)-\phi^T\left[K_f\right]\left\{\theta_o\right\} \qquad \theta<-\theta_o$$

where F_a is the generalized aerodynamic forces, $\bar{K}_f$ and $\bar{K}_o$ are the generalized stiffness matrices associated with $|\theta|\geq\theta_0$ and $|\theta|\leq\theta_0$ respectively. The constant forces, in this case, are represented by the offset moment $\phi^T\left[K_f\right]\left\{\theta_0\right\}$ where K_f is the physical stiffness matrix without freeplay.





It can be seen that without the constant force the resulting torque vs. deflection angle is a straight line which is incorrect in the presence of freeplay.

5. For NEWMODE= “YES” and GSETMAT= “NO”, the generalized mass matrix is computed based on the modal matrix defined in the PHI entry. The program will transform the generalized mass matrix from the generalized coordinates of PHI to those of the baseline structure. It should be noted that if the nonlinear parameter ($LABELA_i$ and $LABELB_i$ in the **NLFLTR** bulk data card) represents forces or moments at structural grid points, GSETMAT = “YES” should be used.

OMITMOD

Delete Structural Modes

Description: Delete structural modes from the database permanently.

Format and Example:

1	2	3	4	5	6	7	8	9	10
OMITMOD	SYMM	MAXMOD							CONT
CONT	MODE1	MODE2	...	-etc-	...				

OMITMOD	ANTI	20							+OMT
+OMT	1	5	7						

Field	Contents
SYMM	Character string to specify the boundary condition of which the structural modes are to be deleted. SYMM = “SYMM” for symmetric modes. SYMM = “ANTI” for anti-symmetric modes. SYMM = “ASYM” for asymmetric modes. (Character) (See Remark 1)
MAXMOD	All structural modes whose indices are greater than MAXMOD are deleted. Note that if MAXMOD = 0, no mode is deleted. (Integer ≥ 0, Default = index of the highest mode)
MODE*i*	Optional indices of the structural mode(s) that are to be deleted. In addition to any specified MAXMOD. Note: MODE*i* can be used by itself without specifying a MAXMOD entry. (Integer ≥ 0)

Remarks:

1. The **OMITMOD** bulk data card is not referred to by any other bulk data card. Its existence “triggers” the program to delete some of the modes that are imported by the **‘ASSIGN FEM=’** Executive Control Command. It should be noted that the remaining modes are used by all downstream aeroelastic analysis; **FLUTTER**, **TRIM**, **MLOADS**, **ELOADS**, **GLOADS**, **NLFLTR**, ...etc.

OUTPUT4 — Export a Matrix Data Entity

Description: Exports a matrix data entity in the OUTPUT4 format to a data file. See description of **'ASSIGN MATRIX='** Executive Control Command for the definition of the OUTPUT4 format.

Format and Example:

1	2	3	4	5	6	7	8	9	10
OUTPUT4	MATNAM	FILENM		FORM					
OUTPUT4	UGTKG	UGTKG.DAT		UNFORM					

Field	Contents
MATNAM	The name of the matrix to be exported. (Character) (See Remark 1)
FILENM	Character string specifying the name of the data file in which the data of the matrix is stored. The file name is always in uppercase. In case the input file name is given in lowercase, the program converts it to uppercase. If the first character of FILENM is a dollar sign "$", the rest of the characters must be integers. This integer is the identification number of an **EXTFILE** bulk data card where the filename is specified. This feature allows for filenames up to 56 characters to be input. (Character)
FORM	Character string either "FORMAT", "FORMAT23", or "UNFORM". For FORM = "FORMAT" the output file is in ASCII with 5E16.9 format. FORM = "FORMAT23" the output file is in ASCII with 3D23.16 format. FORM = "UNFORM" the output file is unformatted. (Character, Default = "FORMAT")

Remarks:

1. **OUTPUT4** is not referred to by other bulk data cards. The existence of **OUTPUT4** in the bulk data card "triggers" the program to export the matrix. Multiple **OUTPUT4** bulk data cards can co-exist in the input file.

Matrix Name	**Description**	**Size (row×column)**	**Type**
UGTKG	Spline matrix relates 6 d.o.f. structural displacement at structural grid to the 50% chord of aerodynamic boxes, i.e. $\{x_a\} = [UGTKG]^T \{x_s\}$ Where x_a is the 6 d.o.f., displacements at aerodynamic boxes, x_s is the 6 d.o.f. displacements at structural grid.	Gset × Kset where Gset = 6× numbers of structural grid points	Real

	Note that x_s has 6 d.o.f, namely T_1, T_2, T_3, R_1, R_2, and R_3 at each structural grid, where T_1, T_2, and T_3 are the modal displacement along x, y, and *z* directions of the local coordinates. R_1, R_2, and R_3 are the modal rotation d.o.f. about *x*, *y*, and *z*. On each aerodynamic box, x_a has 6 d.o.f., namely h_x, h_y, h_z, h_y' and h_z', where h_x, h_y and h_z are the displacement along *x, y* and *z* directions of the aerodynamic coordinate. h_z, h_z' and h_z' are the slope of h_x, h_y, and h_z, respectively with respect to the x-axis. i.e. $h_x' = \frac{\partial h_x}{\partial x}$, $h_y' = \frac{\partial h_y}{\partial x}$, $h_z' = \frac{\partial h_z}{\partial x}$. Note that UGTKG could be a highly sparse matrix.		
UGPLT	Spline matrix relates 6 d.o.f. structural displacement at structural grid to the four corner points of aerodynamic boxes, i.e.	Gset × (4× Kset)	Real
UGFRC	Spline Matrix to transfer the aerodynamic forces from the aerodynamic boxes to the structural degrees of freedom. $\{F_s\} = [UGFRC]\{F_a\}$ where $\{F_a\}$ is the Kset aerodynamic forces on the aerodynamic boxes $\{F_s\}$ is the Gset aerodynamics forces at the structural degrees of freedom. Note that $[UGFRC] = [UGTKG]$ if no **SPLINEF** bulk data card is specified	Gset × Kset	Real
SPHI	Symmetric or asymmetric modal matrix imported by the **'ASSIGN FEM='** Executive Control Command.	Gset × HSet where Hset = number of modes	Real
APHI	Same as SPHI but for anti-symmetric modes.	Gset × Hset	Real
SMHH	Generalized symmetric or asymmetric mass matrix imported by the **'ASSIGN FEM='** Executive Control Command.	Hset × Hset	Real
AMHH	Same as SMHH but for the anti-symmetric structures.	Hset × Hset	Real
SKHH	Generalized symmetric or asymmetric stiffness matrix imported by the **'ASSIGN FEM='** Executive Control Command.	Hset × Hset	Real

AKHH	Same as SKHH but for the anti-symmetric structures.	Hset × Hset	Real
SPHIK	Symmetric or asymmetric modal matrix on aerodynamic boxes computed by $[SPHIK]=[UGTKG]^T[SPHI]$	Kset × Hset	Real
APHIK	Same as SPHIK but for the anti-symmetric modes	Kset × Hset	Real
SMGH	Symmetric or asymmetric modal mass matrix $[SMGH]=[M_{GG}][SPHI]$ where M_{GG} is the g-set mass matrix.	Gset × Hset	Real
AMGH	Same as SMGH but for the anti-symmetric mode.	Gset × Hset	Real
SCNTLK	Symmetric or asymmetric control surface modes at aerodynamic boxes.	Kset × NCS where NCS is the number of **AESURFZ** bulk data cards with entry SYM = “SYMM”	Real
SCNTLG	Same as SCNTLK but at structural grid.	Gset × NCS	Real
SKJ	Integration matrix converts J-set C_p to forces, $\{F_{1/2c}\}$, at ½ chord of aerodynamic boxes $\{F_{1/2c}\}=[SKJ]^T\{C_p\}$. Note the J-set combines the upper and lower boxes of **CAERO7** elements to one box.	Jset × Kset	Real
SKJUL	Integration matrix converts J*-set C_p to forces, $\{F_{1/2c}\}$, at ½ chord of aerodynamic boxes $\{F_{1/2c}\}=[SKJUL]^T\{C_p\}$. Note the J*-set represents the upper and lower boxes of **CAERO7** elements separately.	J*set × Kset	Real
LMODEK	Displacements and slopes defined at K-set d.o.f. due to the load modes specified in **LOADMOD** bulk data cards. Sectional loads are given by $\{L\}=[LMODEK]^T[SKJ]^T\{C_p\}$	Kset × NLM where NLM is the number of **LOADMOD** bulk data cards	Real
LMODEG	Translational and rotational displacements defined at G-set d.o.f. due to the load modes specified in **LOADMOD** bulk data cards. Sectional loads are given by $\{L\}=[LMODEG]^T[MGG]\{\ddot{x}_s\}$, where $\ddot{x}_s$ is the 6 d.o.f. accelerations at the structural grids.	Gset × NLM	Real

PAFOIL7 Airfoil Section Property

Description: Defines the airfoil cross sections at the root and tip of a wing-like aerodynamic component; referenced by the **CAERO7** bulk data card.

Format and Example:

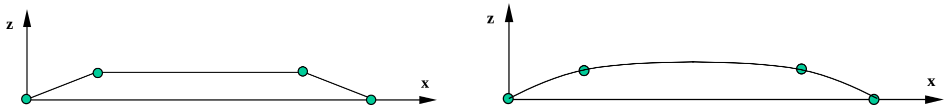
1	2	3	4	5	6	7	8	9	10
PAFOIL7	ID	ITAX	ITHR	ICAMR	RADR	ITHT	ICAMT	RADT	

PAFOIL7	1	-201	202	203	0.1	211	212	0.1	

Field	Contents
ID	**PAFOIL7** identification number (Integer > 0)
ITAX	Identification number of an **AEFACT** bulk data card used to specify the x-coordinate locations, in percentage of the chord length, where the thickness and camber are specified. ITAX can be a negative number (where ABS(ITAX) = **AEFACT** bulk data card identification number) to request linear interpolation. (Integer) (See Remark 1)
ITHR	Identification number of an **AEFACT** bulk data card used to specify the half thickness of the airfoil at the wing root. (Integer ≥ 0)
ICAMR	Identification number of an **AEFACT** bulk data card used to specify the camber of the airfoil at the wing root. (Integer ≥ 0)
RADR	Leading edge radius at the root normalized by the root chord. (Real ≥ 0.0)
ITHT	Identification number of an **AEFACT** bulk data card used to specify the half thickness at the wing tip. (Integer ≥ 0)
ICAMT	Identification number of an **AEFACT** bulk data card used to specify the camber at the wing tip. (Integer ≥ 0)
RADT	Leading edge radius at the tip normalized by the tip chord. (Real ≥ 0.0)

Remarks:

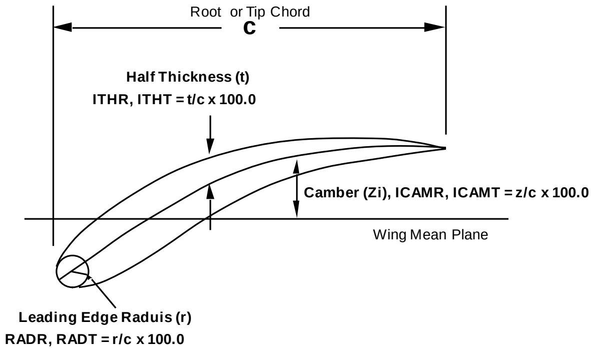
1. The ITAX x-coordinate values listed in the **AEFACT** bulk data card must start with 0.0 and end with 100.0. If ITAX is a positive integer, then a cubic interpolation is used between the airfoil points established by the ITAX, ITHR, ICAMR, RADR, ICAMT, and RADT entries. However, ITAX can be a negative number which implies that a linear interpolation is used between the airfoil points. For example, if the desired airfoil shape at the wing root is shown in Figure(a) below, and a positive value for ITAX was used, the resulting airfoil shape would be that shown in Figure(b) which is incorrect. In this case, a negative value for ITAX is required to generate the airfoil shape shown in Figure(a).



(a) (b)

Note: The number of x-coordinate values must be at least 3.

2. The values listed in the **AEFACT** bulk data cards with identification numbers of ITH(R)/(T), ICAM(R)/(T) and RAD(R)/(T) are in percentage of the root/tip chord lengths (c), respectively. For instance, in the following figure, the ITHR represents the half thickness distribution computed by (t/c) × 100, where *t* is the half thickness and c is the chord at the root. The ITHT represents similar values at the tip chord. ICAM(R)/(T) and RAD(R)/(T) similarly denote camber and leading edge radius, computed by their respective equations shown in the figure below.



Note: The positive camber is in the same direction of the normal vector of the CAERO7 macroelement. See Remark 6 of the **CAERO7** bulk data card for the definition of the normal vector.

3. The number of values listed in the **AEFACT** cards for ITAX, ITHR, ICAMR, ITHT, and ICAMT must be the same.

4. The camber and thickness distributions are computed by linear interpolation from the wing root to the wing tip.

5. To print out the resulting slopes, the user can set a negative integer in the PRINT entry of the **MKAEROZ** bulk data card.

PAFOIL8 — Airfoil Section Property

Description: Defines an NACA series type of airfoil section at the root and tip of a wing-like aerodynamic component referenced by the **CAERO7** bulk data card. Note that the **PAFOIL8** bulk data card is an alternative form of the **PAFOIL7** bulk data card except for defining an NACA series type of airfoil section.

Format and Example:

1	2	3	4	5	6	7	8	9	10
PAFOIL8	ID	RADR	IROOT	RADT	ITIP	PRINT	INTERP	REVERSE	

PAFOIL8	100	1.0	101	1.5	102	1	LINEAR	YES	

Field	Contents
ID	Identification number that is referred to by a **CAERO7** bulk data card. (Integer > 0) (See Remark 1)
RADR	Leading edge radius at the root normalized by the root chord in percentage of the chord length. (Real ≥ 0.0)
IROOT	Identification number of an **FOILSEC** bulk data card to define the airfoil section at root chord. (Integer > 0)
RADT	Same as RADR except for the tip chord. (Real ≥ 0.0)
ITIP	Same as IROOT except for the tip chord. (Integer > 0)
PRINT	Flag for printing out the airfoil shape on the standard output file. PRINT=1 for printing. (Integer)
INTERP	Character string either “LINEAR” or “CUBIC”. For INTERP = “LINEAR,” use linear interpolation to interpolate the airfoil thickness distribution to the CAERO7 macroelement. Otherwise, cubic spline is used. (Character, Default = “CUBIC”)
REVERSE	Character string either “Yes” or “No”. For REVERSE = “YES”, the resulting airfoil shape of the upper and lower surface is reversed. (Character, Default = “NO”) (See Remark 2)

Remarks:

1. The **PAFOIL8** bulk data card is an alternative form of the **PAFOIL7** bulk data card.

2. If the CAERO7 macroelement is located on the left hand side and is modeled from the wing root to wing tip, the airfoil shape must be upside down to follow the normal vector convention of the **CAERO7** bulk data card (See Remark 5 of the **CAERO7** bulk data card). In this case, REVERSE = “YES” must be used.

PANLST1 Set of Aerodynamic Boxes

Description: Defines a set of aerodynamic boxes.

Format and Example:

1	2	3	4	5	6	7	8	9	10
PANLST1	SETID	MACROID	BOX1	BOX2					
PANLST1	100	111	111	118					

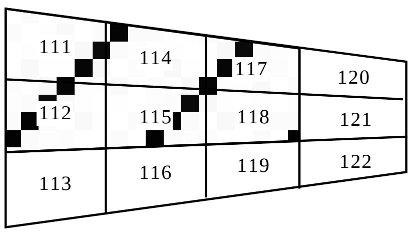
Field	Contents
SETID	Unique set identification number. (Integer > 0) (See Remark 1)
MACROID	Identification number of a **CAERO7** bulk data card to which the aerodynamic boxes listed in the set belong. (Integer ≥ 0) (See Remark 2)
BOX1	Identification number of the first aerodynamic box. (Integer > 0)
BOX2	Identification number of the last aerodynamic box. (Integer > BOX1) (See Remark 3)

Remarks:

1. **PANLST1** is referred to by **SPLINEi**, **ATTACH**, **LOADMOD**, **CPFACT**, and/or **AESURFZ** bulk data card.

2. MACROID is used to define a spline plane for the infinite plate spline method (**SPLINE1** bulk data card).

3. The following sketch shows the boxes identified via BOX1 and BOX2 entries, if BOX1 = 111, BOX2 = 118 and MACROID = 111.



PANLST2 Set of Aerodynamic Boxes

Description: Defines a set of aerodynamic boxes.

Format and Example:

1	2	3	4	5	6	7	8	9	10
PANLST2	SETID	MACROID	BOX1	BOX2	BOX3	BOX4	BOX5	BOX6	CONT
CONT	BOX7	-etc-							

PANLST2	100	101	101	THRU	200				

Field	Contents
SETID	Set identification number. (Integer > 0) (See Remark 1)
MACROID	Identification number of a **CAERO7** or **BODY7** bulk data card to which the aerodynamic boxes listed in the set belong. (Integer ≥ 0) (See Remark 2)
BOXi	Identification number of aerodynamic boxes. (Integer > 0) (See Remark 3)

Remarks:

1. **PANLST2** is referred to by **SPLINEi**, **ATTACH**, **LOADMOD**, **CPFACT**, and/or **AESURFZ** bulk data card.

2. MACROID is used to define a spline plane for the infinite plate spline method (**SPLINE1** bulk data card).

3. Field number 5 can be a character string “THRU”. This implies that all aerodynamic boxes with identification numbers starting with BOX1 and ending with BOX3 are included in the list.

4. If **PANLST2** is not referred to by the **SPLINE1** bulk data card, multiple **PANLST2** bulk data cards with the same SETID are allowed. In this case, all aerodynamic boxes listed in all **PANLST2** with the same SETID are included in the set. For instance, the following two **PANLST2** bulk data cards with the same SETID = 10:

PANLST2	10		1	THRU	3				

PANLST2	10		104	25					

yield 5 aerodynamic boxes with identification numbers of 1, 2, 3, 25, and 104, respectively.

PANLST3 — Set of Aerodynamic Boxes

Description: Defines a set of aerodynamic boxes by the **LABEL** entry in **CAERO7** or **BODY7** bulk data cards.

Format and Example:

1	2	3	4	5	6	7	8	9	10
PANLST3	SETID	LABEL1	LABEL2	LABEL3	…	-etc-	…		

PANLST3	100	WING							

Field	Contents
SETID	Unique set identification number. (Integer > 0) (See Remark 1)
LABELi	Character string that matches the entry **LABEL** in the **CAERO7** or **BODY7** bulk data cards. (Character) (See Remark 2)

Remarks:

1. **PANLST3** is referred to by **SPLINEi**, **ATTACH**, **LOADMOD**, **CPFACT**, and/or **AESURFZ** bulk data card.

2. All aerodynamic boxes of the **CAERO7** or **BODY7** macroelement (with **LABEL** defined in the **CAERO7** or **BODY7** bulk data card) are included in the set.

   Note: If **PANLST3** is referred to by the **SPLINE1** bulk data card, only one LABEL entry is allowed.

PARAM

Values of Global Parameters

Description: Alter the values of global parameters used in the computation.

Format and Example:

1	2	3	4	5	6	7	8	9	10
PARAM	NAME	VALUE							

PARAM									

Field	Contents
NAME	Parameter name. (Character) (See Remark 1)
VALUE	Parameter value based on the parameter type. (Integer, Real, or Character)

Remarks:

1. The list of all parameters is shown in the following table.

Name	Parameter Type	Default	Description
ALFREFD	Real	1.0	Pitch amplitude in degrees as a small perturbation to extract the aerodynamic stability derivatives for trim analysis and as the oscillation amplitude to extract the frequency-domain aerodynamic forces for flutter analysis.
BODYCPX	Integer	0	If VALUE = 1, then apply a correction scheme to obtain the pressure coefficients from the rectangular box to the actual surface of the body.
CHIMERA	Real	1.0	A factor to increase the allocated memory used in overset hole-cutting computation.
DAMPING	Real	0.25	Damping ratio used in the composite sinusoidal excitation technique.
FOLLOWER	Integer	1	Follower force option. If the follower force option is enabled, the aerodynamic forces are applied on the instantaneous deformed shape of the aerodynamic surface mesh at every time step. 1 Follower force enabled 0 Follower force disabled
FREECOMP	Character	NO	For VALUE = “NO”, all **CAERO7**/**BODY7** must be included in the computational mesh via the **BLOCK**, **BLOCKT**, or **BLOCK1** bulk data cards, and a fatal error message is generated if this condition is not satisfied. For VALUE = “YES”, a warning message is generated. The computational mesh is generated without modeling of those **CAERO7**/**BODY7**. This computational mesh can lead to a flow solution except that the aerodynamic effect from those **CAERO7**/**BODY7** are excluded. This functionality can be used to find which component leads to convergence failure.

IBLANK	Integer	0	Generates a PLOT3D file to store the cell center points of the overset mesh including the IBLANK array. If VALUE is 0 or blank, the default file name is “CONNP3DMOD.DAT”. Otherwise, VALUE is the identification number of an **EXTFILE** bulk data card in which the file name is specified.
ISXK	Integer	1	1 Computes the frequency-domain generalized aerodynamic forces at k = 0.0 using the quasi- static computation. 2 Computes the frequency-domain generalized aerodynamic forces at k = 0.0 using the composite sinusoidal excitation.
LFAKERUN	Integer	0	MLOADS fake run option 1 Fake MLOADS run 0 Normal MLOADS run For MLOADS analysis, if the code stops accidentally and no output file is generated from that specified in **MLDPRNT** bulk data card, the fake run option can be used to recover the output file generated by the **MLDPRNT** bulk data card with the help of the generalized coordinate time history data stored in the file “QSHIST_0.dat”.
LYSLP	Integer	60	The Y-slope effect is ignored for horizontal surfaces with sweep angle larger than LYSLP degrees.
MAXTRIM	Integer	3	Maximum number of iterations in TRIM analysis.
N2CYC	Integer	2	Number of cycles for the sinusoidal excitation to compute the harmonic response for generating the frequency domain generalized aerodynamic forces in the FLUTTER module.
VROLLUP	Integer	1	For a surface that is physically connected to another surface but these two surfaces belong to different blocks of mesh, the mesh lines along the connection of these two surface may not be completely sealed. This could potentially induce flow "leakage problem" due to which the flow could pass through the connection between these two surfaces and create artificial vortex roll up. Specifying VROLLUP=1 can mitigate this flow leakage problem.
STRELAX	Real	0.5	Relaxation factor for static aeroelastic/trim analysis. During iteration between aerodynamic and structure coupling procedure, the modal coordinate solution (XS) is updated by the following equation to scale the current modal coordinate solution with that of the previous iteration (XN): $XS = STRELAX \times XN + (1.0 - STRELAX) \times XS$ Usually, if the static aeroelastic analysis fails to converge, using larger STRELAX (but cannot be greater or equal to 1.0) can improve the convergence.
STIFFER	Real	1.0	The square of STIFFER is multiplied to the generalized stiffness matrix to create a stiffer structure. Used only for the trim analysis.
TRMREDUC	Integer	3	The trim analysis first computes the mean flow solution of the trim variable THKCAM using TRMSTEP number of pseudo time steps, where TRMSTEP is specified in the **MKPARAM** bulk data card. Based on this mean flow solution as the initial flow condition, the trim analysis computes the aerodynamic stability derivatives of other trim variables such as angle of attack, control surface, etc. In these computations, the number of pseudo time steps is reduced to TRMSTEP/TRMREDUC.
TRIMCNV	Real	0.03	At each trim iteration, if all trim variables are smaller than TRIMCNV times their respective value at the first trim iteration, the

			trim iteration is converged.
TWOD	Character	NO	Model dimensions "NO" 3-D model "YES" 2-D model If TWOD is set to be "YES", the Euler solver performs a two-dimensional flow computation. For 2-D model input, the span-wise cuts have to be set to 2 for all CAERO7 components.
WTWALL	Character	NO	If VALUE = "YES", the far field boundary condition of the global block of mesh is treated as the wall boundary condition to simulate the wind tunnel wall effects.

PBODY7 Aerodynamic Body Wake/Inlet Property

Description: Defines the inlet aerodynamic boxes of an aerodynamic body; referenced by the **BODY7** bulk data card.

Format and Example:

1	2	3	4	5	6	7	8	9	10
PBODY7	ID								CONT
CONT	IDP1	FLOWRT1	IDP2	FLOWRT2	-etc-				

PBODY7	2								+BC
+BC	101	0.0	103	100.					

Field	Contents
ID	**PBODY7** identification number. (Integer > 0)
IDPi	Body box identification numbers where the flow is allowed to penetrate into the body; denoted as "inlet boxes". (Integer > 0)
FLOWRTi	Amount of flow in percentage of the flow contained in the stream tube in front of the inlet aerodynamic box which penetrates into to the body. FLOWRT*i* = 100.0 represents the entire flow-through condition. (Real, Default = 0.0)

PCHFILE

Imports a NASTRAN Punch File

Description: Imports a NASTRAN Punch output file that contains the modal values of the element forces, stresses, strains, etc.

Format and Example:

1	2	3	4	5	6	7	8	9	10
PCHFILE	IDPCH	FILENM							CONT
CONT	$ELLST_1$	$FIELD_1$	$LABEL_1$	$REMARK_1$	$ELLST_2$	$FIELD_2$	$LABEL_2$	$REMARK_2$	CONT
CONT	…	-etc-	…						

1	2	3	4	5	6	7	8	9	10
PCHFILE	100	NASTRAN.PCH							+P
+P	1000	2	ELFRC	BEAM	1001	3	ELFRC	QUAD4	+P
+P	2001	1	ELSTRN	TRIA3					

Field	Contents
IDPCH	Unique identification number. (Integer > 0) (See Remark 1)
FILENM	Character string specifying the name of the file that is generated by NASTRAN in the punch format. If the first character of FILENM is a dollar sign "$", the rest of the characters must be integers. This integer is the identification number of an **EXTFILE** bulk data card where the filename is specified. This feature allows for filenames up to 56 characters to be input. (Character) (See Remark 2)
$ELLST_i$	Identification number of a structural element whose modal values (forces, stresses, strains, etc.) is to be read from the punch file. (Integer > 0)
$FIELD_i$	The FIELD'th component of the modal values of the element is to be read from the punch file. (Integer > 0) (See Remark 3)
$LABEL_i$	Character string to define a label for describing these modal values. For output, this label consists of the first six characters of $LABEL_i$ and the last two characters are replaced by the integer defined by $FIELD_i$. For instance, for $LABEL_i$ = "ELFORCE" and $FIELD_i$ = 2, the output label becomes "ELFORC02".
$REMARK_i$	Not used.

Remarks:

1. The **PCHFILE** bulk data card imports the modal values of a structural parameter that can be element forces, stresses, strains, etc., from the NASTRAN punch file. These modal values are used

to compute the resulting structural parameter by the superposition of modal values and the generalized modal coordinates.

2. To generate a NASTRAN punch file, the user must specify a **NASTRAN** Case Control Command such as:

   FORCE (PUNCH) = ALL

   or

   STRESS (PUNCH) = n

   in the NASTRAN Case Control Section for a modal analysis where *n* is the identification number of the **SET NASTRAN** Case Control Command to list a set of element identification numbers for output. Note that the "= ALL" option is not recommended because it produces a large amount of data which could significantly increase the ZEUS computational time.

3. A typical NASTRAN punch file is shown as follows:

```
$TITLE   = AC02 MODAL ANALYSIS                                               1
$SUBTITLE= LANCZOS                                                           2
$LABEL   =                                                                   3
$ELEMENT FORCES                                                              4
$REAL OUTPUT                                                                 5
$SUBCASE ID =          1                                                     6
$ELEMENT TYPE =         34                                                   7
$EIGENVALUE = -0.2910688E-03  MODE =     1                                   8
     21000             1.268963E-06     -1.242571E-04      1.035389E-05      9
-CONT-                -1.243219E-02     -4.542464E-09      6.153965E-06     10
-CONT-                -1.456141E-04     -2.811976E-08                       11
     21020             1.036433E-05     -1.251435E-02      3.007680E-05     12
-CONT-                -2.564698E-02     -9.856237E-09      6.566319E-06     13
-CONT-                -2.457500E-04      5.680340E-08                       14

     ............................................................
     ............................................................

     27535            -2.303272E-04      5.782545E-04      1.750886E-07    258
-CONT-                 7.776543E-08     -4.610047E-07      1.156353E-06    259
-CONT-                -3.834066E-08      1.687854E-04                      260
$TITLE   = AC02 MODAL ANALYSIS                                             292
$SUBTITLE= LANCZOS                                                         293
$LABEL   =                                                                 294
$ELEMENT FORCES                                                            295
$REAL OUTPUT                                                               296
$SUBCASE ID =          1                                                   297
$ELEMENT TYPE =         34                                                 298
$EIGENVALUE = -0.2734564E-03  MODE =     2                                 299
     21000            -2.032439E-07     -2.753735E-05     -1.646139E-06    300
-CONT-                 2.614506E-01      7.214478E-10     -1.307391E-04    301
-CONT-                 1.409557E-05     -3.745799E-08                      302
     21020            -1.775028E-06      2.614317E-01     -4.958345E-06    303
-CONT-                 7.044209E-01      1.591658E-09     -2.214946E-04    304
-CONT-                 2.378039E-05     -1.662002E-07                      305
     ............................................................
     ............................................................
     27535            -6.439090E-04     -7.998943E-05     -2.486631E-07    549
-CONT-                 1.699664E-08     -1.287321E-06     -1.600129E-07    550
-CONT-                -2.293527E-07      4.723957E-04                      551
```

   In the example shown above, each element has 8 components of modal values. The entry $FIELD_i$ is used to select a particular component for output.

PLTAERO

ASCII Text File Generation for Plotting the Aerodynamic Model

Description: Defines the name of a data file on which the data for plotting the aerodynamic mesh is stored.

Format and Example:

1	2	3	4	5	6	7	8	9	10
PLTAERO	SETID	FEMGRID	OFFSET	FORM	FILENM		THICK	MESH	

PLTAERO	100	YES	100000	TECPLOT	AERO.PLT		YES	YES	

Field	Contents
SETID	Identification number (Integer > 0) (See Remark 1)
FEMGRID	Flag for the choice of inclusion of structural grid points as part of the plot file. FEMGRID = “YES” or “NO” (Character) (See Remark 2)
OFFSET	Active only if FEMGRID = “YES”. The identification numbers of all structural grid points are increased by OFFSET. (Integer ≥ 0, or Blank) (See Remark 3)
FORM	Must be FORM = “TECPLOT”. (Character)
FILENM	The name of the data file in which the data for plotting the aerodynamic model is stored. This file name is always in the upper case. In case the input file name is given in the lower case, the program converts it to the upper case. If the first character of FILENM is a dollar sign “$”, the rest of the characters must be integers. This integer is the identification number of an **EXTFILE** bulk data card where the filename is specified. This feature allows for filenames up to 56 characters to be input. (Character)
THICK	Character string either “YES” or “NO”. For THICK = “YES”, the upper and lower surfaces of the CAERO7 macroelement are included in the data file. Otherwise, the CAERO7 macroelement is considered as a flat plate for plotting (Character, Default = “YES”)
MESH	Character string either “YES” or “NO”. For MESH = “YES”, the entire mesh generated by the **BLOCK**, **BLOCKT** and **BLOCK1** bulk data cards including the surface mesh is stored in the data file (Character, Default = “NO”)

Remarks:

1. SETID is not referred to by other bulk data cards. The existence of each **PLTAERO** in the bulk data input “triggers” the generation of a data file for the purpose of plotting the aerodynamic model. SETID is used for error message output only.

2. Often times the user wishes to graphically display the aerodynamic and structural models together. Setting FEMGRID = “YES” writes the structural grid points in the aerodynamic coordinates along with the aerodynamic model data in the output data file. This option is useful to assist in setting up the spline input.

3. Since the structural model and the aerodynamic model may contain grids that have the same identification numbers, inclusion of the structural grids in the aerodynamic grids creates problems for plotting. OFFSET is used to circumvent this problem by offsetting all structural grid point identification numbers with the integer of OFFSET.

PLTCP

ASCII Text File Generation for Plotting the Steady/Unsteady Pressure Coefficients

Description: Defines the name of a data file in which the data for plotting the steady/frequency-domain unsteady aerodynamic pressure coefficient that is generated by an **MKAEROZ** bulk data card.

Format and Example:

1	2	3	4	5	6	7	8	9	10
PLTCP	SETID	SYM	IDMK	IK	MODE	FORM	FILENM		CONT
CONT	$IDWING_1$	$YWPOS_1$	$IDWING_2$	$YWPOS_2$	…	etc.	…		

PLTCP	100	ANTI	10	-1	STEADY		PLOTCP.DAT		+P
+P	200	10.0	200	30.0	500	50.0			

Field	Contents
SETID	Identification number. (Integer > 0) (See Remark 1)
SYM	Symmetry condition of the configuration for which the steady/unsteady aerodynamics are generated by the **MKAEROZ** bulk data card. (Character) SYM = “SYM” for symmetric condition SYM = “ANTI” for anti-symmetric condition SYM = “ASYM” for asymmetric condition
IDMK	Identification number of the **MKAEROZ** bulk data card whose generated steady/unsteady aerodynamic pressures will be output to file FILENM for plotting. (Integer > 0) (See Remark 2).
IK	For output of frequency-domain unsteady aerodynamic pressures, IK is used to specify the index of the reduced frequency in the **MKAEROZ** bulk data card with identification number = IDMK. The frequency-domain unsteady aerodynamic pressures associated with this reduced frequency will be output to file FILENM for plotting in either 3D contour format or in 2D x-y curve format based on the sign of entry MODE (i.e., MODE>0 = output in 3D contour format and MODE<0 = output in 2D x-y curve format) For output of steady aerodynamic pressures, IK can be arbitrary integer whose sign determines whether the data file is stored in 3D contour format (IK>0) or 2D x-y curve format (IK<0). (Integer) (See Remark 3)
MODE	For output of frequency-domain unsteady aerodynamic pressures, MODE represents the index of the structural mode whose associated unsteady pressures will be exported to the

file FILENM.

For output of <u>steady aerodynamic pressures</u>, MODE is a character string that must be “STEADY”.

(Integer or character=“STEADY”) (See Remark 3)

FORM — The 3D unsteady aerodynamic pressure contour data file can be generated for the following formats: TECPLOT, PATRAN, IDEAS, FEMAP, ANSYS, NASTRAN, ESA, and NASTL. PATRAN requires that the aerodynamic model be stored in a neutral file whose name is internally assigned as "AEROGEOM.PAT". For steady pressure output or unsteady pressure output in 2D x-y curves, FORM="TECPLOT" is the only available option.

(Character, Default = “TECPLOT”)

FILENM — The name of a data file in which the data for plotting the aerodynamic pressures is stored. This file name is always in the upper case. In case the input file name is given in the lower case, the program converts it to the upper case. If the first character of FILENM is a dollar sign “$”, the rest of the characters must be integers. This integer is the identification number of an **EXTFILE** bulk data card where the filename is specified. This feature allows for filenames up to 56 characters to be input. (Character)

IDWING$_i$ — *Active only if IK < 0 for steady pressure output or MODE < 0 for unsteady pressure output.*

IDWING$_i$ is the identification number of **CAERO7** bulk data card on which the pressure coefficients of specified span locations are stored on FILENM. (Integer ≥ 0)

YWPOS$_i$ — *Active only if IK < 0 for steady pressure output or MODE < 0 for unsteady pressure output.*

The spanwise location in percentage of the wingspan measured from the root of the **CAERO7** bulk data card with identification number = IDWINGi along which the pressure coefficients are stored on FILENM. (0.0≤REAL≤10 0.0)

<u>Remarks:</u>

1. The purpose of the **PLTCP** bulk data card is for the plotting of steady or frequency-domain unsteady pressure on the aerodynamic surface mesh. It generates a data file that stores the pressure coefficients due to:

   - Frequency-domain pressure distribution generated by a **FLUTTER** or **GENGAF** Case Control Command.
   - Steady pressure distribution on rigid aircraft computed by the steady aerodynamic analysis (that is performed prior to the unsteady computation due to the **FLUTTER/GENGAF** Case Control Command).
   - Steady pressure distribution on the flexible aircraft computed by the static aeroelastic analysis that is performed prior to the unsteady aerodynamic analysis due to the **MLOADS**, **GLOADS**, **ELOADS**, and **NLFLTR** Case Control Command.

2. The **MKAEROZ** bulk data card, referred to by a **FLUTTER**, **MLOADS**, **GLOADS**, **ELOADS**, or **NLFLTR** bulk data card, defines the flow condition at which the pressure distribution is stored in the data file (FILENM).

3. Both steady and unsteady aerodynamic pressure coefficient can be exported to file FILENM in either 3D contour format or 2D x-y curve format.

   For both cases, negative sign means the pressure coefficients along the specified wing spanwise positions are exported in x-y curve format, where x (0<x<1.0) is the x-location of the aerodynamic surface panel normalized by the local wing chord length.

   Each CAERO7 with any number of pairs of IDWINGi / YWPOSi specified will result in one and only one ZONE in the TECPLOT file.

PLTFLUT

ASCII Text File Generation for Plotting the Flutter or ASE Mode

Description: Defines the name of a data file in which the data for plotting the flutter mode represented by the aerodynamic model are stored.

Format and Example:

1	2	3	4	5	6	7	8	9	10
PLTFLUT	SETID	IDFLUT	MODE	NTIME	MAXDISP	FORM	FILENM		CONT
CONT	AERONM								

PLTFLUT	100	10	1	10	0.3	FEMAP	FLUTMODE.NEU		

Field	Contents
SETID	Identification number. (Integer > 0) (See Remark 1)
IDFLUT	Identification number that matches a **FLUTTER** Case Control Command for which the computed flutter mode is stored in the data file FILENM for graphical display. (Integer > 0) (See Remark 2)
MODE	Index of the flutter modes. The first flutter crossing is denoted as MODE = 1, the second flutter crossing as MODE = 2, and so on. (Integer > 0) (See Remark 3)
NTIME	Number of deformed aerodynamic models within one cycle of oscillation that are generated for the animation of the flutter mode. (Integer > 0, Default = 1) (See Remark 4)
MAXDISP	A fraction of the reference chord defined by the REFC entry in the **AEROZ** bulk data card to define the maximum displacement of the flutter mode. (Real > 0.0, Default = 1.0) (See Remark 5)
FORM	FORM = “TECPLOT” for generating the TECPLOT file. FORM = “PATRAN” for generating the PATRAN neutral/results file. FORM = “IDEAS” for generating an I-DEAS universal file. FORM = “FEMAP” for generating a FEMAP neutral file. FORM = “ANSYS” for generating an ANSYS supported neutral file. FORM = “NASTRAN” for generating a NASTRAN bulk data deck. FORM = “NASTL” for generating a NASTRAN bulk data deck with **GRID** entries in large field format. (i.e., allows for higher degree of numerical accuracy over the FORM=”NASTRAN” option) (Character, Default = “TECPLOT”) (See Remark 6)
FILENM	The name of the data file in which the data for plotting the flutter mode is stored. This file name is always in the upper case. In case the input file name is given in the lower case, the program converts it to the upper case. If the first character of FILENM is a

dollar sign "$", the rest of the characters must be integers. This integer is the identification number of an **EXTFILE** bulk data card where the filename is specified. This feature allows for filenames up to 56 characters to be input. (Character)

AERONM The name of a data file in which the aerodynamic model is stored in a PATRAN neutral file. ONLY USED IF FORM="PATRAN". If the first character of AERONM is a dollar sign "$", the rest of the characters must be integers. This integer is the identification number of an **EXTFILE** bulk data card where the filename is specified. This feature allows for filenames up to 56 characters to be input. (Character, Default = "AEROGEOM.PAT") (See Remark 7)

Remarks

1. SETID is not referred to by other bulk data cards. The existence of each **PLTFLUT** in the bulk data input "triggers" the generation of a data file for the purpose of plotting the flutter mode on the aerodynamic model. SETID is used for error message output only.

2. The flutter mode is defined as the structural mode whose damping crosses the (velocity/VREF) axis with zero damping. It is generated by the **FLUTTER** Case Control Command with the identification number = IDFLUT.

3. Since each flutter analysis (or ASE analysis) may contain many flutter modes that have zero damping, the selection of the flutter modes is dependent on MODE.

4. The flutter mode is represented by the deformed aerodynamic model. Since the flutter mode does not have a nodal line, it is necessary to generate many deformed aerodynamic models for animating the flutter mode. The magnitude of the deformation of each aerodynamic model is associated with $\sin(\omega_f \times t_i + \phi)$, where ω_f is the flutter frequency, t_i is the time of the i^{th} deformed aerodynamic model and ϕ is the phase angle. t_i is computed by the following equation:

   $$t_i = (i \times 2\pi / \omega_f) \quad \text{where } i=0, 1, \ldots \text{NTIME}$$

   Multiple output of the deformed aerodynamic model at each time step is only employed in the TECPLOT, IDEAS, and NASTRAN output formats. Animations in PATRAN, FEMAP, and ANSYS are supported directly by these software.

5. MAXDISP × REFC will be the maximum displacement of the flutter mode.

6. The format of the data file is defined by the entry FORM. The flutter (or ASE) modal data are added to the x, y, and z values of the aerodynamic grids to create deformed aerodynamic models. Using the TECPLOT, IDEAS, or NASTRAN software (depends on FORM = "TECPLOT", = "IDEAS", etc.) to display one deformed aerodynamic model at a time can create the animation effect of the flutter mode. For the IDEAS universal file output, data sets 781 and 780 are used for displaying the aerodynamic grids and boxes, respectively. A data set 55 is output at each time step containing the six-degree-of-freedom motion for all aerodynamic grids. For the FEMAP neutral file format, Data Blocks 403 and 404 are used for displaying the aerodynamic grids and boxes, respectively. Data Block 451 is used for displaying the flutter mode (TOTAL Translation), X-axis translation (T1), Y-axis translation (T2), and Z-axis translation (T3). The flutter mode shape can either be statically deformed or animated. The ANSYS output is a FEMAP neutral file that can be read in by an ANSYS neutral file translator developed by PADT Inc.

7. PATRAN requires that the aerodynamic model be stored in a neutral file and that analysis results be stored in a results file. Therefore, the AERONM entry is used to assign a name for a neutral file that contains the aerodynamic model, while the FILENM entry specifies a file that will contain the displacement results. For more details, please see section 7.3, PATRAN Compatible Output.

PLTMODE

ASCII Text File Generation for Plotting the Interpolated Structural Mode on Aerodynamic Model

Description: Defines the name of a data file in which the data for plotting the interpolated structural mode on the aerodynamic model are stored.

Format and Example:

1	2	3	4	5	6	7	8	9	10
PLTMODE	SETID	SYM	MODE	TYPE	MAXDISP	FORM	FILENM		CONT
CONT	AERONM								

PLTMODE	100	SYM	10		0.2	PATRAN	PLOTMODE.DAT		
	AEROMODE.PAT								

Field	Contents
SETID	Identification number. (Integer > 0) (See Remark 1)
SYM	Symmetry condition of the structural modes corresponding to the BOUNDARY entry in the **'ASSIGN FEM ='** Executive Control Statement. (Character) SYM = "SYM" for symmetric condition SYM = "ANTI" for anti-symmetric condition SYM = "ASYM" for asymmetric condition
MODE	The absolute value of MODE represents the index of the structural modes. (Integer) (See Remark 2)
TYPE	Not used.
MAXDISP	A fraction of the reference chord defined by the REFC entry in the **AEROZ** bulk data card to define the maximum displacement of the mode. (Real > 0.0, Default = 1.0) (See Remark 3)
FORM	Must be FORM = "TECPLOT"
FILENM	The name of the data file in which the data for plotting the interpolated structural mode is stored. This file name is always in the upper case. In case the input file name is given in the lower case, the program converts it to the upper case. If the first character of FILENM is a dollar sign "$", the rest of the characters must be integers. This integer is the identification number of an **EXTFILE** bulk data card where the filename is specified. This feature allows for filenames up to 56 characters to be input. (Character)
AERONM	Not used.

Remarks:

1. SETID is not referred to by other bulk data cards. The existence of each **PLTMODE** in the bulk data input “triggers” the generation of a data file for the purpose of plotting the interpolated structural mode on the aerodynamic model. SETID is used for error message output only. **PLTMODE** generates a data file that contains one interpolated structural mode with index = MODE. This structural mode is defined in the ′**ASSIGN FEM =**′ Executive Control Statement with BOUNDARY = “SYM”. The interpolation of structural modes from the structural grid points to the aerodynamic model is performed by the SPLINE module. Graphical display of the interpolated mode is useful to detect any error in the spline input.

2. If MODE is a positive integer, the interpolated structural mode is plotted on the surface grid points generated by the **CAERO7** and **BODY7** bulk data cards. If MODE is a negative integer, the surface mesh generated by **BLOCK**, **BLOCKT** and **BLOCK1** bulk data card will be plotted.

3. Since the structural mode is the eigenvector obtained by the structural analysis, the magnitude of the mode may not be of the same order as the size of the aerodynamic model. To circumvent this problem, it is recommended to define the maximum displacement of the mode by MAXDISP × REFC.

PLTSLP

ASCII Text File Generation for Plotting the Slopes on the Computational Surface Mesh

Description: Defines the name of a data file that contains the slopes on the computational surface mesh.

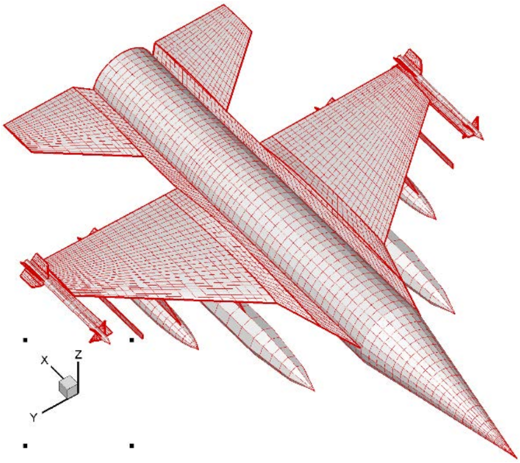
Format and Example:

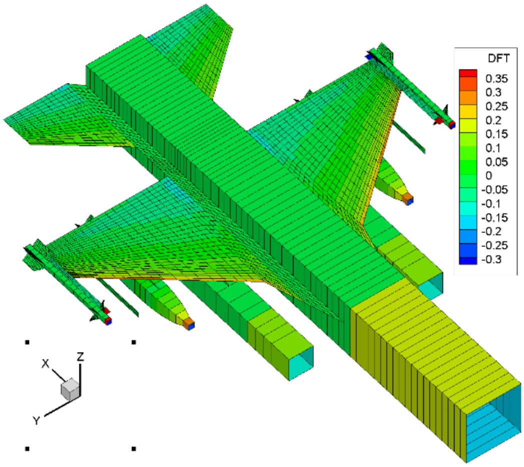
1	2	3	4	5	6	7	8	9	10
PLTSLP	SETID	SLPWMAX	SLPBMAX	FORM	FILEXY		FILE3D		
PLTSLP	100	1.5	0.5	TECPLOT	SLOPESXY.PLT		SLOPES3D.PLT		

Field	Contents
SETID	Identification number. (Integer > 0) (See Remark 1)
SLPWMAX	Threshold slope value for printing warning messages to the standard output file when the slope on a **CAERO7** element with respect to the flow direction exceeds SLPWMAX. (Real > 0.0, Default = 1.5) (See Remark 2)
SLPBMAX	Threshold d(slope)/dx value for printing warning messages to the standard output file when the change in slope along the **BODY7** element with respect to the flow direction exceeds SLPBMAX. (Real > 0.0, Default = 0.5) (See Remark 3)
FORM	Must be FORM = “TECPLOT”
FILEXY	The name of the data file in which the slopes of the aerodynamic boxes are represented in an XY line style plot. This file name is always in upper case. In case the input file name is given in lower case, the program converts it to upper case. If the first character of FILEXY is a dollar sign “$”, the rest of the characters must be integers. This integer is the identification number of an **EXTFILE** bulk data card where the filename is specified. This feature allows for filenames up to 56 characters to be input. (Character, Default = Blank)
FILE3D	The name of the data file in which the slopes of aerodynamic boxes are plotted on the surface mesh as a contour plot. This file name is always in upper case. In case the input file name is given in lower case, the program converts it to upper case. If the first character of FILE3D is a dollar sign “$”, the rest of the characters must be integers. This integer is the identification number of an **EXTFILE** bulk data card where the filename is specified. This feature allows for filenames up to 56 characters to be input. (Character, Default = Blank)

Remarks:

1. SETID is not referred to by any other bulk data card. The existence of a **PLTSLP** in the bulk data input triggers the generation of a data file for the purpose of plotting the slopes on the computational surface mesh. SETID is used for error message output only. Only one **PLTSLP** bulk data card is allowed.
   The transpiration boundary condition of the Euler solver only requires the slope on the computational surface mesh, which is interpolated from the surface panel model (i.e. **CAERO7** and **BODY7** macroelements). The **PLTSLP** bulk data card can assist the user to ensure the correctness of the surface panel model, as well as the interpolated slope. If the program encounters a convergence problem, it is highly recommended to use the **PLTSLP** bulk data card for verifying the correct slopes from the interpolation.
   Shown in the left figure below is the surface panel model of which the fuselage and stores are modeled by **BODY7** macroelements and all lifting surfaces are modeled by **CAERO7** macroelements. In the computational surface mesh shown in the right figure below, the **BODY7** macroelements are modeled by square boxes with the slope distribution with respect to the x-axis represented by a color contour. It is of importance to ensure the smoothness of the slope distribution because that is involved in the transpiration boundary condition.
   See Section 8.5 for details of the output file from the **PLTSLP** bulk data card.





2. For calculated slopes greater than SLPWMAX on **CAERO7** boxes, a warning message is printed to the standard output file. Slopes that are too large on **CAERO7** boxes may prevent the solution from converging. In this case, it is recommended to modify the camber and/or thickness of the airfoil to reduce the maximum slope.

3. For calculated differences of slopes of adjacent boxes greater than SLPBMAX on **BODY7** boxes, a warning message is printed to the standard output file. Large jumps in the slope along a **BODY7** element may prevent the solution from converging and may cause very high and low Cp distributions. In this case, it is recommended to smooth out the geometry of the surface mesh, or, in the case of engine inlet boxes, to specify the flow through condition using the **PBODY7** bulk data card.

PLTSURF

ASCII Text File Generation for Plotting the Aerodynamic Control Surface

Description: Defines the name of a data file in which the data for plotting the deflected aerodynamic control surface on the aerodynamic model are stored.

Format and Example:

1	2	3	4	5	6	7	8	9	10
PLTSURF	SETID	LABEL	MAXDISP	FORM	FILENM		AERONM		
PLTSURF	100	RUDDER	2.0	PATRAN	PLOT.PLT		AEROMODE.PLT		

Field	Contents
SETID	Identification number. (Integer > 0) (See Remark 1)
LABEL	Character string that matches the LABEL entry of an **AESURFZ** bulk data card. (Character) (See Remark 2)
MAXDISP	A physical displacement in the model units (i.e., FMLUNIT entry specified on the **AEROZ** card) at the furthest distance of the control surface from the hinge line. Used only for visualization. (Real, Default = 1.0)
FORM	FORM = “TECPLOT” for generating the TECPLOT file. FORM = “PATRAN” for generating the PATRAN neutral/results file. FORM = “IDEAS” for generating an I-DEAS universal file. FORM = “FEMAP” for generating a FEMAP neutral file. FORM = “ANSYS” for generating an ANSYS supported neutral file. FORM = “NASTRAN” for generating a NASTRAN bulk data deck. FORM = “NASTL” for generating a NASTRAN bulk data deck with **GRID** entries in large field format (i.e., allows for higher degree of numerical accuracy over the FORM=”NASTRAN” option). (Character, Default = “TECPLOT”). (See Remark 3)
FILENM	The name of the data file in which the data for plotting the deflected control surface is stored. This file name is always in the upper case. In case the input file name is given in the lower case, the program converts it to the upper case. If the first character of FILENM is a dollar sign “$”, the rest of the characters must be integers. This integer is the identification number of an **EXTFILE** bulk data card where the filename is specified. This feature allows for filenames up to 56 characters to be input. (Character)
AERONM	The name of a data file in which the aerodynamic model is stored in a PATRAN neutral file. ONLY USED IF FORM=”PATRAN”. If the first character of AERONM is a dollar sign “$”, the rest of the characters must be integers. This

integer is the identification number of an **EXTFILE** bulk data card where the filename is specified. This feature allows for filenames up to 56 characters to be input. (Character, Default = "AEROGEOM.PAT") (See Remark 4)

Remarks:

1. SETID is not referred to by other bulk data cards. The existence of each **PLTSURF** in the bulk data card "triggers" the generation of a data file for the purpose of plotting the deflected control surface on the aerodynamic model. SETID is used for error message output only.

2. The control surface is deflected about its hinge line with a unit deflection angle.

3. The format of the data file is defined by the FORM entry. The control surface deflection data are added to the x, y, and z values of the aerodynamic grids to create a deformed aerodynamic model. Using the TECPLOT or PATRAN software (depends on FORM = "TECPLOT" or = "PATRAN"), the deformed aerodynamic model can be displayed graphically. For I-DEAS universal file output, data sets 781 and 780 are used for displaying the aerodynamic grids and boxes, respectively. A data set 55 is used to output the six degree-of-freedom displacements at all aerodynamic grid. For FEMAP neutral file format, Data Blocks 403 and 404 are used for displaying the aerodynamic grids and boxes, respectively. Data Block 451 is used for displaying the deformed mode shape (TOTAL Translation), X-axis translation (T1), Y-axis translation (T2), and Z-axis translation (T3). The interpolated mode shape can either be statically deformed or animated. The ANSYS output is a FEMAP neutral file that can be read in by an ANSYS neutral file translator developed by PADT Inc.

4. PATRAN requires that the aerodynamic model be stored in a neutral file and that analysis results be stored in a results file. Therefore, the AERONM entry is used to assign a name for a neutral file that contains the aerodynamic model, while the FILENM entry specifies a file that will contain the displacement results.

PLTTIME

Generation of an ASCII Text File for the Post-Processing of the Transient Response Analysis

Description: Defines the name of a data file in which the transient deforming aerodynamic model or the transient loads by the transient response analysis is stored.

