FZM-8679
29 June 2001

Large Eddy Simulation for the Analysis of Weapon Bays Flows

Final Technical Report
Contract Number F49620-99-C-0020
CLIN Sequence No. 0002AA

Releasability of this material under the Freedom of Information Act is subject to the restrictions on release in DoD Regulation 5400.4-R and DoD Directive 5230.25.

DISTRIBUTION STATEMENT A: Approved for public release; distribution is unlimited.

Prepared and Approved for Release by:

Brian R. Smith

Dr. Brian R. Smith
Lockheed Martin Aeronautics Company
Air Vehicle Sciences and Systems

REPORT DOCUMENTATION PAGE

1. REPORT DATE *(DD-MM-YYYY)* 29-06-2001	**2. REPORT TYPE** Final Technical Report	**3. DATES COVERED** *(From - To)* May 1999 - June 2001

4. TITLE AND SUBTITLE
Large Eddy Simulation for the Analysis of Weapons Bay Flows

5a. CONTRACT NUMBER
F49620-99-C-0020

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)
Smith, Brian R.; Welterlen, Tracy J.; Domel, Neal D.

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Lockheed Martin Aeronautics Company
Air Vehicle Sciences and Systems
P.O. Box 748
Fort Worth, TX 76101
Attn: Dr. Brian R. Smith, MZ 9333

8. PERFORMING ORGANIZATION REPORT NUMBER
Lockheed Martin Aeronautics Company Report No. FZM-8679

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Air Force Office of Scientific Research
AFOSR/NA
801 N. Randolph St., Rm 732
Arlington, VA 22203-1977
Attn: Dr. Robert Herklotz

10. SPONSOR/MONITOR'S ACRONYM(S)
AFOSR/NA

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION / AVAILABILITY STATEMENT
DISTRIBUTION STATEMENT A. Approved for public release; distribution is unlimited.

13. SUPPLEMENTARY NOTES

14. ABSTRACT

Computational methods for analysis of weapons bay acoustics and weapons separation were developed. Methods were developed for and implemented in Splitflow, a Cartesian based unstructured grid computational fluid dynamics flow solver. Using a Large Eddy Simulation model, a variety of flux discretization approaches were investigated. Moving grid and six degree of freedom dynamics capability were implemented in the parallel processing version of Splitflow. Methods for interfacing multiple Splitflow solutions separated by a common boundary surface were developed. A Mach 1.2 flow over a simple cavity flow was simulated. The inflow boundary layer for the cavity flow solution was obtained from a compressible, LES channel flow solution. A periodic boundary condition between the inflow and outflow of a compressible channel flow simulation was developed. Computational predictions of sound pressure level spectra were compared to experimental data. A preliminary demonstration of the capability to simulate the release of an AIM 120 from an F-22 fighter jet was performed. Robustness of the simulation methodology was assessed.

15. SUBJECT TERMS

weapons bay, cavity, Large Eddy Simulation, store separtion, moving body simulation, unsteady computational fluid dynamics, CFD simulation and modeling

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON** Dr. Brian R. Smith
a. REPORT **Unclassified**	**b. ABSTRACT** **Unclassified**	**c. THIS PAGE** **Unclassified**	UU	58	**19b. TELEPHONE NUMBER** *(include area code)* 817.935.1127

Table of Contents

List of Figures

Section 1: Need

The new generation of fighter aircraft include internally carried stores. Fighter aircraft such as the F-22 and the Joint Strike Fighter must separate weapons over a wide range of Mach numbers and maneuver conditions. The acoustic and aerodynamic loads for low altitude, supersonic weapons release are close to an order of magnitude higher than is found for high altitude, transonic weapons release. As a result, predictions of acoustic loads and store separation weapons bays has taken on new importance. The cost of wind tunnel testing for assessment of acoustic loads and store separation in the development phase of a fighter aircraft program can run into tens of millions of dollars. Subsequent flight testing for certification requires additional tens of millions of dollars. Development of computational methods for acoustic analysis and weapons separation prediction holds the promise of saving many millions of dollars in wind tunnel and flight testing and reducing design cycle time on future aircraft programs.

The internal weapons bay of a modern fighter aircraft is extremely complex geometrically due to the presence of bulkheads, stores, bay doors and the storage of a variety of flight hardware in the bay. The flow in and near the weapons bay is highly unsteady. The flowfield external to the bay is strongly influenced by both the internal and external aircraft geometry. As a result, simulation of the weapons bay flowfield using Computational Fluid Dynamics (CFD) requires large computational meshes in order to adequately resolve highly complex fighter aircraft geometries and flowfields. The goal for this AFOSR research program is to develop computational methods that can be used efficiently in the accurate prediction of acoustic loads in weapons bays and store release.

1.1 Objectives

The program objectives were to develop and demonstrate methods for the accurate simulation of weapons bay acoustic loads and to develop and demonstrate methods for weapons separation from a fighter aircraft weapons bay. The acoustic load test case is a Mach 1.2 Weapons in Cavity (WICS) study. The store release simulation demonstration is for an AIM-9 missile release from an F-22 at Mach 1.6. The methods developed for this program and the demonstration cases form a foundation for the development of engineering design tools for the accurate simulation of the weapons bay environment and store release.

These objectives proved challenging given the program budget. All of the key goals for methodology development were met. Acoustic load and store separation demonstration cases were run, and preliminary results were obtained. The demonstration cases identified parts of the methodology that need to be streamlined in order to achieve more rapid turnaround of the solutions.

For acoustic predictions, the LES methods implemented in the parallel version of Splitflow were tested on a simple cavity flowfield. Numerical diffusion effects and numerical flux schemes were developed and investigated. A method was developed for simulation of an essentially equilibrium compressible turbulent boundary layer for input to an LES cavity solution.