Format and Example:

1	2	3	4	5	6	7	8	9	10
PLTTIME	IDPLT	IDMLD	TS	TE	NDT	TYPE	FORM	SCALE	+P
+P	FILENM		AERONM						

PLTTIME	10	20	-2.0	1.0	10	ELASTIC	TECPLOT	1.0	+P
+P	TECPLOT.PLT								

Field	Contents
IDPLT	Identification number. (Integer > 0) (See Remark 1)
IDMLD	Identification number of an **MLOADS, ELOADS**, **GLOADS**, or **NLFLTR** Case Control Command. (Integer > 0) (See Remark 2)
TS	Starting time; TS must be greater than or equal to the TSTART entry in the **MLDTIME** bulk data card. (Real)
TE	Ending time; TE must be greater than TS and less than or equal to the TEND entry in the **MLDTIME** bulk data card. (Real)
NDT	Skip factor for output. The time step for output is NDT × OUTDT × DT, where OUTDT and DT are defined in the **MLDTIME** bulk data card. (Integer > 0, Default = 1)
TYPE	Character string (Character) TYPE = "FORCE": Stores the flight loads that are obtained by the mode displacement method in terms of NASTRAN **FORCE** and **MOMENT** bulk data cards at the structural finite element grid points in an ASCII text file. The user can insert this file into a NASTRAN model to obtain a detailed stress analysis by performing a static structural analysis. (See Remark 2) TYPE = "FORCESOF": Same as TYPE = "FORCE" but using the summation of force method.
TYPE	Warning: specifying TYPE = "FORCE" or "FORCESOF" triggers the program to compute the forces at all finite element grid points. For a large structural model, this may significantly increase the

		computational time.
	TYPE = "MANEUVER"	Stores the time history of the moving aerodynamic model on the ASCII file. This moving model includes the motions due to the forward velocity, control surface deflections and the structural deformation.
	TYPE = "ELASTIC"	Stores the time history of the deforming aerodynamic mode on the ASCII file. This deformation includes only the structural rigid body and elastic deformations. The motions due to the forward velocity are excluded. (See Remark 3)
	TYPE = "NORIGID"	Same as TYPE = "ELASTIC" but excludes the structural rigid body modes.
	TYPE = "UCP"	For FORM = "OUTPUT4" Stores the time history of the pressure coefficients on aerodynamic boxes in the OUTPUT4 format. The number of rows of the OUTPUT4 matrix is the number of aerodynamic boxes and the number of columns is the number of time steps defined by TS, TE, and NDT. Note that for TYPE = "UCP", the entry SCALE is not used. For FORM ≠ "OUTPUT4" Same as TYPE = "NORIGID" but the pressure coefficient contour is also displayed on the deforming aerodynamic model.
	TYPE = "UCF"	Same as "UCP" but for the skin friction coefficients.
FORM	Character string (Character)	
	FORM = "TECPLOT"	for generating the TECPLOT file
	FORM = "PATRAN"	for generating the PATRAN neutral/results file
	FORM = "IDEAS"	for generating an I-DEAS universal file
	FORM = "FEMAP"	for generating a FEMAP neutral file
	FORM = "ANSYS"	for generating an ANSYS supported neutral file
	FORM = "OUTPUT4"	for outputting the pressure coefficients in the OUTPUT4 format if TYPE = "UPC" or "UCF"
	FORM = "NASTRAN"	for generating a NASTRAN bulk data deck. This option will generate a file containing NASTRAN **FORCE** and **MOMENT** bulk data cards if TYPE = FORCE.
	FORM = "NASTL"	for generating a NASTRAN bulk data deck with **GRID** entries in large field format (i.e., allows for higher degree of numerical accuracy over the FORM = "NASTRAN" option)

Note: If TYPE = "FORCE", only FORM= "NASTRAN" and FORM= "IDEAS" are

	supported. (Default = "TECPLOT") (See Remark 3)
SCALE	Multiplier to the structural deformation. This is to amplify the structural deformation for the visualization of the deforming aerodynamic model. (Real, Default = 1.0)
FILENM	The name of the file that stores the generated data. This file name is always in the uppercase letters. In case the input file name is given in the lowercase letters, the program will convert it to the uppercase. If the first character of FILENM is a dollar sign "$", the rest of the characters must be integers. This integer is the identification number of an **EXTFILE** bulk data card where the filename is specified. This feature allows for filenames up to 56 characters to be input. (Character)
AERONM	The name of a data file in which the aerodynamic model is stored in a PATRAN neutral file. Only used if FORM="PATRAN". If the first character of AERONM is a dollar sign "$", the rest of the characters must be integers. This integer is the identification number of an **EXTFILE** bulk data card where the filename is specified. This feature allows for filenames up to 56 characters to be input. (Character, Default = "AEROGEOM.PAT") (See Remark 4)

Remarks:

1. IDPLT is not referred to by other bulk data card. The existence of each **PLTTIME** in the bulk data card "triggers" the generation of a data file for the post-processing of the transient response analysis.

2. The flight loads at the structural grid points for all time steps between TS and TE are written on the ASCII file. The LOADSET identification number of the NASTRAN **FORCE** and **MOMENT** bulk data cards starts from IDMLD (at t = TS) and incrementally increased by one for each time step. It is the user's choice to select the critical loads (its corresponding LOADSET identification number) for static structural analysis.

3. The generated ASCII file is for the animation of the motions of the aerodynamic model. For a large aerodynamic model it is recommended that large time step (large NDT) be used to minimize the size of the ASCII file.

4. PATRAN requires that the aerodynamic model be stored in a neutral file and that analysis results be stored in a results file. Therefore, the AERONM entry is used to assign a name for a neutral file that contains the aerodynamic model, while the FILENM entry specifies a file that will contain the displacement.

PLTTRIM

Generation of an ASCII Text File for the Post-Processing of the Static Aeroelastic/Trim Analysis

Description: Defines the name of a data file in which the aerodynamic pressure distribution, deformed aerodynamic model or flight loads generated by the static aeroelastic/trim analysis are stored.

Format and Example:

1	2	3	4	5	6	7	8	9	10
PLTTRIM	IDPLT	IDTRIM	FLEX	TYPE	FORM	FILENM		SCALE	CONT
CONT	AERONM								

PLTTRIM	100	10	FLEX	DEFORM	TECPLOT	PLTTRIM.DAT			

Field	Contents
IDPLT	Identification number. (Integer > 0) (See Remark 1)
IDTRIM	Identification number of a **TRIM** bulk data card. (Integer > 0) (See Remark 2)
FLEX	Character String must be “FLEX” (Character, Default = “FLEX”) FLEX = “FLEX” for the results of the flexible aircraft If the user wishes to obtain the rigid structure results, please specify the bulk data card “**PARAM STIFFER**” to increase the stiffness of the structure.
TYPE	Character string (Character) TYPE = “FORCE” Stores the flight loads (including the aerodynamic loads and the inertial loads) in terms of NASTRAN **FORCE** and **MOMENT** bulk data cards at the structural finite element grid point on the ASCII file. The user can insert this file into the NASTRAN model for detailed stress analysis by performing a static structural analysis. TYPE = “AERO” Same as TYPE = “FORCE” but only for the aerodynamic loads. TYPE = “INERTIAL” Same as TYPE = “FORCE” but only for the inertial loads. TYPE = “CP” Stores the distributed aerodynamic pressure distribution of the aerodynamic model on the file for graphic display. TYPE = “DEFORM” Stores the deformed aerodynamic model on the file for graphic display. If FLEX = ‘FLEX’, the deformation also

		includes the structural deflection.
	TYPE = "ELASTIC"	Stores the deformed aerodynamic model on the file for graphic display. The deformation includes only the structural deflection (no rigid body motion).
FORM	Character string (Character)	
	FORM = "TECPLOT"	for generating the TECPLOT file
	FORM = "PATRAN"	for generating the PATRAN neutral/results file
	FORM = "IDEAS"	for generating an I-DEAS universal file
	FORM = "FEMAP"	for generating a FEMAP neutral file
	FORM = "ANSYS"	for generating an ANSYS supported neutral file
	FORM = "ABAQUS"	for generating an ABAQUS supported file
	FORM = "NASTRAN"	for generating a NASTRAN bulk data deck. This option will generate a file containing NASTRAN **FORCE** and **MOMENT** bulk data cards if TYPE=FORCE.
	FORM = "NASTL"	for generating a NASTRAN bulk data deck with **GRID** entries in large field format (i.e., allows for higher degree of numerical accuracy over the FORM="NASTRAN" option)
	Note: If TYPE = "FORCE", "AERO" and "INERTIAL", only FORM=NASTRAN, FORM=IDEAS and FORM = ABAQUS are supported. (Default = "TECPLOT") (See Remark 3)	
FILENM	The name of the file that stores the generated data. This file name is always in the uppercase letters. In case the input file name is given in the lowercase letters, the program will convert it to the uppercase. If the first character of FILENM is a dollar sign "$", the rest of the characters must be integers. This integer is the identification number of an **EXTFILE** bulk data card where the filename is specified. This feature allows for filenames up to 56 characters to be input. (Character)	
SCALE	Amplification factor of the deformation. (Real, Default=1.0)	
AERONM	The name of a data file in which the aerodynamic model is stored in a PATRAN neutral file. Only used if FORM="PATRAN". If the first character of AERONM is a dollar sign "$", the rest of the characters must be integers. This integer is the identification number of an **EXTFILE** bulk data card where the filename is specified. This feature allows for filenames up to 56 characters to be input. (Character, Default = "AEROGEOM.DAT") (See Remark 4)	

Remarks:

1. IDPLT is not referred to by other bulk data card. The existence of each **PLTTRIM** in the bulk data input "triggers" the generation of a data file for the post-processing of the static aeroelastic/trim analysis.

2. If no **TRIM** bulk data card with IDTRIM existing in the Bulk Data Section, the ASCII file will not be generated. But this does not result a fatal error.

3. IDEAS output of **FORCE** and **MOMENT** are stored in universal dataset 782 for both Left-Hand-Side (LHS) and Right-Hand-Side (RHS) load sets. The ANSYS output is a FEMAP neutral file that can be read in by an ANSYS neutral file translator developed by PADT Inc.

4. PATRAN requires that the aerodynamic model be stored in a neutral file and that analysis results be stored in a results file. Therefore, the AERONM entry is used to assign a name for a neutral file that contains the aerodynamic model, while the FILENM entry specifies a file that will contain the displacement or steady pressure results (depending on whether TYPE=DEFORM or TYPE=CP, respectively). For more details, please see Section 7.2, PATRAN Compatible Output.

PLTVG

X-Y Plot File (ASCII text) Generation for Plotting Flutter Frequency and Damping Curves

Description: Defines a name of a data file in which the frequency and damping curve data are stored.

Format and Example:

1	2	3	4	5	6	7	8	9	10
PLTVG	SETID	IDFLUT	NMODE	XAXIS	FORM	FILENM		REFRHO	

PLTVG	100	10	5	V/VR		VG.DAT	1.14E-7	

Field	Contents
SETID	Identification number (Integer > 0) (See Remark 1)
IDFLUT	Identification number of the **FLUTTER** Case Control Command of which the flutter frequency and damping are stored in a data file for generation of an x-y plot (Integer > 0) (See Remark 2)
NMODE	Number of modes (Integer, Default = all modes)
XAXIS	Character string to represent the x-axis of the x-y data. = M Mach numbers = R densities = Q dynamic pressures = H altitudes = V/VR normalized speeds (V/V_{REF}) = V speeds = EQUV equivalent air speed defined as $\frac{V}{V_{REF}}\sqrt{\frac{\rho}{\rho_{REF}}}$, where ρ is the air density and ρ_{REF} is equal to the value specified in the RHOREF entry and V_{REF} is defined in the **FIXMACH**, **FIXMATM**, or **FIXMDEN** bulk data card.
FORM	FORM = "TABLE" for general x-y tabular output FORM = "IDEAS" for generating an I-DEAS universal file FORM = "FEMAP" for generating a FEMAP neutral file FORM = "ESA" for generating a PEGASUS readable file (Character, Default = "TABLE"). (See Remark 2)
FILENM	The name of the file on which the data for x-y plot of the flutter frequencies and damping are stored. This file name is always in the upper case. In case the input file name is given in the lower case, the program converts it to the upper case. If the first character of FILENM is a dollar sign "$", the rest of the characters must be integers.

This integer is the identification number of an **EXTFILE** bulk data card where the filename is specified. This feature allows for filenames up to 56 characters to be input. (Character)

REFRHO Reference density for computing the equivalent airspeed. Used only if XAXIS=EQUV. Note that the units of REFRHO must be in FTMUNIT/FTLUNIT^3, where FTMUNIT and FTLUNIT are defined in the **FIXMACH**, **FIXMATM**, or **FIXMDEN** bulk data card. (Real > 0.0, Default = 0.0023769 slug/ft^3)

Remarks:

1. SETID is not referred to by any other bulk data card. The existence of each **PLTVG** in the Bulk Data Section "triggers" the generation of a data file with name FILENM. SETID is used for error message output only.

2. The x-y data can be exported to EXCEL, Lotus or any other similar spreadsheet application for x-y plot. For I-DEAS universal file output, data set 58 is used to output the damping and frequency values as functions of Mach number, density, dynamic pressure, altitude, or speed (specified by the XAXIS entry). For FEMAP neutral file output, Data Block 451 is used to output damping and frequency values as functions of Mach number, density, dynamic pressure, altitude, or speed (specified by the XAXIS entry). X-axis values are stored as nodal x, y and z coordinates in order to generate XY vs. Position plots within FEMAP. The ESA format allows for viewing of the damping and frequency plots in the PEGASUS software.

3. A Windows plotting application called VGPlot.exe has been developed to allow for automated plotting of the PLTVG output. This program, along with its User's Manual, can be found in the ZEUS installation directory under the \miscel\\VGPlot folder. VGPlot can be used to generate damping and frequency curves versus the parameter you specify for the XAXIS entry. Note that the FORM must be set to TABLE (the default) in order to generate the appropriate output file format that is readable by VGPlot.

POD

Defines POD Parameters

Description: Defines the parameters for the generation of proper orthogonal decomposition (POD) modes.

Format and Example:

1	2	3	4	5	6	7	8	9	10
POD	IDPOD	NMODE	EPS	MAXIT	FILENM		FILFLW		CONT
CONT	FILECF								

POD	100	10	0.0001	500	POD.PLT		FILFLW.PLT		+POD
+POD	PODCF.PLT								

Field	Contents
IDPOD	Identification number (Integer > 0). (See Remark 1).
NMODE	Number of POD modes to be generated (Integer, Default = all modes). (See Remark 2).
EPS	A small tolerance to delete a column that has insignificant contribution to the POD modes. (Real, Default = 0.001).
MAXIT	Maximum iteration. (Integer > 0, Default = 300).
FILENM	Character string to define a file name on which the POD modes of the surface pressure are stored in the TECPLOT format. If FILENM is blank, then no such file will be generated. (Character)
FILFLW	Character string to define a file name on which the POD modes of the flowfield pressure are stored in the TECPLOT format. If FILFLW is blank, then no such file will be generated. (Character)
FILECF	Character string to define a file name on which the POD modes of the surface skin friction coefficients are stored in the TECPLOT format. If FILECF is blank or the **VISCOUS** bulk data card does not exist, then no such file will be generated. (Character)

Remarks:

1. The POD processes a matrix S to extract its POD coefficients and POD modes.

2. For a matrix that has NR row and NC columns, the POD method first solves the eigenvalues λ and eigenvectors ϕ such that

$$\left[S^{T}S-\lambda I\right][\phi]=0$$

Then POD method keeps NMODE of the eigenvectors whose corresponding eigenvalues λ are higher than other eigenvalues. Thus, the size of ϕ is NMODE by NCOL. Finally, the POD modes ψ are computed as:

$$\psi = S\phi$$

PZTMODE

Trim Analysis
Smart Structural Mode

Description: Defines a structural deformation due to smart structural actuation for static aeroelastic/trim analysis or the transient response analysis.

Format and Example:

1	2	3	4	5	6	7	8	9	10
PZTMODE	LABEL	TYPE	MNAME	ACTID					

PZTMODE	PZT1	SYM	INPUT4						

Field	Contents
LABEL	Unique alphanumeric string of up to eight characters used to identify the smart structural modes. (Character) (See Remark 2)
TYPE	Type of boundary condition. (Character) (See Remark 2) SYM symmetric ANTISYM anti-symmetric ASYM asymmetric
MNAME	Matrix name that is imported by the **"ASSIGN MATRIX="** Executive Control Command or **DMI** bulk data cards. (Character) (See Remark 3)
ACTID	Not used.

Remarks:

1. **PZTMODE** is equivalent to the **AESURFZ** bulk data card, except that **AESURFZ** provides the aerodynamic control forces due to control surface deflection, whereas **PZTMODE** gives the aerodynamic control forces due to the structural deformation. This structural deformation can be induced by a smart structural type of actuator.

2. Among all **PZTMODE**, **AESURFZ, AESLINK**, and **GRIDFRC** bulk data cards, no duplicated LABEL is allowed.

3. The matrix imported by the **"ASSIGN MATRIX="** Executive Control Command must have one column and G-set number (6 × number of structural grid points) of rows. The elements of the matrix are the structural deformation in six degrees of freedom at all structural finite element grids.

RESTART

Save or Retrieve Entire Flow Solution of a Transient Analysis

Description: Save or retrieve the entire flow solution at the last time step of a transient analysis

Format and Example:

1	2	3	4	5	6	7	8	9	10
RESTART	ID	NSTEP	SAVRST	FILNM					
RESTART	100	300	ACQU	FLOW.DAT					

Field	Contents
ID	Unique identification number that is referred to by a **MLOADS**, **ELOADS**, **GLOADS**, or **NLFLTR** bulk data card (Integer ≥ 0) (See Remark 1)
NSTEP	Incremental time steps at which the entire flow solution of the last time step is saved (used only if SAVRST = “SAVE” or “ACQUSAVE”). (Integer > 0, default = 100) (See Remark 2)
SAVRST	Character string “SAVE”, “AQUIRE”, or “ACQUSAVE” to save, retrieve or first retrieve then save, respectively, the entire flow solution. (Character) (See Remark 3)
FILENM	File name to specify the file name on which the entire flow solution is saved or retrieved. If the first character of FILENM starts with a dollar sign “$”, the rest of the characters must be integers. This integer is the identification number of an **EXTFILE** bulk data card where the filename is specified. This feature allows for filenames up to 56 characters to be input. (Character or Blank)

Remarks:

1. The **RESTART** bulk data card allows the user to save or retrieve the entire flow solution at the last time step of the transient analysis to a file. In the event of an abnormal termination during the transient analysis, or if the user wishes to extend the physical computational time from a previous transient analysis, the entire flow solution at the last time step saved on the file can be retrieved as the initial flow condition for a restarted transient analysis.

2. The entire flow solution is saved at the last time step of NSTEP incremental time steps. For instance, the entries TSTART = 0.0 seconds (starting time), TEND = 1.0 second (ending time) and DT = 0.001 seconds (the time step) are specified in the **MLDTIME** bulk data card. Therefore the total number time steps of the transient analysis is 1001. If NSTEP = 300 is specified, the entire flow solution is saved at time = 0.9 second. However, if the transient analysis is terminated abnormally at time = 0.7 second the entire flow solution is saved at time = 0.6 second.

3. For SAVRST = “ACQUIRE”, the starting time is the last saved time step by a previous transient analysis with the entry SAVRST = "SAVE" in respect to the starting time (the entry TSTART)

specified in the **MLDTIME** bulk data card. However, the ending time is the ending time specified by the entry TEND in the **MLDTIME** bulk data card. For instance, the last physical time of a previous transient analysis for which the entire flow solution is saved at 0.6 second and the new ending time is 2.0 seconds, the transient analysis of the restarted job is performed from 0.6 second to 2.0 seconds.

For SAVRST = “ACQUSAVE”, the entire flow solution is retrieved first by the restarted transient analysis from a previous transient analysis then the entire flow solution during the restarted transient analysis is continuously saved on file “FILENM” at every NSTEP time steps.

4. The following example shows how to use the **RESTART** bulk data card:

For cold start analysis:

MLDTIME	100	0.0	8.0	0.01					

RESTART	150	200	SAVE	FLOW.DAT				

If the transient analysis is terminated abnormally at 0.5 second and the user wishes to continue the analysis up to 8.0 second, the **MLDTIME** and **RESTART** bulk data cards should be:

MLDTIME	100	0.0	8.0	0.01					

RESTART	150	200	ACQU	FLOW.DAT				

If the transient analysis is terminated normally at a 8.0 second and the user wishes to extend the analysis up to 12.0 seconds, the **MLDTIME** and **RESTART** bulk data cards should be:

MLDTIME	100	0.0	12.0	0.01					

RESTART	150	200	ACQU	FLOW.DAT				

RETINAS

ROM Generation

Description: Computes the training data for the generation of a Reduced Order Model (ROM) of the unsteady aerodynamics.

Format and Example:

1	2	3	4	5	6	7	8	9	10
RETINAS	IDRETI	IDMLD	IDPOD	DAMPING	TYPE	IDFACT	IDGEOM	IDEXTS	CONT
CONT	$IDBLK_1$	$ISTART_1$	$IEND_1$	$JSTART_1$	$JEND_1$	$KSTART_1$	$KEND_1$	$IDEXTF_1$	CONT
CONT	$IDBLK_2$	$ISTART_2$	$IEND_2$	$JSTART_2$	$JEND_2$	$KSTART_2$	$KEND_2$	$IDEXTF_2$	CONT
CONT			...	-etc-	...				

RETINAS	100	200	0	-0.5	SET1	20	33	40	+R
+R	2100	10	30	6	6	30	70	100	+R
+R	200	20	40	10	10	20	50	101	

Field	Contents
IDRETI	Identification number. IDRETI is referred to by **RETINAS** Case Control Command (Integer $\neq 0$). Note that IDRETI can be a negative integer. In this case, a normal transient response analysis is performed. (See Remark 1).
IDMLD	If IDRETI > 0, IDMLD is the identification number of a **MLOADS** bulk data card. If IDRETI < 0, IDMLD is the identification number of either a **MLOADS** bulk data card or an **NLFLTR** bulk data card. (Integer $\neq 0$). Note that the ROM of all control surfaces defined in the **MLOADS** bulk data card via the **MLDCOMD** bulk data card will be generated. This ROM will generate the unsteady aerodynamic C_p due to control surface motion in the FRM analysis.
IDPOD	Identification number of a **POD** bulk data card. (Integer). (See Remark 2).
DAMPING	Damping coefficient for the Filtered Impulse Method (FIM). (Real < 0.0, Default = -0.1). (See Remark 3).
TYPE	Character string either “AEFACT” or “SET1” to indicate the type of entry IDFACT. (Character).
IDFACT	For TYPE = “AEFACT”, IDFACT is the identification number of an **AEFACT** bulk data card. (See Remark 4).

	For TYPE = "SET1", IDFACT is the identification number of a **SET1** bulk data card to list the identification numbers of a set of **TABELD1** bulk data cards. (Integer > 0). (See Remark 5).
IDGEOM	Identification number of an **EXTFILE** bulk data card to define a filename on which the geometric data is stored. (Integer > 0).
IDEXTS	Identification number of an **EXTFILE** bulk data card to define a filename on which the Proper Orthogonal Decomposition (POD) coefficients and POD modes of the surface unsteady pressure are stored. (Integer > 0). (See Remark 6).
IDBLK*i*	Identification number of a **BLOCK** bulk data card from which the computed training data at the flow field mesh is output. IDBLK*i* can be zero to deactivate this option. (Integer ≥ 0).
ISTART*i*, IEND*i*	The indices of the mesh along the X-axis between which the resulting data is output.
JSTART*i*, JEND*i*	Same as ISTART*i* and IEND*i* except for the mesh index is along the Y-axis.
KSTART*i*, KEND*i*	Same as ISTART*i* and IEND*i* except for the mesh index is along the Z-axis.
IDEXTF*i*	Identification number of an **EXTFILE** bulk data card to define a file name on which the POD coefficients and modes of the flow field pressures are stored. (Integer > 0). (See Remark 7).

Remarks:

1. The **RETINAS** bulk data card computes the training data that are exported to a MATLAB code to generate the ROM for the unsteady aerodynamics.

   If IDRETI is negative, a normal transient response analysis (called a *trial case* as opposed to a *training case*); either MLOADS or NLFLTR analysis is performed, according to the sign of IDMLD. If IDMLD is positive, then it is MLOADS analysis; if IDMLD is negative, then it is a NLFLTR analysis.

   The benefits of using negative IDRETI to perform normal transient analysis includes additional outputs, e.g., Tecplot format plotting files for flow solutions on the flow-field blocks.

2. The transient unsteady pressure coefficient will be processed by a Proper Orthogonal Decomposition (POD) method. Thus RETINAS outputs the POD coefficients and POD modes for the generation of ROM.

   Note that POD is only performed for *training case* (i.e., IDRETI is positive).

   A special case arises for a *trial case* (i.e., IDRETI is negative). If IDPOD is negative, only one run (time marching process) is conducted. In other words, a negative IDRETI and a negative IDPOD would be identical to a normal MLOADS or NLFLTR case. The underlying two transient runs are meant to take out the slowly-developing-steady component from the solution (the legacy of OVERCAP). In a *training case*, if IDPOD >, then the dynamic results of the subtraction of the two runs and; if IDPOD < 0, the static components are obtained from the final results of the "static" analysis stage before the transient analysis. (See Table 1.1)

Table 1.1 Corresponding Cases for Various Sign Combinations of IDRETI, IDMLD and IDPOD

IDRETI	IDMLD	IDPOD	DESCRIPTION
> 0	> 0 < 0	> 0 < 0	Training case. POD analysis performed.
< 0	> 0	< 0	Trial case. Identical to MLOADS.
		> 0	Trial case. Similar to MLOADS, but final solutions are the subtraction of two time marching processes to take out steady component.
	< 0	< 0	Trial case. Identical to NLFLTR
		> 0	Trial case. Similar to NLFLTR, but final solutions are the subtraction of two time marching processes to take out steady component.

3. The FIM signal is given by:

$$
\begin{aligned}
u(t) &= A_i e^{a_0(\omega_i t - \omega_i t_{0_i} - \pi)^2} \sin(\omega_i t - \omega_i t_{0_i}), \quad \text{when } t \ge t_{0_i} \quad \text{for } i = 1, 2, \cdots, N \\
&= 0, \qquad\qquad\qquad\qquad\qquad\qquad\qquad\quad \text{when } t < t_{0_i}
\end{aligned}
$$

where a_0 = DAMPING

ω_i is the natural frequency of the i-th mode,

t_{0_i} is the starting time of the i-th mode,

A_i is the amplitude of the i-th mode,

and N is the number of modes.

u(t) represents a staggered sequences FIM input of the flap defections and the generalized coordinates for the training signals.

4. The values listed in the **AEFACT** bulk data card must be 2 × N, where N is the number of modes. Thus, there is one pair of values for each mode. This first value of the pair is the amplitude (A_i shown in the above equation) and the second value is the starting time (t_{0_i}, shown in the above equation). Note that t_{0_i} can be negative. In this case, the best t_{0_i} is automatically determined by the program.

5. The number of identification numbers of the **TABELD1** bulk data card must be N, where N is the number of modes. Each **TABELD1** defines the time history of the input signal u(t). In this case, the FIM is not used; i.e. the equation shown above is replaced by a user defined input signal.

6. When **IDRETI** > 0 (a *training case*), the file linked to IDEXTS stores the matrices in OUTPUT4 and ASCII format. The matrices include (in order) the time information matrix (time step size, etc.), the generalized coordinates, the flap deflection, POD modes for surface C_p, POD coefficients for surface C_p, steady surface C_p, steady components of the generalized coordinates, and if viscous effects are considered, POD modes for surface C_f, POD coefficients for surface C_f, and steady surface C_f.

   When **IDRETI** < 0 (a *trial case*), the matrices stored in order are, the time information matrix (time step size, etc.), the generalized coordinates, the flap deflection, the surface C_p , and if applicable, the surface C_f.

7. When **IDRETI** > 0 (a *training case*), the file linked to IDEXTF*i* stores the matrices in OUTPUT4 and ASCII format. The matrices include (in order) the time information matrix (time step size, etc.), the generalized coordinates, the flap deflection, POD modes for C_p in flow-field block *i*, POD coefficients for C_p in flow-field block *i*, steady C_p in flow-field block *i*, and steady components of the generalized coordinates.

When **IDRETI** < 0 (a *trial case*), the matrices stored in order are, the time information matrix (time step size, etc.), the generalized coordinates, the flap deflection, the C_p solutions in flow-field block *i*.

8. One can use "PARAM, GUSTROM, *n*", in the input file to activate the discrete gust consideration. The integer *n* in the **PARAM** bulk data card refers to a **DGUST** bulk data card identification number. If IDRETI > 0 (a training case), all the generalized coordinates will be cleared to be zero regardless of what type of excitation for the generalized coordinates being specified. It's meant for gust ROM development. Note that, if one is interested in the GAF results. The time history of the generalized coordinates and normalized GAF are recorded in text file "GFHIST_1.DAT" and "GFHIST_2.DAT" (if applicable), where 1 and 2 refer to the first and second run, respectively.

   If IDRETI < 0 (a trial case), it corresponds to a MLOADS/NLFLTR analysis with gust consideration.

SEGMESH

Body Segment Definition

Description: Defines a grid system for a body segment; referenced by the **BODY7** bulk data card.

Format and Example:

1	2	3	4	5	6	7	8	9	10
SEGMESH	IDMESH	NAXIS	NRAD	YRSHRINK	YLSHRINK	ZTSHRINK	ZBSHRINK	REDUCE	CONT
CONT	ITYPE1	X1	CAM1	YR1	ZR1	IDY1	IDZ1	$SHRINK_1$	CONT
CONT	ITYPE2	X2	CAM2	YR2	ZR2	IDY2	IDZ2	$SHRINK_2$	CONT
CONT	ITYPE3	X3	CAM3	YR3	ZR3	IDY3	IDZ3	-etc-	

SEGMESH	2	3	6						+BC
+BC	1	0.0	0.0	0.0					+EF
+EF	1	1.0	0.0	0.5					+HI
+HI	3	2.0				103	104		

Field	Contents
IDMESH	Body segment mesh identification number. (Integer > 0) (See Remark 1)
NAXIS	Number of axial stations (i.e., divisions) of the segment. (Integer ≥ 2) (See Remark 2)
NRAD	Number of circumferential points of the segment. (Integer ≥ 3)
YRSHRINK YLSHRINK	Y value of the RHS (YRSHRINK) and LHS (YLSHRINK) of a BODY7 macroelement to define a region on the project X-Y plane. Within YLSHRINK and YRSHRINK, the spanwise X-lines are sheared to accommodate the variation of width of the BODY7 macroelement. If YRSHRINK = YLSHRINK = 0.0, this shearing process is not activated. In addition, YRSHRINK and YLSHRINK are not used if the **BODY7** bulk data card is referred to by a **BLOCK1** bulk data card. Note that both YRLSHRINK and YLSHRINK must be ≥ 0. (Real ≥ 0.0, Default = 0) (See Remark 3)
ZTSHRINK ZBSHRINK	Same as YRSHRINK and YLSHRINK except for the *z* value above (ZTSHRINK) and below (ZBSHRINK) of a BODY7 on the projected x-z plane. (Real ≥ 0.0, Default = 0)
REDUCE	A factor to define the rate of grid shearing. (Real ≥ 0.0, Default = 0.1)
ITYPE*i*	Type of input used to define the circumferential box cuts; = 1 body of revolution, = 2 elliptical body, = 3 arbitrary body. (Integer 1, 2 or 3) (See Remark 4)
X*i*	x-location of the axial station; X*i* must be in ascending order. (i.e., Xi+1 > Xi) (Real)

CAM*i*	Body camber at the X*i* axial station. (Real)
YR*i*	Body cross-sectional radius if ITYPEi = 1 or the semi-axis length of the elliptical body parallel to the y-axis if ITYPE*i* = 2. (Real)
ZR*i*	The semi-axis length of the elliptical body parallel to the z-axis. (Real)
IDY*i*	For ITYPE*i* =3, IDY*i* is the identification number of the **AEFACT** bulk data card that specifies NRAD number of the y coordinate locations of the circumferential points at the Xi axial station. For ITYPE*i* =1 or 2, - IDY*i* = 0, NRAD number of evenly distributed circumferential grid points are generated. - IDY*i* > 0, IDY*i* is the identification number of the **AEFACT** bulk data card that specifies NRAD number of circumferential angles in degrees which must start from 0.0 and end with 360.0 for a full-span model or 180.0 for a half-span model. (Integer ≥ 0) (See Remark 5)
IDZi	Identification number of **AEFACT** bulk data card that specifies NRAD number of the *z* coordinate locations of the circumferential points at the *Xi* axial station, used only if ITYPE*i* = 3. (Integer ≥ 0)
SHRINK$_i$	Character sting either "NONE", "ALL", or any combination of "YR", "YL", "ZT", and "ZB", for instance, SHRINK$_i$ = "YRYL", or "YRZB", etc. For SHRINK$_i$ = "NONE", the shearing on both x-y plane and x-z plane is not activated. For SHRINKi = "ALL", the shearing on both x-y plane and x-z plane is activated. For SHRINK$_i$ = "YR", only the spanwise x-line on the RHS of the BODY are sheared. For SHRINK$_i$ = "YL", only the spanwise x-lines on the LHS of the body are sheared. For SHRINK$_i$ = "ZT", only the horizontal x-lines below the body are sheared. (Character, Default = "NONE")

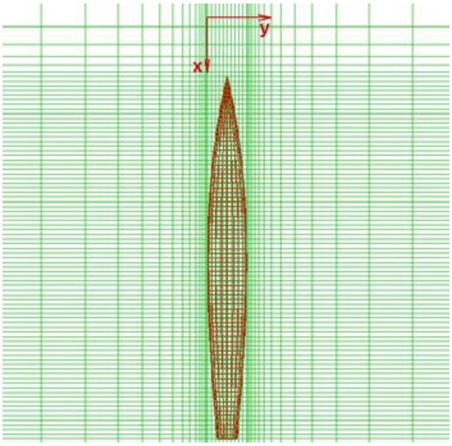
Remarks:

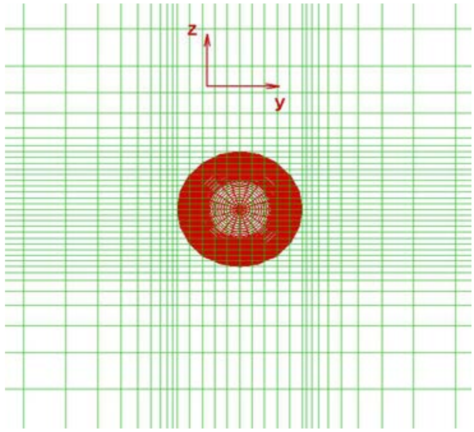
1. The **SEGMESH** bulk data card is referred to by a **BODY7** bulk data card to define the surface mesh of the BODY7 macroelement. All coordinates are in the local coordinates defined by the **ACOORD** bulk data card of the **BODY7** bulk data card.

2. The number of aerodynamic grids and boxes generated by each segment is NAXIS $\times$ NRAD and $(\text{NAXIS} - 1) \times (\text{NRAD} - 1)$ respectively; therefore, there are $\sum_{i=1}^{\text{NSEG}} \text{NAXIS}_i \times \text{NRAD}_i$ and $\sum_{i=1}^{\text{NSEG}} (\text{NAXIS}_i - 1) \times (\text{NRAD}_i - 1)$ number of grids and boxes, respectively, for each **BODY7** bulk data card.

3. The surface boundary of the flowfield mesh of the BODY7 macroelement where the boundary conditions of the Euler equations are applied is approximated by a prismatic rectangular box that embodies the actual surface mesh. The width and height of the rectangular box are the maximum width and height of the body, respectively, and are constant along the entire body (see the figure

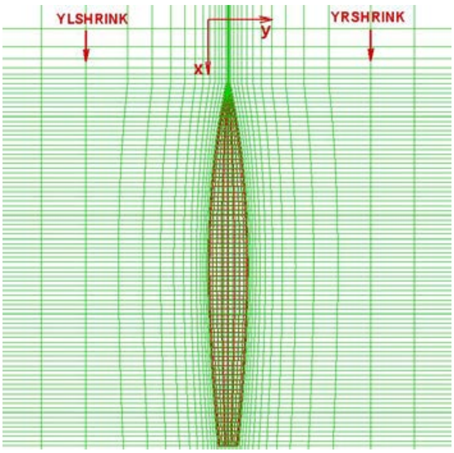
below). A correction parameter using a slender body theory to account for the spatial difference between the prismatic and actual surface is employed to correct the boundary condition applied on the prismatic box.

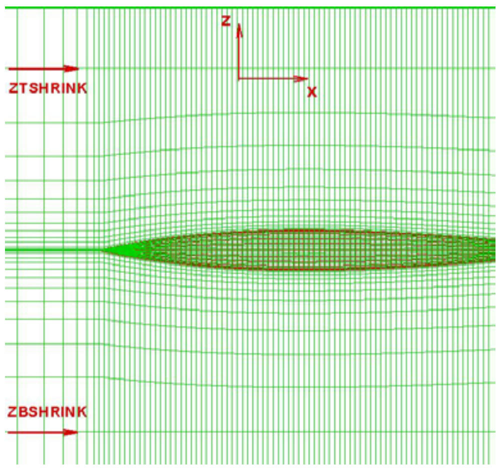




However, this restriction of constant cross section of the rectangular box can be relaxed to achieve a varying width and height rectangular box along the body by activating YRSHRINK, YLSHRINK, ZTSHRINK, and ZBSHRINK.

In the following figure, YRSHRINK and YLSHRINK define a region on the X-Y plane within which the spanwise X-lines are sheared to accommodate the variation of the body width at each axial station. The spanwise X-lines closest to the body surface are sheared exactly (DY) by following the width variation of the

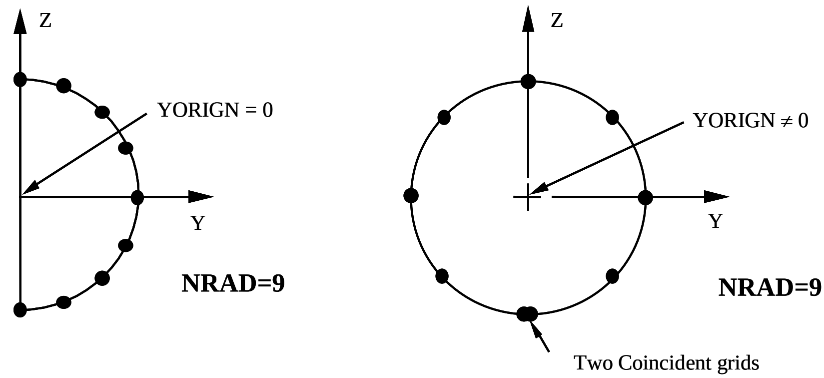




body. For those spanwise X-lines away from the body surface, the amount of shearing (shift of y position of the grid points on those spanwise X-lines) is reduced to DY/(1.0 + N × REDUCE), where REDUCE is specified in the entry REDUCE, and N is the numbering index of X-line away from the body surface. The same shearing procedure is also applied on those horizontal X-lines on the X-Z plane except within the region defined by ZTSHRINK and ZBSHRINK. The resulting flowfield mesh is a shearing Cartesian mesh on both X-Y plane and X-Z plane with a varying width and height rectangular box along the body to embody the actual body surface mesh.

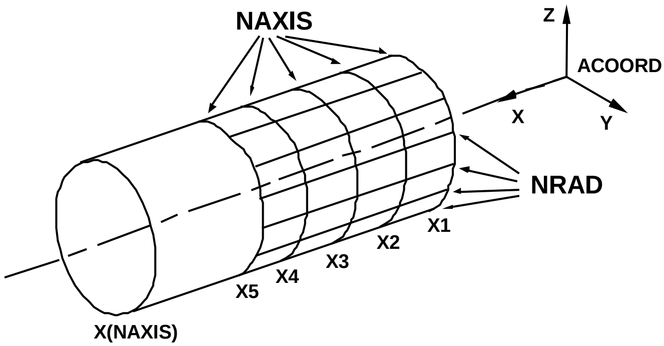
4. For a <u>body of revolution</u> or <u>elliptical body</u>, the circumferential points are distributed evenly for the body. If YORIGN defined in the **ACOORD** bulk data card in which the body refers to is zero and the XZSYM entry of the **AEROZ** bulk data card is YES, only half of the body (on the positive Y side) is generated.

Conversely, if YORIGN is not zero and the XZSYM entry of the **AEROZ** bulk data card is YES, the points must be distributed over the entire circumference of the body (e.g., an underwing store). For this case, the first and last points are coincident points. (See figures below) However, if the XZSYM entry of the **AEROZ** bulk data card is NO, then the entire body must be input (i.e., all circumferential points defined), regardless of the value of YORIGN.



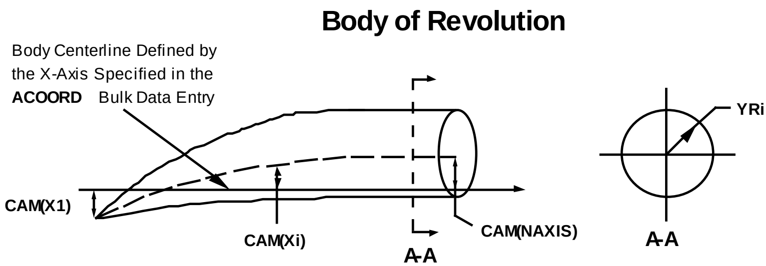
For an <u>arbitrary body,</u> the circumferential points must be entered in a counterclockwise direction (as viewed along the negative x-axis) looking at the y-z plane (in local body coordinates). If YORIGN defined in the **ACOORD** bulk data card to which the body refers is zero and the XZSYM entry of the **AEROZ** bulk data card is YES, only half of the body (on the positive y side) is generated. Conversely, if YORIGN is not zero and the XZSYM entry of the **AEROZ** bulk data card is YES, the points input must be distributed over the entire circumference of the body. For both of these cases, the y values listed in the **AEFACT** bulk data card must start with zero and end with zero. (See the figures above) However, if the XZSYM entry of the **AEROZ** bulk data card is NO, then the entire body must be input (i.e., all circumferential points defined), regardless of the value of YORIGN.

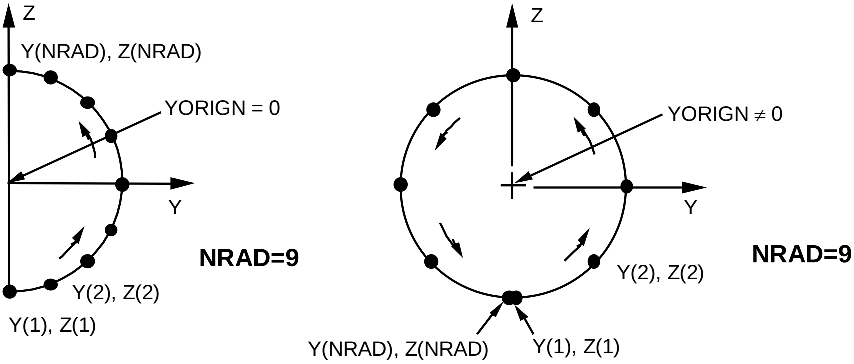
5. ITYPE*i* through IDZ*i* entries must be repeated for each axial station of the body segment (i.e., NAXIS times). Therefore, CAM*i*, YR*i*, ZR*i*, IDY*i*, and IDZ*i* represent the circumferential points at X*i*.



6. There are three methods to define the circumferential points at a given axial station:

   1) Body of Revolution (using ITYPE*i* = 1, and X*i*, CAM*i*, and YR*i* entries).
   2) Elliptical Body (using ITYPE*i* = 2, and X*i*, YR*i*, and ZR*i* entries).
   3) Arbitrary Body (using ITYPE*i* = 3, and X*i*, IDY*i*, and IDZ*i* entries).




Z
Y(NRAD), Z(NRAD)
YORIGN = 0
Y
NRAD=9
Y(2), Z(2)
Y(1), Z(1)
Z
YORIGN ≠ 0
Y
NRAD=9
Y(2), Z(2)
Y(NRAD), Z(NRAD)
Y(1), Z(1)

SET1 Set Definition for Aerodynamic Analysis

Description: Defines a set of integers by a list.

Format and Example:

1	2	3	4	5	6	7	8	9	10
SET1	SID	G1	G2	G3	G4	G5	G6	G7	CONT
CONT	G8	-etc-							

SET1	3	31	62	93	124	16	17	18	+BC
+BC	19								

Alternate Form:

1	2	3	4	5	6	7	8	9	10
SET1	SID	G1	THRU	G2					

Field	Contents
SID	Set identification number. (Integer > 0)
Gi	List of integers. (Integer $\neq$ 0)

Remarks:

1. When using the THRU option, all intermediate quantities are assumed to exist.

2. **SET1** is a general purpose bulk data card to define a set of integers. It is referred to by many other bulk data cards for defining the list of bulk data card identification numbers, indices of modes, aerodynamic box divisions, etc.

3. When using the THRU option, entries G4 through G7 must be blank. Thus, the following input set-up is not allowed.

SET1	3	5	THRU	10	110	120			

Instead, the following two **SET1** bulk data cards with the same SID = 3 are allowed and give the G_i as 5, 6, 7, 8, 9, 10, 110, and 120.

SET1	3	5	THRU	10					

SET1	3	110	120						

SET2 Grid Point List

Description: Defines a set of structural grid points in terms of aerodynamic macroelements.

Format and Example:

1	2	3	4	5	6	7	8	9	10
SET2	SID	SP1	SP2	CH1	CH2	ZMAX	ZMIN		

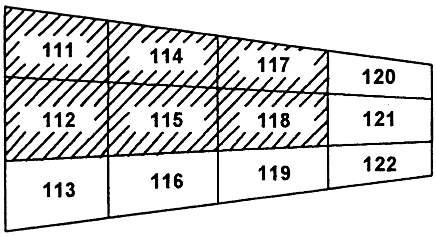
SET2	3	0.0	.73	0.0	.667	1.0	-3.51		

Field	Contents
SID	Set identification number (Integer > 0)
SP1, SP2	Lower and higher span division points defining the surface containing set (1.01 > Real > - 0.01).
CH1, CH2	Lower and higher chord division points defining the surface containing set (1.01 > Real > - 0.01).
ZMAX, ZMIN	Z-coordinates of top and bottom (using right-hand rule with the order of the corners as listed on a CAEROi entry) of the surface containing set (Real). Usually ZMAX > 0.0, ZMIN < 0.0.

Remarks:

1. These entries are referenced by the **SPLINE1** bulk data card.

2. Every grid point, within the defined surface and within the height range, will be in the set. For example:



The shaded area in the figure defines the surface containing the aerodynamic boxes for the sample data given above. Points exactly on the boundary may be missed, hence, to get all the grid points within the area of the macroelement, use SP1 = -0.01, SP2 = 1.01, etc.

3. A zero value for ZMAX and ZMIN implies infinity is to be used.

SETADD Set Definition

Description: Defines a set of integers as a union integer set defined on the **SET1** bulk data cards.

Format and Example:

1	2	3	4	5	6	7	8	9	10
SETADD	SETID	$SET1_1$	$SET1_2$	$SET1_3$	...	-etc-	...		

SETADD	10	101	200	300	400				

Alternate Form:

SETADD	10	$SET1_1$	THRU	$SET1_2$					

Field	Contents
SETID	Unique identification number. No duplicated identification number between **SETADD** and **SET1** bulk data card is allowed. (Integer > 0) (See Remark 1)
$SET1_i$	Identification number of a **SET1** bulk data card.

Remarks:

1. The **SETADD** bulk data card defines a list of integers by collecting all integers listed in the **SET1** bulk data cards with identification numbers being equal to $SET1_i$.

SPLINE0 Zero Displacement of Aerodynamic Boxes

Description: Imposes a zero displacement condition on aerodynamic boxes.

Format and Example:

1	2	3	4	5	6	7	8	9	10
SPLINE0	EID	MODEL	CP	SETK					

SPLINE0	100			20					

Field	Contents
EID	Unique element identification number. (Integer > 0) (See Remark 1)
MODEL	Not used.
CP	Not used.
SETK	Refers to a **PANLST1**, **PANLST2** or **PANLST3** bulk data card that lists the aerodynamic box identification numbers. (Integer > 0)

Remarks:

1. EID is only used for error output.

2. A typical case of imposing the zero displacement condition on aerodynamic boxes is the modeling of the wind tunnel wall by the **CAERO7** macroelement on which a zero-displacement condition is desired. Since the **CAERO7** macroelement representing the wind tunnel wall is not attached to the structural model, the zero displacement condition can be specified by using the **SPLINE0** bulk data card.

SPLINE1 Surface Spline Method

Description: Defines an infinite plate spline method for displacements and loads transferal between **CAERO7** macroelement and structural grid points.

Format and Example:

1	2	3	4	5	6	7	8	9	10
SPLINE1	EID	MODEL	CP	SETK	SETG	DZ	EPS		

SPLINE1	100			20	30	0.0			

Field	Contents
EID	Unique element identification number. (Integer > 0) (See Remark 1)
MODEL	Not used.
CP	Identification number of a **CORD2R** bulk data card that is defined in the structural basic coordinate system. The X-Y plane of this **CORD2R** bulk data card defines the spline plane. All structural grid points listed by the entry SETG are projected onto this spline plane for performing the infinite spline method. (Integer ≥ 0 or Blank) (See Remark 2)
SETK	The identification number of a **PANLST1**, **PANLST2** or **PANLST3** bulk data card that lists the aerodynamic box identification numbers. (Integer > 0)
SETG	The absolute value of SETG refers to the identification number of a **SETi/SETADD** bulk data card that lists the structural grid points to which the spline is attached. (Integer ≠ 0) (See Remark 3)
DZ	Linear attachment flexibility. (Real ≥ 0.0) (See Remark 4)
EPS	Multiplication factor to obtain a small tolerance to detect any duplicated location of structural grid points. The tolerance is computed by EPS × REFC, where REFC is the reference chord defined in the **AEROZ** bulk data card. (Real ≥ 0.0, Default = 0.01). (See Remark 5)

Remarks:

1. EID is only used for error output.

2. If no CP is specified, the plane defined by the macroelement specified in the **PANLSTi** bulk data card is used for the spline plane.

3. SETG can be a negative integer. In this case, the sign of the structure grid point ids listed in the **SETi/SETADD** is reversed.

4. The attachment flexibility (units of area) is used for smoothing the interpolation. If DZ = 0.0, the spline will pass through all deflected grid points. If DZ is much greater than the spline area, a least square plane fit will be applied. Intermediate values will provide smoothing.

5. If any two or more structural point locations projected on the spline plane are nearly the same, the spline matrix is singular. EPS is used to detect this condition.

SPLINE2 — Beam Spline Method

Description: Defines a beam spline method for the **BODY7** or **CAERO7** macroelement.

Format and Example:

1	2	3	4	5	6	7	8	9	10
SPLINE2	EID	MODEL	SETK	SETG	DZ	EPS	CID	CURV	

SPLINE2	100		10	20	0.0	0.02		0.0	

Field	Contents
EID	Unique element identification number (Integer > 0) (See Remark 1)
MODEL	Not used.
SETK	The identification number of a **PANLST1**, **PANLST2** or **PANLST3** bulk data card that lists the aerodynamic box identification numbers. (Integer > 0)
SETG	The absolute value of SETG refers to the identification number of a **SETi/SETADD** bulk data card that lists the structural grid points to which the spline is attached. (Integer ≠ 0) (See Remark 2)
DZ	Linear attachment flexibility. (Real ≥ 0.0)
EPS	Multiplication factor to obtain a small tolerance to detect any duplicated location of structural grid points. The tolerance is computed by EPS × REFC, where REFC is the reference chord defined in the **AEROZ** bulk data card. (Real ≥ 0.0, Default = 0.01)
CID	Identification number of a **CORD2R** bulk data card that is defined in the structural basic coordinate system whose y-axis defines the spline axis, i.e., the line of the beam. All structural grid points listed by the entry SET6 are projected onto this axis for performing the beam spline method. (Integer ≥ 0 or Blank; not used for **BODY7**) (See Remark 3)
CURV	Curvature effects of the torsion stiffness. (Real ≥ 0.0, Default = 1.0) (See Remark 4)

Remarks:

1. Unlike **SPLINE1** and **SPLINE3**, that require only the transitional degrees of freedom (d.o.f) of the structural grid, the beam spline method also requires the rotational d.o.f for both accurate displacement and slope spline at the aerodynamic boxes. Therefore, the user must ensure that the structural grid (defined by entry SETG) have no unwanted constraints at their rotational degrees of freedom.

   Warning: The beam spline method can accurately transfer the displacement from the structural grid to the aerodynamic grid. But when transferring the aerodynamic forces back the structured grid, it does not ensure the conservation of forces. Thus, if the user wishes to obtain the loads at the structural grid using the **PLTTRIM** or **PLTTIME** bulk data cards. **SPLINE2** is not recommended. The user can add additional grid points in the structural model and connect those grid points to the beam structure by rigid elements then uses **SPLINE1** or **SPLINE3** bulk data card for spline.

2. SETG can be a negative integer. In this case, the sign of the structure grid point ids listed in the **SETi**/**SETADD** is reversed.

3. If the macroelement specified in the **PANLSTi** bulk data card is a **CAERO7**, the spline axis is the y-axis of the coordinate system **CORD2R** with identification number = CID. In this case, the y-axis represents a line along which the original structural grid points are located. Note that the structure grid point locations are those in the structural finite element model before the ACSID and the FLIP entries of the **AEROZ** bulk data card are applied. If the macroelement is a **BODY7**, CID is not used and the spline axis is the x-axis of the **ACOORD** bulk data card associated with the **BODY7** macroelement.

4. Specifying CURV = 0.0 gives the agreement with the SPLINE2 of MSC.Nastran because that of MSC.Nastran does not include the curvature effect of the torsion stiffness of the beam.

SPLINE3 — 3D Spline Method

Description: Defines a 3-D spline for the **BODY7** and **CAERO7** macroelement.

Format and Example:

1	2	3	4	5	6	7	8	9	10
SPLINE3	EID	MODEL	CP	SETK	SETG	DZ	EPS		

SPLINE3	100			1	10	0.0			

Field	Contents
EID	Unique element identification number. (Integer > 0)
MODEL	Not used.
CP	Not used.
SETK	The identification number of a **PANLST1**, **PANLST2** or **PANLST3** bulk data card that lists the aerodynamic box identification numbers. (Integer > 0)
SETG	The absolute value of SETG refers to the identification number of a **SETi/SETADD** bulk data card that lists the structural grid points to which the spline is attached. (Integer ≠ 0) (See Remark 3)
DZ	Not used.
EPS	Multiplication factor to obtain a small tolerance to detect any duplicated location of structural grid points. The tolerance is computed by EPS × REFC, where REFC is the reference chord defined in the **AEROZ** bulk data card. (Real ≥ 0.0, Default = 0.01)

Remarks:

1. **SPLINE3** employs the Thin Plate Spline (TPS) method. Unlike the infinite plate spline method employed by the **SPLINE1** bulk data card, the **SPLINE3** does not require that a spline plane be defined. All structural grid points are located in 3-D space. Therefore, the TPS method can be considered as a 3D spline method.

2. Two restrictions are associated with the 3D spline method:
   (a) Similar to **SPLINE1**, no two or more structural points can be at the same location.
   (b) All of the structural points cannot be located in the same plane.
   EPS is the tolerance used to detect the above two conditions.

3. SETG can be a negative integer. In this case, the sign of the structure grid point ids listed in the **SETi/SETADD** is reversed.

SPLINEF

Spline Matrix for Force Mapping

Description: Generates the force spline matrix to map the aerodynamic forces at the aerodynamic grids to the structural grids by altering the **SPLINE1**, **SPLINE2** or **SPLINE3** bulk data card.

Format and Example:

1	2	3	4	5	6	7	8	9	10
SPLINEF	EID	IDSPLINE	SET1						

SPLINEF	100	200	300						

Field	Contents
EID	Identification number that is used only for error message output (Integer >0) (See Remark 1)
IDSPLINE	Identification number of a **SPLINE1**, **SPLINE2** or **SPLINE3** bulk data card, whose entry SETG is replaced by the SET1 entry of the **SPLINEF** bulk data card. (Integer >0) (See Remark 2)
SET1	Identification number of a **SET1** or **SETADD** bulk data card to list a set of identification numbers of structural grid points that are used to generate the force spline matrix. (Integer >0) (See Remark 3)

Remarks:

1. The **SPLINEF** bulk data card is optional. Its existence “triggers” the program to generate a different force spline matrix from the displacement spline matrix. There are two spline matrices generated by the spline module:

$$\{h\} = [UGTKG]^T \{x\}$$
$$\{F_s\} = [UGFRC]\{F_a\}$$

where *{x}* is the G-set displacement at the structural grid points
{h} is the k-set displacement at the aerodynamic boxes
$\{F_a\}$ is the aerodynamics forces at the aerodynamic boxes
$\{F_s\}$ is the G-set forces at the structural grid points
[UGTKG] is the displacement spline matrix
and *[UGFRC]* is the force spline matrix

If there is no **SPLINEF** bulk data card specified, then *[UGFRC] = [UGTKG]*

2. The spline module first generates the *[UGTKG]* matrix by processing all **ATTACH**, **SPLINE0**, **SPLINE1**, **SPLINE2**, and **SPLINE3** bulk data cards. Then the spline module processes the

SPLINEF bulk data cards to alter the **SPLINE1**, **SPLINE2** or **SPLINE3** bulk data cards by a new set of structural grid points involved in the force spline. The new set of spline bulk data cards along with all of the rest of the unaltered spline bulk data cards (not referred to by the **SPLINEF** bulk data card) are used to generate the *[UGFRC]* matrix.

3. To ensure a continuous displacement and slopes at the aerodynamic grid points by the displacement spline matrix, the generation of *[UGTKG]* matrix may need more structural grid points. However, to achieve a good force spline, it is recommended to select less structural grid points involved in the *[UGFRC]* matrix. This is because one aerodynamic box produces only one aerodynamic force. If there are more than one structural grid points located on one aerodynamic box, the *[UGFRC]* matrix needs to split one aerodynamic force at more than one structural grid points. This may result in an irregular distribution of the force distribution at the structural grid points. Note that based on the principle of virtual work, the conservation of the total force is ensured by the *[UGFRC]* matrix, but it may result a in a poor distribution of forces if the structural grid points involved in the force spline are not carefully selected.

SPLINEM Save or Retrieve the Spline Matrix

Description: Saves the spline matrix on an external file for the cold start job or retrieves the spline matrix from the external file for the restart job.

Format and Example:

1	2	3	4	5	6	7	8	9	10
SPLINEM	SAVE	FILENM							

SPLINEM	ACQU	spline.dat							

Field	Contents
SAVE	Character strings either “SAVE” or “ACQUIRE”. For SAVE = “SAVE”, save the spline matrix on the file “FILENM”. For SAVE = “ACQUIRE”, retrieve the spline matrix from the file “FILENM”. (Character) (See Remark 1)
FILENM	File name to specify the file name on which the spline matrix is saved or retrieved. If the first character of FILENM is a dollar sign “$”, the rest of the characters must be integers. This integer is the identification number of an **EXTFILE** bulk data card where the filename is specified. This feature allows for filenames up to 56 characters to be input. (Character) (See Remark 2)

Remarks:

1. The SPLINEM bulk data card is not referred to by any other bulk data card. Its existence in the input file “triggers” the program to save/retrieve the spline matrix. Computation of the spline matrix for a large number of FEM grid points could be time-consuming. The SPLINEM bulk data card can avoid the recomputation of the spline matrix if both the aerodynamic and the structural finite element grid points are unchanged.

2. Because the spline matrix is independent of Mach number, the spline matrix can be first saved in the cold start job and then retrieved for other Mach numbers in the restart job.

SPLNDOF Changes Spline d.o.f. of Structural Model

Description: Relates a dependent degree of freedom (d.o.f.) in the structural model to the other two independent degrees of freedom for spline.

Format and Example:

1	2	3	4	5	6	7	8	9	10
SPLNDOF	DEPGRD	DEPDOF	INDGRD1	INDDOF1	A1	INDGRD2	INDDOF2	A2	

SPLNDOF	1	5	1	3	0.1	2	3	-0.1	

Field	Contents
DEPGRD	Identification number of dependent structural grid point. (Integer > 0) (See Remark 1)
DEPDOF	The displacement component of the structural grid point DEPGRD at which the modal displacements are related to that of the other two independent degrees of freedom. (1 ≤ Integer ≤ 6) (See Remark 2)
INDGRD1	Identification number of the first independent structural grid point. (Integer > 0)
INDDOF1	The displacement component of the structural grid points INDGRD1. (1 ≤ Integer ≤ 6)
A1	A multiplication factor applied to the modal displacement at the d.o.f. of INDDOF1. (Real)
INDGRD2	Identification number of the second independent structural grid point. (Integer > 0)
INDDOF2	The displacement component of the structural grid points INDGRD2. (1 ≤ Integer ≤ 6)
A2	A multiplication factor applied to the modal displacement at the d.o.f. of INDDOF2. (Real) (See Remark 3)

Remarks:

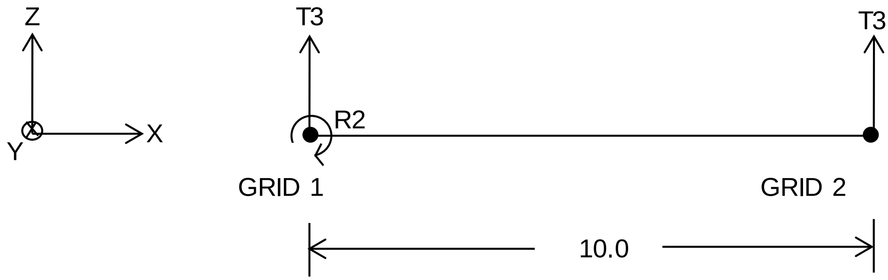
1. The **SPLINDOF** bulk data card is not referred to by other bulk data cards. Its existence "triggers" the program to change the spline matrix so that the structural d.o.f. involved in the spline matrix is related to the other two structural degrees of freedom. Note that multiple SPLNDOF bulk data cards can be specified.

2. DEPDOF, INPDOF1, and INPDOF2 are defined in the output displacement local coordinates.

3. An example to show the application of the **SPLNDOF** bulk data card is described as follows:
The following figure shows a beam element located along the x-axis and connected by two grid points with identification numbers being 1, and 2. The distance between these two grid points in 10.0. Supposedly, the displacement at the R2 degree of freedom (the rotation about the y-axis) of the grid point 1 is constrained in the structural analysis so that the modal displacement at R2 is zero. In this case, the beam spline method (the **SPLINE2** bulk data card) will give incorrect

splined mode shapes on the aerodynamic model because the beam spline method requires both translation d.o.f. (T3) and the rotation d.o.f. (R2). To circumvent this problem, the user can specify a **SPLNDOF** bulk data card to relate the modal displacement at R2 of the grid point, 1 to the modal displacement at T3 of the grid point 1 and 2 as:

$$(R2\,of\ grid\ point\,1) = \frac{(T3\,of\ grid\ point\,1) - (T3\,of\ grid\ point\,2)}{10.0}$$



The corresponding **SPLNDOF** bulk data card is:

SPLNDOF	1	5	1	3	0.1	2	3	-0.1	

STABDRV Aerodynamic Stability Derivatives

Description: Generates the aerodynamic stability derivatives using the linearized Euler solver.

Format and Example:

1	2	3	4	5	6	7	8	9	10
STABDRV	ID	MACH	IDSTFLW	GAMMA	EPSC	PRINT			CONT
CONT	CFL	NSTEP	NEWTN	NCYC	LVRSMOD	TVDCOEF	VIS2	VIS4	

STABDRV	100	0.9	100		0.00001	1			+BC
+BC	7.5				0.001		1.1		

Field	Contents
ID	Unique identification number that is referred to by the **STABDRV** case control command. (Integer > 0) (See Remark 1)
MACH	Mach number. (Real ≥ 0.0)
IDSTFLW	Identification number of an **EXTFILE** bulk data card in which the specified file name contains the steady flow solution. (Integer>0) (See Remark 2)
GAMMA	Specific heat ratio. (Real>0.0, Default=1.4)
EPSC	Convergence control for the linearized Euler run. (Real≠0.0, Default = 0.0001) (See Remark 3)
PRINT	CFD convergence history print flag. (Integer, Default = 2) PRINT = 0 No print. PRINT = 1 Print out the convergence history of the last EULER cycle of the last sub-iteration. PRINT = 2 Print out the full convergence history.
CFL	Courant-Friedrichs-Lewy (CFL) number for the EULER solver. (Real>0.0, or Integer > 0 Default = 7.0) (See Remark 4)
NSTEP	Number of time steps to compute the Linearized Euler solution. (Integer ≥ 0, Default = 300) (See Remark 5)
NEWTN	The number of Newton sub-iterations per time step. (Integer, Default = 4) (See Remark 5)
NCYC	The number of Euler cycles per Newton sub-iteration. (Integer, Default = 2) (See Remark 5)
LVRSMOO	Residual smoothing for the Euler solver. (Integer, Default = 1) (See Remark 5) LVRSMOO = 0 Turn off the residual smoothing option. LVRSMOO = 1 Turn on the residual smoothing option.
TVDCOEF	A parameter involved in the TVD form of pressure sensor of Jameson-Schmidt-Turkel (JST) scheme. It can be used to smooth the undesired zigzag pressure distribution that could occur on lower surface of the wing at high angle of attack flow condition (0.0 ≤ Real < 1.0, Default = 0.0) (See remark 6)

VIS2	Second order artificial dissipation parameter for the EULER solver. (Real > 0.0, or Integer > 0, Default = 1.0) (See Remark 7)
VIS4	Fourth order artificial dissipation parameter for the EULER solver. (Real > 0.0, or Integer > 0, Default = 1.0).

Remarks:

1. **STABDRV** bulk data card is referred to by the **STABDRV** case control command to compute the aerodynamic stability derivatives that are generated from the frequency-domain generalized aerodynamic forces of six internally created rigid-body modes at a small reduced frequency. These frequency-domain generalized aerodynamic forces are computed using the linearized Euler solver or the K-expansion method (**KEXPAND** bulk data card).
   A typical output of those stability derivative is shown below:

```
       * * * AERODYNAMIC STABILITY DERIVATIVES COMPUTED FROM THE GENERALIZED AERODYNAMIC FORCES OF 6 RIGID BODY MODES * * *

 REFERENCE AREA, CHORD (C), SPAN (B)=  1.0000E+00  2.1960E+01  1.0000E+00, MOMENT CENTER=  0.0000E+00  0.0000E+00  0.0000E+00

      ALL STABILITY DERIVATIVES ARE IN AERODYNAMIC COORDINATES AT MACH=   0.300
      I.E. SIGNS OF ROLL AND YAW RATES AS WELL AS ROLLING AND YAWING MOMENTS ARE OPPSITE FROM THE STABILITY AXIS.

      AERODYANIC MODEL IS A HALF SPAN MODEL BUT ALL STABILITY DERIVATIVES ARE OF THE FULL SPAN MODEL

                                CD           CY           CL           CR           CM           CN

U                        0.96364E+00  0.00000E+00  0.59841E-07  0.00000E+00 -0.71884E-07  0.00000E+00

ALPHA (PER RAD)         -0.27092E-06  0.00000E+00  0.36351E+04  0.00000E+00 -0.33596E+04  0.00000E+00

BETA (PER RAD)           0.00000E+00  0.30139E-07  0.00000E+00  0.91642E-15  0.00000E+00  0.13627E-05

PRATE (PER P*B/2V)       0.00000E+00 -0.51144E-05  0.00000E+00 -0.13140E+07  0.00000E+00 -0.19967E-03

QRATE (PER Q*C/2V)      -0.45800E-06  0.00000E+00  0.92910E+04  0.00000E+00 -0.96150E+04  0.00000E+00

RRATE (PER R*B/2V)       0.00000E+00 -0.55570E+02  0.00000E+00  0.68351E-04  0.00000E+00 -0.21544E+04

ALDOT (PER ALDOT*C/2V)   0.20908E-06  0.00000E+00 -0.54697E+03  0.00000E+00  0.43261E+03  0.00000E+00

BTDOT (PER BTDOT*B/2V)   0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00

PDOT (PER PDOT*B/2V)     0.00000E+00  0.25809E-05  0.00000E+00 -0.14843E+08  0.00000E+00 -0.40719E-03
```

2. The steady flow solution must be saved by the **STFLOW/MLDTRIM** bulk data card of ZEUS or the A**EROGEN/MLDTRIM** bulk data card of ZESt prior to the **STABDRV** analysis. Therefore, the file specified by the **EXTFILE** bulk data card must exist. Otherwise, a fatal error occurs.

3. At the end of each time step, for each BLOCK, the ratio of the averaged residual over that at the last iteration of the first time step is compared to EPSC. If the ratios are less than the absolute value of EPSC for all the blocks, the convergence criteria are met, and the computation will stop even before NSTEP number of time steps are finished. The time history of the LOG10 of the residual ratio for each block will be plotted. Note that for EPSC>0.0, the linearized Euler solver (see **MKPARAL** bulk data card of ZEUS) is used to compute the aerodynamic force /moment derivatives and damping derivatives. For EPSC<0.0, those derivcatives are computed by the K-expansion method (See **KEXPAND** bulk data card of ZEUS).

4. CFL is the Courant-Friedrichs-Lewy (CFL) number for the linesarized Euler solver. If CFL is an integer, this integer is the identification number of the **MKBLOCK** bulk data card that allows the different CFL numbers being used by the different blocks of mesh.

5. NSTEP is the number of iteration steps to achieve a converged solution. There is no physical time step involved in Linearized Euler runs, and thus NSTEP is actually not necessary and can be simply set to 1. If there is only one block of mesh, then the value of NEWTN doesn't matter either, which means only the final product of TRMSTEP × NEWTN × NCYC counts. If overset mesh is involved, the value of NEWTN matters as overset interpolation is performed at the end of each Newton sub-iteration.

6. The pressure sensor switch function used in the JST scheme is defined as:

$$v_j = \left| \frac{p_{j-1} - 2p_j + p_{j+1}}{p_{j-1} + 2p_j + p_{j+1}} \right|$$

   Using the TVD concept, an alternative for the pressure switch can be:

$$v_j = \frac{\left| p_{j+1} - 2p_j + p_{j-1} \right|}{(1-\omega)\rho + \omega\rho_{TVD}}, \ \rho_{TVD} = \left| p_{j+1} - p_j \right| + \left| p_j - p_{j-1} \right|, \ \rho = p_{j+1} + 2p_j + p_{j-1}$$

   where ω is called TVDCOEF, and if ω =0.0, the original JST pressure switch is recovered.

   Note: that using a large ω could potentially smear out the pressure jump across the shock. For cases involving transonic shock, TVDCOEF < 0.8 is recommended.

7. Larger VIS2/VIS4 adds more artificial dissipation to the Euler solver scheme that provides more stability to the solver. However, this could smear out the discontinuity of the flow solution across the shock if too much artificial dissipation is added. The default values of VIS2 and VIS4 are the optimal values and work the best for most cases except for extreme conditions such as high supersonic flow at high angle of attack where solution diverges and gradually increase of VIS2/VIS4 is recommended for a remedy try. If VIS2/VIS4 is an integer, this integer is the identification number of the **MKBLOCK** bulk data card that allows the different VIS2/VIS4 numbers being used by the different blocks of mesh.

STFLOW Save/Retrieve Steady Flow Solution

Description: Saves or retrieves the steady flow solutions and outputs of the steady flow solution and the mesh in the PLOT3D format on external files.

Format and Example:

1	2	3	4 5	6 7	8 9	10
STFLOW	IDBULK	SAVE	SAVFILE	GRDFILE	SOLFILE	

STFLOW	100	SAVESTOP	SAVE.DAT	F5MESH.PLT	F5SOL.PLT	

Field	Contents
IDBULK	Identification number that matches the identification number of a **FLUTTER**, **MLOADS**, **ELOADS**, **GLOADS**, **TRIM**, **GENGAF** or **NLFLTR** Case Control Command. (Integer > 0) (See Remark 1)
SAVE	Character string either "SAVE", "ACQU", "SAVESTOP", or "ACQUSTOP". For SAVE = "SAVE" — Saves the steady flow solution including all parameters of the governing equation on file "SAVFILE". SAVE = "ACQU" — Retrieves the steady flow solution from the file "SAVFILE". SAVE = "SAVESTOP" or "ACQUSTOP" — Same as "SAVE" and "ACQU", respectively, except terminates the computation of the current subcase after the steady aerodynamic computation is completed to avoid the unsteady aerodynamic computation. SAVE = "ACQUSAVE" — Retrieves the steady flow solution from the file "SAVFILE" as the initial flow solution and save the flow solution on the same file "SAVFILE" after the steady solution of the current flight condition is obtained. SAVE = "ACQUSAVS" — Same as "ACQUSAVE" except terminates the computation for the current subcase after the steady flow computation is completed. (Character) (See Remark 2)
SAVFILE	Character string to specify the file name on which the steady flow solution is stored.
GRDFILE	Character string up to 16 characters to specify the name of an external file on which the mesh is stored in the ASCII PLOT3D format for plotting. (Character) (See Remark 3)
SOLFILE	Character string up to 16 characters to specify the name of an external file on which the flow solution is stored in the ASCII PLOT3D format for plotting. (Character) (See Remark 4)

Remarks:

1. The **STFLOW** bulk data card is referred to by a **FLUTTER**, **MLOADS**, **ELOADS**, **GLOADS**, **TRIM**, or **NLFLTR** Case Control Command to save or retrieve the steady or static aeroelastic flowfield solution. For **FLUTTER**, the steady flowfield solution is computed by the steady aerodynamic computation on the rigid configuration prior to the unsteady aerodynamic computation. For **TRIM**, The steady flowfield solution is computed by the static aeroelastic computation on the flexible configuration for the THKCAM trim variable. For **MLOADS**, **ELOADS**, **GLOADS**, **TRIM**, or **NLFLTR**, the steady flowfield solution is computed by the static aeroelastic computation on the flexible configuration prior to the time integration of the transient response. Note that if IDBULK does not match **FLUTTER**, **MLOADS**, **ELOADS**, **GLOADS** or **NLFLTR** Case Control Command, this **STFLOW** bulk data card is inactive.

2. If SAVE = “ACQU” or “ACQUSTOP”, the steady aerodynamic computation is avoided. Rather, the steady aerodynamic solution of all parameters involved in the governing equation is retrieved from the file “SAVFILE”. For SAVE= “ACQUSAVE”, the steady flow solution retrieved from the file “SAVESILE” is used as the initial flow condition to compute the flow solution of the current subcase. For instance, the convergence of the steady flow computation at angle of attack=2 degrees can be achieved faster if the initial flow solution is retrieved from that of the solution at angle of attack=0 degree.

3. The mesh is stored in the formatted PLOT3D form. The PLOT3D format is shown as follows:

Card Set 1	**NBLK**
NBLK	Number of blocks on the CFD mesh (Integer > 0)
Example	2

Card Set 2	**(IMAX(L), JMAX(L), KMAX(L), L = 1, NBLK)**
IMAX(L) JMAX(L) KMAX(L)	IMAX, JMAX, and KMAX are the number of grid points along the *I*, *J*, and *K* directions of each block, respectively. (Integer > 0)

L = 1

Card Set 3	**(((x(i,j,k), i=1, IMAX(L)), j=1, JMAX(L)), k=1, KMAX(L)), (((y(i,j,k), i=1, IMAX(L)), j=1, JMAX(L)), k=1, KMAX(L)), (((z(i,j,k), i=1, IMAX(L)), j=1, JMAX(L)), k=1, KMAX(L)), (IBLANK(i,j,k), i=1, IMAX(L), j=1, JMAX(L), k=1, KMAX(L))**
x(i,j,k) **y(i,j,k)** **z(i,j,k)** and **IBLANK(i,j,k)**	x(i,j,k), y(i,j,k), and z(i,j,k) are the x, y, and z locations of the grid points (Real). IBLANK(i,j,k) are the indices of each grid point for blanking (Integer)

Repeat Card Set 3 NBLK times.

L = L+1

Note that if the GRDFILE is blank, no data will be generated.

4. The flowfield solution is stored in the formatted PLOT3D form shown as follows:

Card Set 1	**NBLK**	**Format**
NBLK	Number of blocks on the CFD mesh	Integer > 0

L = 1

Card Set 2	**(IMAX(L), JMAX(L), KMAX(L), L = 1, NBLK)**
IMAX(L), JMAX(L), KMAX(L)	IMAX, JMAX, and KMAX are the number of grid points along the *I*, *J*, and *K* directions of each block, respectively. (Integer > 0)

Card Set 3	**FMACH, ALPHA, RE, TIME**	**Format**
FMACH ALPHA RE TIME	Mach number Angle of Attack Reynolds number Time Step	Real

Card Set 4	**(((RHO(i,j,k), i=1, IMAX(L)), j=1, JMAX(L)), k=1, KMAX(L)), (((RU(i,j,k), i=1, IMAX(L)), j=1, JMAX(L)), k=1, KMAX(L)), (((RV(i,j,k), i=1, IMAX(L)), j=1, JMAX(L)), k=1, KMAX(L)), (RW(i,j,k), i=1, IMAX(L), j=1, JMAX(L), k=1, KMAX(L)), (E(i,j,k), i=1, IMAX(L), j=1, JMAX(L), k=1, KMAX(L))**	**Format**
RHO(i,j,k) RU(i,j,k) RV(i,j,k) RW(i,j,k) and E(i,j,k)	Non-dimensionalized density Non-dimensionalized momentum along x Non-dimensionalized momentum along y Non-dimensionalized momentum along z Non-dimensionalized total energy	Real

Repeat Card Sets 2 and 3 NBLK times.

L = L+1

Note that if the SOLFILE is blank, no data will be generated.

SURFLST — Control Surface Definition

Description: Defines the set of control surfaces.

Format and Example:

1	2	3	4	5	6	7	8	9	10
SURFLST	SID	SURF1	SURF2	SURF3	SURF4	SURF5	SURF6	SURF7	CONT
CONT	SURF8	-etc-							

SURFLST	30	SURF2	SURF4	SURF6	SURF3	SURF1			

Field	Contents
SID	Set identification number. (Integer > 0) (See Remark 1)
SURFi	The name of the i^{th} control surface. (Character) (See Remark 2)

Remarks:

1. SID is selected in the **GENGAF** bulk data card.

2. The SURFi labels must match the label entry of an **AESURFZ, AESLINK,** or **PZTMODE** bulk data card. Note that the ACTID entry in these bulk data cards (referring to an **ACTU** bulk data card) cannot be blank.

TABDMP1 — Modal Damping Table

Description: Defines modal damping as a tabular function of frequency.

Format and Example:

1	2	3	4	5	6	7	8	9	10
TABDMP1	TID	TYPE							CONT
CONT	f1	g1	f2	g2	f3	g3	-etc-		

TABDMP1	2								
	2.5	.01057	2.6	.01362	-etc-				

Field	Contents
TID	Table identification number. (Integer > 0)
TYPE	Character string which indicates the type of damping units, G, CRIT, Q or blank. Default is G.
fi	Frequency value in cycles per unit time (Real ≥ 0.0)
gi	Damping value (Real)

Remarks:

1. The fi must be in ascending order.
2. Discontinuity may be specified between any two points, except the two starting points and the two end points.
3. At least two entries must be present.
4. The **TABDMP1** mnemonic infers the use of the algorithm:

$$g = g_t(f)$$

where f is input to the table and g is returned. The table look-up g_t(f) is performed using linear interpolation within the table and linear extrapolation outside the table using the last two end points at the appropriate table end. There are no error returns from this table look-up procedure.

5. If TYPE is CRIT, the damping values are in the units of fraction of critical damping C/C_o, where. C_o is the critical damping. If TYPE is G or blank, the damping values are in structural damping units, that is, $2 \times C/C_o$. If TYPE is Q, the damping values are in the units of the amplification or quality factor, Q. These constants are related by the following equations:

$$C / C_o = g / 2$$

$$Q = \left\{ \begin{array}{c} \dfrac{1}{\left(\dfrac{2C}{C_o}\right)} \\ \dfrac{1}{g} \end{array} \right\}$$

TABLED1

Time-Dependent or Frequency-Dependent Table

Description: Defines a tabular function for use in generating frequency-dependent or time-dependent table.

Format and Example:

1	2	3	4	5	6	7	8	9	10
TABLED1	TID	XAXIS	YAXIS	EXTPXL	EXTPXH	FACTOR			CONT
CONT	x1	y1	x2	y2	x3	y3	-etc-		

TABLED1	32	LOG	LINEAR	NO	NO				
	1.0	6.9	2.0	5.6	3.0	5.6			

Field	Contents
TID	Table identification number. (Integer > 0)
XAXIS	Specifies a linear or logarithmic interpolation for the x-axis. (Character, “LINEAR” or “LOG”; Default = “LINEAR”) (See Remark 6)
YAXIS	Specifies a linear or logarithmic interpolation for the y-axis. (Character, “LINEAR” or “LOG”; Default = “LINEAR”) (See Remark 6)
EXTPXL	Character string either “YES” or “NO”. If EXTPXL = “YES” and XAXIS = YAXIS = “LINEAR”, then extrapolation is performed for any given x value that is less than the smallest xi. (Character, Default = “NO”)
EXTPXH	Same as EXTPXL, but for the given *x* value that is greater than the largest x_i . (Character, Default = “NO”)
FACTOR	Multiplication factor to all yi. (Real, Default = 1.0).
xi, yi	Tabular values (Real)

Remarks:

1. xi must be in ascending order.

2. Discontinuity may be specified between any two points, except the two starting points and the two ending points.

3. At least one continuation must be specified.

4. **TABLED1** uses the algorithm:

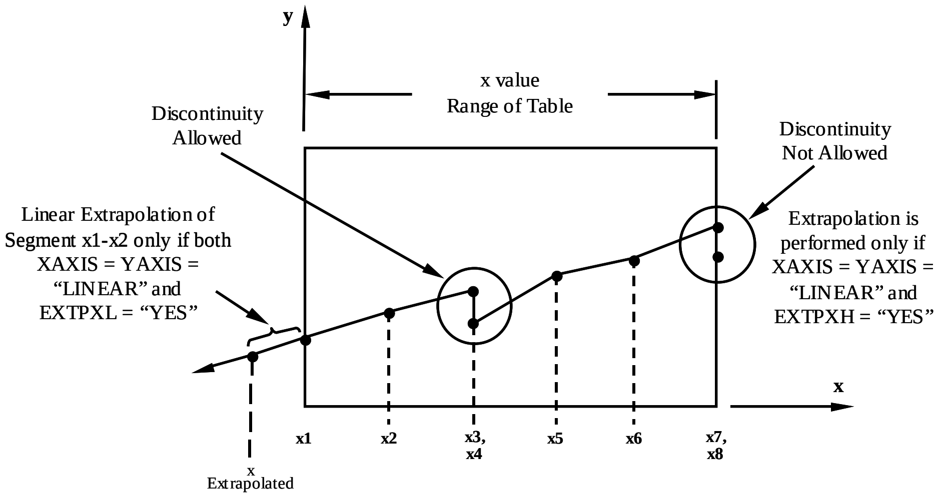
$$y = y_T(x)$$

where x is input to the table and y is returned. The table look-up is performed using interpolation within the table and extrapolation outside the table using the last two starting or ending points. The algorithms used for the interpolation or extrapolation are:

XAXIS	YAXIS	$y_T(x)$
LINEAR	LINEAR	$\dfrac{x_{i+1} - x}{x_{i+1} - x_i} y_i + \dfrac{x - x_i}{x_{i+1} - x_i} y_{i+1}$
LOG	LINEAR	$\dfrac{\ln (x_{i+1}/x)}{\ln (x_{i+1}/x_i)} y_i + \dfrac{\ln (x/x_i)}{\ln (x_{i+1}/x_i)} y_{i+1}$
LINEAR	LOG	$\exp\left[\dfrac{x_{i+1} - x}{x_{i+1} - x_i} \ln y_i + \dfrac{x - x_i}{x_{i+1} - x_i} \ln y_{i+1}\right]$
LOG	LOG	$\exp\left[\dfrac{\ln (x_{i+1}/x)}{\ln (x_{i+1}/x_i)} \ln y_i + \dfrac{\ln (x/x_i)}{\ln (x_{i+1}/x_i)} \ln y_{i+1}\right]$

Note: If NO extrapolation is performed, and y is assumed to be zero if x is outside the table.

5. The following figure shows a typical example of the table.



6. x_i must be positive if XAXIS = "LOG". Likewise, y_i must be positive if YAXIS = "LOG".

TRIM

Static Aeroelastic / Trim Analysis

Description: Defines the flight condition, rigid body mass matrix, trim degrees of freedom and trim variables. If the structural finite element model involves rigid body modes, the SUPORT entry of the **'ASSIGN FEM='** Executive Control Command must be specified. Otherwise, the trim analysis may encounter a singularity in the static aeroelastic analysis. .This is because the stiffness matrix of a free-free structure is singular. The program can remove this singularity only if the SUPORT entry is specified.

Format and Example:

1	2	3	4	5	6	7	8	9	10
TRIM	TRIMID	IDMK	QINF	IDOBJ	IDCONS	RHOX	RHOY	RHOZ	CONT
CONT	WTMASS	WEIGHT	IXX	IXY	IYY	IXZ	IYZ	IZZ	CONT
CONT	TRNACC	NX	NY	NZ	PDOT	QDOT	RDOT	LOADSET	CONT
CONT	IDVAR1	VAL1	IDVAR2	VAL2	-etc-				

TRIM	100	90	1200.0	10	20	2.7	0.0	1.0	+T1
+T1	0.03108	1600.00	50+05	-2.9+05	1.9+06	-8.7+03	0.0	2.4+06	+T2
+T2	G	NONE	0.0	FREE	0.2	0.0	FREE	100	+T3
+T3	100	25.0	300	FREE	400	FREE	200	FREE	+T4
+T4	500	1.0	900	0.0					

Field	Contents
TRIMID	Unique set identification number. (Integer > 0) (See Remark 1)
IDMK	Identification number of an **MKAEROZ** bulk data card that defines the Mach number and its associated aerodynamics used for the static aeroelastic / trim analysis. (Integer > 0) (See Remark 2)
QINF	Dynamic pressure. (Real > 0.0) (See Remark 3)
IDOBJ	Identification number of a **TRIMOBJ** bulk data card that defines the objective function to be minimized. Active only for an over-determined trim system. (Integer ≥ 0)
IDCONS	Identification number of a **TRIMCON** bulk data card that defines a set of constraint functions to be satisfied. Active only for an over-determined trim system. (Integer ≥ 0)
RHOX, RHOY, RHOZ	*x*, *y*, and *z* components, respectively, of a vector from the aerodynamic moment center (REFX, REFY and REFZ in the **AEROZ** bulk data card) to the center of gravity (C.G.) of the configuration. Thus, the center of gravity is computed by:

$x_{C.G.}$ = REFX + RHOX
$y_{C.G.}$ = REFY + RHOY
$z_{C.G.}$ = REFZ + RHOZ
(Real)

WTMASS	Factor to convert weight to mass. WTMASS = 1 / g, where g is the gravitational acceleration. (Real > 0.0)
WEIGHT	The weight of the whole aircraft. (Real > 0.0) (See Remark 4)
IXX, IXY, IYY, IXZ, IYZ, IZZ	The weight moment of inertia about the center of gravity (C.G.) of the whole aircraft, where the x, y and z denote the rotational axis that are associated with the aerodynamic model. (Real) (See Remark 4) <u>Note</u>: IXX, IYY and IZZ must be greater than zero.
TRNACC	Character string to specify the units of the accelerations (NX, NY, NZ, PDOT, QDOT and RDOT) of the trim degrees of freedom. (Character, Default = 'G') TRNACC = "TRUE", The units of the acceleration are FMLUNIT/sec^2, (where FMLUNIT is the length unit defined by the **AEROZ** bulk data card) for NX, NY and NZ, and rad/sec^2 for PDOT, QDOT and RDOT. TRNACC = "G", NX, NY and NZ are specified in terms of the gravity (g), where PDOT, QDOT and RDOT in terms of rad/FMLUNIT.
NX, NY, NZ	Translational accelerations along the *x, y* and *z* axis, respectively, of the aerodynamic model. (Character or Real) (See Remark 5) Three options are available: Characters "NONE" The trim degree of freedom associated with the translational acceleration is eliminated from the trim system. Characters "FREE" The translational acceleration is a "FREE" trim d.o.f. The value of the translational acceleration is unknown and to be solved by the trim system. Real Value The translational acceleration is fixed and given by the real value.
PDOT, QDOT, RDOT	Angular acceleration about the center of gravity (CG) at x, y and z axis, respectively, of the aerodynamic model. (Character or Real) (See Remark 5) Similar to NX, NY and NZ, characters "NONE", "FREE", or real values can be specified.
LOADSET	Identification number of a **SET1** or **SETADD** bulk data card that specifies a set of identification numbers of **TRIMFNC** or **TRIMADD** bulk data card. All values of the trim functions defined by the **TRIMFNC** or **TRIMADD** bulk data card are computed and printed out. (Integer ≥ 0)
IDVARi	Identification number of a **TRIMVAR** bulk data card to define a trim variable. (Integer > 0) (See Remark 6)
VALi	Value of the trim variable IDVARi. (Character or Real) (See Remark 6) Two options are available: Characters "FREE" The value of the trim variable is an unknown and to be solved by the trim system. Real Value The value of the trim variable is fixed and given by the real value.

<u>Remarks:</u>

1. For the static aeroelastic / trim analysis, the **TRIM** discipline must be selected in the Case Control Section with TRIM = TRIMID.

   Note 1: To compute the distributed inertial loads of a free-free structure (i.e. with rigid body vibration modes), it is required to:
   - specify the rigid body d.o.f. in the "SUPORT" entry of the **'ASSIGN FEM='** Executive Control Command
   - import the SMGH (from symmetric/asymmetric finite element modal analysis) or/and the AMGH (from anti-symmetric modal analysis) matrices by the **'ASSIGN MATRIX='** Executive Control Command.

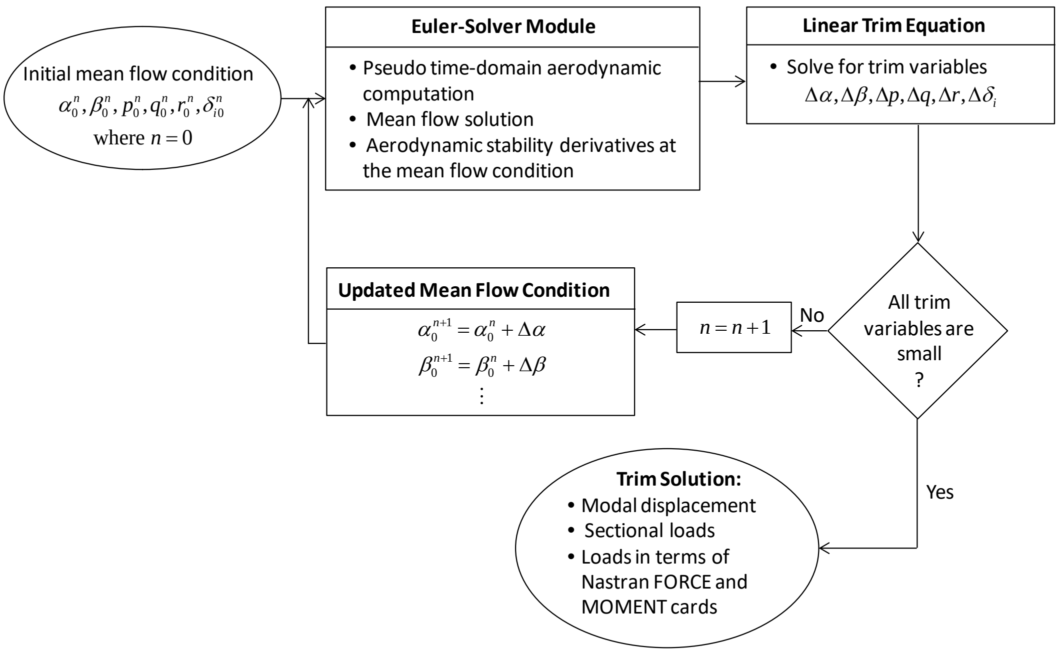
   Note 2: If the static aeroelastic iteration fails to converge, it is recommended to alter the default value of STRELAX using the **PARAM** bulk data card.

2. The ZEUS TRIM module employs the ZAERO linear trim analysis module as the core solver to obtain a trim solution. The ZAERO linear trim analysis module uses the aerodynamic stability derivative of each trim variable to solve the trim equation for satisfying the trim condition. The user should refer to the ZAERO theoretical manual for the formulation of the ZAERO linear trim analysis In this linear trim analysis, all aerodynamic stability derivatives are assumed to be constant; i.e. independent of the mean flow condition. However, in the real flow, the aerodynamic stability derivatives are functions of the mean flow condition. Because ZEUS can capture this nonlinear effect, an iterative procedure is developed in the ZEUS TRIM module that iterates around the ZAERO linear trim module until a converged trim solution is achieved. Such an iterative procedure is depicted in Figure below. An initial mean flow condition (α, β, p, q ,r and control surface deflection angles,.. etc) is first specified by a **TRIMFLT** bulk data card. Based on these this mean flow condition, the Euler solver module first computes the mean flow solution in terms of the aerodynamic forces and moments using the pseudo time-domain aerodynamic computation then computes the aerodynamic stability derivatives from the mean flow solution for each trim variable. The number of the pseudo time step is 100 as a default value but that can be altered using the **MKPARAM** bulk data card with entry TRMSTEP.

   Two methods are available to compute the aerodynamic stability derivatives. The first method is the finite difference method that introduces a small perturbation due to the change of the trim variable to the mean flow condition then calculates the derivative using the difference between the perturbed and mean flow solutions. The pseudo time-domain aerodynamic computation is again employed for this perturbed flow computation except using the mean flow solution as the initial flow condition. To increase the computational efficiency, the number of pseudo time steps for this perturbed flow computation is reduced from TRMSTEP by a factor of 3 but that can be altered using a **PARAM** bulk data card with entry NAME = TRMREDUC. The second method is to use the linearized Euler solver that computes the derivatives by solving the linearized Euler equation. To invoke the linearized Euler solver, it requires to specify the entry METHOD=3 in the **MKPARAM** bulk data card and as an option, to add a **MKPARAL** bulk data card if a different set of parameters is to be used for the linearized Euler solver.

   The mean flow solution and the aerodynamic stability derivatives of trim variables are used by the ZAERO linear trim module to compute the trim variable solutions. These trim variable solutions are then added to the initial mean flow condition to update the mean flow condition. Then the Euler solver module re-computes the mean flow solution and the aerodynamic stability derivatives for the next iteration. Based on the new mean flow solution and aerodynamic stability derivatives, the ZAERO linear trim module solves the trim solution again. If all trim variable solutions are smaller than 3% of their respective values at the first trim iteration (this 3% can be altered by a **PARAM** bulk data card with entry NAME="TRIMCNV"), a converged solution is achieved because the mean flow solution in this mean flow condition itself satisfies the trim condition. Otherwise, these trim variable solutions are added to the mean flow condition of the previous iteration for the next iteration. The maximum number of iterations that loops around the ZAERO linear trim module is 3 but that can be altered using a **PARAM** bulk data card with entry NAME = "MAXTRIM".

The final trim solution can be saved on a file using the **MLDTRIM** bulk data card. This final trim solution can be retrieved back via the **MLDTRIM** bulk data card as the initial flow condition for transient response analysis such as MLOADS, GLOADS, ELOADS, and NLFLTR.



3. The units of the dynamic pressure must be consistent with the mass and length units specified in the FMMUNIT and FMLUNIT entries of the **AEROZ** bulk data card. In fact, all mass and length units involved in the **TRIM** bulk data card must be consistent with FMMUNIT and FMLUNIT, respectively.

4. WEIGHT, IXX, IYY, … are multiplied by WTMASS to convert weight to mass. These values define a 6×6 rigid body mass matrix such as:

$$\begin{bmatrix} \text{WEIGHT} & & & & & \\ & \text{WEIGHT} & & & & \\ & & \text{WEIGHT} & & & \\ & & & \text{IXX} & \text{IXY} & \text{IXZ} \\ & & & \text{IXY} & \text{IYY} & \text{IYZ} \\ & & & \text{IXZ} & \text{IYZ} & \text{IZZ} \end{bmatrix} \times \frac{1}{g}$$

It should be noted that WEIGHT, IXX, IYY, … must account for two sides of aircraft even if only half of the configuration (XZSYM = "YES" in the **AEROZ** bulk data card) is modeled. Also, unlike the conventional definition, the coupling terms IXY, IXZ, and IYZ have NO negative sign in the above matrix.These mass of moment interia values can be optained from the NASTRN output file such as follows.

```
O U T P U T   F R O M   G R I D   P O I N T   W E I G H T   G E N E R A T O R
..........................................................................
..........................................................................
                                  I(S)
              *  7.639202E+05 -1.757623E+05 -1.441086E+04 *
              * -1.757623E+05  1.942568E+06  4.185864E+04 *
              * -1.441086E+04  4.185864E+04  2.684407E+06 *
```

5. If NX, NY, NZ, PDOT, QDOT, and/or RDOT = "NONE", then its associated degree of freedom of the 6 rigid body motions (three translational and three rotational degrees of freedom) is eliminated from the trim system. The remaining degrees of freedom are defined as "*trim degrees of freedom*" of the trim system. Among these trim degrees of freedom, if "FREE" is specified, then the associated degrees of freedom are defined as "*free trim degrees of freedom*". If real values are given, the associated trim degrees of freedom are defined as "*given trim degrees of freedom*".

6. If "FREE" is specified, the trim variable is defined as "*free trim variable*". If real value is given, the trim variable is defined as "*given trim variable*". The number of free trim degrees of freedom plus the number of free trim variables are the total number of unknowns of the trim system. If the total number of unknowns of the trim system is equal to the number of trim degrees of freedom (including both free and given trim degrees of freedom), the trim system is defined as "*determined trim system*" since the number of equations and unknowns are equal. If the total number of unknowns is greater than the number of trim degrees of freedom, the trim system is defined as "*over-determined trim system*". Solving the over-determined trim system requires the minimization of a user defined objective function (a **TRIMOBJ** bulk data card referred to by the IDOBJ entity of the **TRIM** bulk data card) while subjected to a set of user defined constraint functions (a **TRIMCON** bulk data card that is referred to by the IDCONS entity of the **TRIM** bulk data card). If the total number of unknowns is less than the number of trim degrees of freedom, the trim system is unsolvable and a fatal error will occur.

TRIMADD

Defines a Trim Function as a Function of Other Trim Functions

Description: Defines a trim function as a function of the other trim functions. The function is expressed as:

$$F = \left\{ \left\{ \left[\left(S_0\, F_0^{C_0} \oplus S_1\, F_1^{C_1} \right)^{E_1} \oplus S_2\, F_2^{C_2} \right]^{E_2} \oplus S_3\, F_3^{C_3} \right\}^{E_3} \oplus S_4\, F_4^{C_4} \right\}^{E_4} + \ldots$$

where $\oplus$ represents '+', '-', '$*$', or '/ '.

Format and Example:

1	2	3	4	5	6	7	8	9	10
TRIMADD	IDFNC	S_0	F_0	C_0					CONT
CONT	$SYMBOL_1$	S_1	F_1	C_1	E_1				CONT
CONT	$SYMBOL_2$	S_2	F_2	C_2	E_2	-etc-			

TRIMADD	100	1.0	101	-0.5					+T1
+T1	+	2.0	102	1.0	2.0				+T2
+T2	-	-1.0	111	2.0	-1.0				

Field	Contents
IDFNC	Unique identification number. (among all **TRIMFNC** and **TRIMADD** bulk data cards) (Integer > 0) (See Remark 1)
S_0	Real coefficients shown in the above equation. (Real)
F_0	Identification number of a **TRIMFNC** bulk data card whose value is represented by the symbol F_0 shown in the above equation. (Integer > 0)
C_0	Real coefficients shown in the above equation. (Real, Default = 1.0)
$SYMBOL_i$	Character string either '+', '-', '$*$', or '/ ' (see the symbol '$\oplus$' shown in the above equation) (Character)
F_i	Identification number of a **TRIMFNC** bulk data card whose value is represented by the symbol F_i shown in the above equation. If F_i is zero, the value is assumed to be zero. (Integer ≥ 0)
C_i	Real coefficients shown in the above equation. (Real, Default = 1.0)
E_i	Real coefficients shown in the above equation. (Real, Default = 1.0)

Remarks:

1. IDFNC is referred to by the **TRIMOBJ** and **TRIMCON** bulk data cards to define the objective function and constraint functions for over-determined trim systems. IDFNC can also be referred to by the **TRIM** bulk data card through the **SET1** or **SETADD** bulk data card to print out the values of the trim functions.

2. **TRIMADD** can be used to construct a trim function by a complex expression that cannot be defined by a single **TRIMFNC** bulk data card. The following example shows how to construct the Von Mises stress formula by the **TRIMADD** bulk data card.

   The Von Mises stress formula is expressed as follows:

$$g = \left[\left(\frac{\sigma_x}{A_x}\right)^2 + \left(\frac{\sigma_y}{A_y}\right)^2 - \frac{\sigma_x \ \sigma_y}{A_x \ A_y} + \left(\frac{\tau_{xy}}{A_{xy}}\right)^2 \right]^{1/2}$$

   where σ_x, σ_y and τ_{xy} are the stresses of an element in the finite element mode, and A_x, A_y and A_{xy} are constants.

   To construct the Von Mises stress formula by the **TRIMADD** bulk data card, it is required first to specify three **TRIMFNC** bulk data cards which define three trim functions referring to σ_x, σ_y and τ_{xy} stresses of an element, respectively. The identification numbers of these **TRIMFNC** cards, for instance, are ID_1, ID_2 and ID_3. The entries of the **TRIMADD** bulk data card are

   for the term $-\dfrac{\sigma_x \ \sigma_y}{A_x \ A_y}$:

$$S_0 = -\frac{1}{A_x}, \qquad F_0 = ID_1, \qquad C_0 = 1.0$$

$$SYMBOL_1 = '*', \qquad S_1 = \frac{1}{A_y}, \qquad F_1 = ID_2, \qquad C_1 = 1.0, \qquad E_1 = 1.0$$

   for the term $\left(\dfrac{\sigma_x}{A_x}\right)^2$:

$$SYMBOL_2 = '+', \qquad S_2 = \frac{1}{A_x^2}, \qquad F_2 = ID_1, \qquad C_2 = 2.0, \qquad E_2 = 1.0$$

   for the term $\left(\dfrac{\sigma_y}{A_y}\right)^2$:

$$\text{SYMBOL}_3 = '+', \qquad \text{S}_3 = \frac{1}{A_y^{\ 2}}, \qquad \text{F}_3 = \text{ID}_2, \qquad \text{C}_3 = 2.0, \qquad \text{E}_3 = 1.0$$

for the term $\left(\frac{\tau_{xy}}{A_{xy}}\right)^2$:

$$\text{SYMBOL}_4 = '+', \qquad \text{S}_4 = \frac{1}{A_{xy}^{\ 2}}, \qquad \text{F}_4 = \text{ID}_3, \qquad \text{C}_4 = 2.0,$$

and finally

$$\text{E}_4 = 0.5$$

TRIMCON

Constraint Functions for the Static Aeroelastic/Trim Analysis

Description: Defines a set of constraint functions (G_i) to be satisfied for solving the over-determined trim system. G_i is defined as:

$$G_i = (F_i - S_i)^{E_i} \quad < \quad \text{or} \quad > \; V_i$$

where F_i represents the value of a trim function.

Format and Example:

1	2	3	4	5	6	7	8	9	10
TRIMCON	IDCONS								CONT
CONT	$IDFNC_1$	S_1	E_1	$GTORLT_1$	$VALUE_1$				CONT
CONT	$IDFNC_2$	S_2	E_2	$GTORLT_2$	$VALUE_2$				CONT
CONT	-etc-								

TRIMCON	101								+T1
+T1	100	0.0	1.0	GT	100.0				+T2
+T2	200	1.0	2.0	LT	200.0				+T3
+T3	205	0.0	2.0	GT	0.33				

Field	Contents
IDCONS	Unique identification number. (Integer > 0) (See Remark 1)
$IDFNC_i$	Identification number of a **TRIMFNC** or **TRIMADD** bulk data card whose value is represented by the symbol F_i shown in the above equation. (Integer > 0)
S_i and E_i	Real coefficients shown in the above equation (Real) Note: E_i cannot be zero.
$GTORLT_i$	Character string either “GT” or “LT” (Character) “GT” represents that $(F_i - S_i)^{E_i}$ must be greater than V_i “LT” represents that $(F_i - S_i)^{E_i}$ must be less than V_i
$VALUE_i$	Constraint value represented by the symbol V_i shown in the above equation. (Real)

Remarks:

1. IDCONS is referred to by the **TRIM** bulk data card. The **TRIMCON** bulk data card is active only for the over-determined trim system. All G_i serve as a set of constraint functions that must be satisfied simultaneously by the trim variables.

2. Since $(F_i - S_i)$ could be negative, the user must select proper values of E_i to avoid complex number resulting from the constraints functions.

TRIMFLT

Mean Flow Condition Specification

Description: Specifies the mean flow conditions for unsteady aerodynamics computation.

Format and Example:

1	2	3	4	5	6	7	8	9	10
TRIMFLT	IDFLT	IDVIS	ALPHA	BETA	PRATE	QRATE	RRATE	THERMO	CONT
CONT	LABEl1	VAL1	LABEL2	VAL2	LABEL3	VAL3	LABEL4	VAL4	-etc-

TRIMFLT	1	100	13.5	0.0	0.0	0.0	0.0		+BC
+BC	ELEV	9.0	RUDDER	3.0					

Field	Contents
IDFLT	Identification number (Integer > 0)
IDVIS	Identification number of a **VISCOUS** bulk data card to define the parameters for the inclusion of boundary layer effects. (Integer ≥ 0) (See Remark 2)
ALPHA	Angle of attack in degrees (Real)
BETA	Side slip angle in degrees (Real) (See Remark 3)
PRATE, QRATE, RRATE	Non-dimensional Roll, Pitch, and Yaw rates (Real)
THERMO	Not Used
LABELi	Label of the control surfaces defined in the **AESURFZ, AESLINK, PZTMODE**, or **GRIDFRC** bulk data card (Character) (See Remark 4)
VALi	Control surfaces deflection angle in degrees (Real)(See Remark 5)

Remarks:

1. The **TRIMFLT** bulk data card is referred to by the **MKAEROZ** bulk data cards for unsteady aerodynamic data generation. In this case, ALPHA, BETA, PRATE, QRATE, and RRATE and the control surface deflections define the mean flow conditions. The unsteady aerodynamic data is computed by the perturbation about the mean flow conditions. This implies that the unsteady aerodynamics is coupled with the steady mean flow aerodynamics.

2. If IDVIS = 0, the inviscid analysis is performed.

3. The non-dimensional roll, pitch, and yaw rates are defined as:

PRATE =	(roll rate)	×	(REFB/2.0) / V
QRATE =	(pitch rate)	×	(REFC/2.0) / V
RRATE =	(yaw rate)	×	(REFB/2.0) / V

where *V* is the free stream velocity. The quantities REFB and REFC are the reference span and reference chord, respectively, specified in the **AEROZ** bulk data card with units specified in the FMLUNIT entry.

4. LABEL must be defined in the **AESURFZ, AESLINK**, **PZTMODE**, or **GRIDFRC** bulk data cards.

5. If LABEL refers to an AESURFS or AESLINK bulk data card. The unit of VALi is in degrees. Otherwise, the unit is defined by the user.

TRIMFNC

Trim Function

Description: Defines a trim function whose value is dependent on the trim variables and trim degrees of freedom.

Format and Example:

1	2	3	4	5	6	7	8	9	10
TRIMFNC	IDFNC	TYPE	LABEL	RHS	ISSET	IASET	REMARK		
TRIMFNC	10	MODAL	DMI	LHS	MATRIXR	MATRIXL	STRESS.AT.CBAR		

Field	Contents
IDFNC	Unique identification number (Integer > 0) (See Remark 1)
TYPE	Character string. One of "AERO", "FEM" or "MODAL" (Character) TYPE = "AERO" the trim function is evaluated based on the aerodynamic model. TYPE = "FEM" the trim function is evaluated based on the structural finite element model. TYPE = "MODAL" the function is evaluated based on the user supplied modal data.
LABEL	Character string that must match one of the following characters:

For TYPE = "AERO"

Characters	Description
CDL	Induced drag coefficient. Please see the definition of C_d in the **TRIMVAR** bulk data card.
CDT	Total drag coefficient including induced drag, wave drag, and friction drag.
CY	Side force coefficient. Please see the definition of C_y in the **TRIMVAR** bulk data card.
CL	Lift coefficient. Please see the definition of C_L in the **TRIMVAR** bulk data card.
CR	Roll moment coefficient. Please see the definition of C_l in the **TRIMVAR** bulk data card.
CM	Pitch moment coefficient. Please see the definition of C_m in the **TRIMVAR** bulk data card.
CN	Yaw moment coefficient. Please see the definition of C_n in the **TRIMVAR** bulk data card.
CP	Center of aerodynamic pressure. $CP = \frac{CM}{CL}$

NX, NY, NZ, PDOT, QDOT, or RDOT	The acceleration of the trim degrees of freedom is defined as a trim function.
TRIMVAR	The value of a trim variable is defined as the trim function. The identification number of the trim variable is specified in the ISSET entry of the **TRIMFNC** bulk data card.
LOADMOD	The component loads due to the aerodynamic loads at a set of aerodynamic boxes that are specified in SETK entry of the **LOADMOD** bulk data card is defined as the trim function. The identification number of the **LOADMOD** bulk data card is defined in the ISSET entry of the **TRIMFNC** bulk data card. (See Remark 2)

For TYPE = "FEM"

LABEL can either be "LOADMOD", "GRIDDISP", or "FORCE".

Characters	Description
LOADMOD	The component loads due to the aerodynamic loads and inertial loads at a set of structural finite element grid points that are specified in the SETG entry of the **LOADMOD** bulk data card is defined as the trim function. The identification number of the **LOADMOD** bulk data card is defined in the ISSET entry of the **TRIMFNC** bulk data card. (See Remark 2)
GRIDDISP	The displacement at a structural finite element grid point is defined as the trim function. The grid point identification number is specified in the ISSET entry and the component number is specified in the IASET entry. (see Remark 2).
FORCE	The force at a structural finite element grid point is defined as the trim function. The grid point identification number is specified in the ISSET entry and the component number is specified in the IASET entry.

For TYPE = "MODAL"

The resultant value from the superposition of the modal data of the flexible aircraft is defined as a trim function. LABEL must be either "AEFACT" or "DMI".

Characters	Description
LABEL="AEFACT"	The modal data is specified by the **AEFACT** bulk data card. The identification number of the **AEFACT** bulk data card for the symmetric (or asymmetric) modal data is specified in the ISSET entry whereas the anti-symmetric modal data in the IASET entry. (See Remark 3)
LABEL = "DMI"	The modal data is imported either by the **DMI** bulk data card or the **'ASSIGN MATRIX='** Executive Control Command. The name of the matrix that contains the symmetric (or asymmetric) modal data is specified in the ISSET entry whereas the anti-symmetric modal data in the IASET entry. (See Remark 3)

LABEL = "PCHFILE"	For the structural parameters defined by a **PCHFILE** bulk data card. ISSET and IASET are the identification number of a **PCHFILE** bulk data card that imports a NASTRAN punch output file containing the symmetric and anti-symmetric modal values of element forces, stresses, or strains, respectively. The trim results of all structural parameters listed in the $ELLST_i$ and $FIELD_i$ entries of the **PCHFILE** bulk data card are printed out. Note that for output, the LABEL and ISSET entries of the **TRIMFNC** bulk data card are replaced by the $LABEL_i$ and $ELLST_i$ entries of the **PCHFILE** bulk data card, respectively.

RHS	Character string to specify whether the trim function is evaluated on the Right Hand Side (RHS) or the Left Hand Side (LHS) of the configuration. (Character) (See Remark 4) Two options are available: RHS = "RHS" on the right side of the configuration RHS = "LHS" on the left hand side of the configuration
ISSET	ISSET is used only for: (1) TYPE = "AERO" and LABEL = "TRIMVAR" or "LOADMOD", (2) TYPE = "FEM" and LABEL = "LOADMOD", or (3) TYPE = "MODAL". ISSET is active only if the trim system is asymmetric or symmetric. (Integer or Character) (1) TYPE = "AERO" LABEL = "TRIMVAR" ISSET is an integer that is the identification number of a **TRIMVAR** bulk data card listed in the **TRIM** bulk data card. LABEL = "LOADMOD" ISSET is an integer that is the identification number of a **LOADMOD** bulk data card. (2) TYPE = "FEM" LABEL = "LOADMOD" ISSET is an integer that is the identification number of a **LOADMOD** bulk data card. LABEL = "GRIDDISP" or LABEL = "FORCE" ISSET is an integer that is the identification number of a structural finite element grid point. (3) TYPE = "MODAL" LABEL = "AEFACT" ISSET is an integer that is the identification number of the **AEFACT** bulk data card containing the modal data associated with the symmetric modes. The number of data must be the same as the number of the symmetric modes. Used only for symmetric or asymmetric trim system. LABEL = "DMI" ISSET is a character string that is the name of the matrix imported either by the **DMI** bulk data card or **'ASSIGN MATRIX='** Executive Control Command. The modal data contained in the matrix is associated with symmetric modes. The number of rows of the matrix must be the same as the symmetric modes. Used only for symmetric or asymmetric trim system. LABEL = "PCHFILE" ISSET is an integer that is the identification number

		of a **PCHFILE** bulk data card to specify the symmetric (or asymmetric) modal data.
IASET	For TYPE = "MODAL", IASET	is active only for anti-symmetric or asymmetric trim system.
	LABEL = "AEFACT"	IASET is an integer that is the identification number of the **AEFACT** bulk data card containing the modal data associated with the anti-symmetric modes. The number of modal data must be the same as the number of anti-symmetric modes.
	LABEL = "DMI"	IASET is a character string that is the name of the matrix imported either by the **DMI** bulk data card or **'ASSIGN MATRIX='** Executive Control Command. The modal data contained in the matrix is associated with the anti-symmetric modes. The number of rows of the matrix must be the same as the number of anti-symmetric modes.
	LABEL = "PCHFILE"	Same as ISSET but for the anti-symmetric modal data.
	For TYPE = "FEM", LABEL =	"GRIDDISP" or LABEL = "FORCE", IASET is the component number of the displacement at the structural finite element grid point with identification number being ISSET (Integer, either 1, 2, 3, 4, 5, or 6).
REMARK	Character String up to 16 with <u>no</u> embedded blanks to give description of the trim function (Character) (See Remark 5)	

<u>Remarks:</u>

1. IDFNC is referred to by the **TRIMOBJ** and **TRIMCON** bulk data cards to define the objective function and constraint functions for over-determined trim systems. IDFNC can also be referred to by the **TRIM** bulk data card through the **SET1** or **SETADD** bulk data card to print out the values of the trim functions.

2. For TYPE = "FEM", the component loads include the aerodynamic and inertial loads. In this case, the matrix [SGMH] must be imported by the **'ASSIGN MATRIX='** Executive Control Command with MNAME = 'SMGH' for symmetric trim system (trim degrees of freedom involving only NX, NY and/or QDOT). For anti-symmetric trim system (trim degrees of freedom involving only NY, PDOT and/or RDOT), the matrix [AMGH] must be imported by the **'ASSIGN MATRIX='** Executive Control Command with MNAME = 'AMHG'. For asymmetric trim system (trim degrees of freedom involve both symmetric and anti-symmetric trim systems), both [SMGH] and [AMGH] matrices must be imported. It should be noted that if the computation of inertial loads is invoked, the SUPORT entry in the **'ASSIGN FEM='** Executive Control Command must be specified to define the degrees of freedom of the rigid body modes of the structural finite element model.

   <u>Note</u>: The sign of the component loads is defined in the structural finite element basic coordinate that is specified by the ACSID entry of the **AEROZ** bulk data card.

3. Since the ZEUS static aeroelastic/trim analysis employs the modal approach to solve the trim system of the flexible aircraft, any structural quantities such as element stresses, forces, displacements, etc, can be obtained by the superposition of their respective modal data of each mode. These modal data must be imported from the structural finite element analysis. For instance, to obtain the modal data of stress by NASTRAN, the user can use the **NASTRAN** Case Control Command such as STRESS=ALL in the NASTRAN free vibration analysis. The user can select the modal stresses of a particular element or a group of element of interest and import these data to ZEUS by the **AEFACT** bulk data card (for one element), **DMI** bulk data card or **'ASSIGN MATRIX='** Executive Control Command (for a group of elements).

   Note: If LABEL = "DMI" or "PCHFILE" is specified, the **TRIMFNC** bulk data card can represent many trim functions. The number of trim functions depends on the number of columns of the matrix.

4. For a symmetric configuration (XZSYM = "YES" in the **AEROZ** bulk data card), ZEUS requires only the modeling of half of the configuration. For asymmetric trim system, ZEUS superimposes the results of the symmetric trim system and the anti-symmetric trim system to obtain the results on both sides of the configuration. The entry RHS is used only if LABEL = "TRIMVAR" or "LOADMOD" for TYPE = "AERO", TYPE = "FEM" and TYPE = "MODAL". For asymmetric configuration (XZSYM = "NO" in the **AEROZ** bulk data card), RHS must be "RHS".

5. Since all entries of the bulk data cards cannot have embedded blanks, the blanks for separating words will lead to a fatal error. For instance, the description "STRESS AT CBAR" has embedded blanks which is not allowed. To circumvent this problem, it is recommended to use period (".") between words such as "STRESS.AT.CBAR".

TRIMLNK

Trim Variable Linking

Description: Defines a set of coefficient and trim variable identification number pairs for trim variable linking.

Format and Example:

1	2	3	4	5	6	7	8	9	10
TRIMLNK	IDLINK	SYM	$COEFF_1$	$IDVAR_1$	$COEFF_2$	$IDVAR_2$	$COEFF_3$	$IDVAR_3$	CONT
CONT	$COEFF_4$	$IDVAR_4$	-etc-						

TRIMLNK	10	SYM	1.0	100	0.5	200			

Field	Contents
IDLINK	Unique identification number. (Integer > 0) (See Remark 1)
SYM	Character string to define the type of aerodynamic stability derivatives that are generated by the trim variable linking. (Character) (See Remark 2) SYM = "SYM" for longitudinal stability derivatives SYM = "ANTI" for lateral stability derivatives SYM = "ASYM" for both longitudinal and lateral stability derivatives
$COEFF_i$	Coefficient to define the linear relationship between the dependent and independent trim variables. (Real) (See Remark 3)
$IDVAR_i$	Identification number of a **TRIMVAR** bulk data card to define a dependent trim variable. (Integer > 0) (See Remark 3)

Remarks:

1. IDLINK is referred to by the TRIMLNK entry in the **TRIMVAR** bulk data card. The trim variable defined in the **TRIMVAR** bulk data card that refers to IDLINK is called "independent trim variable" whereas the trim variables whose identification numbers are listed in $IDVAR_i$ entries of the **TRIMLNK** bulk data cards are called "dependent trim variables." The **TRIMLNK** bulk data card provides a feature that allows the user to establish a linear relationship between the dependent trim variables and the independent trim variable. For instance, the deflections of the leading edge and trailing edge flaps of fighters are often scheduled according to the angle of attack for optimum lift to drag ratio. To model a so-called "flap-scheduling" control surfaces, the user can specify ALPHA to be the independent trim variable and the leading and trailing edge flaps as the dependent trim variables.

2. The type of aerodynamic stability derivative generated by both independent and dependent trim variables must be the same. Thus, the SYM entry in the **TRIMLNK** bulk data card serves as input error detector. If the SYM entry is different from the SYM entries specified in the **TRIMVAR** bulk data cards, a fatal error occurs.

3. The resulting aerodynamic stability derivatives of the variable-linked trim variable are computed based on the following equation:

$$\begin{pmatrix} \text{Resulting Aerodynamic} \\ \text{Stability Derivatives} \end{pmatrix} = \begin{pmatrix} \text{Aerodynamic Stability} \\ \text{Derivatives of the} \\ \text{Independent} \\ \text{Trim Variable} \end{pmatrix} + \sum_{\mathrm{i}} \mathrm{Coeff}_{\mathrm{i}} \begin{pmatrix} \text{Aerodynamic Stability} \\ \text{Derivatives of the} \\ \mathrm{i}^{\mathrm{th}} \text{ Dependent} \\ \text{Trim Variable} \end{pmatrix}$$

TRIMOBJ

Objective Function for the Static Aeroelastic/Trim Analysis

Description: Defines an objective function to be minimized for solving the over-determined trim system. The objective function (*OBJ*) is defined as:

$$OBJ = \sum_{i=1} \left[C_{1i}(F_i - S_{1i})^{E_{1i}} + C_{2i}(F_i - S_{2i})^{E_{2i}} \right]^{E_i}$$

where F_i is the value of a trim function.

Format and Example:

1	2	3	4	5	6	7	8	9	10
TRIMOBJ	IDOBJ								CONT
CONT	$IDFNC_1$	$C1_1$	$S1_1$	$E1_1$	$C2_1$	$S2_1$	$E2_1$	E_1	CONT
CONT	$IDFNC_2$	$C1_2$	$S1_2$	$E1_2$	$C2_2$	$S2_2$	$E2_2$	E_2	CONT
CONT	-etc-								

TRIMOBJ	10								+T1
+T1	100	0.001	0.0	2.0	0.0	0.0	0.0	1.0	+T2
+T2	201	1.0	100.0	1.0	1.0	90.0	1.0	2.0	

Field	Contents
IDOBJ	Unique Identification number (Integer > 0) (See Remark 1)
$IDFNC_i$	Identification number of an **TRIMFNC** or **TRIMADD** bulk data card whose value is represented by the symbol F_i shown in the above equation. (Integer > 0)
$C1_i$, $S1_i$, $E1_i$, C2i, $S2_i$, $E2_i$, and E_i	Real coefficients shown in the above equation. (Real) (See Remark 2) Note: Only E_i cannot be zero.

Remarks:

1. IDOBJ is referred to by the **TRIM** bulk data card. The **TRIMOBJ** bulk data card is active only for the over-determined trim system. The resulting objective function is the summary of a set of trim functions combined according to the equation shown above.

2. Since $(C_{1i}\ F_i - S_{1i})$ or $(C_{2i}\ F_i - S_{2i})$ could be negative, the user must select proper value of $E1_i$, $E2_i$ and E_i to avoid complex number resulting from the objective function.

TRIMVAR

Trim Variable Bulk Data Card

Description: Defines a trim variable for the static aeroelastic/trim analysis.

Format and Example:

1	2	3	4	5	6	7	8	9	10
TRIMVAR	IDVAR	LABEL	LOWER	UPPER	TRIMLNK	DMI	SYM	INITIAL	CONT
CONT	DCD	DCY	DCL	DCR	DCM	DCN			

TRIMVAR	100	ALPHA	-50.0	50.0	10	AFORCE	SYM		+T1
+T1	0.01	NONE	NONE	NONE	NONE	NONE			

Field	Contents
IDVAR	Unique identification number. (Integer > 0) (See Remark 1)
LABEL	Character string to define the trim variable. (Character, cannot be blank) (See Remark 2)
LOWER	The lower limit of the trim variable. Active only for the over-determined trim system. (Real, Default = - 1.0 × 10^{30})
UPPER	The upper limit of the trim variable. Active only for the over-determined trim system. (Real, Default = + 1.0 × 10^{30}) (See Remark 3) Note: UPPER must be greater than LOWER.
TRIMLNK	Identification number of a **TRIMLNK** bulk data card for trim variable linking. (Integer ≥ 0) (See Remark 4)
DMI	Optional input. Character string that matches the NAME entry of a **DMI** bulk data card or the MNAME entry of an **'ASSIGN MATRIX='** Executive Control Command. The matrix contains the derivative of the steady pressure distribution with respect to the trim variable. (dC_p / d (trim variable)) (Character, or Blank) (See Remark 5)
SYM	Character string to define the types of the aerodynamic stability derivative generated by the trim variable. (Character) (See Remark 6) SYM = "SYM" for longitudinal stability derivative SYM = "ANTI" for lateral stability derivative SYM = "ASYM" for both longitudinal and lateral stability derivatives
INITIAL	Initial guess of the trim variable for the minimization computation of an over-determined trim system. (Real)

DCD, DCY, DCL, DCR, DCM, DCN	User input aerodynamic stability derivatives. (Character or Real, Default = "NONE") (See Remark 7) Two options are available: Character "NONE" use program-computed value Real Value user input value to replace the program-computed value.

Remarks:

1. IDVAR is referred to by the IDVARi entry in the **TRIM** bulk data card.
2. There are three types of trim variable:

- *The Program-Assigned Trim Variables*

The program-assigned trim variables are those variables whose aerodynamic stability derivatives and the derivatives of the distributed aerodynamic pressures are computed internally by the program. Each program-assigned trim variable has a "hot-wired" label. If the character string specified in the LABEL entry matches the hot-wired label, the program internally computed aerodynamic stability derivatives are used for solving the trim system. These program-assigned trim variables are listed as follows:

Hot-Wired Label	Description	Unit	Type of Aerodynamic Stability Derivatives
ALPHA	Angle of Attack	degree	Longitudinal Stability Derivative
BETA	Side Slip Angle	degree	Lateral Stability Derivative
PRATE	Roll Rate	pb / 2V	Lateral Stability Derivative
QRATE	Pitch Rate	qc / 2V	Longitudinal Stability Derivative
RRATE	Yaw Rate	rb / 2V	Lateral Stability Derivative
THKCAM	Mean Flow Condition defined in the **TRIMFLT** bulk data card and the airfoil camber effects defined in **PAFOIL7/ PAFOIL8** bulk data card are used to provide the aerodynamic forces as a trim variable.	None	Longitudinal Stability Derivative It is required that THKCAM always be included for the symmetric trim system with entry VAL = 1.0 in the **TRIM** bulk data card.

where *p*, *q* and *r* are the roll rate, pitch rate and yaw rate (in rad/sec), respectively, about the aerodynamic moment center REFX, REFY and REFZ defined in the **AEROZ** bulk data card. *b* and *c* are the reference span (REFB) and reference chord (REFC) defined in the **AEROZ** bulk data card. *V* is the free-stream velocity and is not required for input.

The longitudinal aerodynamic stability derivatives are:

$$\frac{d(C_d)}{d(\text{trim variable})} \qquad \frac{d(C_L)}{d(\text{trim variable})} \qquad \frac{d(C_m)}{d(\text{trim variable})}$$

The lateral aerodynamic stability derivatives are:

$$\frac{d(C_y)}{d(\text{trim variable})} \qquad \frac{d(C_l)}{d(\text{trim variable})} \qquad \frac{d(C_n)}{d(\text{trim variable})}$$

where $C_d = \dfrac{D}{q_\infty S}$, *D* is the drag force,

$C_L = \dfrac{L}{q_\infty S}$, is the lift force,

$C_m = \dfrac{M_y}{q_\infty S c}$, M_y is the pitch moment about REFX, REFY and REFZ,

$C_y = \dfrac{Y}{q_\infty S}$, *Y* is the side force,

$C_l = \dfrac{M_x}{q_\infty S b}$, M_x is the roll moment about REFX, REFY and REFZ,

$C_n = \dfrac{M_z}{q_\infty S b}$, M_z is the yaw moment about REFX, REFY and REFZ,

and

q_∞ is the dynamic pressure, *S* is the reference area (REFS),
C is the reference chord (REFC)
and b is the reference span (REFB)

Note: All aerodynamic stability derivatives are for both sides of the configuration even if only half of the configuration (XZSYM = "YES" in the **AEROZ** bulk data card) is modeled.

- *Control Surface Type of the Trim Variables*
  The control surface type of the trim variables are those defined in the **AESURFZ**, **AESLINK, PZTMODE**, and **GRIDFRC** bulk data cards. If the character string specified in the LABEL entry of the **TRIMVAR** bulk data card matches the LABEL entry of **AESURFZ**, **AESLINK, PZTMODE**, or **GRIDFRC**, the program-computed aerodynamic stability derivatives of the control surfaces (**AESURFZ**, **AESLINK, PZTMODE**, or **GRIDFRC**) are used for solving the trim system.

The type of the aerodynamic stability derivatives depend on the TYPE entry in the **AESURFZ**, **AESLINK, PZTMODE**, or **GRIDFRC** bulk data cards. For TYPE = "SYM", they are the longitudinal aerodynamic stability derivatives. For TYPE = "ANTISYM", they are the lateral aerodynamic stability derivatives. For TYPE = "ASYM", they include both longitudinal and lateral aerodynamic stability derivatives.

Note: The unit of the aerodynamic control surface (**AESURFZ** and **AESLINK**) is degrees. The unit of **PZTMODE** and **GRIDFRC** is defined by the users and is marked as "N/A" in the output.

- *User-Defined Trim Variables*
  The character string specified in the LABEL entry that does not match any of the program-assigned and control surface type of the trim variables is classified as user-defined trim variable. For the user-defined trim variables, the entries SYM, DCD, DCY, DCL, DCR, DCM, and DCN in the **TRIMVAR** bulk data card must be specified.

  Note: The unit of the user-defined trim variables is defined by the user and is marked as "N/A" in the output.

3. LOWER and UPPER are the so-called "side constraints" for solving the over-determined trim system. Thus, the solution of the free trim variables (defined as "FREE" in the VALi entry of the **TRIM** bulk data card) must be within LOWER and UPPER.

4. If TRIMLNK = 0, then the trim variable is not linked with other trim variable. For description of trim variable linking, please see **TRIMLNK** bulk data card.

5. **DMI** provides a feature that allows the user to replace the program-computed dC_p / d(trim variable) by those computed by other aerodynamic methods or wind tunnel measurement. The matrix, either imported by the **DMI** bulk data card or the **'ASSIGN MATRIX='** Executive Control Command, must have one column and J*-set rows, where J*-set is the number of aerodynamic boxes of the aerodynamic model. The sequence of the J*-set is: started from the body boxes from the lowest identification number to the highest identification number then followed by the upper surfaces of the CAERO7 boxes from the lowest identification number to the highest identification number, and finally the lower surfaces of the CAERO7 boxes from the lowest identification number to the highest identification number.

   Note: If DMI entry is blank, the program-computed dC_p / d(trim variable) is used for the program-assigned trim variables and control surface type of trim variables. For the user-defined trim variables, dC_p / d(trim variable) is assumed to be zero.

6. For the program-assigned trim variable and the control surface type of trim variables, the SYM entry is ignored since the types of the aerodynamic stability derivatives are already defined by the trim variables. For the user-defined trim variables, the SYM entry must be specified.

7. $$\mathrm{DCD} = \frac{d(C_d)}{d(\text{trim variable})}, \qquad \mathrm{DCY} = \frac{d(C_y)}{d(\text{trim variable})},$$
   $$\mathrm{DCL} = \frac{d(C_L)}{d(\text{trim variable})}, \qquad \mathrm{DCR} = \frac{d(C_l)}{d(\text{trim variable})},$$

$$DCM = \frac{d(C_m)}{d(\text{trim variable})}, \qquad DCN = \frac{d(C_n)}{d(\text{trim variable})}$$

Note: For the user-defined trim variables, DCD, DCY, DCL, DCR, DCM, and DCN cannot be "NONE". Thus, all aerodynamic stability derivatives of the user-defined trim variables must be specified by real values.

VISCOUS — Defines Boundary Layer Parameter

Description: Defines boundary layer parameter for viscous computation analysis.

Format and Example:

1	2	3	4	5	6	7	8	9	10
VISCOUS	IVISC	REYINF	TINF	PRT	RF	ICOUP	S0	XKINT	CONT
CONT	XKINTH	GAIN1	BRK1	BRK2	GAIN2	BRK3	BRK4	PXIMAX	CONT
CONT	PSIMIN	AWK	WKSTRT	WKSTP	IUPDT	LTRANSIT	LBODYBL	LWINGBL	

VISCOUS	100	10E6	360.0	0.90	0.15	1	120.0	0.0001	+ABL
+ABL	0.28	1.0	0.012	0.05	-0.95	-0.0002	0.005	1.0	+ABL
+ABL	-1.0	6.0	0.33	0.666	1	10			

Field	Contents
IVISC	Identification number. The **VISCOUS** bulk data card is referred to by the **TRIMFLT** bulk data card.
REYINT	Reynolds number based on wing reference chord REFC in **AEROZ** bulk data card and freestream conditions. (Real > 0.00, Default = 1.0×10^6)
TINF	Freestream temperature in kelvins. (Real, Default = 300.0)
PRT	Turbulent Prandtl (Prt) number. (Real > 0.0, Default = 0.9)
RF	Relaxation factor for Carter's IBL coupling method. (Real > 0.0, Default = 0.1)
ICOUP	Viscous coupling method 0 - Edwards' variable gain IBL coupling method. 1 - Carter's IBL coupling method. Usually, Carter's method converges faster but is less robust than Edwards' method. For a case with large separation bubble, Edwards's method is recommended. (Integer, Default = 0)
S0	Sutherland law viscosity constant in kelvins. (Default value is for air). (Real, Default = 110.0)

XKINT	Variable gain IBL coupling method gain factor. The input to the integrator is the error between the inner, viscous edge-velocity, U_e^v, and the outer, inviscid velocity, U_e^i. The output of the integrator is the displacement thickness, δ^*. This is the primary control over the viscous-inviscid interaction. Too large a gain will cause code failure. Too low a gain will give a sluggish boundary layer behavior. Set XKINT at approximately 1/2 – 3/4 of the gain causing code failure. A reasonable range for XKINT is 0.00005 – 0.00030. (Real, Default = 0.00015)
XKINTH	Low pass filter difference equation time constant for estimate of momentum thickness. (Real, Default = 0.25)
GAIN1, BRK1, BRK2, GAIN2, BRK3, BRK4	These six parameters define the nonlinear scheduling of the boundary layer integrator gain which is implemented as XKINT × t2 × (1.0 + t2 × t3). t2 = f(GAIN1, BRK1, BRK2, δ^*) is a nonlinear gain scheduled on δ^* rising from a value of 1.0 for values of δ^* less than BRK1 to a value of GAIN1 for δ^* greater than BRK2. t3 = f(GAIN2, BRK3, BRK4, C_f) is a nonlinear gain scheduled on C_f falling from a value of 0.0 for C_f greater than BRK4 to a value of GAIN2 for C_f less than BRK3. The intent of the first schedule, t2, is to increase the integrator gain in regions of large displacement thickness in order to capture cases of large dynamic boundary layer motions such as self-excited transonic shock oscillations. Such calculations for airfoils have been successful with GAIN1 = 5.0, BRK1 = 0.03, BRK2 = 0.05, and t3 = 0.0. The intent of the second schedule, t3, is to decrease the integrator gain in regions of fully separated flow. In such regions, the inverse form of the boundary layer equations become insensitive to δ^* and the integrator operates essentially 'open-loop' and can drive δ^* to unrealistically large values. (Real) (default: GAIN1 = 1.0, BRK1 = 0.01, BRK2 = 0.05, GAIN2 = -0.95, BRK3 = -0.0002 and BRK4 = 0.005)
PXIMAX	Not used.
PXIMIN	Not used.
AWK WKSTRT WKSTP	Wake modeling parameters. The wake displacement thickness for the upper and lower wake surfaces are modeled as decaying exponentially from their value at the trailing edge to a final value (fraction of δ^*_{TE}) which is scheduled on the trailing-edge shape factor, H_{TE}. The exponential decay constant is AWK. For H_{TE} less than 2.0 the final value is given by WKSTRT, for H_{TE} great than 4.0 it is given by WKSTP, and a linear variation is applied between these limits. (Real, Default: AWK = 6.0, WKSTRT = 0.333 and WKSTP = 0.666)
IUPDT	Not used.
LTRANSIT	Transition point from laminar to turbulent flow in percentage of chord. (Integer > 0, Default = 20)

LBODYBL	Boundary-layer option for the body 1 boundary-layer calculation on the body is turned on for all **BODY7** bulk data cards 0 boundary-layer calculation on the body for all **BODY7** bulk data cards is turned off If LBODYBL is an negative integer, it's absolute value refers to the identification number of a **SET1** bulk data card in which a set of identification numbers of the **BODY7** Bulk data card is listed. On those **BODY7,** the boundary layer calculation is turned off. For all other **BODY7,** the boundary layer calculation is turned on (Integer, Default = 0)
LWINGBL	Same as LBODYBL except for **CAERO7** bulk data cards. (Integer, Default = 1)

YZONEY — *y* Location

Description: Defines a *y* location by referring to a **CAERO7** or **BODY7** bulk data card.

Format and Example:

1	2	3	4	5	6	7	8	9	10
YZONEY	IDY	IDI	RORL						

YZONEY	100	2000	R						

Field	Contents
IDY	Identification number that is referred to by a **BLOCK** bulk data card. (Integer > 0) (See Remark 1)
IDI	Identification number of a **CAERO7** or **BODY7** bulk data card. (Integer > 0)
RORL	Character either “R” or “L”. The *y* location is selected from the smallest y location for RORL = “L” and the largest *y* location for RORL = “R” of the surface grid point of the selected **CAERO7**/**BODY7** bulk data card. (Character) (See Remark 2)

Remarks:

1. **YZONEY** is referred to by the entry YR*i*/YL*i* in the **BLOCK** bulk data card or the YZONEL/YZONER entry of a **GAP1** bulk data card to provide a convenience for defining the *y* value of a y-zone. This can ensure the exact alignment of the side edge of a **CAERO7**/**BODY7** bulk data card with the inboard/outboard of a y-zone. In addition, if the side edge of the **CAERO7**/**BODY7** bulk data card is changed, the inboard/outboard of a y-zone is changed automatically.

2. All surface grid points of the **CAERO7**/**BODY7** bulk data card are defined in the local coordinate system specified by the **ACOORD** bulk data card which is referred to by the **BLOCK** bulk data card.

Chapter 5

AERODYNAMIC MODELING GUIDELINES

There are generally two steps involved in generating a mesh system for executing ZEUS, namely the surface mesh generation and the flowfield mesh generation. ZEUS solves the Euler equation on the flowfield mesh whereas the surface mesh is used only as a reference mesh for the automated mesh generation scheme to generate the flowfield mesh. However, after the solution is obtained on the flowfield mesh, ZEUS interpolates the solution from the flowfield mesh to the surface mesh so that the final solution is always presented on the surface mesh. The input of the surface mesh is nearly identical to that of ZAERO. However, because ZEUS employs the finite volume method to solve the Euler equation on the flowfield mesh, whereas ZAERO uses the panel method to solve the potential equation on the surface mesh, some of the modeling guidelines for the ZEUS surface mesh input are different from those of ZAERO. Once the surface mesh is defined, ZEUS can automatically generate a block of flowfield mesh by growing the mesh from the surface mesh if all aerodynamic components such as fuselage, wing, and tail can be fitted into such a block of flowfield mesh. For complex configuration such as aircraft with stores, multiple blocks of flowfield mesh are required to generate an overset mesh system where overlapping of flowfield meshes among different blocks of mesh are allowed. During each time integration step, a solution at each block of mesh is first computed by running the Euler-solver module independently. Flow communication between the blocks of mesh is obtained by interpolating flow solutions in the overlapping region. Solution convergence within each time step is achieved by Newton sub-iterations.

5.1 SURFACE MESH INPUT

To establish a ZEUS surface mesh for an aircraft configuration, it requires dividing the configuration into wing-like and body-like components. The wing-like components are the lifting surfaces whose chordwise cross-sections can be represented by airfoil-like thickness distribution. These types of components include wings, tails, fins, pylons, and launchers. The body-like components are the non-lifting types of body such as fuselage, nacelles, missile bodies, and stores.

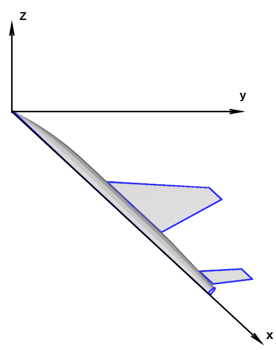


Figure 5.1 The Global Aerodynamic Coordinates of ZEUS

ZEUS inherently assumes the flow direction at zero angle of attack to be along the X-axis of the global aerodynamic coordinates. Thus, as shown in

Figure 5.1, the fuselage must be located along the X-axis and the canopy, from the pilot's view, must be located towards the positive Z-axis. If only a half-span model is required, the model must be located on the positive Y-axis.

5.1.1 MODELING OF WING-LIKE COMPONENTS

The wing-like component is defined by specifying a **CAERO7** bulk data card. Each **CAERO7** defines a trapezoidal flat sheet of surface in which the inboard and outboard edges are parallel to the X-axis. The thickness distribution of the wing-like component is defined by the **PAFOIL7** bulk data card which does not affect the position of the flat sheet of surface, as will be discussed later because the thickness effects are only introduced in the boundary condition. This trapezoidal flat sheet of surface is then further divided into several strips by user-specified spanwise divisions. Each strip is then divided into several boxes (called wing boxes) by chordwise divisions specified at the inboard and outboard edges. Thus, each **CAERO7** represents a wing macroelement comprising of $(n - 1) \times (m - 1)$ wing boxes (where n = the number of spanwise division and m = the number of chordwise divisions). A typical wing macroelement by specifying the **CAERO7** bulk data card is shown in Figure 5.2.

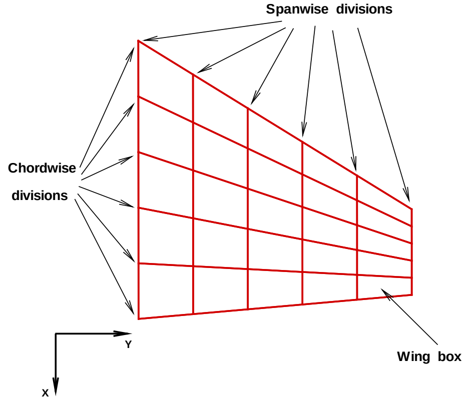


Figure 5.2 Wing Macroelement by CAERO7 Bulk Data Card

It should be noted that the flat sheet of surface defined by **CAERO7** bulk data card represents the mean plane of a wing-like component. This implies that ZEUS does not physically model the thickness of the wing-like components in the mesh; only the mean plane of the wing-like component needs to be fitted into the flowfield mesh. The thickness effects are introduced in the boundary condition using the transpiration boundary condition technique and applied on the mean plane. This technique can be illustrated in Figure 5.3 where only the mean plane of the wing-like component is required to align with a line of the flowfield mesh. Boundary condition with thickness effects is applied on the upper and lower surfaces of the mean plane. This can significantly reduce the mesh generation effort because conformal mesh (gridline conformed to the exact upper and lower surfaces) for the wing-like component can be avoided. The use of the transpiration boundary condition is one of the reasons why the automated mesh generation capability of ZEUS is able to be developed.

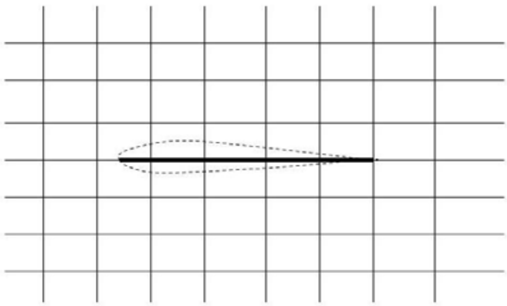


Figure 5.3 Small Disturbance Boundary Condition Technique for Thickness Effects on Lifting Surfaces

The transpiration boundary condition actually works very well even in handling wings with a thick airfoil section. Shown in Figure 5.4 is the steady C_p on the LANN wing along six span stations at M = 0.822 and angle of attack (AoA) = 0.6°. The LANN wing has a supercritical airfoil section with the maximum thickness being equal to 12% of the chord. The aspect ratio, taper ratio and swept angle of the LANN wing are 7.92, 0.4 and 25°, respectively. In addition, the LANN wing has a varying twist angle from 2.6° at wing root to -2.0° at wing tip. However, even for such a thick and twist wing in transonic flow, the ZEUS computed C_p distribution, as shown in Figure 5.4, correlates very well with the experimental data, demonstrating the applicability of the transpiration boundary condition in handling the thick-wing components.

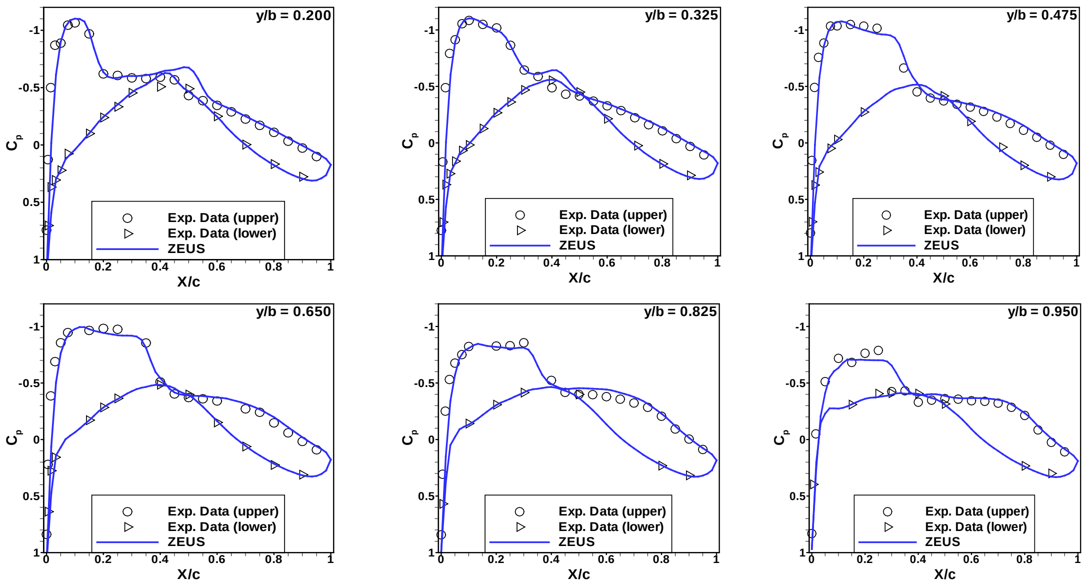
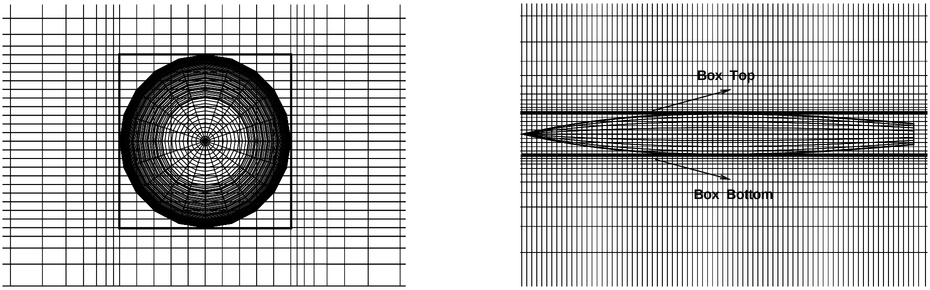


Figure 5.4 Steady C_p on LANN Wing at M= 0.822, AOA = 0.6°

5.1.2 MODELING OF BODY-LIKE COMPONENTS

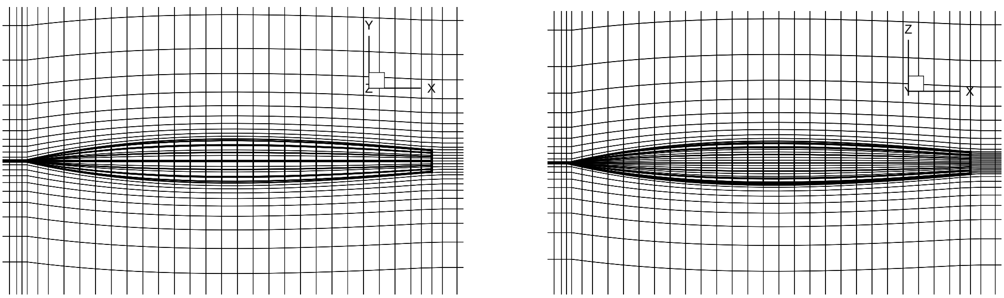
The body-like component is defined by specifying a **BODY7** bulk data card that further refers to a **SEGMESH** bulk data card in which the surface grid points of the body are specified. Unlike ZAERO where multiple **SEGMESH** bulk data cards can be referred to by one **BODY7** bulk data card, ZEUS allows only one **SEGMESH** bulk data card, per **BODY7** bulk data card. The reason behind this is that if multiple **SEGMESH** bulk data cards are allowed, ZEUS would have difficulty to interpolate the solution from the flowfield mesh to the surface mesh of the body.



(a) Y-Z Plane *(b) X-Z Plane*

Figure 5.5 Modeling of Body by a Rectangular Box

ZEUS can automatically generate a block of Cartesian mesh using the **BLOCK** bulk data card that approximates the exact surface geometry of bodies by a rectangular box, such as the one shown in Figure 5.5(a) and 5.5(b). In so doing, the rectangle box can be fitted into an orthogonal mesh on a Y-Z plane and the X-Z plane. Using a slender body theory, the exact surface geometry of the body is mapped onto the surface of the rectangular box where the boundary condition of the body is applied. This significantly reduces the grid generation effort for modeling bodies because no conformal mesh is required. However, as an option, ZEUS also allows a sheared mesh on the X-Y plane and X-Z plane near the body surface such as the one shown in



(a) X-Y Plane *(b) X-Z Plane*

Figure 5.6 Sheared Mesh Near the Body Surface

Figure 5.6 to partially accommodate the variation of the height and width along the body. Using this option, at each X station, a rectangular box is formed into which the height and width of the body at this X station is fitted. Thus, this option can form a set of rectangular boxes with varying height and width along the body. The strategy of modeling the body surface by a rectangular box using the slender body theory actually works surprisingly well. Shown in Figure 5.7 is the pressure coefficient correlation between the ZEUS solution and an Euler solution with the exact boundary condition along a Hemisphere-Cylinder body at AOA = 5°, M = 0.9, and M = 1.2. The good correlations between these two sets of solution validate the rectangular box modeling technique for bodies.

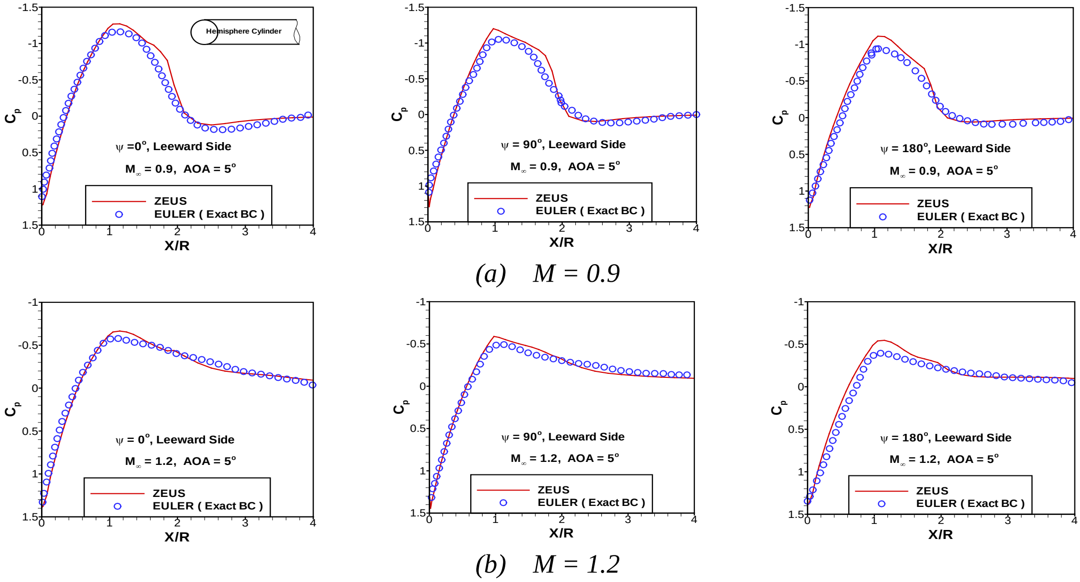
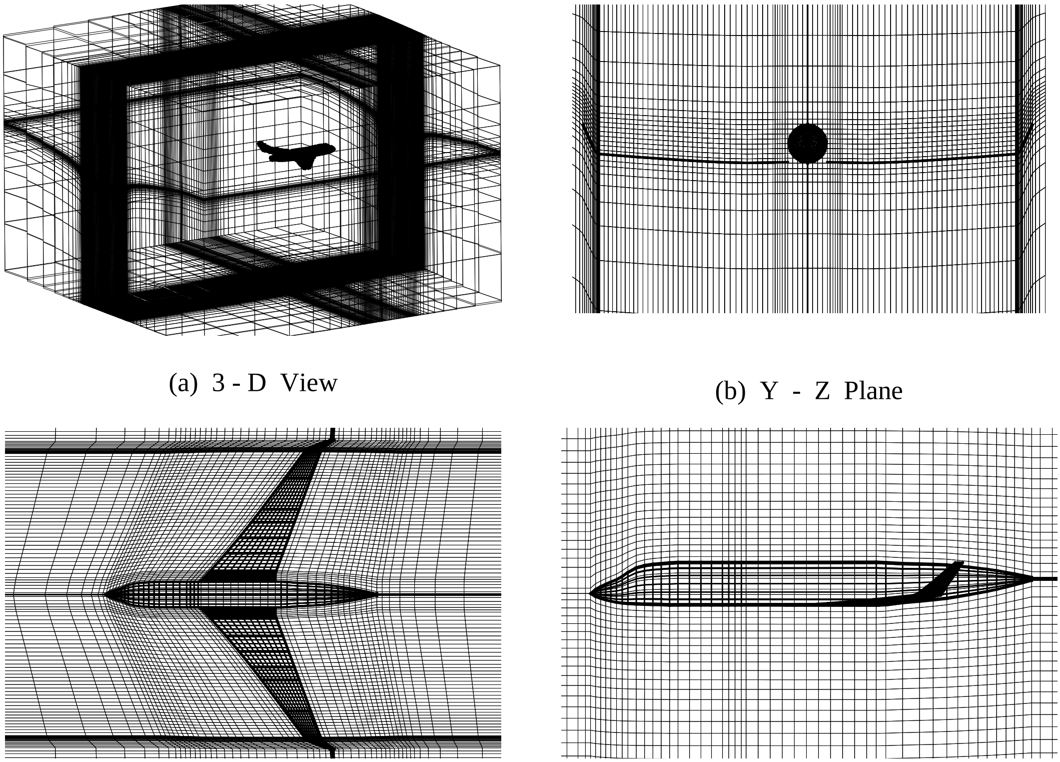


Figure 5.7 Pressure Coefficients Along a Hemisphere – Cylinder Body

As an option, the BODY7 macroelement can be also modeled by a **BLOCK1** bulk data card that automatically generates a body-fitted mesh around the body. In this case, exact Euler boundary condition is applied on the body surface mesh to provide more accurate solution. However, the drawback of the **BLOCK1** bulk data card is that only one BODY7 macroelement is allowed to be embedded in this body-fitted mesh. For a wing-body configuration, the wing mesh must be generated by the **BLOCK/BLOCKT** bulk data card; rendering an overset mesh which increases the number of grid points and computational time.

5.2 GENERATION OF A BLOCK OF FLOWFIELD MESH

A typical flowfield mesh of a fuselage and wing with winglet configuration is shown in Figure 5.8. In order to clearly describe how such a block of mesh is generated, let us first classify the definition of lines that construct the mesh. Note that the X, Y, and Z axis in which the mesh is generated are actually defined in a local coordinate system specified by an **ACOORD** bulk data card. If no **ACOORD** bulk data card is referred to by the **BLOCK/BLOCKT** bulk data card, the global aerodynamic coordinate system is assumed.



(a) 3 - D View

(b) Y - Z Plane

(c) X - Y Plane

(d) X - Z Plane

Figure 5.8 A Block of Flowfield Mesh of a Fuselage and Wing with Winglet Configuration

(1) Mesh Projected on an X-Y Plane

Chordwise Y-line: The line runs from the left wing to the right wing at each X station. If the wing is un-swept, the chordwise Y-line is parallel to the Y-axis. However, the chordwise Y-line can be sheared to accommodate the sweep angle of the wing leading and trailing edge.

Spanwise X-line: The line runs from upstream to downstream at each spanwise location. The spanwise X-line is normally parallel to the X-axis. However, it can be sheared to accommodate the variation of the body width along the streamwise direction.

Figure 5.9 shows how the chordwise Y-lines and the spanwise X-line jointly construct a mesh on a X-Y plane. Note that this mesh on the X-Y plane is invariant even at different Z station because all vertical lines in the 3D mesh are all parallel to the Z-axis. If there are *m* numbers of chordwise Y-lines from upstream to downstream and *n* number of spanwise X-lines from the left wing to the right wing, then the number of grid points on the X-Y plane is $m \times n$.

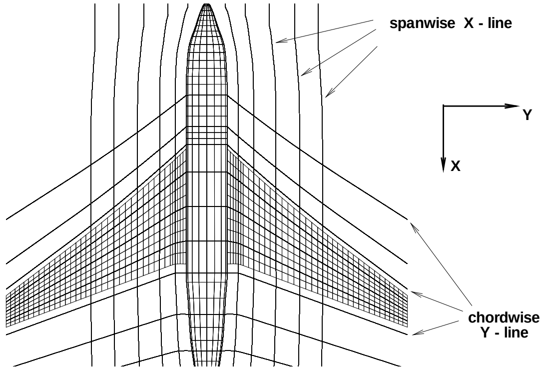


Figure 5.9 Chordwise Y-Lines and Spanwise X-Lines on the X-Y Plane

(2) Mesh projected on a Y-Z Plane:

Horizontal Y-line: The line runs from the left wing to the right wing at each Z station. If the wing has no dihedral angle, the horizontal Y-line is parallel to the Y-axis. However, the horizontal Y-line can be sheared to accommodate the dihedral angle of the wing.

Spanwise Z-line: The line runs from bottom to top at each spanwise location. The spanwise Z-line is always parallel to the Z-axis.

Figure 5.10 shows how the horizontal Y-lines and spanwise Z-lines jointly construct the mesh on a Y-Z plane. The mesh on the Y-Z plane could be varying at each X station because of the sheared spanwise X-lines on the X-Y plane. If there are *l* number of horizontal Y-lines and *n* number of spanwise Z-lines, the number grid points on the Y-Z plane is $l \times n$. Note that the number of spanwise Z-lines on the Y-Z plane must be equal to the number spanwise Y-line on the X-Y plane.

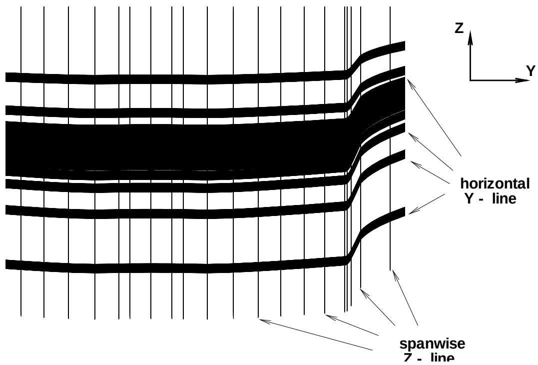


Figure 5.10 Horizontal Y-lines and Spanwise Z-lines on the Y-Z Plane

(3) Mesh Projected on an X-Z Plane:

Horizontal X-line: The line runs from upstream to downstream at each Z station. The horizontal X-line is normally parallel to the X-axis. However, it can be sheared to accommodate the variation of the height of the body along the streamwise direction.

Chordwise Z-line: The line runs from bottom to top at each X station. They are parallel to the Z-axis.

Figure 5.11 shows how the horizontal X-lines and the chordwise Z-lines jointly construct a mesh on an X-Z plane. Note that the mesh on the X-Z plane could be varying at each span station because of the sheared chordwise Y-lines on the X-Y plane. The horizontal X-lines are obtained by sweeping the horizontal Y-lines on the Y-Z plane from upstream to downstream. Therefore, if there is *l* number of horizontal Y-lines on the Y-Z plane, the number of horizontal X-lines on the X-Z must be *l*. Likewise, the chordwise Z-lines are obtained by sweeping the chordwise Y-lines on the X-Y plane from bottom to top. Therefore, the number of chordwise Z-lines must be equal to the number of chordwise Y- lines on the X-Y plane; i.e., *m* number of chordwise Z-lines on the X-Z plane. The total number of grid points on an X-Z plane is $m \times l$.

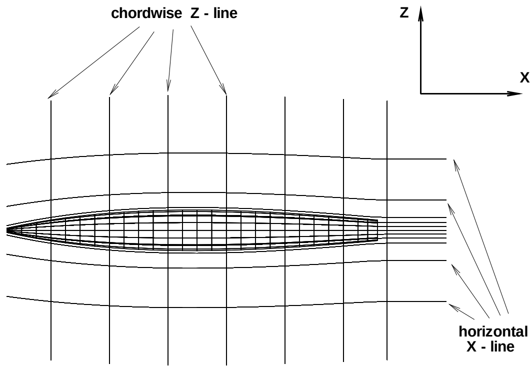


Figure 5.11 Horizontal X-Lines and Chordwise Z-Lines on the X-Z Plane

5.3 AUTOMATED MESH GENERATION OF A BLOCK OF FLOWFIELD MESH

ZEUS employs an automated mesh generation scheme that requires only the surface mesh of the lifting surfaces (by the **CAERO7** bulk data card) and the bodies (by the **BODY7** bulk data card) as input. This automated mesh generation is accomplished by specifying a **BLOCK/BLOCKT** bulk data card that first defines the outer boundary of the flowfield mesh, i.e., the size of the flowfield mesh. Then it refers to a set of **CAERO7** and **BODY7** bulk data cards from whose surface meshes the flowfield mesh is generated. For the rule of selecting those **CAERO7** and **BODY7** that belong to a single **BLOCK/BLOCKT** bulk data card, please see the remarks of the **BLOCK/BLOCKT** bulk data card. In addition, to enhance the quality of the mesh or to add more grid points to the mesh, the user can specify a **GAP, GAP1** and/or **GAPZ** bulk data cards to define a fictitious surface on the X-Y plane and/or Y-Z plane, respectively, on which the grid lines can be added or adjusted. The **BLOCK/BLOCKT** bulk data card also refers to an **ACOORD** bulk data card to define a local coordinate system. In fact, the flowfield mesh is first generated in such a local coordinate system then transform back to the global coordinate system.

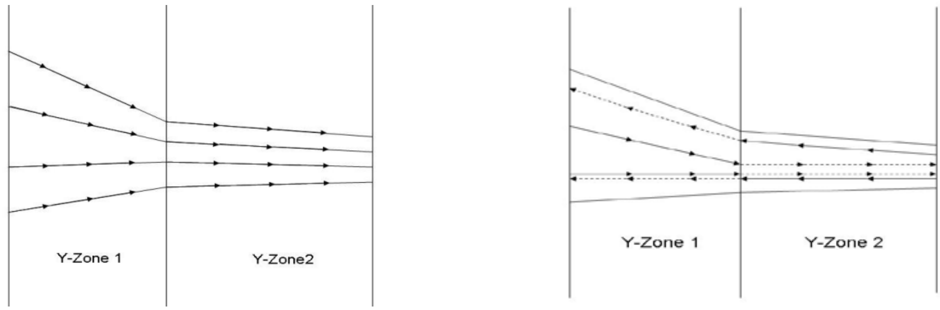
The **BLOCK/BLOCKT** bulk data card activates a Y-zone technique that divides those selected **CAERO7** and **BODY7** into several spanwise zones, called the Y-zones. In each Y-zone, a set of chordwise Y-lines on the X-Y plane and horizontal Y-lines on the Y-Z plane are first generated. Connections of those chordwise Y-lines from the adjacent Y-zone are accomplished by a line-tracing method (see remarks of the **BLOCK** bulk data card). The same line-tracing method is also applied on the mesh projected on the Y-Z plane to connect the horizontal Y-lines from the adjacent Y-zone. The

Y-zone technique enables the generation of a sheared mesh on the X-Y plane to accommodate the sweep angle of the wing-like components and a sheared mesh on the Y-Z plane to accommodate the dihedral angle of the wing-like components.

However, the Y-zone technique can not generate a sheared mesh on the X-Z plane. This implies that a vertical tail with sweep angle cannot be fitted into such a mesh. For a T-tail configuration, if both the horizontal and vertical tails are included in one **BLOCK** bulk data card, the vertical tail is approximated by an unswept rectangular surface which may lead to inaccurate result. One can use two **BLOCK** bulk data cards to model the T-tail configuration; one for the horizontal tail and the other for the vertical tail except in a local coordinates with 90 degrees dihedral angle. In this local coordinates, the vertical tail becomes a horizontal surface so that a sheared mesh on the local X-Y plane can be generated to accommodate the sweep angle of the vertical tail. However, this results an overset mesh which also sometimes decreases the solution accuracy. To circumvent the T-tail issue, the **BLOCKT** bulk data card is developed to generate only one block of mesh to model the T-tail configuration; thereby avoiding the use of overset mesh.

5.3.1 SHEARED MESH ON THE X-Y PLANE

The Y-zone technique first generates a set of chordwise Y-lines in each Y-zone on the surface mesh of the wing-like component according to the chordwise divisions defined in the **CAERO7** bulk data card. Each chordwise division of the surface mesh generates a chordwise Y-line. The connection of those chordwise Y-lines between adjacent Y-zones is achieved by a line-tracing method. Shown in Figure 5.12 is a two adjacent **CAERO7** macroelements in two Y-zones, respectively, where the line-tracing method “traces” the chordwise Y-lines (defined by chordwise divisions of the **CAERO7** bulk data card) of the Y-zone 1 to connect those of Y-zone 2. In Figure 5.12(a), because of the coherent chordwise Y-lines between the two **CAERO7** macroelements (chordwise divisions are aligned with each other along the juncture of the two **CAERO7** macroelements), the chordwise Y-lines can be well connected across the two Y-zones, and no extra chordwise Y-lines on the surface are generated. Otherwise, extra chordwise Y-lines will be generated (shown by the dashed-lines in Figure 5.12(b)) if the surface meshes between two macroelements are not coherent. Extra lines can adversely affect the smoothness of the mesh and also increase the computational time. Therefore, it is highly recommended that all adjacent microelements either between two **CAERO7**’s or **CAERO7** and **BODY7** have a coherent surface mesh at their juncture. Note that the coherence of the chordwise Y-line is detected by a small tolerance, XTOL × DXMIN, where DXMIN is the smallest distance between chordwise cuts of **CAERO7**/**BODY7** macroelements, and XTOL=0.25 as the default value but it can be altered by the **MESHPRM** bulk data card. For any two chordwise Y-lines whose maximum distance is smaller than this small tolerance, one of the chordwise Y-lines will be removed from the mesh.

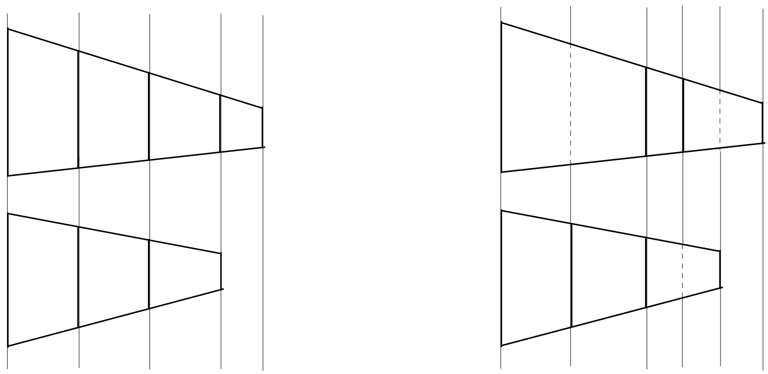


(a) Coherent Meshes between Two CAERO7 *(b) Non Coherent Meshes between Two CAERO7*

Figure 5.12 Line-Tracing Method Across Y-Zones

After the chordwise Y-lines on the surface mesh are generated, the chordwise Y-lines ahead of the leading edge of **CAERO7** (or nose of **BODY7**) and behind the trailing edge of **CAERO7** (or end of **BODY7**) in the flowfield are automatically generated by growing the chordwise Y-lines from the wing leading and trailing edges along the upstream and downstream directions, respectively, to the outer boundary of the block. The growth rate of those chordwise Y-lines in the flowfield is 1.3 as the default value. The default value can be changed using the **MESHPRM** bulk data card.

The spanwise X-lines on the surface mesh are generated according to the spanwise divisions of **CAERO7**. Each spanwise division of the surface mesh generates a spanwise X-line. Shown in Figure 5.13 is the spanwise X-lines generated on a wing-tail configuration. If the spanwise divisions between the wing and tail are aligned with each other, no extra Spanwise X-line is generated (5.13(a)). However, if they are not aligned, extra spanwise X-lines are generated as shown by those dash-lines in Figure 5.13(b). Again, the detection of the spanwise alignment is determined by a small tolerance, YTOL × DYMIN, where DYMIN is the smallest distance between spanwise cuts of **CAERO7**/**BODY7** macroelements, and YTOL=0.25 as the default value but it can be altered by the **MESHPRM** bulk data card. If the maximum distance between two adjacent spanwise X-lines is smaller than this small tolerance, one of the spanwise X-lines will be removed from the mesh.



(a) Aligned Spanwise Divisions *(b) Non-Aligned Spanwise Divisions*

Figure 5.13 Spanwise X-Lines Generation on a Wing-Tail Configuration

After the spanwise X-lines are generated on the surface mesh, the spanwise X-lines in the flowfield are generated by growing from the right and left wing tip to the outer boundary of the block of mesh. The growth rate is 1.3 as a default, but it can be changed using the **MESHPRM** bulk data card.

The spanwise X-lines can be also sheared if the X-Y plane Shrink-body option in the **SEGMESH** bulk data card is activated. In such case, the spanwise X-lines near the body surface will be sheared to accommodate the body width variation along the body. This results in a sheared mesh due to the sheared chordwise Y-lines (due to wing sweep angle) and the sheared spanwise X-lines (due to width variation of body) on the X-Y plane such as the one shown in Figure 5.8(c).

5.3.2 SHEARED MESH ON THE Y-Z PLANE

Within each Y-zone, a set of horizontal Y-lines are first generated by growing the horizontal Y-lines from the mean plane of **CAERO7** and the top and bottom surface of the rectangular box **BODY7** along the Z-axis to the outer boundary of the block. Figure 5.14 shows the grid space of the horizontal Y-line near the surface which is determined by DZFACT × REFC, where DZFACT = 0.01 as the default value but it can be altered by the **MESHPRM** bulk data card. Please note, the first horizontal Y-line grid space towards the top and bottom of the body is 1.5 times that of the horizontal surface. The growth of the horizontal Y-lines from the surface to the outer boundary of the entire Z-domain is accomplished by a growth rate of 1.3. This growth rate can be altered using the **MESHPRM** bulk data card. After the horizontal Y-lines in all Y-zones are generated, the Y-zone technique connects the horizontal Y-lines across Y-zones using the line-tracing method. For any two horizontal Y-lines whose maximum distance is within a small tolerance, ZTOL × DZMIN, one of the horizontal lines will be removed from the mesh. The definition of DZMIN is similar to DXMIN, and ZTOL=0.25 as the default value but it can be altered by the **MESHPRM** bulk data card.

The spanwise Z-lines on the Y-Z plane are obtained by simply sweeping the spanwise X-lines on the X-Y plane vertically, i.e. parallel to the Z-axis. The intersection between the horizontal Y-lines and the spanwise Z-lines forms the mesh on the Y-Z plane.

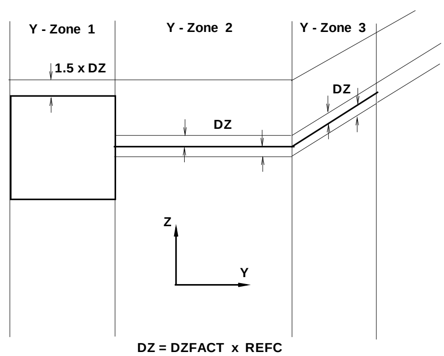


Figure 5.14 Grid Space of the Horizontal Y-Lines Near the Surface

5.3.3 SHEARED MESH ON THE X-Z PLANE

The horizontal X-lines on the X-Z plane are obtained by sweeping the horizontal Y-lines on the Y-Z plane from upstream to downstream. However, as an option that is defined in the **SEGMESH** bulk data card, these horizontal X-lines can be sheared to accommodate the variation of body height along the body.

The chordwise Z-lines are obtained by simply sweeping the chordwise Y-lines on the X-Y plane vertically, i.e. along the Z-axis. The intersection between the horizontal X-lines and chordwise Z-lines forms the mesh on the X-Z plane.

5.3.4 THE FINAL MESH FOR A BLOCK

If there are *m* number of chordwise Y-lines on the X-Y plane, *n* number of spanwise X-lines on the X-Y plane and *l* number of horizontal Y-line on the Y-Z plane, the total number of grid points and the total number of grid cells of a block of mesh are $m \times n \times l$ and $(m-1) \times (n-1) \times (l-1)$, respectively. ZEUS applies the finite volume method to evaluate the in-flux and out-flux of each cell and computes the flow solution at each grid cell center. After the solution is obtained, ZEUS interpolates the solution at those grid cells near the surface mesh to the centroid of **CAERO7** and **BODY7** panels. Therefore, the final solution of ZEUS is always presented on the **CAERO7** and **BODY7** panel model.

5.4 OVERSET MESH SCHEME

ZEUS offers an overset mesh scheme to model the complex configuration whose components cannot be entirely fit into a single block of mesh. To examine whether the overset mesh scheme must be activated or not, the user must project the whole configuration onto the X-Y plane and Y-Z plane of the aerodynamic coordinates. If there is any overlapping of components on the projected X-Y plane or Y-Z plane, then those overlapped components should be fitted into different blocks of mesh by specifying multiple **BLOCK** bulk data cards. For instance, for a wing with underwing store configuration as shown in Figure 5.15, the wing and store are overlapped on the projected X-Y plane. Therefore, the wing and store must be fitted into two different blocks of mesh. For a wing and V-tail configuration as shown in Figure 5.16, because the V-tail has a large dihedral angle that intersects the wing on the projected Y-Z plane, the wing and the V-tail must be fitted into two different blocks of mesh, too.

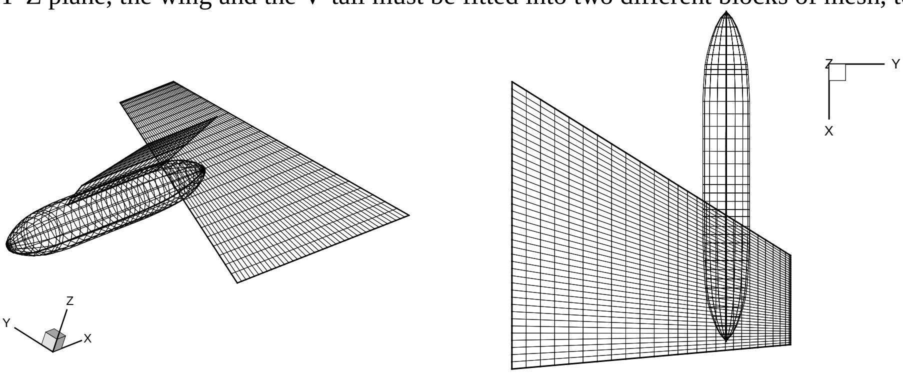


Figure 5.15 Overlapped Wing and Store on the Projected X-Y Plane

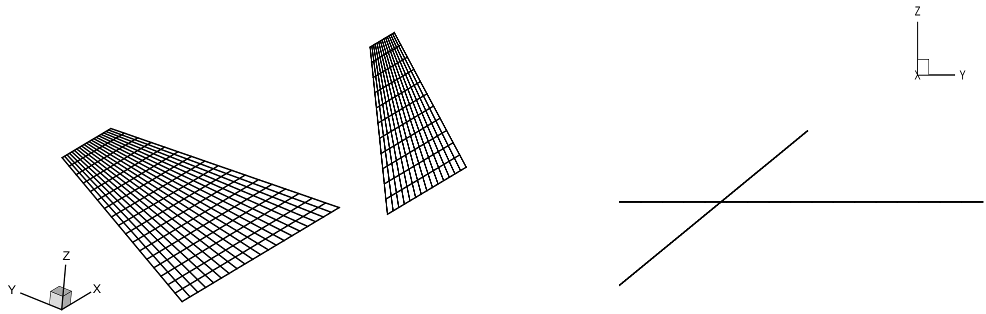


Figure 5.16 Intersected Wing and V-Tail on the Projected Y-Z Plane

The overset mesh scheme first performs a “hole-cutting” procedure in the overlapping regions of the overset mesh, which establishes a so-called “IBLANK” array for each block to “blank” out the flow solution in the overlapped regions that will receive data from other blocks. The interpolation coefficients are also figured out during this “hole-cutting” procedure using a connectivity algorithm. Solution convergence within each time step is achieved by Newton sub-iterations. During each Newton sub-iteration, the flow solution in each block of mesh is first computed independently. Then flow communication among blocks is obtained by interpolating flow solution in the overlapping regions. Note that the default number of the Newton sub-iterations is 4, but it can be altered using a **MKPARAM** bulk data card with entry NEWTN.

In principle, each component of a configuration can be defined in an individual block of mesh and relies on the overset mesh scheme to establish the flow communication with other components. However, this is not recommended because the interpolation of flow solution in the overlapping region introduces error that reduces the accuracy of the solution. Furthermore, overset mesh increases the number of grid points and requires Newton sub-iterations for solution convergence which could lead to a much longer computational time. Therefore, the user should try to fit all components into a single block of mesh if this is possible. In other words, overset mesh scheme should not be activated if all components of a configuration can be fitted into a single block of mesh.

5.4.1 AN EXAMPLE OF OVERSET MESH

In this section, an example of a half model aircraft is illustrated in Figure 5.17. The aircraft has one underwing tank and two underwing missiles, for which it will require three blocks for the underwing bodies and a total of eight blocks for the canards and fins as shown in Figure 5.18 and as illustrated and explained subsequently. As a rule of thumb, the fuselage and all the horizontal surfaces in the plane of the main wing is usually accommodated in the first block, which is the biggest of blocks and extends to the far field of the flow. This block is also called the global block and is shown in Figure 5.19.

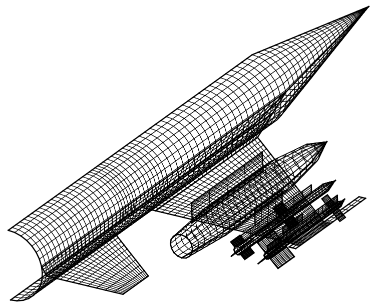


Figure 5.17 A Half Model Aircraft with Tip Missile Launcher, One Underwing Tank, and Two Underwing Missiles

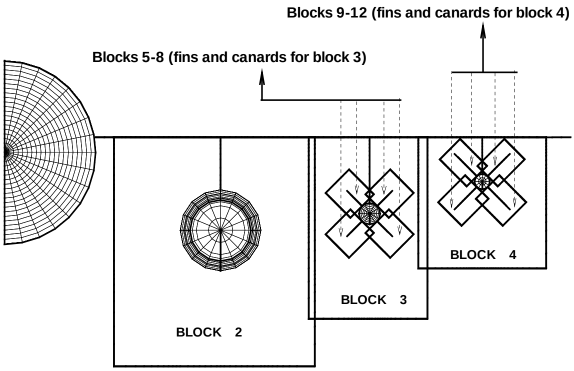


Figure 5.18 Y-Z View of a 12 Block Modeling of a Half Aircraft with Tip Missile Launcher, Underwing Tank and Missiles Configuration

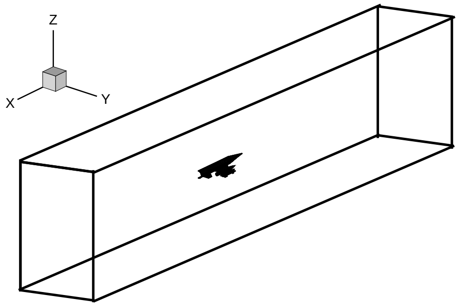


Figure 5.19 Block 1 Consisting of the Fuselage and all the Horizontal Surfaces in the Same Plane of the Main Wing

Block 1 consists of fuselage, wing and horizontal tail, tip launcher, and strakes. They are all defined in the global aerodynamic coordinates, i.e., IACORD = 0 in the **BLOCK** bulk data card.

Block 2 consists of the underwing tank and supporting pylons for the tank. Note that the pylon is a vertical lifting surface that normally cannot be fitted into an X-Z plane of mesh if the pylon has a non-zero sweep angle. It must be treated as a horizontal surface and fitted into an X-Y plane of a separated block of mesh. This mesh is defined in a local coordinate system using an **ACOORD** bulk data card with the entry THETA=90°. However, in the present example, because the pylons have no sweep angle, they all can be fitted into block 2.

Blocks 3 and 4 are the blocks for the underwing missile bodies, excluding canards and fins.

Blocks 5-8 consist of the canards and fins for the inboard missile. The local coordinates of these four blocks will be 45, 135, 225, and 315° of dihedral, which can be achieved by the **ACOORD** bulk data card.

Blocks 9-12 consist of the canards and fins for the outboard missile. The local coordinates of these four blocks will be similar to the earlier four blocks.

It should be noted that the overset mesh scheme generally requires the smaller block of mesh to be embedded in the bigger block of mesh. In such case, the bigger block will provide solution of flow variables for the outermost two layers of grid cells of the smaller block as its "far-field" boundary condition, while the resolved flow variables of the inner part of the smaller block feed back to the bigger block. Therefore, for the example shown in Figure 5.18, blocks 2, 3 and 4 should be embedded in block 1 and blocks 5-12 should be embedded in blocks 3 and 4, respectively. This sequence of blocks for the overset mesh strategy is specified by the **BLKSEQ** bulk data card that requires a list of the identification numbers of the **BLOCK, BLOCKT** and **BLOCK1** bulk data cards from the biggest global block to the smallest block of the mesh. If this sequence is not provided, a default **BLKSEQ** bulk data card will be used, and the default sequence of the overset mesh strategy goes from the **BLOCK/BLOCKT** bulk data card with the smallest identification number to the largest identification number in the ascending order then followed by the **BLOCK1** bulk data cards, again, in the ascending order of the identification numbers. On top of **BLKSEQ** bulk data card, **BLKLAY** bulk data card can be used to impose additional control of the overset scheme. The **BLKLAY** bulk data card specifies a layer number for the corresponding **BLOCK**, **BLOCKT** and **BLOCK1** bulk data card listed in the **BLKSEQ** bulk data card. If no **BLKLAY** bulk data card exists, the overset strategy is performed only according to the block sequence specified by the **BLKSEQ** bulk data card. With a **BLKLAY** bulk data card, the overset strategy can perform another check for the layering number of each block. Basically, the blocks with the same layer number will not communicate with each other even though there are overlapping regions among them; a block can only be "hole-cut" by blocks with larger layer number; and the outermost two layers of cells of a block can only receive solutions from blocks with smaller layer numbers. For the example shown in Figure 5.18, blocks 2, 3 and 4 can be assigned the same layer number which should be larger than that of the global block. Blocks 5-12 can be also assigned a single layer number that is again larger. Therefore, the overset mesh has 3 layers of blocks, i.e. the first layer with only the global block, the second layer with blocks 2-4, and the third layer with blocks 5-12. All blocks can only communicate with blocks belonging to the other two layers. So, the **BLKLAY** bulk data card is essentially a mechanism to assign blocks into different layers and avoid interactions between blocks bearing the same layer number.

5.4.2 HOW TO GENERATE A GOOD OVERSET MESH

Making a good mesh is very critical for any CFD solver. ZEUS is no exception. To make a good overset mesh using ZEUS, one needs to have some understanding as to what is a good overset mesh and what is a bad one. Taking the most common case of a smaller block with under-wing store embedded in a bigger main wing block as an example, the smaller block needs to receive interpolated solutions of flow variables from the bigger block

for two layers of its boundary cells. In order to minimize the interpolation errors, an overset mesh with comparable grid densities at the boundary of the smaller block is obviously necessary. ZEUS has got a feature which really helps in frequently changing some parameters and regenerating the mesh for review. In the Executive Control Section, there is a command which stops the code right after the grid is generated. The command is “SOL -1”. If the user specifies a **PLTAERO** bulk data card, then ZEUS creates a grid file for the user to look at the mesh. For overset meshes, then, the user should judge whether at the boundary of the smaller block, the two blocks have comparable grid cell sizes. It doesn’t have to be a 1:1 ratio, but 1:10 is not quite acceptable and that will introduce large errors in the interpolation. The way that a user can judge whether the two blocks of mesh are comparable is by looking at different views (X-Y, Y-Z, and X-Z) in the Tecplot file generated by the **PLTAERO** bulk data card. If the grid cell sizes differ too much, the user can change some of the parameters in the **MESHPRM** and/or **BLOCK/BLOCKT** bulk data card or carefully redefine the chordwise/ spanwise cuts of those involved components. For simplicity, changing the smaller block is always easier, and that should be tried first.

X-Y View:

If the X-Y view suggests that the distances between chordwise Y-lines of the two blocks are not comparable, then tweaking XGROWTH or XLEAD and XTRAIL (defined in the **MESHPRM** and **BLOCK/ BLOCKT** bulk data cards) of the smaller block to improve the mesh. A good overset grid system of two blocks from the X-Y view is shown in Figure 5.20.

XLEAD and XTRAIL for the under-wing store block can be chosen arbitrarily, but as a guideline, it is not necessary to extend it too far from the store in either direction.

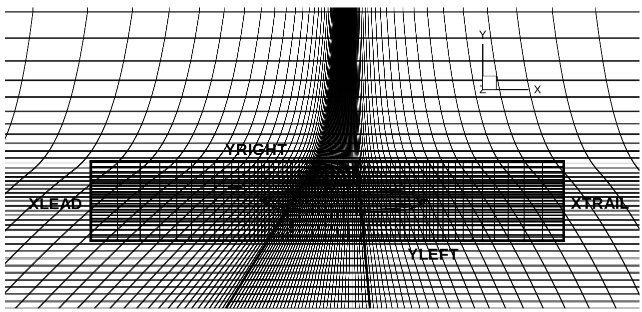
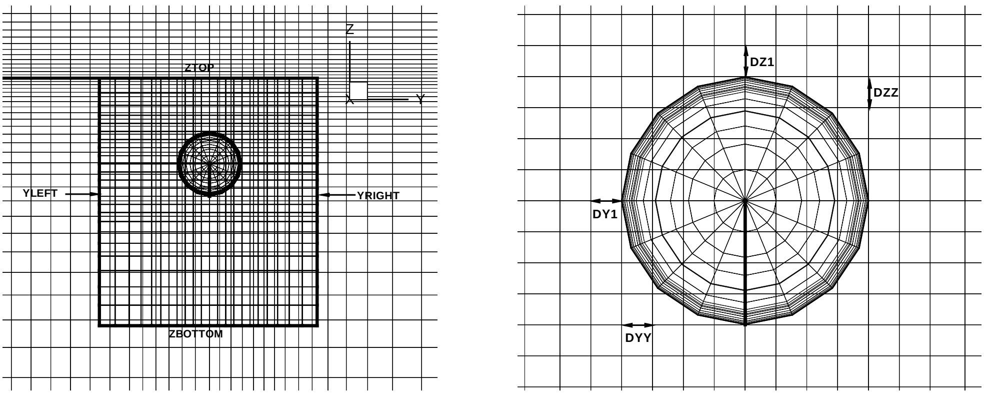


Figure 5.20 An Example of a Good Overset Mesh with Two Blocks in X-Y View

Y-Z view:

Similarly, the quality of the overset mesh in terms of comparable spacing for spanwise Z-lines and horizontal Y-lines can be easily seen from the Y-Z view. If the Y-Z view shows that the distances between spanwise Z-lines are not comparable, then the parameters to tweak are DYFACT, YGROWTH, YLEFT, and YRIGHT, whereas parameters DZFACT, ZGROWTH, ZBOT, and ZTOP can be used to tweak the spacing of horizontal Y-lines of the smaller block. DYFACT is used to control the spacing (DY1) of the outer first spanwise Z-lines close to the body's left and right surfaces, but this is also dependent on the spacing (DYY) of the inner first spanwise Z-lines close to the body's surfaces. ZEUS will pick the smaller of DYY and 1.5×DYFACT×REFC for the value of DY1. As for the value of DYY, it is determined by the entry FACTDY of the **BODY7** bulk data card together with the entry NRAD of the **SEGMESH** bulk data card. Please refer to these two bulk data cards in Chapter 4 for details. The spacing of the outer first horizontal Y-lines close to the body's top and bottom surfaces is affected by parameter DZFACT as well as NRAD and FACTDZ in a similar way. Figure 5.21(a) shows an example of good comparison of gridline spacing in both Y and Z directions. In the zoom-in Figure 5.21(b), the meaning of DY1/DYY, and DZ1/DZZ is depicted. For this example, DY1=DYY, and DZ1=DZZ because the values of 1.5×DYFACT×REFC and 1.5×DZFACT×REFC are larger than DYY and DZZ, respectively.



(a) Y-Z Plane *(b) Zoom-in Figure*

Figure 5.21 An Example of a Good Overset Grid with Two Blocks in Y-Z View

5.5 USE OF BLOCK1 FOR BODY-FITTED GRID MODELING OF BODY

In Section 5.1.2, it is mentioned that ZEUS can automatically generate a block of Cartesian mesh using the **BLOCK** bulk data card that approximates the exact surface geometry of a body by a rectangular box. In this way, the body is basically modeled as four surfaces on the mean planes of which the transpiration boundary condition is applied just as in the modeling of wing-like components. Although it has been demonstrated that this approximate method yields reasonable results for some cases with the help of slender body theory, a more accurate modeling of the body-like component is desirable for cases that the body has dominant aerodynamic effects for the flowfield characteristics. For this reason, ZEUS provides an option of using **BLOCK1** instead of **BLOCK** bulk data card to model the body geometry exactly with an automatically generated body-fitted mesh around the body surface. The drawback of the **BLOCK1** bulk data card is that it only models one **BODY7** and cannot model wing-body configuration unless overset mesh is employed in which the wing is modeled by another **BLOCK** bulk data card.

With **BLOCK1** bulk data card, the exact boundary condition is applied on the exact body surface, and the true angle of attack freestream flow condition is imposed . However, for a overset mesh generated by multiple **BLOCK** and **BLOCK1** bulk data cards ZEUS has to include the angle of attack effects in the transpiration boundary conditions rather than apply the true angle of attack freestream flow condition.

The bent-nose Compact Kinetic Energy Missile (CKEM) with eight wrap-around fins configuration is such a case that the missile body is aerodynamically dominant and thus a **BLOCK1** bulk data card is preferred. In the rest of the section, the modeling of the CKEM case is presented in details as an example of proper application of this body-fitted grid modeling strategy of ZEUS.

The bent-nose CKEM configuration has a centerbody with a length of 52.07 inches and a diameter of 2.214
inches. The centerbody is characterized by a 4.0-caliber, 7.0 power series nose with a cylindrical afterbody and eight wrap-around fins. The geometry of one of the wrap-around fins is depicted in Figure 5.22. The maneuverability of CKEM is achieved by the bent-nose control via a single joint by which the 4.0-caliber nose is projected at an angle relative to the missile centerbody to provide a control deflection.

The centerbody is modeled by a **BODY7** bulk data card that refers to a **SEGMESH** bulk data card to define the centerbody as a body of revolution. The centerbody is divided into 60 axial stations and 65 circumferential points, and at each axial station, a radius and a camber are specified. A 2-degree bent-nose angle is introduced by the camber line in the **SEGMESH** bulk data card. Due to the use of **BLOCK1** bulk data card for the centerbody, these 65 circumferential points is necessary in order to generate a refined mesh whose grid size is compatible with that of the mesh for the eight wrap-around fins in the overlapping regions.

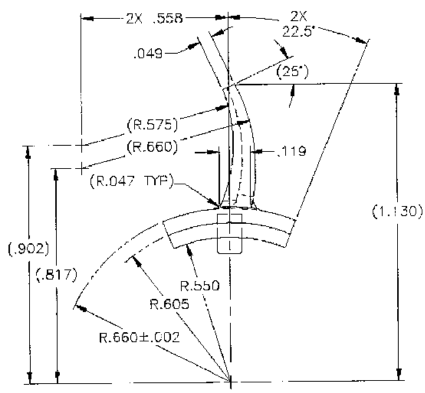


Figure 5.22 Geometry of the Wrap-Around Fin

The eight wrap-around fins are modeled by the **CAERO7** bulk data cards. Each fin is modeled by 9 **CAERO7** bulk data cards with varying dihedral angles to approximate the wrap shape of the fin. Note that the 9 **CAERO7** bulk data cards of each fin are located in a local coordinate system defined by an **ACOORD** bulk data card where a dihedral angle corresponding to that at the root of each fin is specified. In this local coordinate system, the geometry of the eight fins becomes identical to each other so that the set-up of the 9 **CAERO7** bulk data cards for each fin is much simpler. Figure 5.23 shows the surface panel model of the CKEM configuration.

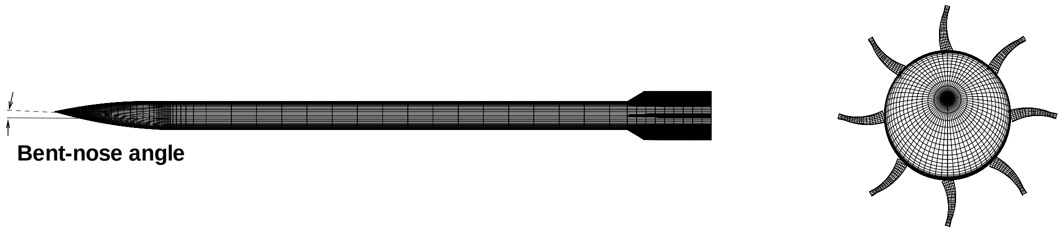


Figure 5.23 Surface Panel Model of the CKEM Configuration with a 2-Degree Bent-Nose Angle

Because there are overlapping regions among the eight wrap-around fins and the centerbody on the projected X-Y plane and Y-Z plane, the CKEM configuration cannot be fitted into one block of mesh. To model the CKEM configuration, it is required to use nine blocks of mesh; one block for the centerbody and the other eight blocks for the eight wrap-around fins, respectively.

The block of mesh for the centerbody is generated using a **BLOCK1**. The **BLOCK1** bulk data card generates an O-type body-fitted mesh around the centerbody by referring to the corresponding **BODY7** bulk data card. The **BLOCK1** bulk data card can model only one BODY7 macroelement and cannot model the wing-body configuration whereas the **BLOCK** bulk data card can. The domain of this body-fitted mesh starts from X=-10.0 and ends at X=70.0 inches with a maximum radius of 15.0 inches. 30 grid points along each of the 65 radial lines originated from the 65 circumferential points on the centerbody surface mesh are specified. The locations of these 30 grid points along each radial line in terms of the percentage of the total length of the radial line are specified by an **AEFACT** bulk data card which will be referred by the **BLOCK1** bulk data card. In this **AEFACT** bulk data card, a very small mesh size near the centerbody surface (0.25% of the total length of the radial line) is specified, and the first several values are also carefully picked to make sure the growth rate is not too large close to the body surface. The reason for doing this is, in conjunction with the 65 radial lines, to keep the centerbody mesh size in the overlapping regions to be compatible with that of the eight wrap-around fins' mesh. A compatible mesh size in the overlapping region between the different blocks of mesh can minimize the interpolation errors involved in the overset mesh scheme.

The mesh of the eight wrap-around fins is generated by eight **BLOCK** bulk data cards. To generate a block of mesh into which the nine CAERO7 macroelements for modeling each fin can be fitted, it is required to specify nine Y-Zones in each **BLOCK** bulk data card. Each Y-Zone occupies a CAERO7 macroelement by referring to the ID of the corresponding **CAERO7** bulk data card. All eight **BLOCK** bulk data cards have the same values of those entries because they are defined in a local coordinates specified by their respective **ACOORD** bulk data cards. It should be noted that the values of ZBOT and ZTOP are properly calculated so that the outer boundary of the mesh projected on the Y-Z plane of the global coordinates does not intersect with the surface mesh of other fins. In the **MESHPRM** bulk data card for the fin block, a negative ZGROWTH is specified to trigger the automated mesh generation scheme to generate a block of mesh whose outer boundary in Z direction is following the wrapped shape of the fin. On the other hand, if ZGROWTH is positive, the outer boundary of the mesh will be a rectangular box. This rectangular box renders the mesh size between the surface mesh of the wrapped fin and the rectangular outer boundary to be non-uniformly distributed along the fin span and thus affects the smoothness of the mesh. This is the reason of specifying a negative ZGROWTH to avoid the non-uniformly distributed mesh.

The resulting overset mesh for the CKEM configuration is presented in Figure 5.24. It can be seen in the projected mesh on the Y-Z plane, the mesh size of the centerbody mesh in the radial direction is comparable to that of the fin block mesh along the spanwise direction. As already mentioned earlier, the mesh size in the radial direction of the centerbody block can be controlled by an **AEFACT** bulk data card. The comparable mesh sizes among the different blocks of mesh in the overlapping region can enhance the accuracy of the flow solution interpolated by the overset mesh scheme.

It is important to know that the overset mesh scheme requires the information about the sequence of the blocks of mesh from the global mesh to the smallest embedded mesh to be listed in a **BLKSEQ** bulk data card. If there is no **BLKSEQ** card specified, the default overset strategy assumes the mesh generated by the **BLOCK** bulk data card with the smallest identification number to be the global mesh and the mesh generated by the **BLOCK1** bulk data card to be the last embedded mesh. Apparently, this default overset strategy is deadly wrong for the CKEM case, and the correct one should be that the **BLOCK1** bulk data card is listed first and then followed by those of the eight **BLOCK** bulk data cards.

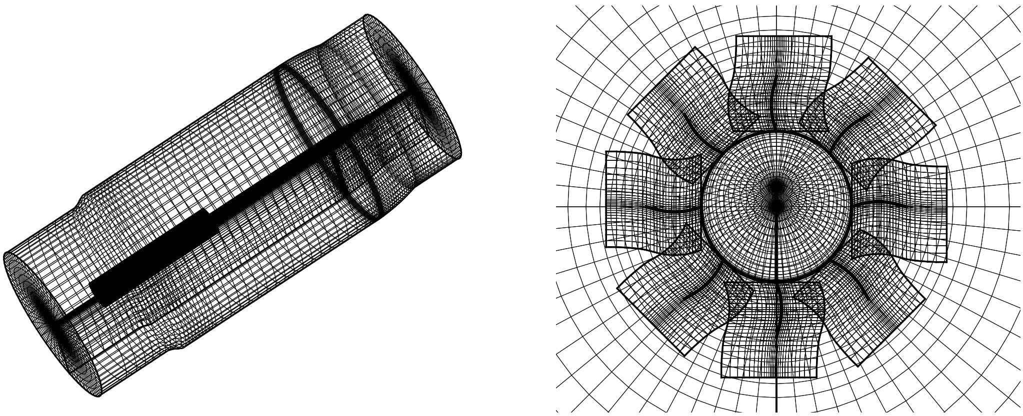


Figure 5.24 Overset Mesh for Modeling the CKEM Configuration

The lift and moment coefficients of the CKEM configuration with bent nose angle=2.0°at M=6.0, and at angles of attack=0°, 2°, 4°, and 6° generated by ZEUS are compared to the wind-tunnel data as well as the results computed by the Missile DATCOM. These comparisons are shown in Figure 5.25. In the same figure, the ZEUS solution and the wind-tunnel data of the CKEM configuration with bent node angle=0.0° are also presented. The ZEUS solution of the bent nose angle=0.0° configuration is obtained by specifying a zero camber line in the **SEGMESH** bulk data card. It can be seen that the ZEUS results compared very well with the wind-tunnel data, which verified the proper modeling of the CKEM configuration by using a **BLOCK1** bulk data card for its centerbody.

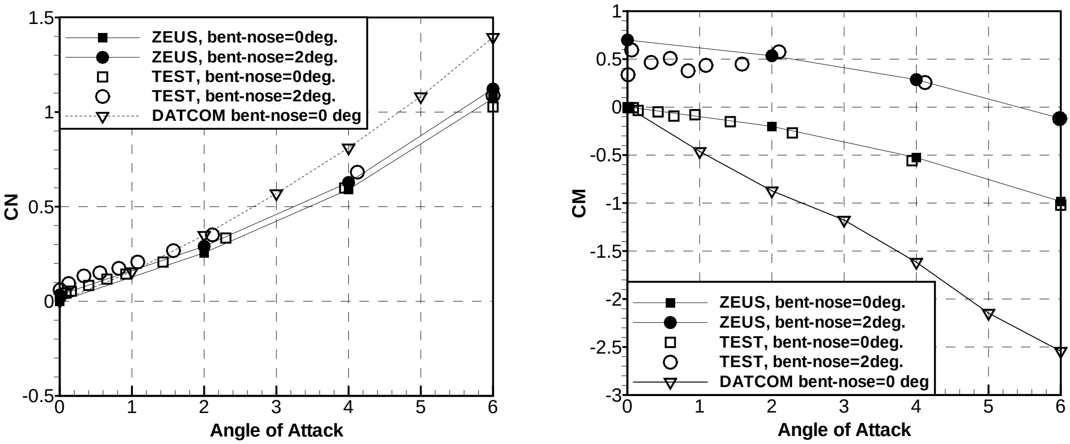


Figure 5.25 Lift and Moment Coefficients of the CKEM Configurations with Bent-Nose Angles=0.0° and 2.0° at M=6.0 and Various Angles of Attack

Please note that the use of **BLOCK1** bulk data card should be limited to missile-fin cases such as the CKEM configuration and under-wing store with canard and/or fins. For fuselage and main wing case, the fuselage should be modeled together with the main wing using a **BLOCK** bulk data card.

5.6 MODELING TIPS

Some modeling tips for the automated mesh generation are presented in this section.

5.6.1 VERTICAL LIFTING SURFACE WITH SWEEP ANGLE

The vertical lifting surfaces are those with 90° dihedral angle such as vertical tails and pylons. If a vertical lifting surface has a non-zero sweep angle, it cannot be fitted into the X-Z plane of a block of mesh because all chordwise Z-lines projected on the X-Z plane are parallel to the Z-axis and thus cannot be sheared to accommodate the sweep angle of the vertical lifting surface.

Such vertical lifting surfaces with sweep angle must be fitted into a separate block of mesh or added to the **BLOCKT** mesh as part of the T-tail component. In this block of mesh, the vertical lifting surface is treated as a horizontal surface lying on the X-Y plane of a local coordinate system so that the mesh on the local X-Y plane can be sheared to accommodate the sweep angle of the lifting surface. This can be achieved by defining the block of mesh to be located in a local coordinate system using an **ACOORD** bulk data card with the entry THETA set to 90°. Figure 5.26 shows an example of a swept vertical tail modeled in a block of mesh located in a local coordinates whose x-y plane is lying on the plane of the vertical tail. In this local x-y plane, a sheared mesh can be generated to accommodate the sweep angle of the vertical tail. Note that if the user insists on fitting a swept vertical lifting surface into an X-Z plane of the mesh, ZEUS will model the vertical lifting surface as a straight wing without sweep angle. This may lead to large errors in the computational result.

Vertical lifting surfaces located along the centerline should not be included in half-span aerodynamic models. Because half-span models only allow symmetric analysis, and a centerline vertical tail does not influence a symmetric analysis, they should not be included in the aerodynamic model.

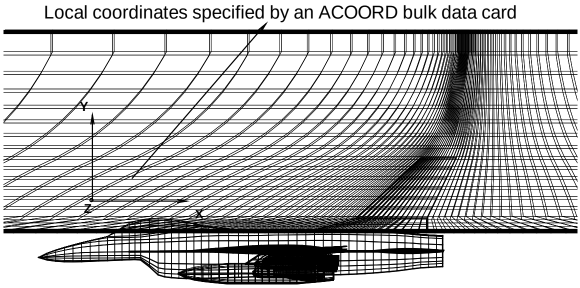


Figure 5.26 Vertical Tail with Sweep Angle Modeled as Horizontal Surface in a Separated Block of Mesh

5.6.2 WING-BODY JUNCTION

In the Cartesian mesh generated by the **BLOCK** bulk data card, the body is modeled as a rectangular box whose width and height coincide with the body's maximum dimension in both directions. For configurations that have a wing attached to the body but not at the into a separate block of mesh. body's maximum width, the wing has to be fitted. Otherwise, part of the wing will be embedded in the body mesh box. Figure 5.27 shows a wing-body-tail configuration where a portion of the tail is within the body mesh box if the wing, body and tail are modeled in a single block of mesh. The part of the tail embedded in the body mesh box will not be accounted for in the boundary condition of the flow solver, which means the flow solver simply ignores it and provides no solution on those tail grid points embedded in the body. However, at the end of each time step, ZEUS always interpolates/extrapolates the solution from the computational mesh to the surface panels. Therefore, the part of the tail surface panels embedded in the body mesh box will still have flow solution by the extrapolation from the solution on those nearby tail panels.

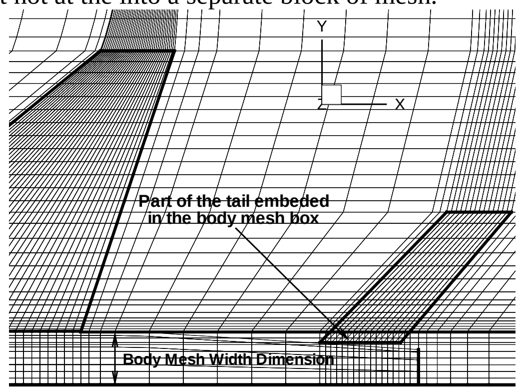


Figure 5.27 Part of Wing Embedded in Body Mesh Box

5.6.3 CHORDWISE Y-LINES AHEAD OF WING LEADING EDGE IN THE PRESENCE OF WING-BODY/WING-TIP LAUNCHER CONFIGURATION

For fighter aircraft like F-16, it is very common to see such configurations with wing-body-tail and wing-tip launcher as shown in Figure 5.28(a). The mesh generation in the region between the leading edges of the main wing and the launcher is complicated. As mentioned in Section 5.3, a line-tracing methodology is employed to generate the surface mesh, and coherent chordwise divisions between adjacent **CAERO7** macroelements are recommended in order to achieve nice mesh without unnecessarily redundant grid lines. For the region between the leading edges of the main wing and the launcher, the line-tracing mechanism is yet to play, and the generation of the chordwise Y-lines is somewhat guided by the chordwise divisions of surface mesh of the fuselage and launcher. However, the fuselage and the launcher are separated by the wings, and there is no way to specify coherent cuts for them beforehand. In most cases, each fuselage chordwise divisions will incur a chordwise Y-line, and so does each launcher chordwise divisions. These two bunches of Y-lines will mix with each other, and some of them will be deleted according to a small tolerance.

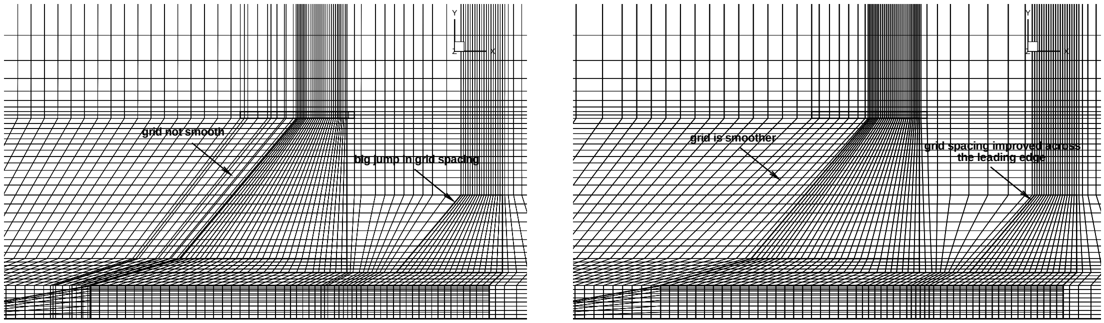
The mesh generation in the region between the main wing and tail is determined by the chordwise divisions of the fuselage surface mesh in the section between the trailing edge of the wing and leading edge of the tail. Therefore, in order to have sufficient chordwise y-lines in this region, more chordwise divisions of the fuselage surface mesh is required.

In the next section, a tool meant to gain control of the mesh generation in the above mentioned "out-of-control" regions will be introduced and explained in details.

5.6.4 USE OF GAP AND GAPZ TO IMPROVE MESH QUALITY

Just like other finite-volume solvers, the convergence speed and solution accuracy of ZEUS highly rely on the quality of the CFD mesh. A good mesh should be smooth and yet skewness in grid cells should be minimized. Ideally, the grid size will grow from the surface to the far field gradually. The rate of grid size growth towards the far field is controlled by the growth parameters set in the **MESHPRM** bulk data card. The **GAP**, **GAPZ**, and **GAP1** bulk data cards provide tools to improve the smoothness and reduce the skewness of the mesh in the regions close to the aerodynamic surface mesh, such as before the leading edge of the wing, after the trailing edge of the wing, and between the wing and tail.

Figure 5.28 shows a model with fuselage, wingtip launcher, main wing, and tail. Without the help of a **GAP** bulk data card, the mesh is poor as shown in Figure 5.28(a). In the region between the leading edge of the main wing and the leading edge of the launcher, the spacing of the chordwise Y-lines is poorly distributed. The reason is that there is no guarantee for the chordwise Y-lines coming from the Y-zone close to the fuselage to match with those coming from the Y-zone close to the launcher. These two bunches of chordwise Y-lines are determined by the chordwise divisions of the surface mesh of the fuselage and launcher, respectively. As mentioned in Section 5.3, the **GAP** bulk data card is a fictitious surface that has chordwise divisions specified to direct the generation of chordwise Y-lines within the **GAP** region. Therefore, by adding GAP 1 and GAP 2 shown in Figure 5.29, and carefully setting up the chordwise divisions to match with the chordwise divisions of the fuselage and launcher, a much smoother mesh as shown in Figure 5.28(b) can be obtained. Similarly, for the region between the main wing and the tail, adding GAP 3 and GAP 4 can help to make the mesh spacing more benign, especially across the leading edge and trailing edge of the wings, where equal spacing is desirable for most CFD solvers.



(a) Without GAP *(b) With GAP*

Figure 5.28 Better Mesh can be Achieved with the Help of GAP

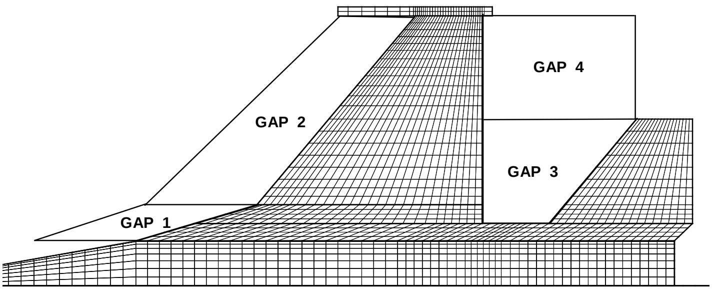
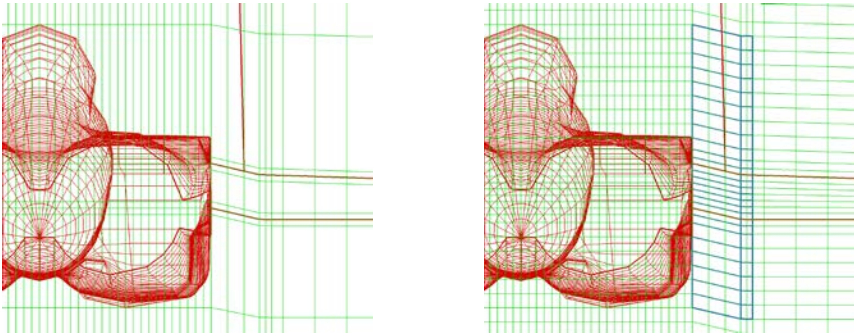


Figure 5.29 Use of GAP

Just as a **GAP** bulk data card offers extra control of the chordwise Y-line spacing, the **GAPZ** bulk data card provides a means to control the spacing of the horizontal Z-lines. Figure 5.30 shows the comparison of Y-Z mesh with or without the use of **GAPZ** bulk data card for a model of fuselage-wing-tail combination. As the wing and tail are not in the same plane, the code will generate horizontal Z-lines immediately above and below the two **CAERO7** surfaces, in addition to gridlines at the top and bottom of the **BODY7** element defining the fuselage. These horizontal Z-lines, together with those generated from the body surface mesh, could yield an inadequate mesh in the Y-Z view. By adding a **GAPZ** bulk data card, the spacing of the horizontal Z-lines is fully in control. Please note, NSEG = - 1 should be specified in the **BODY7** bulk data card in order to avoid the automatic horizontal Z-line generation based on the body surface mesh. The use of the **GAPZ** bulk data card is highly recommended in the case of a fuselage and wing in the same block, and in the case of two **CAERO7** elements separated by a vertical gap embedded in the same block.



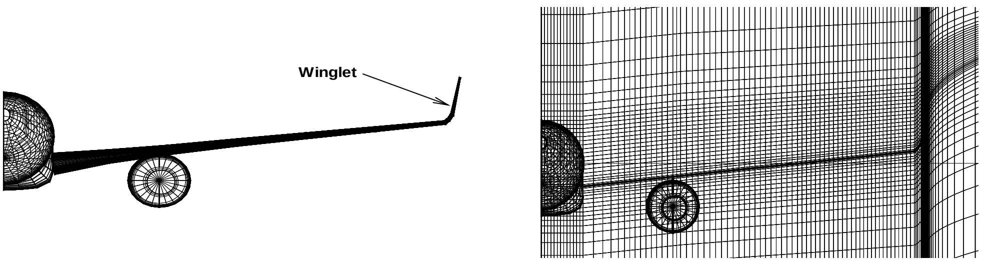
(a) Without GAPZ (b) With GAPZ

Figure 5.30 Better Mesh on the Y-Z Plane can be Achieved with the Help of GAPZ

One difference from the **CAERO7** bulk data card is that the **GAP**, **GAPZ**, and **GAP1** bulk data cards do not need to specify the spanwise divisions. In addition, to specify the chordwise cuts using the **AEFACT** bulk data card, an option of using physical X or Z coordinate values instead of the percentage values of the chord is available.

5.6.5 MODELING OF WING WITH LARGE DIHEDRAL ANGLE

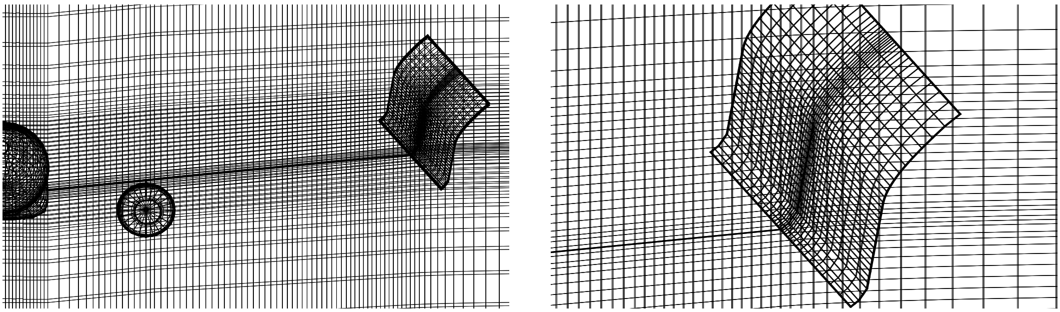
As demonstrated in Section 5.3.2, the automated mesh generation scheme in ZEUS can generate sheared mesh on the Y-Z plane to accommodate the dihedral angle of the wing-like component. However, in the CFD solver point of view, sheared mesh is never recommended because the accuracy of CFD solution is dependent on the orthogonality of the computational mesh. With increase of dihedral angle of the wing, the skewness of the mesh generated is getting worse, and thus the flow solution will be less accurate. ZEUS may refuse to converge or even blow-up with highly skewed mesh in the Y-Z plane.



(a) Aircraft with Winglet (b) Highly Skewed Mesh at Winglet

Figure 5.31 Model the Winglet in the Same Block of Fuselage and Main Wing

A common case involving large dihedral angle wing component is an aircraft with winglet such as the one shown in Figure 5.31(a). For this configuration, if we model the winglet in the global block together with the fuselage and main wing, the resulting mesh is highly skewed in the portion of span of the winglet as shown in Figure 5.31(b). Therefore, it is recommended to model the wing component in a separated block if its dihedral angle exceeds 45°. Figure 5.32 shows the Y-Z plane view of the overset mesh when the winglet is modeled in a second block of mesh. Please note that YLEFT of the winglet block is set right at the root of the winglet in order to avoid the intersection of the winglet block with the surface mesh of the main wing. The main wing is the most critical component of the aircraft for the generation of aerodynamic forces. Any overset interpolation involving the grid cells close to the main wing could potentially undermine the solution accuracy, and thus should be avoided if possible.



(a) Y-Z Plane View (b) Zoom-in of Y-Z View

Figure 5.32 Winglet Modeled in a Separated Block of Mesh

5.6.6 Y-ZONE FOR INDENTED WING-BODY JUNCTION

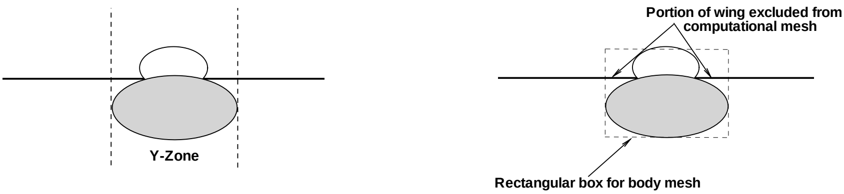


Figure 5.33 Incorrect Y-Zone Definition

Figure 5.33 shows an indented wing-body junction where the body width is wider than the distance between the roots of the right-hand side (RHS) and left-hand side (LHS) wings. If the Y-Zone containing the body is defined according to the width of the body as shown in Fig. 5.33, then this width is used to establish a rectangular box in the computational mesh where the body boundary condition is applied. Any components inside the box are excluded from the computational mesh. For this case, a portion of the wing inside the box is excluded from the computational mesh and cannot generate aerodynamic forces, which leads to incorrect result. The correct Y-Zone definition is shown in Fig. 5.34 where the width of the Y-Zone is defined according to the distance between the roots of the RHS and LHS wings. Then this width is used for the width of the rectangular box to model the body surface mesh. In so doing, the whole wing is outside the box. It should be noted that the slope of true body surface are mapped on the rectangular box to construct the body boundary condition on the box. After the Euler solver solves the aerodynamic pressures on the box, their pressures are mapped back to the true body surface. Therefore, the final aerodynamic forces are calculated using the true area of the body surface; thereby no aerodynamic forces are lost.

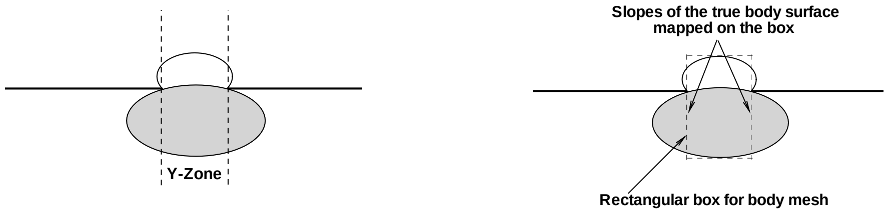


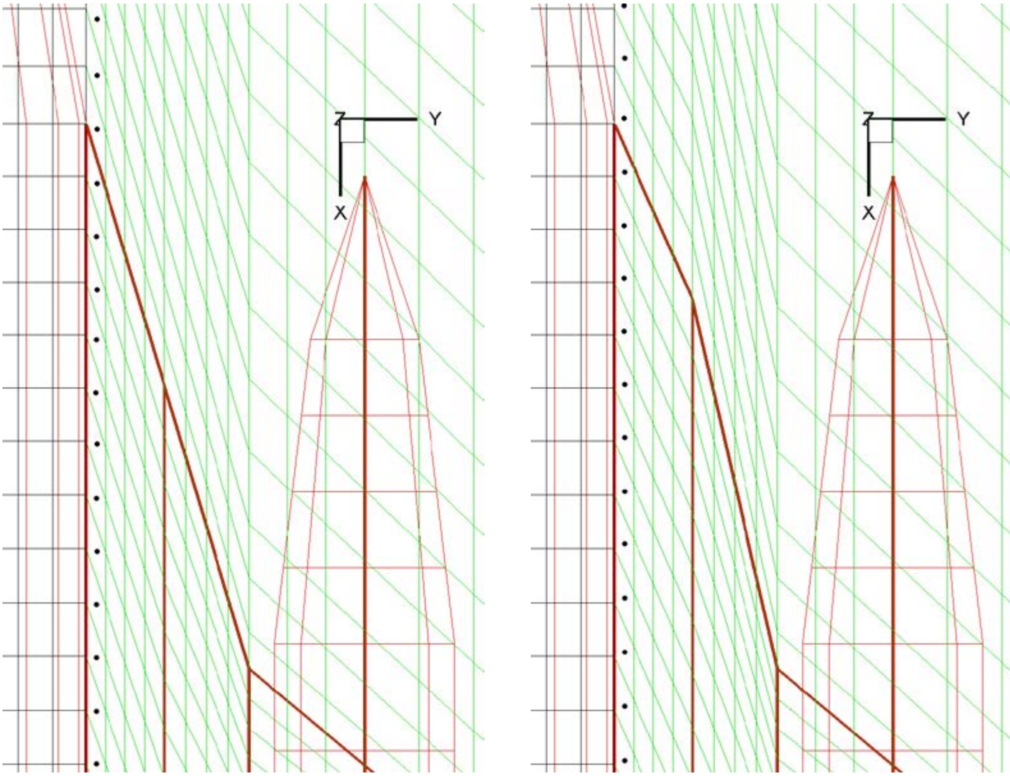
Figure 5.34 Correct Y-Zone Definition

5.6.7 AVOID LARGE SWEEP ANGLE OF CAERO7 ADJACENT TO BODY7

Strakes of fighter aircraft may have a large leading edge sweep angle as shown in Figure 5.35(a). Modeling such a strake by a single CAERO7 macroelement generates highly skewed computation volume cells adjacent to the fuselage box model shown by the black mesh lines in the figure. The boundary condition for solving the Euler equation on the fuselage box model is applied at the centroid of the volume cells adjacent to the fuselage shown by the black solid circles. The ideal volume cells adjacent to the fuselage box are orthogonal to the box surface mesh. Such orthagonality in this case is largely violated to the extent that the flow solver fails to converge.

To circumvent this problem, the user can break the one CAERO7 macroelement into two CAERO7 macroelements, as shown in Figure 3.35(b). The CAERO7 macroelement adjacent to the fuselage has a smaller leading edge sweep angle than the original CAERO7 macroelement. This in turn reduces the skewness of the volume cells adjacent to the fuselage surface mesh.

The allowable sweep angle leading to a converged solution varies case by case. If the sweep angle faithfully representing the true model leads to a convergence problem, a trial and error process to artificially reduce the sweep angle for the Euler solver may be required to reduce the sweep angle by a minimum amount.



(a) ***High-Sweep Strake Leading Edge*** **(b)** ***Reduced-Sweep Strake Leading Edge***

Figure 5.35 Strake Leading Edge Sweep Angle Adjacent to Fuselage

5.6.8 TROUBLESHOOTING FAILURE OF CONVERGENCE OF CFD SOLUTION

It is very important to always ensure convergence of the solution. For this reason, it is recommended to set PRNTCOV = 2 in the **MKPARAM** bulk data card. When checking the convergence history, one should ensure that the residuals drop by a few orders of magnitude. In the case that the solution is not converging, first check for warning messages about the quality of the generated mesh in the output file.

If a warning message specifying slope discontinuity or too large slopes is present, use the **PLTSLP** bulk data card to generate plot files of the slopes of the aerodynamic mesh. See Section 8.5 for a description of the plot files. Using the warning messages in the standard output file as a guide, find the elements causing the warning and identify the cause. For **CAERO7** elements, large slopes are caused by the airfoil definition. Use the **PLTAERO** bulk data card with option THICK = YES to verify the airfoil definition input is correct. For **BODY7** elements, warnings are generated when the change in slope from one box to the next along the flow direction is large. If this is caused by engine inlet areas, ensure the flow through condition is specified using the **PBODY7** bulk data card. If the warning is caused by sharp changes in the surface mesh, such as the fuselage body/canopy intersection, consider smoothing out the area in the surface mesh.

Once the surface mesh is sufficiently smooth, the next step is to ensure the steady flow solution converges. It is desirable for the CFL number (set in the **MKPARAM** bulk data card) to be as large as possible, as larger CFL numbers result in faster solution convergence. However, a larger CFL number can also cause solution divergence, in which case the CFL number should be lowered. In some cases, more artificial dissipation is needed by increasing the VIS2 and VIS4 parameters above their default of 1.0. It is recommended not to raise VIS2 and VIS4 above 1.25.

Most of the convergence problems are caused by a poor computational mesh usually between two aerodynamic components. To find out where the poor quality mesh originates the user can activate the FREECOMP option in the **PARAM** bulk data card and running the steady flow solver with a few aerodynamic components at a time.

Chapter 6

THEORETICAL BACKGROUND OF THE FLOW SOLVER IN ZEUS

6.1 UNSTEADY EULER SOLVER ON STATIONARY CARTESIAN GRID

ZEUS is an unsteady aerodynamics generator based on a stationary Cartesian grid. It solves the time-accurate Euler equations using a cell-centered central-differencing finite-volume method with Jameson-Schmidt-Turkel (JST) artificial dissipation scheme [1] implemented for stability of the flow solver.

6.1.1 TIME-ACCURATE EULER METHOD

The three-dimensional unsteady Euler equations in conservative differential form and in curvilinear coordinates are as follows:

$$\frac{\partial Q}{\partial t}+\frac{\partial H_1}{\partial \xi}+\frac{\partial H_2}{\partial \eta}+\frac{\partial H_3}{\partial \varsigma}=0 \tag{6.1}$$

where Q is the product of conservative flow variables vector **q** and the inverse of the transformation Jacobian **J**, and H_1, H_2, H_3 are the convective fluxes in three curvilinear coordinate directions:

$$Q = Jq = J\begin{bmatrix}\rho \\ \rho u \\ \rho v \\ \rho w \\ e\end{bmatrix}, \qquad H_1 = J\begin{bmatrix}\rho U \\ \rho uU + p\xi_{\mathbf{x}} \\ \rho vU + p\xi_{\mathbf{y}} \\ \rho wU + p\xi_{\mathbf{z}} \\ (e+p)U - p\xi_t\end{bmatrix}$$

$$H_2 = J\begin{bmatrix}\rho V \\ \rho uV + p\eta_{\mathbf{x}} \\ \rho vV + p\eta_{\mathbf{y}} \\ \rho wV + p\eta_{\mathbf{z}} \\ (e+p)V - p\eta_t\end{bmatrix}, \qquad H_3 = J\begin{bmatrix}\rho W \\ \rho uW + p\varsigma_{\mathbf{x}} \\ \rho vW + p\varsigma_{\mathbf{y}} \\ \rho wW + p\varsigma_{\mathbf{z}} \\ (e+p)W - p\varsigma_t\end{bmatrix} \tag{6.2}$$

U, V, W and u, v, w are the three components of the flow velocity in curvilinear and Cartesian coordinates, respectively. They are related by metric terms as follows:

$$U = \xi_x u + \xi_y v + \xi_z w + \xi_t, \quad V = \eta_x u + \eta_y v + \eta_z w + \eta_t, \quad W = \varsigma_x u + \varsigma_y v + \varsigma_z w + \varsigma_t \tag{6.3}$$

Perfect gas is assumed, and the equation of state is:

$$e = \frac{1}{\gamma - 1} p + \frac{1}{2}\rho\left(u^2 + v^2 + w^2\right) \tag{6.4}$$

Applying Equation (6.1) to each finite-volume grid cell, we obtain a set of ordinary differential equations of the form:

$$\frac{d}{dt}\left(q_{i,j,k}\Omega_{i,j,k}\right) + \mathbf{R}\left(q_{i,j,k}\right) = 0 \tag{6.5}$$

where $\Omega_{i,j,k}$ is the volume of the cell with index *(i,j,k)* and the residual $\mathbf{R}\left(q_{i,j,k}\right)$ is obtained by evaluating the flux integral at all the cell surfaces and summing them up. Jameson's artificial dissipation flux is added to the convective flux for stability:

$$\mathbf{R}\left(q_{i,j,k}\right) = C_{i,j,k} - D_{i,j,k} = C_{i,j,k} - (D^{(2)}_{i,j,k} - D^{(4)}_{i,j,k}) \tag{6.6}$$

here, $C_{i,j,k}$ and $D_{i,j,k}$ are the flux integrals of the cell *(i,j,k)* due to convective flux and artificial dissipation flux, respectively, while $D^{(2)}_{i,j,k}$ and $D^{(4)}_{i,j,k}$ are the so-called 2nd and 4th order artificial dissipation fluxes. Both artificial dissipation fluxes are the sum of the artificial dissipation at all six surfaces of the computational cell. For example, they are formulated as follows at the surface (i+1/2):

$$\begin{aligned} d^{(2)}_{i+\frac{1}{2},j,k} &= \sigma_{i+\frac{1}{2},j,k}\,\varepsilon^{(2)}_{i+\frac{1}{2},j,k}\left(q_{i+1,j,k} - q_{i,j,k}\right) \\ d^{(4)}_{i+\frac{1}{2},j,k} &= \sigma_{i+\frac{1}{2},j,k}\,\varepsilon^{(4)}_{i+\frac{1}{2},j,k}\left(q_{i+2,j,k} - 3q_{i+1,j,k} + 3q_{i,j,k} + q_{i-1,j,k}\right) \end{aligned} \tag{6.7}$$

where $\sigma_{i+\frac{1}{2},j,k}$ is the largest eigenvalue, or say spectral radii, of the convective flux Jacobian in the ξ direction, and the 2nd and 4th order artificial dissipation terms, $\varepsilon^{(2)}_{i+\frac{1}{2},j,k}$ and $\varepsilon^{(4)}_{i+\frac{1}{2},j,k}$, are defined as:

$$\begin{aligned} \varepsilon^{(2)}_{i+\frac{1}{2},j,k} &= \kappa^{(2)} \min\left[0.25, \max(\nu_{i+1,j,k}, \nu_{i,j,k})\right] \\ \varepsilon^{(4)}_{i+\frac{1}{2},j,k} &= \max\left[0, \frac{\kappa^{(4)}}{32} - \varepsilon^{(2)}_{i+\frac{1}{2},j,k}\right] \end{aligned} \tag{6.8}$$

where $\kappa^{(2)}$ and $\kappa^{(4)}$ are the two parameters VIS2 and VIS4, respectively, in the **MKPARAM** bulk data card of ZEUS to control the amount of the 2nd and 4th order artificial dissipation to be added in the Euler solver for stability, and the so-called pressure sensor $\nu_{i,j,k}$ is defined as:

$$\nu_{i,j,k} = \left| \frac{p_{i-1,j,k} - 2p_{i,j,k} + p_{i+1,j,k}}{p_{i-1,j,k} + 2p_{i,j,k} + p_{i+1,j,k}} \right| \tag{6.9}$$

The pressure sensor serves as a switching function to turn on and off the 2nd and 4th order artificial dissipations. In the region close to the shock wave, the pressure has a jump and the value of $\nu_{i,j,k}$ and thus $\varepsilon^{(2)}$ is of order 1, so the 2nd order artificial dissipation is at work while the 4th order one is disabled. On the other hand, in the smooth pressure region, $\nu_{i,j,k}$ is of 2nd order and thus $\varepsilon^{(2)}$ is switched off whereas $\varepsilon^{(4)}$ is working to damp the high frequencies that the central-differencing scheme fails to damp.

Using the Total Variation Diminishing (TVD) concept, an alternative form of the pressure switch can be:

$$\begin{aligned}
\nu_{i,j,k} &= \frac{\left| p_{i+1,j,k} - 2p_{i,j,k} + p_{i-1,j,k} \right|}{(1-\theta)\Psi + \theta\Psi_{TVD}} \\
\Psi_{TVD} &= \left| p_{i+1,j,k} - p_{i,j,k} \right| + \left| p_{i,j,k} - p_{i-1,j,k} \right| \\
\Psi &= p_{i+1,j,k} + 2p_{i,j,k} + p_{i-1,j,k}
\end{aligned} \tag{6.10}$$

where θ is called TVDCOEF in ZEUS, and if the default value of θ=0.0 is applied, the original JST pressure switch is recovered. The purpose of the TVD scheme is to prevent the generation of new extrema in the flow solution. It has been noticed that the ZEUS calculated pressure coefficient on the lower surface of the wing may show some jiggles for low Mach number high angle of attack flow conditions. Applying a non-zero θ will help smooth out the jiggles. Please note that using a large value of θ close to 1.0 could potentially smear out the pressure jump across the shock wave. For cases involving transonic shock waves, TVDCOEF < 0.8 is recommended.

If a steady-state flow problem is to be solved, we can treat t as a pseudo-time t^*, and a five-stage explicit Runge-Kutta (R-K) pseudo-time marching scheme can be applied to solve Equation (6.5):

$$
\begin{cases}
q_{i,j,k}^{(0)} = q_{i,j,k}^{(n)} \\
q_{i,j,k}^{(1)} = q_{i,j,k}^{(0)} - \frac{1}{4}\frac{\Delta t^*}{\Omega_{i,j,k}} R(q_{i,j,k}^{(0)}) \\
q_{i,j,k}^{(2)} = q_{i,j,k}^{(0)} - \frac{1}{6}\frac{\Delta t^*}{\Omega_{i,j,k}} R(q_{i,j,k}^{(1)}) \\
q_{i,j,k}^{(3)} = q_{i,j,k}^{(0)} - \frac{3}{8}\frac{\Delta t^*}{\Omega_{i,j,k}} R(q_{i,j,k}^{(2)}) \\
q_{i,j,k}^{(4)} = q_{i,j,k}^{(0)} - \frac{1}{2}\frac{\Delta t^*}{\Omega_{i,j,k}} R(q_{i,j,k}^{(3)}) \\
q_{i,j,k}^{(5)} = q_{i,j,k}^{(0)} - \frac{\Delta t^*}{\Omega_{i,j,k}} R(q_{i,j,k}^{(4)}) \\
q_{i,j,k}^{(n+1)} = q_{i,j,k}^{(5)}
\end{cases}
\tag{6.11}
$$

where the superscripts n and n+1 mean the solution procedure is from pseudo-time level $n\Delta t^*$ to $(n+1)\Delta t^*$.

For time accurate solution, the $\frac{d}{dt}$ operator is approximated by an implicit backward difference formula of second-order accuracy in the following form (dropping the subscripts i, j, k for clarity):

$$
\frac{3}{2\Delta t}\left[q^{n+1}\Omega^{n+1}\right] - \frac{2}{\Delta t}\left[q^n\Omega^n\right] + \frac{1}{2\Delta t}\left[q^{n-1}\Omega^{n-1}\right] + \mathbf{R}\left(q^{n+1}\right) = 0
\tag{6.12}
$$

The above equation can be reformulated into a steady-state problem in the pseudo time t^*:

$$
\frac{d(q^{n+1}\Omega^{n+1})}{dt^*} + \mathbf{R}^*\left(q^{n+1}\right) = 0
\tag{6.13}
$$

where

$$
\mathbf{R}^*\left(q^{n+1}\right) = \mathbf{R}\left(q^{n+1}\right) + \frac{3}{2\Delta t}\left(q^{n+1}\Omega^{n+1}\right) - \frac{2}{\Delta t}\left(q^n\Omega^n\right) + \frac{1}{2\Delta t}\left(q^{n-1}\Omega^{n-1}\right)
\tag{6.14}
$$

Again, a five-stage R-K pseudo-time marching scheme can be applied to solve Equation (6.13). As there are two different time-steps Δt^* and Δt involved in this solution procedure, it is called dual-time stepping method for the solution of time-accurate Euler equations.

Please note that the five-stage R-K pseudo-time marching is an explicit scheme and thus the pseudo-time step size Δt^* has to be less than a certain value for numerical stability. In the CFD community, the Courant-Friedrichs-Lewy (CFL) number is widely used for the time-step size control. Take the one-dimensional case as an example; the CFL number is defined as:

$$CFL = \frac{\Delta t^*}{\Delta t_c} = \frac{\Delta t^*}{L/U} \tag{6.15}$$

where Δt_c represents the time needed for the fluid with velocity U to flow through the computational cell with dimension L. The coefficients of the five-stage R-K scheme in Equation (6.11) are carefully picked up to achieve an optimal CFL number of about 4.0 without residual smoothing. If residual smoothing is applied, the maximum attainable CFL number can be increased to 8.0. Of course, the maximum CFL number, in theory, might not work in the real computation. However, ZEUS should be able to use the CFL number of 7.0 for most cases with residual smoothing turned on.

For the time-accurate flow computation, the dual-time stepping method applied in ZEUS is an implicit scheme in terms of the physical time step size Δt, and thus numerically there is no stability limitation for the physical time step size. Therefore, the physical time-step size Δt is free to pick based on physics of the flow rather than numerical stability concern. A guideline is that at least 50 physical time steps are necessary, for example, within each cycle of the sinusoidal oscillation of the configuration.

An implicit variable-coefficient residual smoothing scheme is incorporated in ZEUS to extend the stability range of the Euler solver. Take the 2D case as an example; the implicit residual smoothing formula is as follows:

$$(1-\beta_\xi \nabla_\xi \Delta_\xi)(1-\beta_\eta \nabla_\eta \Delta_\eta)\bar{R}_{i,j} = R_{i,j} \tag{6.16}$$

where $\mathbf{R}_{i,j}$ and $\bar{\mathbf{R}}_{i,j}$ are the residuals before and after smoothing, respectively. The quantity $\Delta\nabla$ is the standard second-difference operator and β is the residual smoothing coefficient that can be either a constant or a function of the local spectral radii.

The residual smoothing option is always recommended unless the surface mesh has triangular panels in which case ZEUS will turn off residual smoothing and reduce the CFL number accordingly inside the code. Another case for ZEUS to automatically turn off the residual smoothing is the body-fitted grid block modeled by using a **BLOCK1** bulk data card.

6.1.2 TRANSPIRATION BOUNDARY CONDITIONS

To generate a body-fitted grid could be a very difficult task for complex configurations. And the use of deforming mesh for unsteady simulations can cause troubles such as grid cross-over or over-skewed mesh. To circumvent these problems, ZEUS uses stationary Cartesian grid by applying the transpiration boundary conditions to account for both thickness and small amplitude motion of the wing surface. As the grid is fixed, all the metric time-derivative terms such as ξ_t, η_t and ς_t in Equations (6.2) and (6.3) disappear.

A thin wing moving or deforming slightly about its mean position is considered. The mean position of the wing lies in the horizontal plane $z=0$. The shape of the wing is defined as $z=f(x,y)$ and $g(x,y)$ for its upper and lower surfaces, respectively, and the instantaneous position of the wing is described by $z=F(t,x,y)$ and $z=G(t,x,y)$ for the upper and lower surfaces. Under the assumption $\|F\| \ll 1$, the first order approximation of the surface velocity boundary condition on the upper surface of the wing at instant t reads:

$$w(t,x,y,0^+) = u(t,x,y,0^+)F_x + v(t,x,y,0^+)F_y + F_t + O(F) \tag{6.17}$$

where the subscripts x, y, and t denote the partial derivatives with respect to x, y and t, respectively; $O(F)$ represents terms of the same order of magnitude as F or higher. The normal velocity boundary condition on the lower surface is treated similarly.

The boundary condition for pressure can be derived from the normal momentum equation:

$$\vec{n} \cdot \left[\frac{\partial \vec{V}}{\partial t} + \left(\vec{V} \cdot \nabla \right) \vec{V} \right] = \vec{n} \cdot \left(-\frac{\nabla p}{\rho} \right) \tag{6.18}$$

where $\vec{n}$ is the unit normal vector and $\vec{V}$ is the flow velocity vector at the wing surface. On the upper surface of the wing, the pressure gradient in the normal direction can be derived from the above equation:

$$\begin{aligned} p_z(t,x,y,0^+) &= p_x(t,x,y,0^+)F_x + p_y(t,x,y,0^+)F_y - \\ &\qquad \rho(F_{tt} + 2F_{tx}u + 2F_{ty}v + 2F_{xy}uv + F_{xx}u^2 + F_{yy}v^2) + O(F) \end{aligned} \tag{6.19}$$

More details about this unsteady Euler method applied on stationary Cartesian grid can be found in Zhang [2].

6.2 LINEARIZED EULER SOLVER IN ZEUS

Traditionally, to obtain the Generalized Aerodynamic Forces (GAFs) in the frequency domain, a time-domain simulation is performed, and then Fourier transformation is applied on the resulting time histories of the surface pressure. This process is time-consuming because, taking the harmonic forced excitation case, for example, several periods of unsteady flow computation is generally needed in order to achieve truly periodic flow solution. The linearized Euler method proposed by Laschka etc. [3] can reduce the harmonic forced excitation problem to a steady flow problem with regard to the small perturbation flow variables; thereby the frequency-domain unsteady forces can be obtained with only about twice the cost of a steady flow solution.

6.2.1 LINEARIZED EULER EQUATION ON STATIONARY GRID

By assuming the unsteadiness of a flow is small compared to its nonlinear mean flow, an unsteady flow can be decomposed into a steady part and an unsteady small perturbation part:

$$q = \overline{q} + \tilde{q}(t), H_1 = \overline{H}_1 + \tilde{H}_1(t), H_2 = \overline{H}_2 + \tilde{H}_2(t), H_3 = \overline{H}_3 + \tilde{H}_3(t) \tag{6.20}$$

where quantities marked with an over-bar are for mean flow and the ones with over-wave for the perturbation counterparts. For example, the vector of perturbation flow variables is defined as:

$$\tilde{q} = \begin{bmatrix} \tilde{\rho} \\ \widetilde{\rho u} \\ \widetilde{\rho v} \\ \widetilde{\rho w} \\ \tilde{e} \end{bmatrix} \tag{6.21}$$

As ZEUS uses stationary grid, all the metric terms including the inverse of the transformation Jacobian **J** are constant quantities. Therefore, the linearization process is much simpler than that presented in Laschka [3].

Substituting Equation (6.20) into (6.1), we have:

$$\frac{\partial J\left(\overline{q} + \tilde{q}(t)\right)}{\partial t} + \frac{\partial\left(\overline{H}_1 + \tilde{H}_1(t)\right)}{\partial \xi} + \frac{\partial\left(\overline{H}_2 + \tilde{H}_2(t)\right)}{\partial \eta} + \frac{\partial\left(\overline{H}_3 + \tilde{H}_3(t)\right)}{\partial \varsigma} = 0 \tag{6.22}$$

Collecting terms with over-bar in the above equation leads to the nonlinear steady-state Euler equation, while the rest of terms with over-wave form the linearized Euler equation. Considering the following relations:

$$\tilde{H}_1 = \frac{\partial \overline{H}_1}{\partial \overline{q}}\tilde{q} = \overline{A}_\xi \tilde{q}, \quad \tilde{H}_2 = \frac{\partial \overline{H}_2}{\partial \overline{q}}\tilde{q} = \overline{A}_\eta \tilde{q}, \quad \tilde{H}_3 = \frac{\partial \overline{H}_3}{\partial \overline{q}}\tilde{q} = \overline{A}_\varsigma \tilde{q} \tag{6.23}$$

where $\overline{A}_\xi$, $\overline{A}_\eta$ and $\overline{A}_\varsigma$ are the mean flow convective flux Jacobians for three curvilinear coordinate directions, we obtain the linearized Euler equation on stationary grid in the time-domain:

$$\frac{\partial\left(J\tilde{q}(t)\right)}{\partial t} + \frac{\partial\left(\overline{A}_\xi \tilde{q}(t)\right)}{\partial \xi} + \frac{\partial\left(\overline{A}_\eta \tilde{q}(t)\right)}{\partial \eta} + \frac{\partial\left(\overline{A}_\varsigma \tilde{q}(t)\right)}{\partial \varsigma} = 0 \tag{6.24}$$

Performing Fourier transformation on Equation (6.24) yields the linearized Euler equation in the frequency domain:

$$(i\omega)J\tilde{q}(\omega)+\frac{\partial\left(\bar{A}_{\xi}\tilde{q}(\omega)\right)}{\partial\xi}+\frac{\partial\left(\bar{A}_{\eta}\tilde{q}(\omega)\right)}{\partial\eta}+\frac{\partial\left(\bar{A}_{\varsigma}\tilde{q}(\omega)\right)}{\partial\varsigma}=0 \tag{6.25}$$

Equation (6.25) represents a steady-state Euler equation with regard to the vector of perturbation flow variables $\tilde{q}(\omega)$. To solve for $\tilde{q}(\omega)$, we first need to compute the nonlinear steady mean flow vector $\bar{q}$. After the mean flow solution is obtained, $\bar{A}_{\xi}$, $\bar{A}_{\eta}$ and $\bar{A}_{\varsigma}$ can be calculated once and for all, and then Equation (6.25) can be solved for each circular frequency ω by the five-stage R-K pseudo-time marching scheme mentioned earlier. One reason for the linearized Euler solver to be very efficient is that the mean flow need to be solved only once and thereafter the linearized Euler equation can be solved for any circular frequency ω and for different structure modes

6.2.2 BOUNDARY CONDITIONS

The wing surface velocity boundary conditions for the linearized Euler equation can be derived from the linearization of Equation (6.17):

$$\tilde{w}(t,x,y,0^{+})=\tilde{u}(t,x,y,0^{+})\bar{F}_{x}+\tilde{v}(t,x,y,0^{+})\bar{F}_{y}+\bar{u}(x,y,0^{+})\tilde{F}_{x}(t)+\bar{v}(x,y,0^{+})\tilde{F}_{y}(t)+\tilde{F}_{t}(t) \tag{6.26}$$

The frequency-domain counterpart of Equation (6.26) can be written as:

$$\tilde{w}(\omega,x,y,0^{+})=\tilde{u}(\omega,x,y,0^{+})\bar{F}_{x}+\tilde{v}(\omega,x,y,0^{+})\bar{F}_{y}+\bar{u}(x,y,0^{+})\tilde{F}_{x}(\omega)+\bar{v}(x,y,0^{+})\tilde{F}_{y}(\omega)+(i\omega)\tilde{F}(\omega) \tag{6.27}$$

$\tilde{F}$ represents any arbitrary surface motion of small amplitude. To generate GAFs, one can replace $\tilde{F}$ by the structural modes $\left[\phi_h\right]$, the control surface modes $\left[\phi_c\right]$, or the gust modes $\left[\phi_g\right]$.

The GAFs, in fact, represent the generalized aerodynamic forces due to a unit input. Take the sinusoidal gust as an example, the corresponding unit input for the sinusoidal gust in the time-domain is a traveling Dirac delta function shown in Figure 6.1.

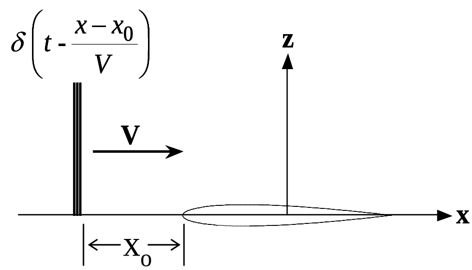


Figure 6.1 Traveling Delta Function of the Sinusoidal Gust

In the definition of the traveling Dirac delta function $\delta\left(t-\frac{x-x_o}{V}\right)$, x_o is the reference point of the gust, and x is any point on the aircraft. The frequency-domain counterpart of the traveling Dirac delta function can be obtained by Fourier transform:

$$\bar{\delta}(i\omega)=\int_{-\infty}^{\infty}\delta(t-\frac{x-x_0}{V})e^{-i\omega t}dt=e^{-i\omega\left(\frac{x-x_o}{V}\right)} \tag{6.28}$$

where $\bar{\delta}(i\omega)$ is the so-called sinusoidal gust. As $\bar{\delta}(i\omega)$ is already normalized by the freestream velocity and thus represents the induced angle of attack, the linearized Euler transpiration velocity boundary condition for the sinusoidal gust can be written as:

$$\tilde{w}(i\omega,x,y,0^{+})=\tilde{u}(i\omega,x,y,0^{+})\bar{F}_x+\tilde{v}(i\omega,x,y,0^{+})\bar{F}_y-\bar{u}(x,y,0^{+})\bar{\delta}(i\omega) \tag{6.29}$$

As for the pressure boundary conditions, the small perturbation pressure is not involved in the solution of the linearized Euler equation except for output purpose, so there is no need of pressure boundary conditions.

6.2.3 GENERATION OF GENERALIZED AERODYNAMIC FORCES (GAFS)

The major outcome from the unsteady CFD solution is the unsteady force which is the integral of the unsteady pressure on the configuration surface panels. To obtain the unsteady pressure from the Linearized Euler solution is very simple.

The linearization of the equation of state (4) yields the small perturbation pressure:

$$\tilde{p}=(\gamma-1)\left[\tilde{e}-\frac{1}{2}(2\bar{u}\widetilde{\rho u}-\bar{u}^2\tilde{\rho}+2\bar{v}\widetilde{\rho v}-\bar{v}^2\tilde{\rho}+2\bar{w}\widetilde{\rho w}-\bar{w}^2\tilde{\rho}\right] \tag{6.30}$$

The real and imaginary components of this perturbation pressure represent the first harmonics of the unsteady pressure. Therefore, the first harmonics of the unsteady pressure coefficient can be calculated as follows:

$$\mathrm{Re}\,C_p^1=\frac{2\,\mathrm{Re}(\tilde{p})}{\gamma M^2},\qquad \mathrm{Im}\,C_p^1=\frac{2\,\mathrm{Im}(\tilde{p})}{\gamma M^2} \tag{6.31}$$

The unsteady pressure coefficients thus obtained can be used to compute the generalized aerodynamic forces needed for analysis such as flutter boundary prediction.

In general, the frequency-domain equations of motion of an aeroelastic system excited by control surface deflections and atmospheric gust can be expressed as:

$$\left[-\omega^2[M_{hh}]+i\omega[C_{hh}]+[K_{hh}]-q_\infty\left[Q_{hh}(i\omega)\right]\right]\{\xi\}=q_\infty\left[Q_{hc}(i\omega)\right]\{\delta\}$$
$$+q_\infty\left[Q_{hg}(i\omega)\right]\frac{w_g}{V_\infty} \tag{6.32}$$

where $[M_{hh}]$, $[C_{hh}]$ and $[K_{hh}]$ are the generalized structural mass, damping and stiffness matrices, respectively. The subscript h denotes the index number of structural natural modes. ξ is the generalized modal coordinates, q_∞ and V_∞ are the dynamic pressure and free stream velocity, respectively, δ is the control surface deflections and w_g is the gust velocity.

There are three types of GAFs involved in Equation (6.32); namely the GAF due to structural modes $[Q_{hh}(i\omega)]$, the GAF due to control surface deflections $[Q_{hc}(i\omega)]$, and the GAF due to a sinusoidal gust $[Q_{hg}(i\omega)]$, where ω is the circular frequency. These three types of GAFs represent the linearized frequency-domain unsteady aerodynamics about a nonlinear steady mean flow condition due to small perturbations of structural deformation, control surface deflection, and gust excitation, respectively, and usually, are computed by an unsteady aerodynamic force generator such as ZEUS.

At a given ω, applying the unsteady transpiration boundary conditions of $[\phi_h]$, $[\phi_c]$ and $[\phi_g]$ to the linearized Euler equation yields three sets of unsteady aerodynamic forces, namely $[F_h]$ due to structural modes, $[F_c]$ due to control surface modes and $\{F_g\}$ due to gust. Then their respective GAFs can be obtained by simply pre-multiplying the transposed modal matrix $[\phi_h]^T$ to the aerodynamic forces:

$$\left[[Q_{hh}(i\omega)],[Q_{hc}(i\omega)],\{Q_{hg}(i\omega)\}\right]=[\phi_h]^T\left[[F_h],[F_c],\{F_g\}\right] \tag{6.33}$$

6.2.4 LINEARIZED EULER SOLVER VS. FULL EULER SOLVER

The Linearized Euler solver implemented in ZEUS provides another option for the generation of GAFs in addition to the full Euler solver. The major advantage of the Linearized Euler solver is its computational efficiency. By applying the linearized Euler method, a time-domain Euler simulation is reduced to a steady Euler solution and thus the computational time could be one order smaller. As the Linearized Euler run is basically a steady flow solution, its convergence is much easier to control compared to the full Euler unsteady run. For the Linearized Euler simulation, one only needs to change the number of pseudo-time steps and see how many orders of the residual drop from the beginning can be achieved. If there are more than three orders of residual drop, the solution should be pretty much converged, which means a further increase of the number of pseudo-time steps will not change the final solution much. On the other hand, to set up a full Euler unsteady run, one has to specify a proper physical time step size, and also enough number of pseudo-time steps for solution convergence within each physical time step. As for how many orders of residual drop within each physical time step are adequate, it is a tough question to answer because the situation is complicated by the physical time-step size employed and also the convergence behavior could be very different for the physical time steps in different periods of the configuration motion.

For the ideal case that the Linearized Euler solver of ZEUS can use the default CFL number of 7.0 with residual smoothing, the Linearized Euler simulation takes only about one-eighth of the computational time needed for the full Euler run. However, it has been noticed that the convergence behavior of the current Linearized Euler solver in ZEUS depends on the density of the farfield computational grid. The computational mesh for ZEUS is getting coarser approaching the farfield boundaries in order to save computational time. Those large grid cells in the farfield of the computational domain generally won't cause any convergence problem for the full Euler solver, while it is not the case for the current Linearized Euler solver. It is found that for frequency-domain flutter analysis the Linearized Euler solver is getting more and more difficult to converge with the increase of reduced frequencies. The Linearized Euler solver will blow up, or CFL number needs to be reduced to a very small value unless a much denser mesh is used by reducing the growth rate parameters in **MESHPRM** bulk data card. Later, it is also found that chopping off several layers of large grid cells in the farfield can improve the convergence of the Linearized Euler solver for high reduced frequency runs. The theory regarding the relationship between the density of the computational mesh in the farfield and the applicable CFL number as well as the reduced frequencies is not clear yet for the current Linearized Euler solver in ZEUS. The option of using 6 integer parameters specified in **MKPARAL** bulk data card to exclude several layers of large grid cells along the farfield boundaries from the Linearized Euler solution procedure is provided temporarily to avoid using too small a CFL number for high reduced frequency runs. Please note that shrinking the computational domain too aggressively may result in loss of solution accuracy. The current Linearized Euler solver, even with CFL=3.0, can still save about three-quarters of the computational time needed for the full Euler run with CFL=7.0.

As frequency-domain flutter analysis is performed frequency by frequency and mode by mode, it is highly recommended that a test run at the largest reduced frequency for one mode is tried first to make sure the Linearized Euler solution converges well before the real flutter run should start. Otherwise, ZEUS may stop at certain high reduced frequency run after successfully finishing a lot of runs at low reduced frequencies. If the default CFL number doesn't work, the user should gradually decrease the CFL number until the Linearized Euler solver converges well for the highest reduced frequency run, and in the meantime adjust parameters NEWTN/NCYC accordingly to make sure that about three orders of residual drop are achieved. If CFL number needs to be reduced below 1.0, which will practically render the Linearized Euler solver less efficient than the full Euler solver, going back to improve the mesh generation is recommended, or simply resort to the full Euler solver instead.

6.3 BOUNDARY-LAYER COUPLING OPTION IN ZEUS

In the last two sections, an unsteady full Euler solver and a Linearized Euler solver available in ZEUS have been presented in details. ZEUS has the capability of automatically generating a Cartesian grid for use by its Euler solver, which relieves the user of the burden to generate a body conforming grid. To develop a Reynolds-Average-Navier-Stokes (RANS) solver on a Cartesian grid is a daunting task, and thus there is no RANS solver option for ZEUS. However, ZEUS provides a boundary-layer coupling capability to account for the viscous effects if desired.

For aerodynamic flows of general concern, the Reynolds number is usually in millions. According to Prandtl, for such high Reynolds number flow, the boundary layer is thin along the solid wall, and thus the viscous effects are considered as generally confined to this thin boundary-layer. Therefore, the flow domain can be decomposed into two zones: an outer inviscid flow and a viscous boundary-layer flow. The method of solving the two parts of the flow field separately and coupling them together has been proven to be a good alternative to the Reynolds-Average-Navier-Stokes (RANS) method when dealing with viscous flow. With regard to the solution of the boundary-layer part, there are both integral method and finite-difference method available. In consideration of computational cost as well as uncertainties of turbulence modeling involved in the finite-difference method, an integral boundary-layer method is chosen to be coupled with the Euler solver of ZEUS.

6.3.1 INTEGRAL BOUNDARY-LAYER METHOD

The integral boundary-layer method in ZEUS is actually applied in a 2-D quasi-steady manner, which means the boundary-layer parameters are solved at each physical time step independently and in the freestream x direction only and then coupled with the inviscid Euler flow solution strip by strip in the y direction.

The integral boundary-layer quantities are governed by a set of ordinary differential equations:

$$\begin{aligned}
\frac{dU_e^v}{dx} &= F_1 + F_2 \frac{1}{\bar{m}} \frac{d\bar{m}}{dx} \\
\frac{d\bar{H}}{dx} &= F_3 + F_4 \frac{1}{\bar{m}} \frac{d\bar{m}}{dx} \\
\frac{dC_E}{dx} &= F_5 + F_6 \frac{dU_e^i}{dx}
\end{aligned} \tag{6.34}$$

where $\bar{H}$ is shape factor, C_E is lag entrainment coefficient, U_e^v is the boundary-layer edge velocity while U_e^i is the Euler solver calculated inviscid flow velocity at the wall where the boundary-layer grows, and F_i are nonlinear functions of the boundary-layer parameters involved. Please refer to [2] for details about the derivation of those equations and definitions of all the parameters. The so-called perturbation mass flow parameter $\bar{m}$ is defined as:

$$\bar{m} = \rho_e U_e \delta^* \tag{6.35}$$

where ρ_e and U_e are the boundary-layer edge density and velocity while δ^* is the boundary-layer displacement thickness.

As Equation (6.34) involves flow quantities such as density and velocity at the edge of the boundary-layer that are not known beforehand, the integral boundary-layer solver needs to couple with the inviscid Euler solver iteratively until the boundary-layer edge velocity obtained from the integral boundary-layer solver matches the wall surface velocity from the Euler solver. The integral boundary-layer coupling procedure goes like this: (1) first obtain the Euler solution with boundary-layer effects included and take the density and velocity U_e^i at the wall as the boundary-layer edge quantities, combined with the boundary-layer displacement thickness from the previous iteration to make an approximated mass flow parameter $\bar{m}$; (2) solve the boundary-layer equations with this guessed mass flow parameter to obtain a new boundary-layer edge velocity U_e^v; (3) compare the two velocities U_e^i and U_e^v, if the difference is within certain tolerance, we say the boundary-layer coupling is converged, otherwise, update the boundary-layer displacement thickness applying a scheme described in the next section and pass it to the Euler solver for use in the next iteration.

The coupling between the Euler solver and integral boundary-layer solver in ZEUS is performed at the end of each Newton sub-iteration. Therefore, the number of NEWTN coupling iterations within each physical time step will be performed for unsteady cases, whereas the number of TRMSTEP x NEWTN (defined in the **MKPARAM** bulk data card) iterations for steady flow computation. There is no boundary-layer coupling convergence control implemented in ZEUS yet; the user needs to carefully set the parameter NEWTN to guarantee a converged solution. Please note that the convergence of the Euler solver itself is progressing with the Newton sub-iterations, which means the Euler solution is not converged when coupled with the integral boundary layer solver in the early stages of Newton sub-iterations.

6.3.2 BOUNDARY-LAYER COUPLING SCHEME

In the third stage of the boundary-layer coupling procedure mentioned in the last section, ZEUS provides two different schemes for the update of the boundary-layer displacement thickness δ^*.

The first one is called Carter's relaxation scheme [4], and it is very simple:

$$\frac{\delta_{n+1}^*}{\delta_n^*} = 1 + \omega\left(\frac{U_e^v}{U_e^i} - 1\right) \tag{6.36}$$

where n is the index of iteration step, and ω is the relaxation factor that is corresponding to entry RF of the **VISCOUS** bulk data card in ZEUS.

After the boundary-layer displacement thickness δ^* is obtained from the integral boundary-layer solver at the end of each Newton sub-iteration, simply add its slope on top of the wing surface slope F_x used in Equation (6.17) to account for the boundary-layer thickening effect in the inviscid Euler solution.
For attached boundary-layer or boundary-layer with small separation bubble, Carter's scheme converges well and fast, while it tends to fail for cases with large separation region. In an effort to extend the applicability range of the integral boundary-layer method, Edwards developed a variable gain integral control coupling scheme [5] which introduces some limiters and gain control functions to help the integral boundary-layer coupling procedure in handling cases with extensive separation region. Basically, a formulation similar to Equation (6.36) is applied:

$$\delta^*_{n+1} = \delta^*_n + XKINT * t_1 * (1.0 + t_1 * t_2)(\frac{U_e^v}{U_e^i} - 1)$$
$$t_1 = f_1(GAIN1, BRK1, BRK2, \delta^*)$$
$$t_2 = f_2(GAIN2, BRK3, BRK4, C_f)$$
(6.37)

where XKINT is the major gain factor, t_1 is a nonlinear gain factor scheduled on δ^* rising from a value of 1.0 for values of δ^* less than BRK1 to a value of GAIN1 for δ^* greater than BRK2, t_2 is another nonlinear gain factor scheduled on skin friction coefficient C_f falling from a value of 0.0 for C_f greater than BRK4 to a value of GAIN2 for C_f less than BRK3. Please refer to [5] for more details of Edwards' coupling scheme.

In ZEUS, Edwards' boundary-layer coupling scheme is the default option with ICOUP=0 in the **VISCOUS** bulk data card. However, if Carter's scheme can achieve a converged solution, it should be preferred because it converges faster than Edwards', and once converged, both coupling methods should yield the same result.

As for those control parameters pertaining to the Edwards' scheme only, the default values are recommended until the effects of changing those parameters are made clear later with more experience.

6.4 LOOPS AND TIME STEPS IN ZEUS

As mentioned in Section 6.1.1, ZEUS solves the time-accurate Euler equations using a dual-time stepping method, and thus there are pseudo-time marching loop and physical time stepping loop in the Euler solver of ZEUS. In addition, for overset interpolation and boundary-layer coupling purpose, the Euler solver needs another loop which is called the Newton sub-iteration loop. Here, the word "Newton" is used for convenience and there is nothing to do with the so-called Newton–Raphson Method.

Figure 6.2 shows the flowchart of the Euler solver of ZEUS. A Euler cycle refers to a 5-stage Runge-Kutta pseudo-time marching scheme presented by Equation (6.11). The loop of Euler cycles combined with the Newton sub-iterations loop make the so-called pseudo-time marching iterations to achieve converged steady flow solution or instantaneous flow solution at each physical time step for the time-accurate case. The pseudo-time marching step size is determined internally and controlled by the CFL number specified in the **MKPARAM** bulk data card. For the rigid-body steady flow simulation, the physical time stepping loop is not necessary, and NSTEP is equal to TRMSTEP specified in the **MKPARAM** bulk data card. In this case, TRMSTEP can be set to 1 , and the solution convergence is determined by the value of NEWTN instead of TRMSTEP. For static aeroelastic **TRIM** analysis, the physical time stepping loop is necessary as the coupling with structural solver needs to be performed iteratively within this loop, although the time step size used for this loop is not really physical time step size just as in the rigid-body steady flow solution. For the truly time-accurate simulation such as

MLOADS run and frequency-domain **FLUTTER** run, the physical time stepping loop is absolutely needed.

The physical time step size is specified by DT of **MLDTIME** bulk data card for **MLOADS** run, and NSTEP is determined accordingly, while for **FLUTTER** run, NSTEP is equal to FLTSTEP of **MKPARAM** bulk data card and the physical time step size is determined by NSTEP together with the reduced frequency used.

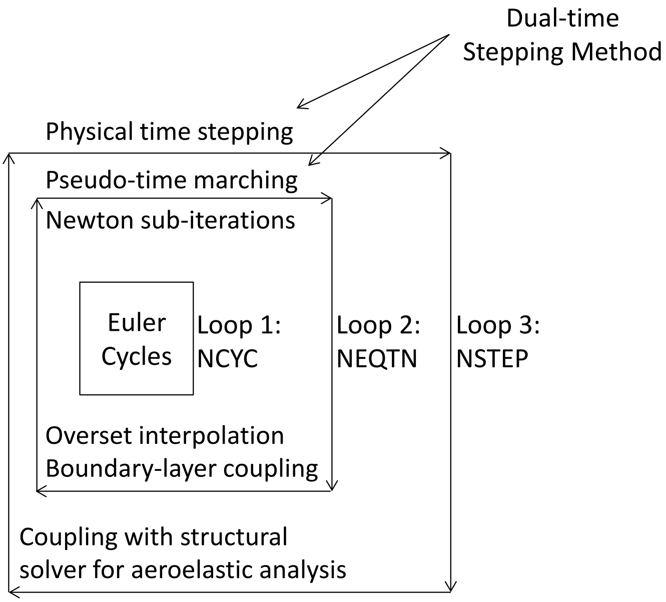


Figure 6.2 Flow Chart of the Euler Solver in ZEUS

REFERENCES

[1] Jameson, A.; Schmidt, Wolfgang; and Turkel, Eli "Numerical Solution of the Euler Equations by Finite Volume Methods Using Runge Kutta Time Stepping Schemes". AIAA 81-1259, June 1981.

[2] Zhang, Z., Liu, F., and Schuster, D. M. "An Efficient Euler Method on Non-Moving Cartesian Grids with Boundary-Layer Correction for Wing Flutter Simulations". AIAA 2006-0884, Presented at the *44th AIAA Aerospace Sciences Meeting and Exhibit*. 2006. Reno, Nevada.

[3] Kreiselmaier E., Laschka B., "Small Disturbance Euler Equations: Efficient and Accurate Tool for Unsteady Load Prediction". Journal of Aircraft, 2000. 37(5): p. 770-778.

[4] J.E. Carter, "A New Boundary-Layer Inviscid Iteration Technique for Separated Flow". AIAA Paper 1979-1450, July 1979.

[5] J.W. Edwards, "Transonic Shock Oscillations Calculated with a New Interactive boundary Layer Coupling Method". AIAA Paper 93-0777, Jan. 1993.

Chapter 7

MODELING GUIDELINES FOR SPLINE

7.1 COORDINATE SYSTEMS OF ZEUS AND STRUCTURAL FINITE ELEMENT MODELS

Aeroelastic analysis involves the coupling of the aerodynamic forces and structural responses. In practice, the aerodynamic model and structural finite element model are constructed by different groups of engineers. This can result in a situation where the aerodynamic model and the structural finite element model are located in different regions within the same coordinate system. In order to transfer the displacements and loads between these two models, the spline module of OVECAP requires a coordinate transformation to align the overall geometry of the aerodynamic and structural models. This is discussed in the following subsections.

7.2 AERODYNAMIC COORDINATES

The aerodynamic coordinate system is the coordinate system in which the entire aerodynamic model geometry is defined. Since the Euler-Solver module inherently defines the x-axis as the compressible direction of the flow, the x-axis of the aerodynamic coordinates must be parallel to the flow direction (Figure 7.1). If a pilot were situated in an aerodynamic model, the y-axis of the aerodynamic coordinates must be in the direction of the pilot's right-hand side (see Figure 7.1). If the aircraft configuration is symmetric about the x-z plane of the aerodynamic coordinates and the structural modes are the symmetric modes, ZEUS only requires modeling of half of the configuration. This is done by setting XZSYM= 'YES' in the **AEROZ** bulk data card. Again, the aerodynamic model must be located on the right-hand side of the pilot (i.e., the positive y-axis direction). In addition, the aerodynamic model must be

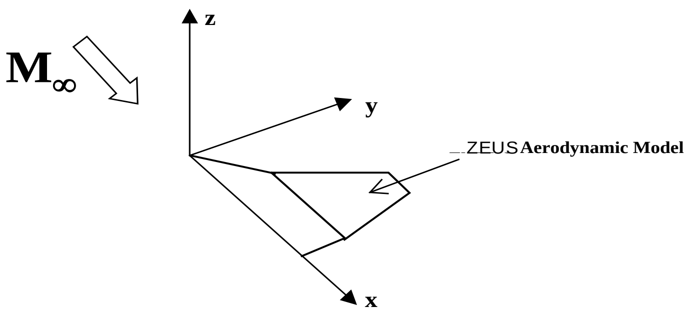


Figure 7.1 Proper Orientation of a ZEUS Model in the Aerodynamic Coordinate System

located at the zero angle-of-attack and zero side-slip angle conditions. The angle-of-attack and side-slip angle effects are introduced through the boundary conditions.

7.3 OVERLAPPING OF THE FINITE ELEMENT MODEL ON THE AERODYNAMIC MODEL THROUGH COORDINATE TRANSFORMATION

The structural finite element model can be oriented in an arbitrary fashion with respect to the aerodynamic coordinates. Therefore, to establish a spline between these two models requires a transformation of the grid point locations of the structural finite element model such that the structural finite element model coincides with the aerodynamic model. Such a transformation is defined by a **CORD2R** bulk data card whose identification number is referred to by the ACSID entry in the **AEROZ** bulk data card. For ease of discussion, the axes of the **CORD2R** system are denoted here as x', y' and z'. (Figure 7.2(b))

Next, let us assume that a finite element model of a half aircraft configuration exists whereby the fuselage is oriented along the negative y-axis, and the wing is parallel to the negative x-axis. The fuselage of the aerodynamic model is always along the x-axis (i.e., the flow direction). When the aerodynamic model and structural finite element model are graphically displayed simultaneously, one will see two aircraft with different orientations. This is shown in Figure 7.2(a). Note that in Figure 7.2(a), the nose of the fuselage of the finite element model is located at x = z = 0 and y = -100 whereas that of the aerodynamic model at x = y = z = 0.

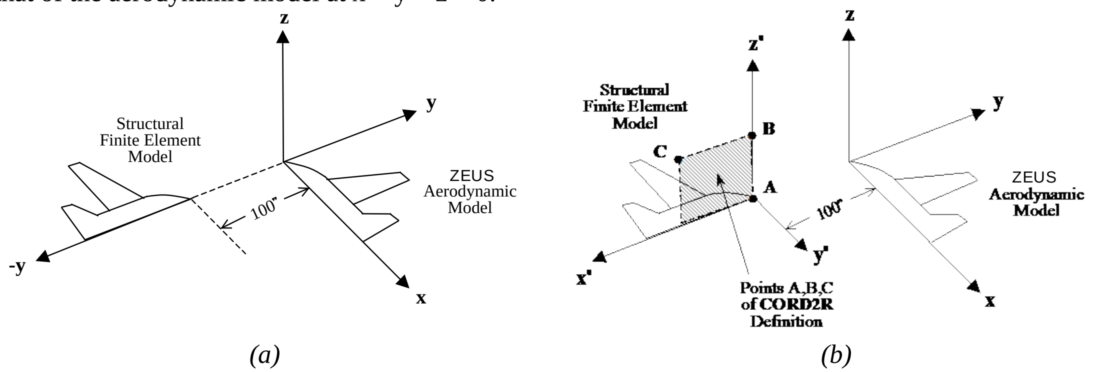


Figure 7.2 Definition of the Local Coordinates Used to Spline the Finite Element and Aerodynamic Models

For a pilot situated in the finite element model, the x'-axis of the **CORD2R** coordinate system is toward the pilot's face and the positive y'-axis is on the pilot's right-hand side. Since the nose of the fuselage of the aerodynamic model is located at x = y = z = 0, the origin of the x', y' and z' axes must be at the nose of the finite element model which corresponds to x = z = 0, and y = -100 with respect to Aerodynamic model coordinate. Such a **CORD2R** coordinate system is shown in Figure 7.2(b). Figure 7.2(b) also indicates that the finite element model is on the pilot's left-hand side. For such a case, the entry FLIP = 'YES' in the **AEROZ** bulk data card must be specified to flip the finite element model from the pilot's left-hand side to the right-hand side.

As an example of Figure 7.2, the following **CORD2R** bulk data card:

CORD2R	50		0.0	-100.0	0.0	0.0	-100.0	1.0	+CRD1
+CRD1	0.0	-101.0	1.0						

could be referred to by an **AEROZ** bulk data card such as:

AEROZ	50	YES	YES	. . .					

The spline module of ZEUS first transforms the finite element model to the aerodynamic coordinates using the coordinate specified by the **CORD2R** bulk data card (in this case, with an identification number of 50) and then flips the finite element model from the pilot's left-hand-side to the right-hand-side (since FLIP = "YES" is specified in the **AEROZ** bulk data card). To verify this CORD2R bulk data input, it is recommended that the **PLTAERO** bulk data card with entry FEMGRID = "YES" be used. The user can graphically display the aerodynamic model and the structural finite element grid points simultaneously to verify the overlapping of these two models

It should be noted that while performing a coordinate transformation for the finite element grid points, the spline module also transforms the degrees of freedom (d.o.f.) of displacements at structural grid points from the structural finite element model to the aerodynamic model. Thus, the spline matrix generated by the spline module of ZEUS establishes a direct link between the displacements at the structural finite element grid points and the aerodynamic boxes.

7.4 MODELING GUIDELINES FOR VARIOUS SPLINE METHODS

Since the requirements to generate the discretized models for the structural analysis and the aerodynamic analysis are subject to different engineering considerations, the grid point locations of these two models may be considerably different. This gives rise to the problem of transferring the displacements and forces between these two grid systems. Four spline methods are incorporated, in the spline module of ZEUS, which generate spline matrices to perform the displacement and force transferal between the structural finite element model and the ZEUS model. These four spline methods are:

- Infinite Plate Spline (IPS) Method by the **SPLINE1** bulk data card
- Beam Spline Method by the **SPLINE2** bulk data card
- Thin Plate Spline (TPS) Method by the **SPLINE3** bulk data card
- Rigid Body Attachment by the **ATTACH** bulk data card

The generation of the spline matrix is performed on a component-by-component basis. The selection of the spline method for a given component depends on the type of component in the ZEUS model (i.e., wing-like or body-like component) and the type of elements (i.e., beam or plate element) used in the finite element model. For instance, if a body-like component is modeled by a **BODY7** in the ZEUS model and if beam-type elements are used for the finite element model, then the beam spline method should be employed. If wing-like components are modeled by a **CAERO7** in the ZEUS model and plate-type elements are used for the finite element model, then the IPS method should be used. The TPS method is a 3-D spline method that can link a set of finite element grid points in 3-D space to either a **BODY7** or **CAERO7** component. The **ATTACH** bulk data card handles the special case in which a component is absent in the finite element model but is present in the ZEUS model. A typical example of such a special case is an underwing store that is represented by a concentrated mass at a single finite element grid point but is completely modeled (by a **BODY7**) in the ZEUS model.

Experience has shown that most of the errors in performing aeroelastic analysis are introduced in the spline procedure. The following modeling guidelines present several situations in which inaccurate spline results are easily introduced due to incorrect input set-up.

7.5 ILL-CONDITIONED SPLINE MATRIX DUE TO COINCIDENT FINITE ELEMENT GRID POINT LOCATIONS

The selection of the finite element grid points that are to be linked to an aerodynamic component is completely at the users' discretion. These grid points are defined by **SET1** or **SET2** bulk data cards. Should two of the selected finite element grid points be located within a small tolerance of one another (tolerance set by EPS defined in the **SPLINE1** and **SPLINE3** bulk data cards), the resultant spline matrix is either singular or ill-conditioned. This input error is automatically detected by the ZEUS spline module. However, certain scenarios exist in which this kind of input error may not be detected by the spline module.

As an example of such a scenario, Figure 7.3 shows the cross-section of a wing-like component in which the solid circles represent the finite element grid points on the upper and the lower skins and the line represents the side view of a **CAERO7** macroelement. All finite element grid points appear to be well separated. If the IPS method is selected as the spline method, the spline module projects the finite element grid points onto the plane of the **CAERO7** macroelement (Figure 7.3(b)). This plane is called the "spline plane."

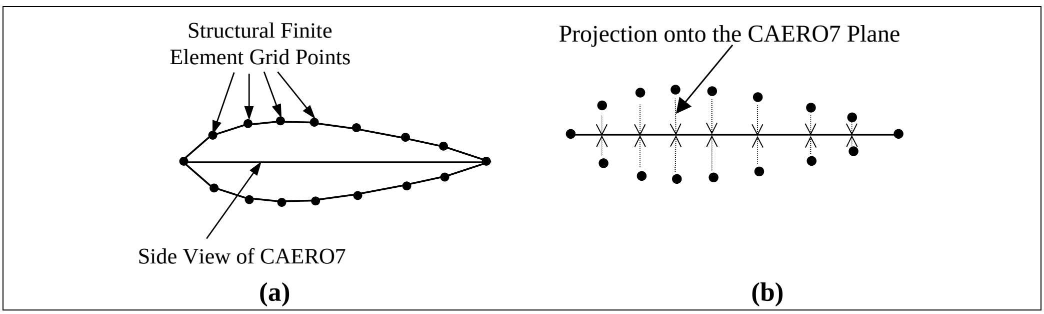


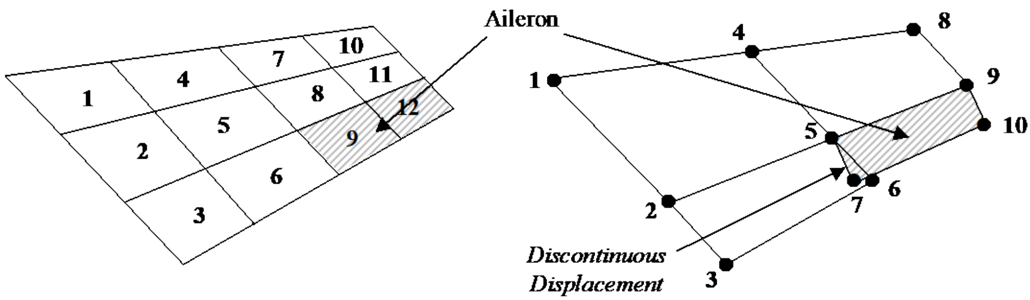
Figure 7.3 Cross-Section of a Wing-Like Component

If the projection of two grid points on the spline plane are too close to one another, an ill-conditioned spline matrix results. In this situation, the error condition may not be detected by the spline module. To avoid this input error, it is recommended that either the upper or the lower grid points, but not both, be included in the **SET1** bulk data card.

The spline case illustrated in Figure 7.3(a) is an ideal case for the TPS method. Since TPS is a 3-D spline method, there is no requirement to define a spline plane for grid point projection. Therefore, all upper and lower grid points can be included in the spline. However, this is true only for a thick wing-like component. As described in the remarks of the **SPLINE3** bulk data card, the structural points used by the TPS method cannot be located close to or within the same plane. Otherwise, an ill-conditioned spline matrix may result. For such a case, where the wing-like component thickness is very thin, the IPS method is recommended, but only with the selection of either the upper skin or the lower skin grid points.

7.6 SPLINE FOR DISCONTINUOUS STRUCTURE

A typical case of a discontinuous structure is a control surface. The control surface creates a discontinuous displacement between its side edges and the main wing as well as the discontinuous slopes along the hinge line, which may have a large impact on the aeroelastic response. For this reason, it becomes important to accurately transfer these discontinuous displacements and slopes from the finite element grid points to the aerodynamic model.



(a) CAERO7 Macroelement (b) Structural Finite Element Model

Figure 7.4 Spline of Discontinuous Structure Due to a Control Surface

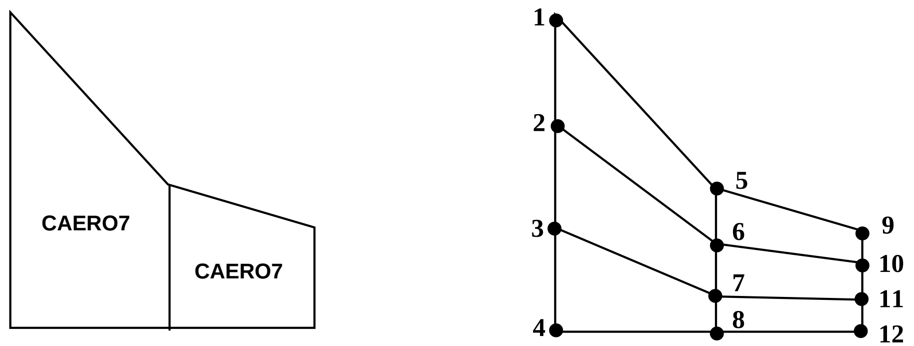
Figure 7.4(a) presents a wing with aileron configuration modeled by a **CAERO7** macroelement that includes 12 wing boxes, denoted as box 1 through box 12. The shaded area represents the aileron, and its corresponding wing boxes are box 9 and box 12. The finite element model shown in Figure 7.4 b) consists of 4 plate-type elements generated by the connection of the ten grid points (represented by the solid circles and denoted as grid points 1 through 10). Discontinuous displacement occurs between the inboard edge of the aileron and the main wing due to the discontinuous structure (between grid points 6 and 7). Because the finite element model exclusively employs plate type elements, the IPS method should be selected for this case.

Since the IPS method is formulated based on the structural equation of an infinite plate, the continuity of displacement is inherently imposed. This indicates that if all of the finite element grid points shown in Figure 7.4(b) are included in the spline, the resultant displacement on the **CAERO7** macroelement are continuous. In this case, failure to transfer discontinuous displacement due to the aileron will lead to incorrect aeroelastic results.

The correct technique to be used in this spline case is to apply the IPS method on the main wing and on the aileron separately by specifying two **SPLINE1** bulk data cards. The first **SPLINE1** established for the main wing should include the wing boxes (boxes 1 – 6, plus 7, 8, 10, and 11) and finite element grid points corresponding to the main wing only (grid points 1 – 6, plus 8 and 9). Likewise, the second **SPLINE1** established for the aileron should include only those wing boxes (boxes 9 and 12) and finite element grid points (5, 7, 9, and 10) associated with the aileron.

7.7 ENSURING CONTINUOUS STRUCTURE ACROSS TWO ADJACENT CAERO7 MACROELEMENTS

One of the modeling restrictions of the **CAERO7** macroelement is that it can only represent trapezoidal types of surfaces, i.e., the inboard and outboard edges must be parallel to the x-axis of the aerodynamic coordinates. Therefore, to model a non-trapezoidal type of wing-like component may require more than one **CAERO7**. Figure 7.5 (a) presents a cranked wing planform that is modeled by two **CAERO7** macroelements; one for the inboard region and one for the outboard region. The plate-type finite element model shown in Figure 7.5 (b) has 12 grid points, denoted as grid point 1 through grid point 12.



(a) Aerodynamic Model *(b) Structural Finite Element Model*

Figure 7.5 Spline for a Cranked Wing Planform

Two **SPLINE1** bulk data cards are required to spline the two **CAERO7** macroelements to the structure. The structural finite element model by itself is a continuous structure and should not incur any discontinuous slopes. Discontinuous slopes across the two **CAERO7** macroelements result if the inboard **CAERO7** only refers to the finite element grid points located in the inboard region (grid points 1 through 8) and the outboard **CAERO7** only refers to the finite grid points located in the outboard region (grid points 5 through 12). Such discontinuous slopes across the two **CAERO7** macroelements are incorrect and will lead to incorrect aeroelastic results.

The correct technique for this spline case is to use the IPS method and to ensure that the inboard and outboard **CAERO7** macroelements refer to <u>all</u> the grid points in the finite element model (grid points 1 through 12). The infinite plates generated by the IPS method for these two **CAERO7** macroelements are then identical, leading to continuous displacements and slopes across these two wing components.

7.8 ACCURATE ROTATIONAL STRUCTURAL DISPLACEMENT FOR BEAM SPLINE METHOD

Unlike the IPS and TPS methods, which adopt only the translational displacements at the structural grid points, the beam spline method requires both the translational and rotational displacements.

Often in structural finite element analysis, the translational displacements are included as the analysis set (i.e., A-set) d.o.f. Since the modal analyses of finite element methods only assure accurate modal displacements for the A-set d.o.f., exclusion of the rotational displacement for A-set d.o.f. in the beam spline method leads to inaccurate spline results on the aerodynamic model.

7.9 INACCURATE SPLINE RESULTS DUE TO EXTRAPOLATION

Since structural grid points are usually placed at major load carrying components, the structural finite element model may appear to be “shorter” than the aerodynamic model. A typical case where this can occur is in modeling the structural wing torque box of a wing component. A finite element wing model that does not fully extend to the leading and trailing edges of the wing may result an inaccurate spline result due to extrapolation. Another typical case is the beam-type element model of a fuselage component. Since the nose section of a fuselage is often considered a non-structural part and, therefore, requires no structural modeling, the beam model may end up shorter than the actual length of the fuselage.

Extrapolation is performed for the spline of aerodynamic boxes located outside the domain of the structural finite element grid points. Both of the plate spline methods (IPS and TPS) and the beam spline method incorporated within the spline module of ZEUS provide a purely linear extrapolation only if the aerodynamic box is located far away from the finite element model. Otherwise, distortions and oscillations may occur in the extrapolation regions. For this reason, extrapolation should be avoided.

To circumvent the extrapolation problem, it is recommended that extra grid points located at the leading and trailing edges of the wing or at the nose of the fuselage be added in the structural finite element model. These grid points can then be connected by rigid elements to their adjacent grid points. Thus, the problem associated with extrapolation can be avoided.

As a final note, graphical display of the displacements on the aerodynamic model for spline verification is highly recommended. It is for this reason that ZEUS provides an option to generate output files containing the aerodynamic box and corresponding displacement data using the **PLTMODE** bulk data card. Visual inspection of the displacements for both the aerodynamic and the finite element models would minimize errors caused by incorrect implementation of the spline.

Chapter 8

PLOT FILES

This section describes the ZEUS output files generated for plotting purposes. Output plot files are generated by the existence of any of the following bulk data cards within the bulk data input section: **PLTAERO**, **PLTCP**, **PLTFLUT**, **PLTMODE**, **PLTSLP**, **PLTTIME**, **PLTTRIM**, and/or **PLTVG** (described in Section 2.5). Table 2.1 from Section 2.4 is once again presented to list the output plot file capability of the ZEUS software system.

Category	**Associated Bulk Data Card**	**Description**	**Software Compatibility**
Aerodynamic Model	**PLTAERO**	Generates an ASCII text file for plotting the aerodynamic model	- TECPLOT
Steady/Unsteady Pressures	**PLTCP**	Generates an ASCII text file for plotting the steady/unsteady pressure coefficients (XY-Curve plot only has TECPLOT format)	- PATRAN - TECPLOT - I-DEAS - FEMAP - ANSYS - NASTRAN - PEGASUS
Flutter Mode Shapes	**PLTFLUT**	Generates an ASCII text file for plotting the flutter mode	- PATRAN - TECPLOT - I-DEAS - FEMAP - ANSYS - NASTRAN
Interpolated Structural Modes	**PLTMODE**	Generates an ASCII text file for plotting the interpolated structural mode on the aerodynamic model	- PATRAN - TECPLOT - I-DEAS - FEMAP - ANSYS - NASTRAN
Slopes of the Aerodynamic Surface	**PLTSLP**	Generates an ASCII text file for plotting the slopes of the aerodynamic surface mesh	TECPLOT
Transient Response Analysis	**PLTTIME**	Generates an ASCII text file for plotting the transient response	- PATRAN - TECPLOT - I-DEAS - FEMAP - ANSYS - NASTRAN

Static Aeroelastic / Trim Analysis Results	**PLTTRIM**	Generates an ASCII text file for the post-processing of the static aeroelastic/trim analysis	- PATRAN - TECPLOT - I-DEAS - FEMAP - ANSYS - NASTRAN - PEGASUS
Flutter Damping and Frequency Results	**PLTVG**	Generates an X-Y plot file (ASCII text) for plotting the flutter frequency and damping curves	- I-DEAS - FEMAP - TABLE - PEGASUS *and most other spreadsheet applications*

All output plot files are saved in ASCII text format and can be directly read in by the graphical software programs listed in the above table (or any equivalent software packages that can process the same data format). The PATRAN output is a combination of neutral file(s) (containing the aerodynamic model) and results file(s) (containing the displacement(s) or pressure results) output. When outputting PATRAN flutter results (PLTFLUT) or transient results (PLTTIME), ZEUS will generate one results file for each time step. **Note that for the PATRAN output option, the aerodynamic models generated by the PLTxxxx bulk data cards (all stored in neutral file format) are all different from one another and cannot be used interchangeably.** This restriction is due to the necessity of duplicating grid points in some of the output plot files to allow for viewing (or animation) of discontinuous components (e.g., a flapping control surface in a modal analysis). In addition, some plot files (like flutter mode shape) require displaying of both sides of the model even though only half of the model may have been specified in the input. For this reason, the user must take extra care when requesting multiple plot files in PATRAN format to ensure that the aerodynamic model names are unique (or they will be overwritten by the default name of 'AEROGEOM.PAT'). The TECPLOT output is in Tecplot's finite element zone input format. The I-DEAS output is in universal data file format. The FEMAP output is in FEMAP neutral file format. The ANSYS format is identical to the FEMAP neutral file output format and can be read into ANSYS via a translator developed by PADT Inc. in Tempe, Arizona (see Appendix A, Version 4.3 enhancements, Item #3). **Note that the ANSYS option description is not included in this Chapter since it is identical to the FEMAP output option.** Excel output is in column format and can be read in by virtually any spreadsheet application. Finally, the NASTRAN supported output is in bulk data format and can be plotted by any graphical software package capable of reading in and displaying NASTRAN bulk data input (e.g., ARIES, FEMAP, etc.)

The following sections describe each of the output plot files listed above. Samples of each output file taken from the cropped wing and trim forward swept wing demonstration test cases are presented along with descriptions of the output file contents.

8.1 Aerodynamic Model (PLTAERO)

An output data file of the aerodynamic model can be generated with the **PLTAERO** bulk data card (see Figure 8.1). Viewing the aerodynamic model is extremely useful in determining if the aerodynamic configuration is modeled properly. Often numeric typos are entered in the bulk data input that can go undetected in the analysis. For example, an aerodynamic coordinate system that is referred to by an underwing store (**BODY7** bulk data card) may be located in the wrong place due to a typo in the **ACOORD** bulk data card. In this case, the store would be located somewhere other than its intended position. This error in the aerodynamic configuration can quickly be detected by viewing the aerodynamic model.

The **PLTAERO** output plot file contains the aerodynamic box corner grid points along with the aerodynamic box connectivity information. The data for this output file is generated in the High-Fidelity

Geometry (HFG) Module based on the aerodynamic wing (**CAERO7**) and body (**BODY7**) macroelements specified in the bulk data section of the input deck. Only quadrilateral aerodynamic boxes are generated. Triangular type elements are generated by having two coincident aerodynamic grid points.

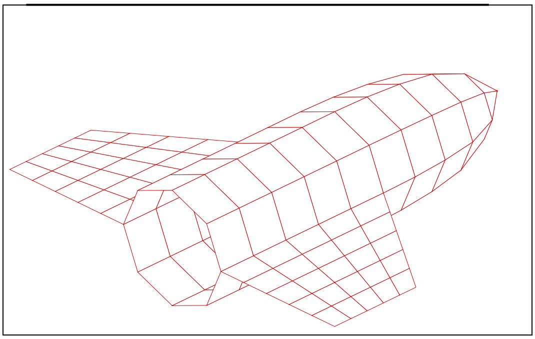


Figure 8.1 Plot of the Aerodynamic Model

- ***Tecplot Compatible Output***

A sample of the Tecplot compatible output is shown as follows:

```
TITLE ="AERO MODEL WITH    91 AERO GRIDS &    17 FEM GRIDS. AERO BOXES=    65"
VARIABLE = X, Y, Z, EXTID
ZONE I=     108 J=      65 F=FEPOINT
 -1.0000E+02  0.0000E+00  0.0000E+00     201
 -1.0000E+02  0.0000E+00  0.0000E+00     202
                  .
                  .
                  .

       2       7       6       1
       3       8       7       2
                  .
                  .
                  .
```

Grid Point Identification Numbers

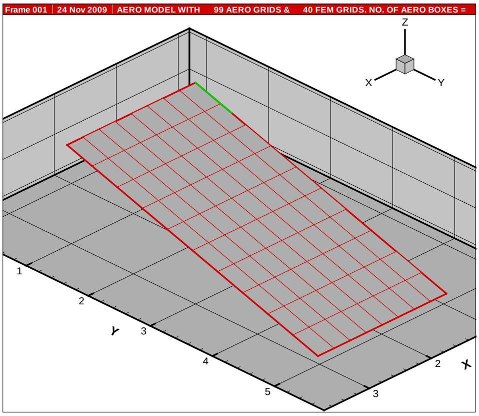
Aerodynamic Grid Points (X,Y,Z)

Aerodynamic Connectivity Information (aero boxes)
e.g., the first line connects the 2nd, 7th, 6th, and 1st aero grid points listed above

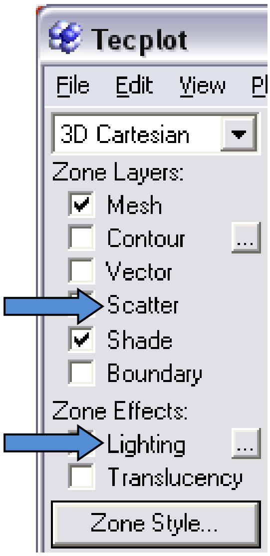
TITLE	Lists the number of aerodynamic grids, structural finite element grids and aerodynamic boxes.
VARIABLE	Defines the variable names associated with the column data.
X	X-coordinate of the aerodynamic grid point.
Y	Y-coordinate of the aerodynamic grid point.
Z	Z-coordinate of the aerodynamic grid point.
EXTID	External grid point identification number.
ZONE	Specifies information for the current zone (the Tecplot input can be broken up into multiple zones; only one ZONE is used to define the aerodynamic model).
I	Number of aerodynamic grid points listed in the plot file.
J	Number of aerodynamic boxes listed in the plot file.
F=FEPOINT	Finite-element zone specification.

- *<u>Steps within TECPLOT to View the Aerodynamic Model & Structural Grid Points</u>*
  (ZEUS output file generated by the PLTAERO bulk data card with FEMGRID='YES')

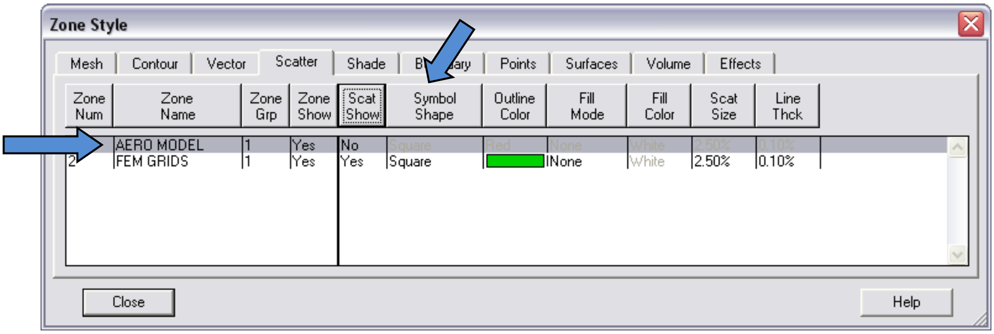
1) Open the ZEUS output Tecplot file of the aerodynamic model.



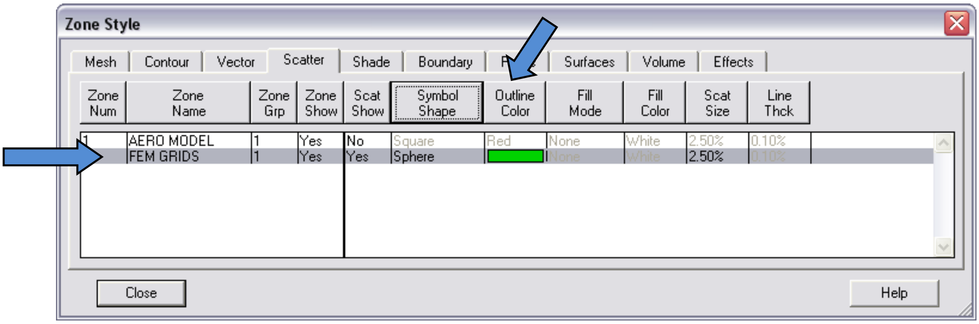
2) Check the "Scatter" box and click on the "Zone Style" button.



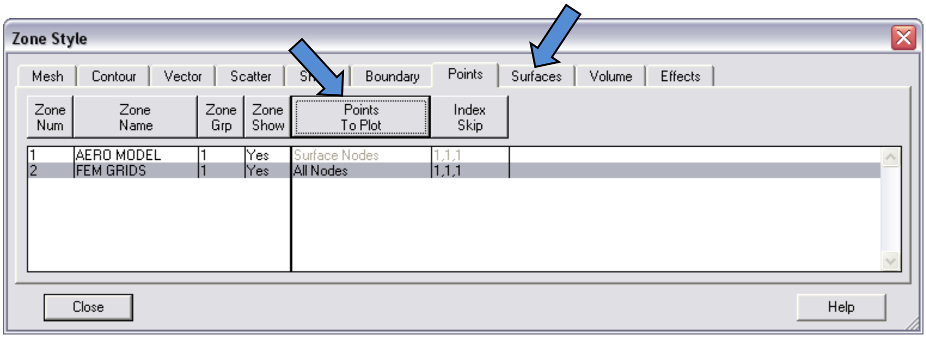
3) In the "Zone Style" window, click on the "Scatter" tab and select "AERO MODEL" zone.
   Click on the "Scat Show" button and select "NO".



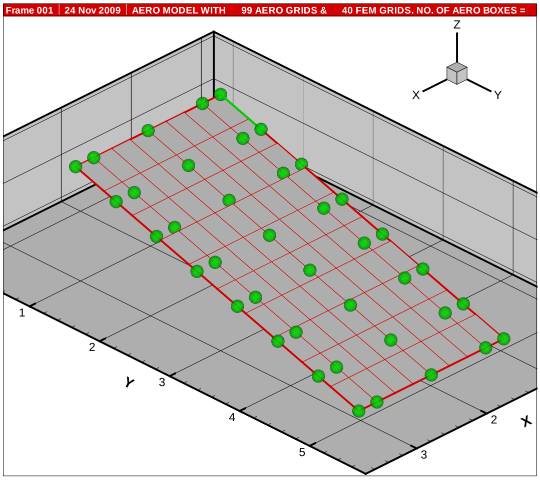
4) Select "FEM GRIDS" zone.
You can change the symbol, color, and symbol size by clicking on the "Symbol Shape", "Outline Color", and "Scat Size" buttons, respectively.



5) Select "Points" tab. Click on the "Points To Plot" button and select "All Nodes".



6) The aerodynamic model is plotted along with the structural grid points.



8.2 Steady/Unsteady Pressure Coefficients (PLTCP)

An output data file of the steady/unsteady pressure coefficients for all aerodynamic boxes in the model can be generated with the **PLTCP** bulk data card (see Figure 8.2).

For steady pressure plotting, the magnitude of pressure coefficient C_p is generated. For unsteady pressure, the real, imaginary, magnitude, and phase angle components of the unsteady pressure coefficients for a specified Mach number, reduced frequency and symmetry condition are generated. For detailed descriptions of these options, please see the **PLTCP** bulk data card description presented in Chapter 4

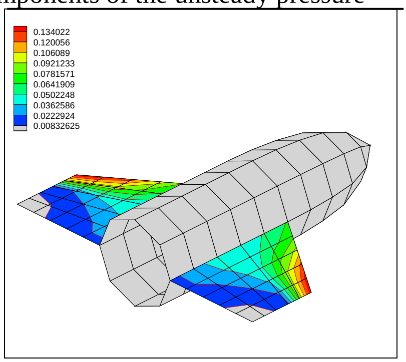


Figure 8.2 Plot of Unsteady Pressure Coefficient (M=0.8, k=0.2, 1st Mode, Re(C_P))

The symmetry condition of the unsteady aerodynamics is established by the **ASSIGN** Executive Control Command of the Executive Control Section. The **MKAEROZ** bulk data card generates unsteady aerodynamic results based on this symmetry condition. Therefore, the SYM entry specified in the **PLTCP** bulk data card <u>must match</u> a symmetry condition specified in the BOUNDARY entry of an **ASSIGN** Executive Control Command (i.e., either SYM for symmetric, ANTI for anti-symmetric, or ASYM for asymmetric). Since two **ASSIGN** statements can be specified in the Executive Control Section (one for symmetric case and one for an anti-symmetric case), all **MKAEROZ** bulk data cards generate both symmetric and anti-symmetric unsteady aerodynamics. In this case, two **PLTCP** bulk data cards could be used to generate two separate files containing the symmetric (with SYM entry set to 'SYM') and anti-symmetric (with SYM entry set to 'ANTI') unsteady aerodynamic pressures of the structural modes.

For the asymmetric case (i.e., full aircraft modeled in the input deck) only one **ASSIGN** Executive Control Command is allowed in the Executive Control Section. In this case, all **PLTCP** bulk data cards must have SYM = ‘ASYM’.

The **PLTCP** output plot file contains the steady/unsteady aerodynamic pressures on each aerodynamic box (for the PATRAN, I-DEAS, FEMAP, and NASTRAN output cases) or at each aerodynamic grid point (for the TECPLOT output case).

- ***PATRAN Compatible Output***

The PATRAN compatible output to display the unsteady pressure results is saved in two separate files. The aerodynamic model is saved in the neutral file format while the unsteady pressure results are saved in a results file. Both files will need to be imported into PATRAN to display the results. A sample of the PATRAN compatible output files are shown in the following figures and are described below:

Neutral File of the Aerodynamic Model

```
25       0       0       1       0       0       0       0       0
ZEUS AERODYNAMIC MODEL - PATRAN NEUTRAL FILE OUTPUT
26       0       0       1      91      65       1       1       0
  08/22/200016:11:23         2.5
 1     201       0       2       0       0       0       0       0
-0.100000000E+03 0.000000000E+00 0.000000000E+00
0G       6       0       0000000
 1     202       0       2       0       0       0       0       0
-0.100000000E+03 0.000000000E+00 0.000000000E+00
0G       6       0       0000000
                              .
                              .
                              .
2     201       4       2       0       0       0       0       0
       4       0       1       0 0.000000000E+00 0.000000000E+00 0.000000000E+00
     202     207     206     201
 2     202       4       2       0       0       0       0       0
       4       0       1       0 0.000000000E+00 0.000000000E+00 0.000000000E+00
     203     208     207     202
                              .
                              .
                              .
```

Aerodynamic Grid Points

Aerodynamic Boxes

Data Packets 1 (Node Data) and 2 (Element Data) are used to output the aerodynamic grid points and aerodynamic boxes, respectively.

Results File of the Unsteady Pressure (Element Results)

```
ZEUS UNSTEADY CP: PLTCP ID=      10, MODE=       2, M= 0.8, K= 0.2
    4
RESULT QUANTITIES: CP REAL, CP IMAG, CP MAGNITUDE, CP PHASE (DEG)
SUBTITLE 2
     201       4
-.5222457E-03-.1103247E-020.1220613E-02-.1153315E+03
     202       4
-.2227817E-03-.4547957E-030.5064295E-03-.1160979E+03
```

C_P Real, C_P Imaginary, C_P Magnitude, C_P Phase Angle

Unsteady Pressure Results

The Element Results File is used with four data quantities specified (C_P Real, C_P Imaginary, C_P Magnitude, and C_P Phase Angle (degrees)).

\- <u>*Steps within PATRAN to View the Aerodynamic Model with Unsteady Pressure*</u>

*(ZEUS output file generated by **PLTCP** bulk data card)*

- Start PATRAN and from *File/New* open a new PATRAN database.
- Read in the geometry file first. Select *File/Import* from the menu, then select, *Object:Model* and *Source:Neutral*, and finally select the appropriate geometry neutral file and click on the "Apply" button.
- Read in the pressure results file next. Select *File/Import* from the menu, then select, *Object:Results* and *Source:Patran2.els*. After selecting the *Patran2.els*, locate the ZEUS pressure results file in whatever directory it has been stored in (installed in the [miscel] folder within the ZEUS home directory). Select the appropriate pressure results file and click on the "Apply" button.
- Verify that the import/read was successful in the Dialog Box. Select *Results* from the toolbar buttons, *Action:Create*, *Object:Quickplot*.
- The Fringe results list should show up with the results that were read in. Select the desired pressure quantity and click on the "Apply" button.

The following PATRAN results template can be used to load the unsteady pressures from ZEUS. This results template (called 'ZEUS_pres.res_tmpl') is installed to the [miscel] folder within the ZEUS home directory.

```
/* ZEUS_pres.res_tmpl */
/* PATRAN 2.5 results file template for ZEUS *.els files */

KEYLOC = 0

TYPE = scalar
COLUMN = 1
PRI = Pressure Coefficient
SEC = Real

TYPE = scalar
COLUMN = 2
PRI = Pressure Coefficient
SEC = Imag.

TYPE = scalar
COLUMN = 3
PRI = Pressure Coefficient
SEC = Magnitude

TYPE = scalar
COLUMN = 4
PRI = Pressure Coefficient

SEC = Phase (deg)

TYPE = END
```

- ***Tecplot Compatible Output***

A sample of the Tecplot output is shown in the following figure and is described below:

```
TITLE ="UNSTEADY CP: PLTCP=    3000 MODE=       1 MACH=   .8000 K=   .2000"
VARIABLE = X, Y, Z, RE(CP), IM(CP) EXTID
```

```
ZONE I=      91 J=      65 F=FEPOINT
  -.1000E+03   .0000E+00   .0000E+00  -.7643E-04  -.2361E-03     201
  -.1000E+03   .0000E+00   .0000E+00  -.5331E-04  -.1644E-03     202
  -.1000E+03   .0000E+00   .0000E+00   .2541E-05   .3619E-05     203
  -.1000E+03   .0000E+00   .0000E+00   .5529E-04   .1649E-03     204
  -.1000E+03   .0000E+00   .0000E+00   .7531E-04   .2300E-03     205
                     .
                     .
                     .
       2       7       6       1
       3       8       7       2
                     .
                     .
                     .
```

Grid Point Identification Numbers

Aerodynamic Grid Points (X,Y,Z)

Unsteady Pressures (Re(CP), Im(CP))

Aerodynamic Connectivity Information (aero boxes) e.g. the first line connects the 2nd, 7th, 6th, and 1st aero grid points listed above

TITLE	Lists the **PLTCP** bulk data card identification number, mode, Mach number, and reduced frequency of the unsteady pressure results.
VARIABLE	Defines the variable names associated with the column data.
X	X-coordinate of the aerodynamic grid point.
Y	Y-coordinate of the aerodynamic grid point.
Z	Z-coordinate of the aerodynamic grid point.
RE(CP)	Real component of the unsteady pressure result.
IM(CP)	Imaginary component of the unsteady pressure result.
EXTID	External grid point identification number.
ZONE	Specifies information of the current zone (the Tecplot input can be broken up into multiple zones; only one zone is used to define the unsteady pressure output).
I	Number of aerodynamic grid points listed in the plot file.
J	Number of aerodynamic boxes listed in the plot file.
F=FEPOINT	Finite-element zone specification.

- ***I-DEAS Compatible Output***

The I-DEAS compatible output is saved in the universal file format. Data sets 781 and 780 are used to output the aerodynamic grid points and boxes, respectively. Data set 56 is used to output the unsteady pressure and is output four times for displaying the real, imaginary, magnitude, and phase angle of the unsteady pressure. The first five ID lines of each data set 56 list the following information:

Line 1: Mode number, Mach number, and reduced frequency, unsteady pressure component of the current data set (i.e., CP(RE), CP(IM), MAGNITUDE, or PHASE ANGLE).
Line 2: **PLTCP** Bulk Data Card identification number.
Line 3: The number of aerodynamic grid points in the model.
Line 4: The number of aerodynamic boxes in the model.
Line 5: The unsteady pressure component of the current data set (i.e., real, imaginary, magnitude, or phase angle) – repeated from Line 1 but more descriptive.

A sample of the I-DEAS compatible output is shown in the following figure:

```
    -1
   781
```

```
       203         0         0        11
  -1.0000000000000000D+02   0.0000000000000000D+00   0.0000000000000000D+00
       204         0         0        11
  -1.0000000000000000D+02   0.0000000000000000D+00   0.0000000000000000D+00
                                                                .
    -1
   780
       202        94         1    100000         1         1         1         4
       203       208       207       202
       203        94         1    100000         1         1         1         4
       204       209       208       203
       204        94         1    100000         1         1         1         4
                                                                .
    -1
    56
ZEUS UNSTEADY PRESSURE - MODE=    1 MACH NO.=  1.2000 K=  0.2000 CP(REAL)
UNSTEADY PRESSURE FOR PLTCP BULK DATA CARD WITH ID:       25
NUMBER OF AERODYNAMIC GRID POINTS IN MODEL =          91
NUMBER OF AERODYNAMIC BOXES IN MODEL =          65
REAL COMPONENT OF PRESSURE - CP(RE)
        1         1         1        18         2         1
        1         1         1
       202        1
  0.00000E+00
       203        1
  0.00000E+00
          .
          .
```

Aerodynamic Grid Points

Quadrilateral Elements

Unsteady Pressures

- ***FEMAP Compatible Output***

The FEMAP compatible output is saved in the FEMAP (Version 7.0) neutral file format. Data Blocks 403 and 404 are used to output the aerodynamic grids and boxes, respectively. Data Block 450 is used to output the unsteady pressure output set definition. Data Block 451 is used to output four data vectors to display the Real (CPRE), Imaginary (CPIM), Magnitude (MAGN) and Phase Angle (PHAS – in degrees). A sample of the FEMAP compatible output is shown in the following figure.

```
-1
   100
<NULL>
       7.
   -1
-1
                                                                .

403
       201,           0,           0,           1,           46,     0,     0,     0,     0,     0,     0,   -
1.0000000000000000D+02,   0.0000000000000000D+00,   0.0000000000000000D+00,        0,
       202,           0,           0,           1,           46,     0,     0,     0,     0,     0,     0,   -
1.0000000000000000D+02,   0.0000000000000000D+00,   0.0000000000000000D+00,        0,
                                                                .

-1
   404
       201,        124,           1,          17,           4,           1,           0,           0,           0,
0,         0,         0,
       202,        207,         206,         201,           0,           0,           0,           0,           0,
0,
         0,           0,           0,           0,           0,           0,           0,           0,           0,
0,
   0.0000000000000000D+00,   0.0000000000000000D+00,   0.0000000000000000D+00,
   0.0000000000000000D+00,   0.0000000000000000D+00,   0.0000000000000000D+00,
    0,     0,     0,     0,     0,     0,     0,     0,     0,     0,     0,     0,          0,          0,          0,
0,
                                                                .

-1
   450
   1,
ZEUS Unsteady Pressure
   0,   0
```

Neutral File Header followed by other required Data Blocks

Aerodynamic Grid Points

Quadrilateral Elements

Unsteady Pressure Output Set Definition

```
   2.0000000298023224D-01,
   1,
 <NULL>
   -1
   -1
   451
   1,   1,   1,
CPRE Mode   1M= 0.8k= 0.2
   0.,   0.,   0.,
  0,  0,  1,  0,  0,  0,  0,  0,  0,  0,
  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
  0,  0,  3,  8,
  0,  1,  1,
       201,  -1.7237762222066522D-04,   <--  Unsteady Pressure
       202,  -7.4403775215614587D-05,   .    Output Data Vector
                                        .    Definition
```

- *Steps within FEMAP to View the Aerodynamic Model with Unsteady Pressure*
  *(ZEUS output file generated by **PLTCP** bulk data card)*

  - Open (via File/Import/FEMAP Neutral) the ZEUS output neutral file of the aerodynamic model with unsteady pressure (select *View/Redraw* if the image does not appear after loading).
  - Aerodynamic grids (nodes) are displayed as green x's.
  - Aerodynamic boxes (elements) are displayed as white quadrilaterals.
  - Rotate, pan or autocenter the model with the Dynamic Rotate function (top left button on the toolbar).
  - Node, Point and Element features (such as id's) can be set in the *View/Options* window.
  - Open the *View/Select* Window.
  - From the Contour Style section, click on the "Contour" button.
  - Click on the "Deformed and Contour Data" button bar.
  - In the window that opens, under *Output Vectors/Contour*, select either CPRE, CPIM, MAGN, or PHAS to be displayed.
  - Click on the "OK" button in both windows.

- ***NASTRAN Compatible Output***

The NASTRAN compatible output is saved in a standard NASTRAN bulk data format. A sample is shown in the following figure and is described below:

```
BEGIN BULK
$ UNSTEADY CP: PLTCP=    4000 MODE=       2 MACH=  1.2000 K=   .1750
$ CP IS ON PLOAD4 BULK CARDS.  WITH SID = 1 FOR REAL CP,  SID = 2 FOR IMAG CP.
$ THERE ARE       91 GRIDS &       65 CQUAD4 TO REPRESENT THE AERO MODEL.
$ WHERE THE PSHELL ENTRIES OF CQUAD4 ARE CAERO7 AND BODY7 IDENTIFICATION NUMBERS
$ THERE ARE     65 PLOAD4 SID=1 FOR REAL CP AND     65 PLOAD4 SID=2 FOR IMAG CP
GRID         201        -1.00+020.000+000.000+00 ]
GRID         202        -1.00+020.000+000.000+00 ]  <-- Aerodynamic
                               .                         Grid Points
                               .
                               .

CQUAD4       201     201     202     207     206     201 ]
CQUAD4       202     201     203     208     207     202 ]  <-- Quadrilateral
                               .                                Elements
                               .
                               .

PLOAD4         1     2010.000+00 ]  <-- Unsteady
PLOAD4         1     2020.000+00 ]      Pressures
                               .
```

```
                                .
                                .
ENDDATA
```

The comment title cards (**$**) list the identification number of the current **PLTCP** bulk data card, mode, Mach number, and reduced frequency of the unsteady pressure results. Standard **PLOAD4** bulk data cards are used for the unsteady pressure output. The **PLOAD4** bulk data card SID entry is used to delineate the real from imaginary components of the unsteady pressure. The number of aerodynamic grid points (**GRID**), quadrilateral (**CQUAD4**) elements and **PLOAD4** bulk data cards within the plot file are also provided.

- ***PEGASUS Compatible Output***

The PEGASUS compatible output is saved in ESA data format. A sample of the PEGASUS compatible output is shown in the following figure and is described below:

```
*ESA
*COM
*COM   UNSTEADY AERODYNAMIC CP BY PLTCP BULK DATA CARD WITH ID=      10
*COM   MACH=   0.700,K= 0.00000E+00, MODE=       1
*COM
*FILE_SCALAR TEXT DATE
08/28/2000
*EOD
RUN1
*RTITLE
CP ON BODY7 ID=     201 NUMBER OF BOXES=      40
*EOD
*FLOAT
           X           Y           Z      RE(CP)      IM(CP)   MAGNITUDE  PHASE(DEG)
 -8.6667E+01  4.0069E+00 -9.6736E+00  5.8109E-04  0.0000E+00  5.8109E-04  0.0000E+00
 -8.6667E+01  9.6736E+00 -4.0069E+00  2.4097E-04  0.0000E+00  2.4097E-04  0.0000E+00
                                                   .
                                                   .
                                                   .
*EOD
RUN2
*RTITLE
CP ON CAERO7 ID=     101 STRIP NO.       1 NUMBER OF CHORDWISE BOXES=       5
*EOD
*FLOAT
          X           Y           Z         X/C      RE(CP)      IM(CP)   MAGNITUDE  PHASE(DEG)
 2.1150E+01  3.7000E+01  0.0000E+00  1.7000E-01  5.4455E-02  0.0000E+00  5.4455E-02  0.0000E+00
 4.0150E+01  3.7000E+01  0.0000E+00  3.7000E-01  2.1080E-02  0.0000E+00  2.1080E-02  0.0000E+00
                                                   .
                                                   .
                                                   .
```

Comment Cards

Run Title

Arrays containing pressure results at aerodynamic box control points (located at X,Y,Z)

The comment title cards (*COM) list the identification number of the current **PLTCP** bulk data card, Mach number, reduced frequency, and mode of the unsteady pressure results. The **BODY7** results are output for all aerodynamic boxes on all body macroelements. The **CAERO7** results are output for each chordwise strip of each wing macroelement starting at the wing root and proceeding towards the wing tip. The *X*, *Y*, and *Z* data correspond to the aerodynamic box control point locations at which the pressure is specified. *X/C* is the x-coordinate position as measured from the strip leading edge divided by the local strip chord length. C_p Magnitude and C_p Phase Angle are found from *SQRT(C_P(Re)**2 + C_P(Im)**2)* and *(ATAN(C_P(Im) / C_P(Re))*, respectively.

8.3 Flutter Mode Shape (PLTFLUT)

An output data file of the flutter mode shapes can be generated with the **PLTFLUT** bulk data card (see Figure 8.3). Each **PLTFLUT** bulk data card refers to a **FLUTTER** bulk data card that specifies the flutter mode to be computed. The flutter mode is represented by the deformed aerodynamic model. Since the flutter mode does not have a nodal line about which the configuration oscillates (as is the case with structural modal results), it is necessary to generate many deformed aerodynamic models for animating the flutter mode. (Note: multiple deformed models are output for the TECPLOT, IDEAS and NASTRAN formats only. Animations in PATRAN, FEMAP, and ANSYS are supported directly by this software through the output of a displacement set(s)). The magnitude of the deformation of each aerodynamic model is associated with sin $(\omega_f * t_I + \phi)$, where ω_f is the flutter frequency, t_I is the time of the i^{th} deformed aerodynamic model and ϕ is the phase angle. t_I is computed by the following equation:

$$t_i = (i * 2\pi / \omega_f) \qquad \text{where } i = 0, 1, \ldots, \text{NTIME}$$

The maximum displacement of the flutter mode shape is controlled by the MAXDISP entry of the **PLTFLUT** bulk data card. MAXDISP is a fraction (i.e., 0.0 – 1.0) of the reference chord length (REFC entry) specified in the **AEROZ** bulk data card.

The **PLTFLUT** output plot file contains the deformed aerodynamic model for a specified flutter mode. This file consists of multiple sections, each containing the complete aerodynamic model configuration within one time step of a flutter mode oscillation cycle. This is done to allow for animation of the flutter mode shape which can be accomplished by successively viewing the deformed shapes through one complete cycle. All aerodynamic box grids and connectivity information of the aerodynamic model for each section of the output (i.e., each deformed aerodynamic model) are generated. Both the right-hand-side and left-hand-side of the model are output if the SYM entry of the **FLUTTER** bulk data card is set to either SYM (symmetric) or ANTI (anti-symmetric). The data for this output file is generated in the flutter (**FLUTTER**) module

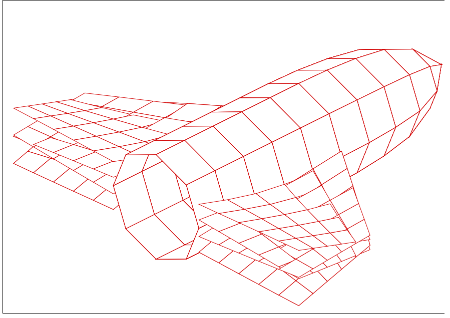


Figure 8.3 Plot of the Flutter Mode Shape at Four Time Steps (NTIME=4)

- ***PATRAN Compatible Output***

The PATRAN compatible output to display the flutter mode shape is saved in multiple files. The aerodynamic model is saved in the neutral file format while the flutter mode shape nodal displacements are saved in the results files. One results file is generated for each time step output and are numbered sequentially from 1 to NTIME+1 by adding a number suffix to your FILENM entry of your **PLTFLUT** bulk data card. For example, if NTIME = 2 and the user specifies FILENM = ‘FLUT.OUT’, then the results files generated would be FLUT.OUT.1, FLUT.OUT.2, and FLUT.OUT.3. In addition to the neutral file and results files output, a session file is also generated to help you expedite the process in loading the results files into PATRAN. The session file name is the FILENM entry with an added extension of (.SES). So using FILENM = 'FLUT.OUT' would produce a session file called FLUT.OUT.SES. Loading the session file into PATRAN is a simple two step process that will automatically load the neutral file and results files and launch the flutter animation for you. However, the neutral file and results files can also be loaded manually. Both methods are described below.

Neutral File of the Aerodynamic Model

```
25       0       0       1       0       0       0       0       0
ZEUS AERODYNAMIC MODEL - PATRAN NEUTRAL FILE OUTPUT
26       0       0       1     520     130       1       1       0
  08/23/200009:10:19         2.5
 1       1       0       2       0       0       0       0       0
-0.100000000E+03 0.000000000E+00 0.000000000E+00
0G       6       0       0000000
 1       2       0       2       0       0       0       0       0
-0.800000000E+02 0.120208158E+02-0.120208149E+02
0G       6       0       0000000
                              .
                              .
2      1       4       2       0       0       0       0       0
       4       0       1       0 0.000000000E+00 0.000000000E+00 0.000000000E+00
       1       2       3       4
 2       2       4       2       0       0       0       0       0
       4       0       1       0 0.000000000E+00 0.000000000E+00 0.000000000E+00
       5       6       7       8
                              .
```

Aerodynamic Grid Points

Aerodynamic Boxes

Results File(s) of the Flutter Mode Shape (Displacement Results)

Multiple files output for each time step listed in the title as T=…

```
ZEUS FLUTTER MODE AT TIME T= 0.00000, FOR PLTFLUT ID=      30
    13648    13648   0.339401E+02    13220        3
FLT FREQ=   5.415 HZ, FLT SPEED= 919.030, FLT MODE=       1, DYN PRES= 470.448
FOR THREE DEGREES OF FREEDOM DX DY DZ
       10.1720123E+000.5134657E+01-.4656522E+00
       20.1965479E+000.5020568E+01-.3835102E+00
```

Aero Grid Id's DX, DY, DZ
Displacements at aero grids

- *<u>Steps within PATRAN to View the Deformed or Animated Flutter Mode Shape</u>*
  (ZEUS output files generated by ***PLTFLUT*** *bulk data card)*

Loading Results Using the Session File

- Start PATRAN.
- *Select File/Session/Play* from the menu.
- Select the session file generated from your run which will be located in your runtime directory and names FILENM.SES (where FILENM is the name you input on the FILENM entry of your PLTFLUT bulk data card).

Loading Results Manually

- Start PATRAN
- Use the *File/New* menu item to open a new PATRAN database

- Read in the geometry neutral file first. Select *File/Import* from the menu, then select *Object:Model* and *Source:Neutral*. Now select the geometry neutral file generated in your runtime directory and click *Apply*.
- Click the *Yes* button on any message boxes that may appear.
- Read in the displacement results files next. Select *File/Import* from the menu, then select *Object:Results* and *Source:Patran2.dis*. A dialog will appear to select a results template. The results template is generated by ZEUS in your runtime directory called 'ZEUS_dis.res_tmpl'. You can also obtain this file from the ZEUS home directory/miscel folder. After selecting the results template, you need to load the flutter mode shape displacement results files from your runtime directory. These files need to be loaded one at a time which can become cumbersome if you have many time steps output.
- Verify that the import/read was successful in the Dialog Box. Select "Results" from the toolbar buttons, *Action:Create*, *Object:Deformation*.
- In the *Select Results* list control, select all of the loaded results file cases (note that you can use the Shift key to select more than 1 results file). Check the Animate checkbox (located on the bottom left of the dialog). Click the *Animation Graphics 3D* radio button. Change the number *Number of Frames* to the number of results files you selected (should be NTIME+1). Click on the "Apply" button and user should see the flutter mode animation.

The following PATRAN results template can be used to load the displacements from ZEUS. This file called 'ZEUS_dis.res_tmpl' is also installed to the ZEUS home directory/miscel folder.

```
/* ZEUS_dis.res_tmpl */
/* PATRAN 2.5 results file template for ZEUS .dis files */

KEYLOC = 0

TYPE = vector
COLUMN = 1, 2, 3
PRI = Displacements
SEC = Translational
CTYPE = nodal

TYPE = END
```

- ***Tecplot Compatible Output***

Tecplot provides an animation feature that allows for animation of multiple zones. This is useful in viewing the flutter mode shapes. An animated movie file can also be generated to allow for continuous looping of the flutter mode shape. A sample of the Tecplot compatible output is shown in the following figure and is described below.

```
TITLE ="FLUTTER MODE  1, PLTFLUT ID=      10 WF=7.965+00HZ, VF=1.295+04 QINF=9.596+00"
VARIABLE = X, Y, Z, EXTID
ZONE T="  .00000 SEC." I=     260 J=      65 F=FEPOINT
 -8.0000E+01  0.0000E+00 -1.7000E+01     206
 -1.0000E+02  0.0000E+00  0.0000E+00     201
 -1.0000E+02  0.0000E+00  0.0000E+00     203
                    .
                    .

      9       10       11       12
     13       14       15       16
```

Grid Point
Identification Numbers

Aerodynamic
Grid Points (X,Y,Z)

Aerodynamic Connectivity
Information (aero boxes)
e.g., the first line connects the 1st, 2nd, 3rd, and 4th aero grid points listed above

```
      17      18      19      20
                   .
                   .
                   .

ZONE T="  .01569 SEC." I=     260 J=      65 F=FEPOINT
 -1.0000E+02  0.0000E+00  0.0000E+00     202
 -8.0000E+01  1.2021E+01 -1.2021E+01     207
 -8.0000E+01  0.0000E+00 -1.7000E+01     206
                   .
```

← 2nd bulk data input containing the deformed aerodynamic model at the second time step

TITLE Lists the requested flutter mode index, associated **PLTFLUT** bulk data card identification number, flutter frequency (in Hz), flutter speed (in units specified in the referenced flutter bulk data cards), and dynamic pressure (in units specified in the referenced flutter bulk data cards; **FIXMATM**, **FIXMACH**, **FIXMDEN**).

VARIABLE Defines the variable names associated with the column data.

X x-coordinate of the aerodynamic grid point.

Y y-coordinate of the aerodynamic grid point.

Z z-coordinate of the aerodynamic grid point.

EXTID External grid point identification number.

ZONE Specifies information of the current zone (NTIME zones are used to animate the flutter mode shape. The NTIME entry is specified on the **PLTFLUT** bulk data card.)

T Title of the current zone; contains the real time (in seconds) of the current deformed plot based on the flutter frequency and is computed from:

$$i \times \frac{1}{\omega_f \times NTIME} \quad \text{for} \quad i = 1, 2, \ldots, NTIME,$$

where ω_f = flutter frequency (in Hz)

I Number of aerodynamic grid points listed in the plot file.

J Number of aerodynamic boxes listed in the plot file.

F=FEPOINT Finite-element zone specification.

- ***I-DEAS Compatible Output***

The I-DEAS compatible output is saved in the universal file format. Data sets 781 and 780 are used to output the aerodynamic grid points and boxes, respectively. Data set 55 is used to output the aerodynamic grid point displacements at each time step. The first five ID lines of each data set 55 list the following information:

Line 1: Flutter mode shape number, flutter frequency in Hertz (WF), flutter speed (VF), and dynamic pressure (QINF).

Line 2: **PLTFLUT** Bulk Data Card identification number.

Line 3: The number of time steps and scale factor specified in the **PLTFLUT** Bulk Data Card.

Line 4: The symmetry condition specified in the **FIXMATM**, **FIXMACH**, or **FIXMDEN** Bulk Data Card(s).

Line 5: The current time step in seconds.

A sample of the I-DEAS compatible output is shown in the following figure:

```
    -1
   781
         2          0          0         11
```

```
   5.1763749122619629D-01   0.0000000000000000D+00   0.0000000000000000D+00
         3         0         0        11
   7.6437747478485107D-01   9.2084997892379761D-01   0.0000000000000000D+00
         4         0         0        11
   2.4673999845981598D-01   9.2084997892379761D-01   0.0000000000000000D+00
                                                    .
                                                    .
    -1
   780
         2        94         1    100000         1       101         1         4
         5         6         7         8
         3        94         1    100000         1       101         1         4
         9        10        11        12
                                                    .
                                                    .
    -1
    55
ZEUS FLUTTER MODE SHAPE, MODE=  1, WF=  680.69 HZ, VF=   6333., QINF=    2.15
FLUTTER MODE SHAPE FOR PLTFLUT BULK DATA CARD WITH ID:       10
NO. OF TIME STEPS=       8 SCALE FACTOR=    0.39
MODEL IS SYMMETRIC ABOUT X-Z PLANE – LEFT HAND SIDE OF MODEL OUTPUT
CURRENT TIME STEP=         0.00000 SEC.
         1         2         3         8         2         6
         1         1         1
  0.00000E+00
         1
  0.00000E+00  0.00000E+00 -0.89259E-02  0.00000E+00  0.00000E+00  0.00000E+00
         2
  0.00000E+00  0.00000E+00  0.27763E-02  0.00000E+00  0.00000E+00  0.00000E+00
         3
  0.00000E+00  0.00000E+00 -0.13783E-01  0.00000E+00  0.00000E+00  0.00000E+00
         4
  0.00000E+00  0.00000E+00 -0.23197E-01  0.00000E+00  0.00000E+00  0.00000E+00
                                                    .
                                                    .
```

Aerodynamic Grid Points

Quadrilateral Elements

Displacements at Aerodynamic Grids

- ***FEMAP Compatible Output***

The FEMAP compatible output is saved in the FEMAP (Version 7.0) neutral file format. Data Blocks 403 and 404 are used to output the aerodynamic grids and boxes, respectively. Data Block 450 is used to output the flutter mode output set definition. Data Block 451 is used to output four data vectors to display the flutter mode shape (TOTAL Translation), X-axis translation (T1), Y-axis translation (T2) and Z-axis translation (T3). Data blocks 450 and 451 are repeated within the neutral file for each time step output. Therefore there will be NTIME+1 blocks 450 and 451 where NTIME is the number of time steps you input on the **PLTFLUT** bulk data card. A sample of the FEMAP compatible output is shown in the following figure.

```
-1
   100
<NULL>
        7.
   -1
-1
                                                    .
                                                    .
403
       203,          0,          0,          1,         46,     0,     0,     0,     0,     0,     0,  -
1.0000000000000000D+02,   0.0000000000000000D+00,   0.0000000000000000D+00,         0,
       204,          0,          0,          1,         46,     0,     0,     0,     0,     0,     0,  -
```

Neutral File Header followed by other required Data Blocks

```
1.0000000000000000D+02,   0.0000000000000000D+00,
                                                    .
                                                    .          Aerodynamic
-1                                                             Grid Points
   404
       201,       124,         1,        17,         4,         1,         0,         0,         0,
0,         0,         0,
       202,       207,       206,       201,         0,         0,         0,         0,         0,
0,
         0,         0,         0,         0,         0,         0,         0,         0,         0,
0,
   0.0000000000000000D+00,   0.0000000000000000D+00,   0.0000000000000000D+00,
   0.0000000000000000D+00,   0.0000000000000000D+00,   0.0000000000000000D+00,
    0,    0,    0,    0,    0,    0,    0,    0,    0,    0,    0,    0,         0,         0,         0,
0,
                                                    .                   Quadrilateral
                                                                        Elements
   -1
   450                                    Flutter Mode
   1,                                     Output Set
Flut Mode    1 T= 0.00000                 Definition
   0,   3
   5.4152030944824219D+00,
   2,
ZEUS FLUTTER MODE AT TIME T= 0.00000, FOR PLTFLUT ID=      20
FLT FREQ=   5.415 HZ, FLT SPEED= 919.030, FLT MODE=       1
   -1                                                  Flutter Mode
   -1                                                  Output Data Vector
   451                                                 Definition
   1,   1,   1,
ZEUS Total Translation                      Flutter Mode
   0.,   0.,   0.,                          Output Set
  2,  3,  4,  0,  0,  0,  0,  0,  0,  0,    Definition
  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
  0,  0,  1,  7,
  1,  1,  1,
         1,   5.1585974693298340D+00,
         2,   5.0390295982360840D+00,
         3,   5.0613470077514648D+00,
         4,   5.1585974693298340D+00,

                                                    .
```

- *<u>Steps within FEMAP to View the Deformed or Animated Flutter Mode Shape</u>*
  *(ZEUS output file generated by **PLTFLUT** bulk data card)*

  - Open (via File/Import/FEMAP Neutral) the ZEUS output neutral file of the aerodynamic model with flutter mode deformation (select *View/Redraw* if the image does not appear after loading).
  - Aerodynamic grids (nodes) are displayed as green x's.
  - Aerodynamic boxes (elements) are displayed as white quadrilaterals.
  - Rotate, pan or autocenter the model with the Dynamic Rotate function (top left button on the toolbar).
  - Node, Point and Element features (such as id's) can be set in the *View/Options* (F6 key) window.
  - Open the *View/Select* Window (F5 key).
  - To <u>statically view</u> the flutter mode at a certain time step, from the Deformed Style section, click on the "Deform" radio button. Then click on the "Deformed and Contour Data" button. From the Output Set section, select the time step you want to view. Then click the "OK" buttons in both screens. You will then see the flutter mode deformation at the selected time step

- To <u>animate</u> the flutter mode through all time steps, from the Deformed Style section, click on the "Animate-MultiSet" radio button. Then click on the "Deformed and Contour Data" button.
- In the window that opens, under Output Set select the first output set (i.e., 1. Flut Mode 1, T= 0.00000). In the Output Vectors/Deformation, select either, TOTAL, T1, T2, or T3 to be displayed. TOTAL is the complete flutter mode shape. The flutter frequency is displayed in the Output Set box and will also be shown on the lower left-hand side of the screen during animation.
- Click on the "OK" buttons from both screens.
- If the animation does not start, select the *View/Advanced Post/Animation* menu item and click on the "Play" button. You can control the animation settings from this dialog.
- FEMAP, by default, animates or deforms the model based on a percentage of the model length. To view the actual displacement based on the ZEUS output (set by the MAXDISP entry of the **PLTFLUT** bulk data card), open the *View/Options* window, select the PostProcessing category, select *Deformed Style* in the Options menu and uncheck the % of Model (Actual) checkbox.

- ***NASTRAN Compatible Output***

The NASTRAN compatible output is saved in standard NASTRAN bulk data format. A sample is shown in the following figure and is described below:

```
BEGIN BULK
$ FLUTTER MODE  1, PLTFLUT ID=      10 WF=8.815+00HZ, VF=1.321+03 QINF=1.859+03
$ AT T=  .00000SEC
GRID           1        -1.00+020.000+000.000+00    Aerodynamic
GRID           2        -8.00+011.202+01-1.20+01    Grid Points
                              .

CQUAD4         1     201       1       2       3       4    Quadrilateral
CQUAD4         2     201       5       6       7       8    Elements
CQUAD4         3     201       9      10      11      12
                              .

ENDDATA

BEGIN BULK
$ FLUTTER MODE  1, PLTFLUT ID=      10 WF=8.815+00HZ, VF=1.321+03 QINF=1.859+03
$ AT T=  .01418SEC
GRID           1        -1.00+020.000+000.000+00
GRID           2        -8.00+011.202+01-1.20+01    2nd zone containing
                              .                     the deformed
                              .                     aerodynamic model
                              .                     at the second time step
ENDDATA
```

The comment title card (**$**) lists the index of the flutter mode shape, the identification of the current **PLTFLUT** bulk data card, the flutter frequency (in Hz), the flutter speed (in units specified in the referenced flutter bulk data cards; **FIXMATM**, **FIXMACH**, and **FIXMDEN**), and the dynamic pressure (in units specified in the referenced flutter bulk data cards). The real time (in seconds) of the current deformed plot based on the flutter frequency computed from is also provided. One complete Bulk Data Section is provided at each time step for one complete cycle.

$$i \times \frac{1}{\omega_f \times NTIME} \quad \text{for} \quad i = 1, 2, \ldots, NTIME, \qquad \text{where } \omega_f = \text{flutter frequency (in Hz)}$$

8.4 Interpolated Structural Mode Shape (PLTMODE)

An output data file of the interpolated structural mode shapes on the aerodynamic model can be generated with the **PLTMODE** bulk data card (see Figure 8.4). Viewing the interpolated structural modes is very useful in determining whether or not the aerodynamic model is properly splined to the structure. Experience has shown that most errors in aeroelastic analysis are a result of incorrect spline input. Therefore, viewing the interpolated structural mode shapes should always be performed for verification purposes whenever the spline input (i.e., **SPLINE1**, **SPLINE2**, **SPLINE3**, and **ATTACH** bulk data cards) is modified.

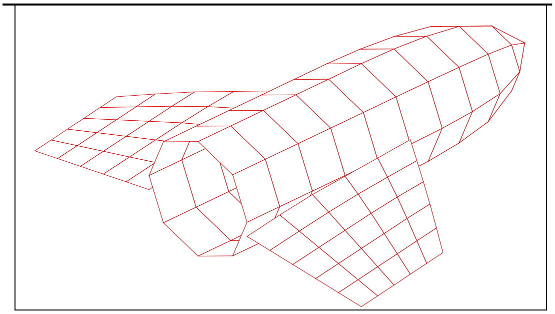


Figure 8.4 Plot of an Interpolated Mode Shape on the Aerodynamic Model (Mode 1)

The maximum displacement of the interpolated structural mode shape is controlled by the MAXDISP entry of the **PLTMODE** bulk data card. MAXDISP is a fraction (i.e., 0.0 – 1.0) of the reference chord length (REFC entry) specified in the **AEROZ** bulk data card.

The **PLTMODE** output plot file contains the deformed aerodynamic model for a specified structural mode. All aerodynamic box corner grid points and connectivity information of the aerodynamic model (i.e., a deformed aerodynamic model) is generated. The magnitude of the displacement is scaled by the SCALE entry of the **PLTMODE** bulk data card. The data for this output file is generated by the spline (**SPLINE**) module.

- ***PATRAN Compatible Output***

The PATRAN compatible output to display the interpolated mode shape is saved in two separate files. The aerodynamic model is saved in the neutral file format while the interpolated mode shape nodal displacements are saved in a results file. Both files will need to be imported into PATRAN to display the interpolated mode shape results. A sample of the PATRAN compatible output files are shown in the following figures and are described below:

Neutral File of the Aerodynamic Model

```
25       0       0       1       0       0       0       0       0
ZEUS AERODYNAMIC MODEL - PATRAN NEUTRAL FILE OUTPUT
26       0       0       1     260      65       1       1       0
  08/23/200009:09:28         2.5
 1       1       0       2       0       0       0       0       0
-0.100000000E+03 0.000000000E+00 0.000000000E+00
0G       6       0       0000000
 1       2       0       2       0       0       0       0       0
-0.800000000E+02 0.120208158E+02-0.120208149E+02
0G       6       0       0000000
                              .
                              .                                    Aerodynamic
                              .                                    Grid Points

2       1       4       2       0       0       0       0       0
       4       0       1       0 0.000000000E+00 0.000000000E+00 0.000000000E+00
       1       2       3       4
 2       2       4       2       0       0       0       0       0
       4       0       1       0 0.000000000E+00 0.000000000E+00 0.000000000E+00
       5       6       7       8
                              .
                              .                                    Aerodynamic
                              .                                    Boxes
```

Results File of the Interpolated Mode Shape (Displacement Results)

```
ZEUS MODE SHAPE: PLTMODE ID=      10, MODE=       1, FREQ=   4.613 HZ
      260      260   0.300000E+02      259        3
FOR THREE DEGREES OF FREEDOM DX DY DZ
SUBTITLE 2
       10.0000000E+000.0000000E+000.0000000E+00
       20.0000000E+000.0000000E+000.0000000E+00
       30.0000000E+000.0000000E+000.0000000E+00
       40.0000000E+000.0000000E+000.0000000E+00

 Aero         DX,          DY,          DZ
 Grid Id's       Displacements at aero grids
```

– <u>*Steps within PATRAN to View the Deformed or Animated interpolated Mode Shape*</u>
  (ZEUS output file generated by ***PLTMODE*** *bulk data card)*

  - Start PATRAN and from *File/New* open a new PATRAN database.
  - Read in the geometry file first. Select *File/Import* from the menu, then select *Object:Model* and *Source: Neutral* select the appropriate geometry neutral file and click on the "Apply" button.
  - Click on the "Yes" button on any message boxes that may appear.
  - Read in the displacement results files next. Select *File/Import* from the menu, then select *Object:Results* and *Source:Patran2.dis*. A dialog will appear to select a results template. You can the results template file from the ZEUS home directory/miscel folder. After selecting the results template, you need to load the flutter mode shape displacement results file from your runtime directory.
  - Verify that the import/read was successful in the Dialog Box. Select *Results* from the toolbar buttons, *Action: Create*, *Object: Deformation*.
  - The Fringe results list should show the results of the data that were read in.

The following PATRAN results template can be used to load the displacements from ZEUS.

```
/* ZEUS_dis.res_tmpl */
/* PATRAN 2.5 results file template for ZEUS .dis files */
```

```
KEYLOC = 0

TYPE = vector
COLUMN = 1, 2, 3
PRI = Displacements
SEC = Translational
CTYPE = nodal

TYPE = END
```

- ***Tecplot Compatible Output***

A sample of the Tecplot compatible output is shown in the following figure and is described below.

```
TITLE ="THE    1TH MODE ON AERO MODEL FROM FILE sample.fre"
VARIABLE = X, Y, Z, EXTID
ZONE I=     260 J=      65 F=FEPOINT
 -1.0000E+02  0.0000E+00  0.0000E+00     202
 -8.0000E+01  1.2021E+01 -1.2021E+01     207
 -8.0000E+01  0.0000E+00 -1.7000E+01     206
 -1.0000E+02  0.0000E+00  0.0000E+00     201
 -1.0000E+02  0.0000E+00  0.0000E+00     203
                    .
                    .
                    .
      1       2       3       4
      5       6       7       8
      9      10      11      12
     13      14      15      16
     17      18      19      20
```

Grid Point
Identification Numbers

Aerodynamic
Grid Points (X,Y,Z)

Aerodynamic Connectivity
Information (aero boxes)
e.g., the first line connects the 1st, 2nd, 3rd
and 4th aero grid points listed above

TITLE	Lists the requested structural mode index, and the name of the structural finite element output file from which the structural finite element modes are read in (i.e., file assigned by the **ASSIGN** Executive Command in the **ASSIGN** Executive Command Section).
VARIABLE	Defines the variable names associated with the column data.
X	X-coordinate of the aerodynamic grid point.
Y	Y-coordinate of the aerodynamic grid point.
Z	Z-coordinate of the aerodynamic grid point.
EXTID	External grid point identification number.
ZONE	Specifies information for the current zone.
I	Number of aerodynamic grid points listed in the plot file.
J	Number of aerodynamic boxes listed in the plot file.
F=FEPOINT	Finite-element zone specification.

- ***I-DEAS Compatible Output***

The I-DEAS compatible output is saved in the universal file format. Data sets 781 and 780 are used to output the aerodynamic grid points and boxes, respectively. Data set 55 is used to output the displacements at the aerodynamic grid points. The first five ID lines of data set 55 list the following information:

Line 1: Structural input data mode shape number
Line 2: Structural input data filename
Line 3: NONE
Line 4: NONE
Line 5: NONE

A sample of the I-DEAS compatible output is shown in the following figure:

```
    -1
   781
         1         0         0        11
  -1.0000000000000000D+02   0.0000000000000000D+00   0.0000000000000000D+00
         2         0         0        11
  -8.0000000000000000D+01   1.2020814895629883D+01  -1.2020814895629883D+01
                                                     .
                                                     .        Aerodynamic
    -1                                                        Grid Points
   780
         1        94         1    100000         1       201         1         4
         1         2         3         4
         2        94         1    100000         1       201         1         4
         5         6         7         8
                                                     .  Quadrilateral
                                                     .  Elements
    -1
    55
ZEUS MODE SHAPE OUTPUT, MODE=        1
FROM FILE: crop.f06
NONE
NONE
NONE
         1         2         3         8         2         6
         1         1         1
  0.00000E+00
         1
  0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00
         2
  0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00
                                                     .                 Displacement on
                                                     .                 Aerodynamic Grids
```

- ***FEMAP Compatible Output***

The FEMAP compatible output is saved in the FEMAP (ver 7.0) neutral file format. Data Blocks 403 and 404 are used to output the aerodynamic grids and boxes, respectively. Data Block 450 is used to output the flutter mode output set definition. Data Block 451 is used to output four data vectors to display the interpolated mode shape (TOTAL Translation), X-axis translation (T1), Y-axis translation (T2), and Z-axis translation (T3). A sample of the FEMAP compatible output is shown in the following figure.

```
-1
   100
<NULL>                        Neutral File Header followed by other
        7.                    required Data Blocks
   -1
-1
                                                     .
                                                     .

403
       201,          0,           0,           1,          46,     0,     0,     0,     0,     0,     0,   -
1.0000000000000000D+02,   0.0000000000000000D+00,   0.0000000000000000D+00,          0,
       202,          0,           0,           1,          46,     0,     0,     0,     0,     0,     0,   -
1.0000000000000000D+02,   0.0000000000000000D+00,   0.0000000000000000D+00,          0,
       203,          0,           0,           1,          46,     0,     0,     0,     0,     0,     0,   -
```

```
1.0000000000000000D+02,   0.0000000000000000D+00,   0.0000000000000000D+00,         0,
       204,           0,            0,            1,           46,     0,     0,     0,     0,     0,     0,   -
1.0000000000000000D+02,   0.0000000000000000D+00,
                                                    .
                                                    .
```

Aerodynamic Grid Points

```
-1
   404
       201,        124,          1,         17,          4,          1,          0,          0,          0,
0,         0,          0,
       202,        207,        206,        201,          0,          0,          0,          0,          0,
0,
         0,          0,          0,          0,          0,          0,          0,          0,          0,
0,
   0.0000000000000000D+00,   0.0000000000000000D+00,   0.0000000000000000D+00,
   0.0000000000000000D+00,   0.0000000000000000D+00,   0.0000000000000000D+00,
   0.0000000000000000D+00,   0.0000000000000000D+00,   0.0000000000000000D+00,
    0,    0,    0,    0,    0,    0,    0,    0,    0,    0,    0,    0,          0,          0,          0,
0,
                                                    .
                                                    .
-1
   450
   1,
Mode     1    4.613 Hz
   0,   2
   4.6127114295959473D+00,
   1,
 <NULL>
   -1
   -1
   451
   1,   1,   1,
ZEUS Total Translation
   0.,   0.,   0.,
  2,  3,  4,  0,  0,  0,  0,  0,  0,  0,
  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
  0,  0,  1,  7,
  1,  1,  1,
         1,   0.0000000000000000D+00,
         2,   0.0000000000000000D+00,
         3,   0.0000000000000000D+00,
         4,   0.0000000000000000D+00,
                                                    .
                                                    .
```

Interpolated Mode Output Set Definition

Interpolated Mode Output Data Vector Definition

- *Steps within FEMAP to View the Deformed or Animated Interpolated Mode Shape* *(ZEUS output file generated by **PLTMODE** bulk data card)*

  - Open (via *File/Import/FEMAP* Neutral) the ZEUS output neutral file of the aerodynamic model with interpolated mode deformation (select *View/Redraw* if the image does not appear after loading).
  - Aerodynamic grids (nodes) are displayed as green x's.
  - Aerodynamic boxes (elements) are displayed as white quadrilaterals.
  - Rotate, pan or autocenter the model with the Dynamic Rotate function (top left button on the toolbar).
  - Node, Point and Element features (such as id's) can be set in the *View/Options* window.

  - Open the *View/Select* Window.
  - To animate the flutter mode, from the Deformed Style section, click on the "Animate" button.
  - To statically view the flutter mode, from the Deformed Style section, click on the "Deform" button.
  - Click on the "Deformed and Contour Data" button bar.

- In the window that opens, under *Output Vectors/Deformation*, select either, TOTAL, T1, T2, or T3 to be displayed. TOTAL is the complete flutter mode shape. The modal natural frequency is displayed in the Output Set box and will also be shown on the lower left-hand side of the screen during animation.
- Click on the "OK" from both screens.
- FEMAP, by default, animates or deforms the model based on a percentage of the model length. To view the actual displacement based on the ZEUS output (set by the MAXDISP entry of the **PLTMODE** bulk data card), open the *View/Options* window, select the PostProcessing category, select *Deformed Style* in the Options menu and uncheck the % of Model (Actual) checkbox.
- The number of frames and display times of the animation sequence can be set by the View/Options/PostProcessing/Animation/Frames and Delay input options.

- ***NASTRAN Compatible Output***

The NASTRAN compatible output is saved in standard NASTRAN bulk data format. A sample is shown in the following figure and is described below.

```
BEGIN BULK
$ DEFORMED AERO MODEL OF THE    2TH MODE (REPRESENTED BY GRID & CQUAD4) FROM FILE:
$      sample.fre
GRID           1        -1.00+020.000+000.000+00    Aerodynamic
GRID           2        -8.00+011.202+01-1.20+01    Grid Points
                              .
                              .

CQUAD4         1     201       1       2       3       4    Quadrilateral
CQUAD4         2     201       5       6       7       8    Elements
                              .
                              .

ENDDATA
```

The comment title card (**$**) list the index of the structural mode shape and the name of the structural finite element output file from which the structural finite element modes are read in (i.e., file assigned by the **ASSIGN** Executive Command in the **ASSIGN** Executive Command Section).

8.5 Slopes of the Aerodynamic Surface (PLTSLP)

Two output files displaying the slopes of the surface boxes can be generated using the **PLTSLP** bulk data card.

- ***Tecplot Compatible Output***

A sample of the Tecplot compatible output of the XY line style plot is shown in the following table and figures and is described in the following figure.

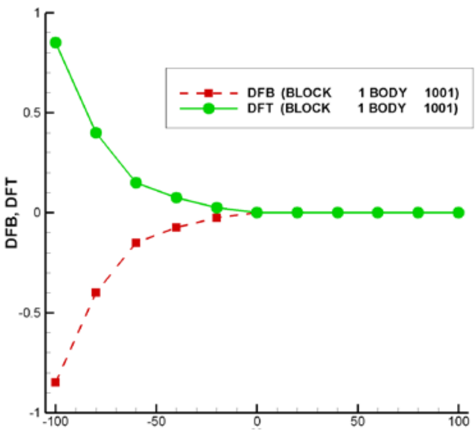
```
TITLE ="SLOPE INFO FOR        1 BLOCKS WITH A TOTAL OF         1 BODYS,         1 HORIZONTAL AND         0
```

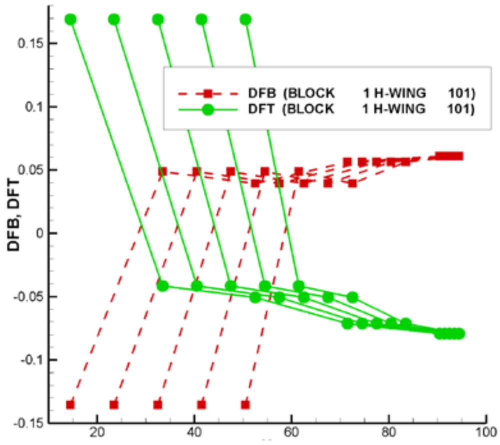
```
VERTICAL SURFACES"
VARIABLES= X, DFB, DFT, DFL, DFR, FB, FT, FL, FR,
ZONE T="BLOCK        1 BODY     1001" I=      11 J=1
 -1.0000E+02 -8.5000E-01  8.5000E-01 -7.4309E-08  8.5000E-01  0.0000E+00  0.0000E+00  0.0000E+00  0.0000E+00
 -8.0000E+01 -4.0000E-01  4.0000E-01 -3.4969E-08  4.0000E-01 -1.7000E+01  1.7000E+01 -1.4862E-06  1.7000E+01
 -6.0000E+01 -1.5000E-01  1.5000E-01 -1.3113E-08  1.5000E-01 -2.5000E+01  2.5000E+01 -2.1856E-06  2.5000E+01
 -4.0000E+01 -7.5000E-02  7.5000E-02 -6.5567E-09  7.5000E-02 -2.8000E+01  2.8000E+01 -2.4478E-06  2.8000E+01
 -2.0000E+01 -2.5000E-02  2.5000E-02 -2.1856E-09  2.5000E-02 -2.9500E+01  2.9500E+01 -2.5790E-06  2.9500E+01
                    .
                    .
                    .

ZONE T="BLOCK        1 H-WING      101" I=       5 J=       5
  1.4500E+01 -1.3562E-01  1.6897E-01  0.0000E+00  0.0000E+00 -3.4749E+00  3.9613E+00  0.0000E+00  0.0000E+00
  3.3500E+01  4.8665E-02 -4.1545E-02  0.0000E+00  0.0000E+00 -3.6831E+00  4.5217E+00  0.0000E+00  0.0000E+00
  5.2500E+01  3.9289E-02 -5.0529E-02  0.0000E+00  0.0000E+00 -2.8950E+00  3.6489E+00  0.0000E+00  0.0000E+00
  7.1500E+01  5.6231E-02 -7.1287E-02  0.0000E+00  0.0000E+00 -1.8801E+00  2.3853E+00  0.0000E+00  0.0000E+00
  9.0500E+01  6.0793E-02 -7.9272E-02  0.0000E+00  0.0000E+00 -7.1527E-01  8.9627E-01  0.0000E+00  0.0000E+00
```

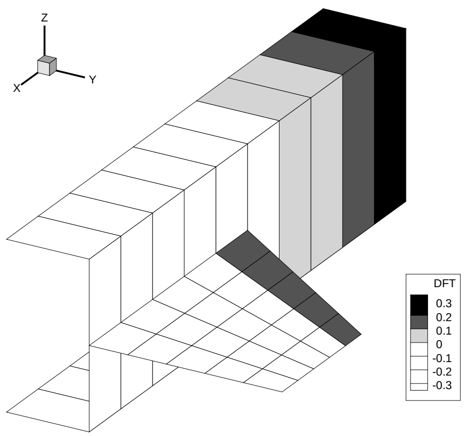
TITLE	Lists the total number of **BLOCK** elements, **BODY7** elements, and **CAERO7** elements.
VARIABLE	Defines the variable names associated with the column data.
X	X-coordinate of the aerodynamic grid point.
DFB	Slope along the bottom surface (0 for vertical lifting surfaces).
DFT	Slope along the top surface (0 for vertical lifting surfaces).
DFL	Slope along the left surface (0 for horizontal lifting surfaces).
DFR	Slope along the right surface (0 for horizontal lifting surfaces).
FB	Z coordinate of the bottom surface.
FT	Z coordinate of the top surface.
FL	Y coordinate of the left surface.
FR	Y coordinate of the right surface.
ZONE	Specifies information for the current zone.
T	Title of the current zone, containing the **BLOCK** ID, element type, and element ID.
I	Number of data points in the current zone.
J	Number of lines in the current zone. For **BODY7** elements this is always 1, and for **CAERO7** elements, it will be equal to NSPAN-1.

The slopes of the top and bottom surfaces for the body and wing are shown in the figures below. If any of the slopes on a lifting surface exceed SLPWMAX in the **PLTSLP** bulk data card, a warning message will be printed to the standard output file. This plot can show where on the wing the slope threshold is exceeded. The slopes are influenced by the airfoil definition of the **PAFOIL7** or **PAFOIL8** bulk data cards. For a BODY7 element, a warning message is printed to the standard output file if the change in slope exceeds the SLPBMAX parameter in the **PLTSLP** bulk data card. The plot can be used to find the approximate location in the body where this occurs.

A 3D contour plot can also be generated. This gives a quick overview of the slopes on the whole aircraft. In this file, there are only two variables, DFT and DFB, which represent the slopes on the top and bottom surfaces, respectively, on **CAERO7** elements. Because the **BODY7** element is represented as a box, and the slopes for the top, bottom, right, and left sides are plotted directly on the box surface, the variable selection does not affect the display on **BODY7** elements.

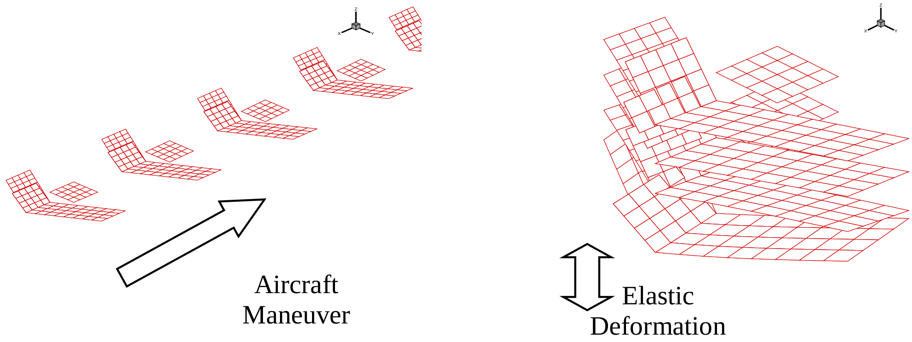






8.6 Transient Response Analysis (PLTTIME)

An output data file of the transient response of maneuver loads analysis can be generated with the **PLTTIME** bulk data card. Two types of output plot files and one ASCII text file can be generated through the use of the **PLTTIME** bulk data card. The plot files can be used to display the aircraft maneuver history (e.g., Figure 8.5(a)) or the elastic deformation history (e.g., Figure 8.5(b)). An ASCII text file can also be output that contains the NASTRAN or I-DEAS **FORCE** and **MOMENT** bulk data cards at the structural finite element grid points. This output can be inserted into the NASTRAN or I-DEAS model input deck to perform a detailed stress analysis using static structural analysis.



(a) Aircraft Maneuver History **(b) Elastic Deformation History**

Figure 8.5 Sample Output Plot Files of the PLTTIME Bulk Data Card (can be animated)

The output format for the maneuver and elastic deformation histories are identical to that of the **PLTFLUT** bulk data card (please see Section 8.3 for a description of this output format). Again, the data output is at each time step (i.e., transient).

- ***NASTRAN Compatible FORCE/MOMENT Output***

The output format of the ASCII text file containing the NASTRAN **FORCE** and **MOMENT** bulk data cards is shown in the following figure.

```
$FORCES & MOMENTS AT FEM GRIDS RESULTING FROM SET =     100
$ THE USER CAN INSERT THIS FILE BACK TO THE FEM MODEL
$ FOR SUBSEQUENT STATIC ANALYSIS AND STRESS CALCULATIONS.
$
$$$$$$$$$$   AT TIME=  0.10000E+00,  LOAD SET =     100 $$$$$$$$$
$
FORCE        100      90       0-1.26-12   1.000   0.000   0.000
MOMENT       100      90       0-2.26-10   0.000   1.000   0.000
FORCE        100      97       0-1.55+02   1.000   0.000   0.000
MOMENT       100      97       05.809-10   0.000   1.000   0.000
                                   .
$
$$$$$$$$$$   AT TIME=  0.12000E+00,  LOAD SET =     101 $$$$$$$$$
$
FORCE        101      90       04.872-13   1.000   0.000   0.000
MOMENT       101      90       0-1.64-10   0.000   1.000   0.000
FORCE        101      97       0-2.91+01   1.000   0.000   0.000
MOMENT       101      97       05.851-10   0.000   1.000   0.000

                                   .
```

Six title lines are output, each beginning with a NASTRAN comment (**$**) card, that list the **MLOADS** (transient maneuver load set) bulk data card ID number, and LOAD SET ID's generated for each time step. Each load set corresponding to a specific time can be inserted into a NASTRAN input deck to perform a static structural analysis.

- ***I-DEAS Compatible FORCE/MOMENT Output***

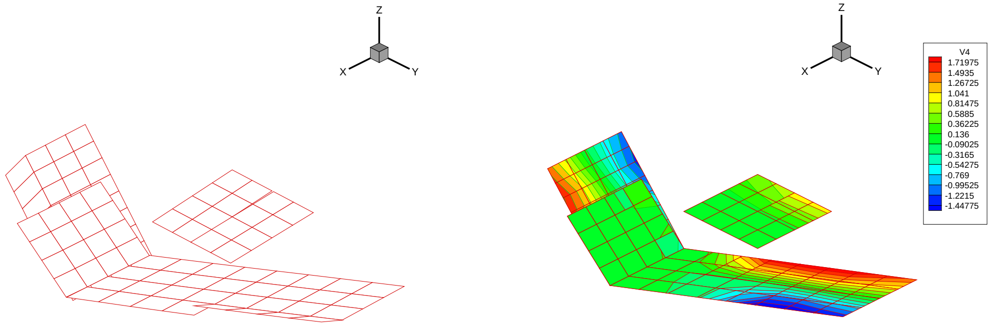
The output format of the ASCII text file containing I-DEAS output of **FORCE** and **MOMENT** stored in universal dataset 782 for each load set as shown in the following figure.

```
  -1
     782
        100          1
 AT TIME =  0.10000E+00 LOAD SET=      100
```

```
        90          4 1 1 1 1 1 1
 -1.26953E-12  0.00000E+00  4.43699E-09  0.00000E+00 -2.26623E-10  0.00000E+00
        97          4 1 1 1 1 1 1
                                 .
                                 .
    -1
    -1
    782
       101          1
AT TIME =  0.12000E+00 LOAD SET=      101
        90          4 1 1 1 1 1 1
  4.87298E-13  0.00000E+00  4.51772E-09  0.00000E+00 -1.64101E-10  0.00000E+00
        97          4 1 1 1 1 1 1
 -2.91847E+01  0.00000E+00 -3.05421E+03  0.00000E+00  5.85172E-10  0.00000E+00
```

8.7 Static Aeroelastic/Trim Analysis Results (PLTTRIM)

An output data file of the static aeroelastic/trim results can be generated with the **PLTTRIM** bulk data card. Two types of output plot files and one ASCII text file can be generated through the use of the **PLTTRIM** bulk data card. The plot files can be used to display the deformed aerodynamic model under flight loads (e.g., Figure 8.6(a)) and/or the resulting steady pressure distribution (e.g., Figure 8.6(b)). An ASCII text file can also be output that contains the NASTRAN or I-DEAS **FORCE** and **MOMENT** bulk data cards at the structural finite element grid points. This output can be inserted into the NASTRAN or I-DEAS model input deck to perform a detailed stress analysis using static structural analysis.



(a) Deformed Aero Model ***(b) Steady Pressure Distribution***

Figure 8.6 Sample Output Plot Files of the PLTTRIM Bulk Data Card

The output format for the deformed aerodynamic model is identical to that of the **PLTMODE** bulk data card (please see Section 8.4 for a description of this output format), except that both sides of the deformed aerodynamic model are included. The output format for the steady pressure distribution is identical to the **PLTCP** bulk data card except that the imaginary component of the unsteady pressure result will always be zero while the real component will reflect the steady pressure (please see Section 8.2 for a description of this output format).

- ***NASTRAN Compatible FORCE/MOMENT Output***

The output format of the ASCII text file containing the NASTRAN **FORCE** and **MOMENT** bulk data cards is shown in the following figure.

```
$FORCES & MOMENTS AT FEM GRIDS RESULTING FROM TRIM =      100 FOR FLEXIBLE MODEL
```

```
$ MACH =  0.9000 DYNAMIC PRESSURE= 0.12000E+04
$FORCES & MOMENTS IN TERMS OF NASTRAN FORCE AND MOMENT BULK DATA CARDS
$FOR TWO SIDES OF THE MODEL
$WHERE LOAD SET=      100 REFERS TO THE GRIDS ON THE RIGHT HAND SIDE OF THE MODEL
$      LOAD SET =     101 REFERS TO THE GRIDS ON THE LEFT HAND SIDE OF THE MODEL
$ THE USER CAN INSERT THIS FILE BACK TO THE FEM MODEL
$ FOR SUBSEQUENT STATIC ANALYSIS AND STRESS CALCULATIONS.
FORCE        100      90       00.000+00   1.000   0.000   0.000
FORCE        100      90       00.000+00   0.000   1.000   0.000
FORCE        100      90       01.665+04   0.000   0.000   1.000
MOMENT       100      90       04.083+04   1.000   0.000   0.000
MOMENT       100      90       04.859+04   0.000   1.000   0.000
MOMENT       100      90       00.000+00   0.000   0.000   1.000
                                 .
                                 .
```

Eight title lines are output, each beginning with a NASTRAN comment (**$**) card, that list the **TRIM** bulk data card ID number, Mach number, dynamic pressure, and LOAD SET ID's within the current file that indicate whether the entries refer to the right-hand or left-hand sides of the model.

- ***I-DEAS Compatible FORCE/MOMENT Output***

The output format of the ASCII text file containing I-DEAS output of **FORCE** and **MOMENT** stored in universal dataset 782 for both Left-Hand-Side (LHS) and Right-Hand-Side (RHS) load sets is shown in the following figure.

```
-1
    782
       100         1
RIGHT-HAND-SIDE OF FLEXIBLE MODEL
        90         4 1 1 1 1 1 1
  0.00000E+00  0.00000E+00  1.66503E+04  4.08355E+04  4.85924E+04  0.00000E+00
        97         4 1 1 1 1 1 1
 -2.93787E-10 -7.39602E-13 -7.20023E+03  0.00000E+00  0.00000E+00  0.00000E+00

                                 .
                                 .
-1
    782
       101         1
LEFT-HAND-SIDE OF FLEXIBLE MODEL
        90         4 1 1 1 1 1 1
  0.00000E+00  0.00000E+00  1.14435E+04  2.51545E+04  3.56345E+04  0.00000E+00
        97         4 1 1 1 1 1 1
 -2.93787E-10 -7.39602E-13 -7.20023E+03  0.00000E+00  0.00000E+00  0.00000E+00
```

8.8 Flutter Frequency and Damping Curves (PLTVG)

An output data file of the flutter frequency and damping curves can be generated with the **PLTVG** bulk data card (see Figure 8.7). The **PLTVG** output data is saved in column format which can be read in by virtually any spreadsheet application. The y-axis values are the frequency and damping results from a referenced **FLUTTER** bulk data card, while the x-axis can be specified as any one of the following:

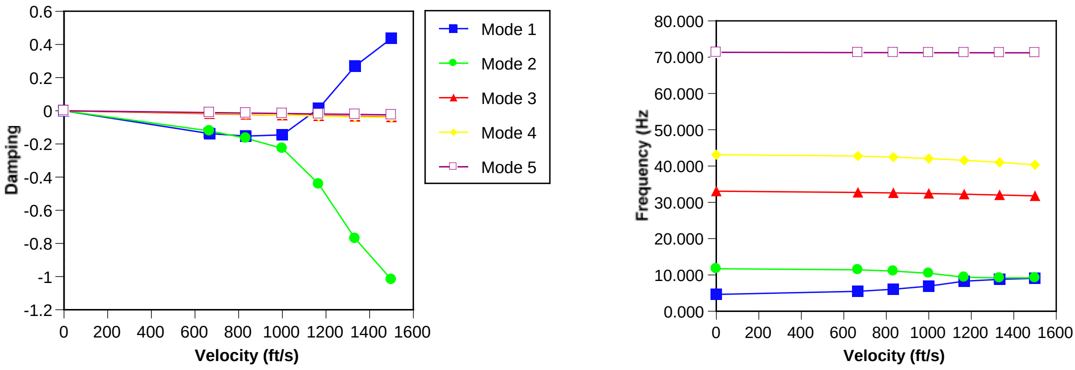


Figure 8.7 Flutter Curves of Damping and Frequency vs. Velocity

Mach number (M), density (R), dynamic pressure (Q), altitude (H), velocity (V) and normalized velocity (V/VR) (i.e., V/Vref, where V is the velocity and Vref is a reference velocity; useful to convert from one set of units to another). Note that VREF is specified on the **FIXxxxx** bulk data card(s). The **PLTVG** output plot file contains the aerodynamic damping and frequency versus velocity results of the flutter analysis. The number of modes output can be specified by the NMODE entry of the **PLTVG** bulk data card. The **PLTVG** output plot file contains the aerodynamic damping and frequency versus velocity results of the flutter analysis. The number of modes output can be specified by the NMODE entry of the **PLTVG** bulk data card.

A Windows plotting application called VGPlot.exe has been developed to allow for automated plotting of the PLTVG output. This program, along with its User's Manual, can be found in the ZEUS installation directory under the \miscel\VGPlot folder. VGPlot can be used to generate damping and frequency curves versus the parameter you specify for the XAXIS entry. Note that FORM must be set to TABLE (the default) in order to generate the appropriate output file format that is readable by VGPlot.

- ***TABLE Output (Microsoft Excel Compatible)***

A sample of the spreadsheet-formatted output is shown in the following figure and is described below.

```
DAMPING & FREQUENCY X-Y PLOT FILE OF PLTVG IDSET=      11 FOR FLUTTER ID=      100 NMODE=    5
      V        G,MODE 1    G,MODE 2    G,MODE 3    G,MODE 4    G,MODE 5        V     W HZ,MODE 1 W HZ,MODE 2 W HZ,MODE 3 W HZ,MODE 4 W HZ,MODE 5
  .0000E+00   .0000E+00   .0000E+00   .0000E+00   .0000E+00   .0000E+00   .0000E+00   .4613E+01   .1170E+02   .3307E+02   .4312E+02   .7132E+02
  .8000E+04  -.1492E+00  -.1134E+00  -.2034E-01  -.1593E-01  -.8797E-02   .8000E+04   .5550E+01   .1127E+02   .3267E+02   .4274E+02   .7116E+02
  .1400E+05   .1686E+00  -.6141E+00  -.3630E-01  -.3589E-01  -.1460E-01   .1400E+05   .8633E+01   .8891E+01   .3212E+02   .4152E+02   .7101E+02
  .1600E+05   .3886E+00  -.9165E+00  -.4148E-01  -.4159E-01  -.1637E-01   .1600E+05   .8914E+01   .8906E+01   .3185E+02   .4093E+02   .7095E+02
  .1800E+05   .5486E+00  -.1167E+01  -.4595E-01  -.4443E-01  -.1803E-01   .1800E+05   .9173E+01   .8966E+01   .3155E+02   .4023E+02   .7089E+02
```

Velocity

Damping
Values

Velocity

Frequency
Values

SETID	Set identification number of the **PLTVG** bulk data card.
FLUTTER ID	Identification number of the referenced **FLUTTER** bulk data card specified in the IDFLUT entry of the **PLTVG** bulk data card.
NMOD	The number of modes for which damping and frequency values are printed.
V	Velocity values in units specified by the FLTUNIT entry of the **FIXMATM**, **FIXMACH**, or **FIXMDEN** bulk data card. The default time unit will be seconds unless the VREF entry in one of these bulk data cards is set to some value to account for the conversion. For example, if the FLTUNIT is set to "FT" and VREF is set to "1.0" then the resulting velocities will be in ft/sec. If the desired velocities are knots, then a value of VREF set to "1.68" should be used. In addition, the XAXIS entry of the **PLTVG** bulk data card should be set to "V/VR" to implement the desired conversion.
G,MODE x	Flutter damping values of mode number 'x'.
W HZ,MODE x	Flutter frequency values (in Hertz) of mode number 'x'.

- ***I-DEAS Compatible Output***

The I-DEAS compatible output is saved in the universal file format. Data set 58 is used to output the damping and frequency curves versus Mach number, density, dynamic pressure, altitude, normalized speed, or velocity. One data set 58 is used to define an abscissa and ordinate data characteristics and therefore represents a single curve of a single plot. The number of data set 58 output will equal 2 x number of modes requested in the **PLTVG** bulk data card (i.e., the first set of output for the damping and the second for the frequency). The first five ID lines of data set 58 list the following:

Line 1:	Standard output title: 'ZEUS FLUTTER RESULTS'
Line 2:	Type of data output (e.g., Density versus Frequency)
Line 3:	The date and time the universal file was created
Line 4:	**PLTVG** bulk data card identification number, referenced **FLUTTER** bulk data card identification number, and flutter mode number
Line 5:	NONE

A sample of the I-DEAS compatible output is shown in the following figure:

```
    -1
    58
ZEUS FLUTTER RESULTS
OUTPUT OF MACH NUMBER VERSUS DAMPING
DATE: 04-08-1999  TIME: 13:03:08
OUTPUT FROM PLTVG WITH SETID=      11 , FOR FLUTTER ID=     100 , MODE NO.     1
NONE
    0         1    1        11    DAMPING         1   0       M          100   0
         2         8         0  0.00000E+00  0.00000E+00  0.00000E+00
         1    0    0    0 MACH NUMBER
         1    0    0    0             DAMPING                    G
         0
         0
  0.00000E+00  0.00000E+00  0.70000E+00 -0.16262E+00  0.80000E+00 -0.16941E+00
  0.90000E+00 -0.65267E-01  0.95000E+00  0.21774E-01  0.10500E+01  0.80173E-01
    -1
    -1
```

```
    58
ZEUS FLUTTER RESULTS
OUTPUT OF MACH NUMBER VERSUS DAMPING
DATE: 04-08-1999  TIME: 13:03:08
OUTPUT FROM PLTVG WITH SETID=      11 , FOR FLUTTER ID=     100 , MODE NO.     2
NONE
    0         2    1        11    DAMPING         2   0        M          100   0
         2         8         0  0.00000E+00  0.00000E+00  0.00000E+00
         1    0    0    0 MACH NUMBER
         1    0    0    0              DAMPING                    G
         0
         0
  0.00000E+00  0.00000E+00  0.70000E+00 -0.10325E+00  0.80000E+00 -0.17427E+00
  0.90000E+00 -0.37126E+00  0.95000E+00 -0.28615E+00  0.10500E+01 -0.34533E+00
    -1
                                                              .
```

- ***FEMAP Compatible Output***

The FEMAP compatible output is saved in the FEMAP (Version 7.0) neutral file format. Data Blocks 403 and 404 are used to output the fictitious aerodynamic grids and bar elements, respectively. Since FEMAP requires XY plots to be functions of ID, Set Values or nodal Positions, the x-coordinate values (i.e., Mach number (*M*), density (*R*) , dynamic pressure (*Q*), altitude (*H*), normalized speed (*V/VR*), or velocity (*V*)) requested by the **PLTVG** bulk data card are saved as *x*,*y*,*z* coordinates in Data Block 403 (i.e., $x = y = z$).

Two output set definitions (Data Block 450) are used to output the damping and frequency results. Data Block 451 is used to output the damping and frequency values for the numbers of modes (NMODE) requested in the **PLTVG** bulk data card(s).

A sample of the FEMAP compatible output is shown in the following figure.

```
-1
   100
<NULL>                        Neutral File Header followed by
       7.                     other required Data Blocks
   -1
-1
                                                                     .
                                                                     .

-1
   403
         1,           0,            0,            1,           46,      0,      0,      0,      0,      0,      0,
0.0000000000000000D+00,   0.0000000000000000D+00,   0.0000000000000000D+00,          0,
         2,           0,            0,            1,           46,      0,      0,      0,      0,      0,      0,
6.9999998807907104D-01,   6.9999998807907104D-01,   6.9999998807907104D-01,          0,
                                                                     .
                                                                     .
                                                                                 Fictitious Aerodynamic
-1                                                                               Grid Points for
   404                                                                           X-coordinate values
         1,       124,           1,          17,           0,           1,           0,           0,           0,
0,         0,         0,
         1,          2,           0,           0,           0,           0,           0,           0,           0,
0,
         0,          0,           0,           0,           0,           0,           0,           0,           0,
0,
   0.0000000000000000D+00,   0.0000000000000000D+00,   0.0000000000000000D+00,
   0.0000000000000000D+00,   0.0000000000000000D+00,   0.0000000000000000D+00,
    0,     0,     0,     0,     0,     0,     0,     0,     0,     0,     0,     0,          0,          0,          0,
0,
                                                                     .                   Fictitious Bar
                                                                     .                   Elements
                                                                     .
```

```
-1
   450
   1,
M vs Damping
   0,   2
   0.0000000000000000D+00,
   1,
 <NULL>
   -1
   -1
   451
   1,   1,   1,
Mode No.   1
   0.,   0.,   0.,
  1, 0,  0,  0,  0,  0,  0,  0,  0,  0,
  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
  0,  0,  1,  7,
  0,  1,  1,
        1,   0.0000000000000000D+00,
        2,  -1.6261789202690125D-01,
                                     .
                                     .
```

Damping/Frequency Output Set Definition

Damping/Frequency Value Output Data Vectors

- *<u>Steps within FEMAP to View the Damping and Frequency Curves</u>* *(ZEUS output file generated by **PLTVG** bulk data card)*

  - Open (via *File/Import/FEMAP* Neutral) the ZEUS output neutral file of the flutter damping and frequency results (select *View/Redraw* if the image does not appear after loading)
  - Ignore any model that is displayed. The model is created for X-axis data points only.
  - Text will appear on the screen to describe how to view the plots. This text can be turned off from the *View/Options/Text/Entity* Off option.
  - Open the *View/Select* Window.
  - In the XY Style section, select the XY vs. Position checkbox.
  - Click on the “XY Data” button bar.
  - In the window that opens, under Output Set, select either xx vs. Damping or xx vs. Frequency to view the Damping or Frequency curves, respectively (where xx is either M, R, Q, H, V/VR, or V as described above).
  - Under Output Vector, select the mode to be displayed as curve 1.
  - Click on the next curve button (Curve 2)

Repeat the above 3 steps for up to 9 curves to be displayed.

  - Click on the “OK” button in both windows.
  - To generate multiple plots, open a new window with *View/New* and repeat the above procedure for the new window. In this fashion, V-G curves can be placed side-by-side.

- ***PEGASUS Compatible Output***

The PEGASUS compatible output is saved in ESA data format. A sample of the PEGASUS compatible output is shown in the following figure and is described below:

```
*ESA
*COM
*COM  G & F X-Y PLOT FILE OF PLTVG SETID=      30 FOR FLUTTER ID=       3 NMODE=    8
*COM  HORIZONTAL AXIS IS     Q
*COM
*FILE_SCALAR TEXT DATE
08/23/2000
*EOD
RUN1
*RTITLE
```

Comment Cards

Run Title

```
 MODE NO.        1
*EOD
*FLOAT
   DYNAMIC P           G       W(HZ)
  0.0000E+00  0.0000E+00  7.5312E+00
  1.7124E+01 -8.4357E-02  8.3428E+00
  4.2810E+01 -1.9205E-01  9.6138E+00
                    .
                    .
*EOD
RUN2
*RTITLE
 MODE NO.        2
*EOD
*FLOAT
   DYNAMIC P           G       W(HZ)
  0.0000E+00  0.0000E+00  1.5057E+01
  1.7124E+01  3.9689E-05  1.4978E+01
  4.2810E+01 -3.7342E-04  1.4861E+01

                    .
                    .
```

Arrays containing x-axis
and y-axis data
(repeated for each mode)

The comment title cards (*COM) list the identification number of the current **PLTVG** bulk data card, identification number of the referenced **FLUTTER** card, mode number, and the x-axis data label. Data arrays are output for each mode containing the x-axis data (Mach number, density, dynamic pressure, altitude, speed, or normalized speed) and y-axis data (damping and frequency).

ZONA TECHNOLOGY INC
9489 E. Ironwood Square Drive
Scottsdale, AZ 85258
Tel (480) 945-9988
Fax (480) 945-6588
info@zonatech.com
www.zonatech.com