To enable store separation simulation, important capabilities were added to Splitflow. Moving grid terms were added for grid translation and rotation. Methods for initialization of computational cells originating at store boundaries due to the motion of the store were developed. Six degree of freedom dynamics (6-DOF) simulation capability was added to Splitflow. The output from the 6-DOF routines was used to specify the motion of the store through the flowfield at each iteration. An interface method was developed to allow regions of the flowfield with a body fitted grid to interface with a cut grid.

1.2 Overview of Weapons Bay and Store Release Flows

The flow in the shear layer over and in an open weapons bay is highly unsteady. Much of the large scale unsteadiness in the weapons bay is triggered by acoustic feedback. Deflections in the separated shear layer over the top of the bay generate pressure disturbances. These disturbances generate acoustic waves that travel upstream through the slow moving flow in the weapons bay. When the upstream running pressure waves reach the front wall of the bay, they reflect off of the front wall of the bay and disturb the separated shear layer near the separation point. The shear layer disturbance grows as it is convected downstream, resulting in the generation of new pressure disturbances. This mechanism can result in a resonant feedback in weapons bays and in simple cavities. In addition to the large scale unsteadiness, the incoming boundary layer and separated shear layer are highly turbulent, include small scale turbulent structures.

Successful simulation of the weapons bay flowfield requires capturing the effects of both the large scale unsteadiness and the turbulent boundary and shear layers.[1] Attempts to use unsteady Reynolds Averaged Navier-Stokes turbulence models results in excessive damping of the larger scale unsteadiness which is largely responsible for the acoustic loads in the weapons bay. Large Eddy Simulation (LES) resolves all unsteady scales larger than the grid related filter length. As a result, not only does LES preserve the large scale acoustic structures, it also simulates the largest turbulent scales and can capture the interaction between these scales and the acoustic structures. In a previous, closely related AFOSR program "Large Eddy Simulation for Analysis of Transonic Cavity Flows[2]," compressible LES methods were developed for application to cavity flows. This technology was implemented in the LM Aeronautics "Falcon" structured grid flow solver and applied to simple cavity flow problems.[1] Good agreement between simulated and experimental acoustic loads was obtained, although the results were shown to be somewhat sensitive to grid resolution.

Store release poses several computational challenges. The simulation of store release requires the ability to modify the computational mesh to capture the moving store. At each computational iteration, aerodynamic loads must be determined on the store. These loads are combined with the store translation rate, rotation rate, mass and moment of inertia in a six degree of freedom dynamics procedure to determine the displacement of the store. The CFD solver must then displace the store computationally by adjusting the computational mesh to accommodate the store's new position. Effects of grid motion must be included in the flow solver.

Both structured and unstructured grid flow solvers have been used to simulate store separation. For structured grid methods, one mesh is generated around the store, and another mesh is generated around the aircraft. As the store mesh moves with the store, the aircraft grid is cut at every iteration at the interface between the store grid and the aircraft grid. At this interface, mass, momentum and energy fluxes are passed between the two grids. Typically, for unstructured grid methods, a single computational mesh is stretched, distorted and possibly refined to capture the motion of the store. Typically, if the store translation or rotation is too great, the unstructured grid will become excessively distorted, and generation of a new grid is required. In contrast, the structured grid method avoids this problem, but the cutting of the aircraft grid on every iteration is complex and can be computationally time consuming.

1.3 Overview of the Technical Approach

A fighter aircraft with an internal weapons bay presents an extreme challenge for CFD simulation. The generation of a structured mesh around a fighter aircraft with an open, fully loaded weapons bay would require many weeks for an expert in grid generation. Unstructured mesh generation methods can significantly reduce the grid generation time required for complex configurations. The Splitflow unstructured grid flow solver has unique capabilities that make it particularly well suited to weapons bays analysis. The grid cutting technique allows complex weapons bay geometries to be handled with relative ease. Starting from an appropriately processed aircraft geometry, the Lockheed Martin Splitflow unstructured grid flow solver can be used to generate an initial mesh for flow solution in a few hours.

Store separation simulation with Splitflow is unique, unlike either structured or unstructured methods. As the store translates through the flowfield, the Cartesian Splitflow mesh is cut by the store geometry. The irregular shaped cells cut by the store are treated as boundary cut cells. As the store translates through the flowfield, the Splitflow mesh is refined in areas close to the store and coarsened in regions which the store has passed through. Thus, unlike other unstructured grid methods, the grid is not excessively distorted by store motion. Splitflow also includes an alternate moving body method where a body conforming mesh is generated around the store, and this grid moves with the store similar to the structured grid store separation approach. The interfacing between the moving store grid and the aircraft grid is highly automated in Splitflow.

Grid resolution is provided by subdivision of the Cartesian cells. Unlike most structured and unstructured flow solvers, individual cells can be subdivided, while neighboring cells are left unchanged. This makes rapid, efficient and conservative grid refinement possible. This approach is extremely well suited to LES, since cells can be refined in all three coordinate directions near walls where eddy sizes demand high resolution. Away from walls, cell spacing in lateral and streamwise directions is not governed by near wall requirements.

1.4 Summary of preceding work

The technical foundation for the current program was developed under predecessor contract research and LM Aero IR&D programs. In the closely related AFOSR program "Large Eddy Simulation for Analysis of Transonic Cavity Flows," LES capability was developed for Splitflow and implemented in a hybrid version of Splitflow that did not include parallel processing capability. Between the completion of this CRAD program and the beginning of the current research program, LES modeling capability was implemented in the parallel version of Splitflow. In addition to parallel processing capability, this version of Splitflow includes new grid generation capabilities and a variety of flow solver enhancements.

In the predecessor program, "Large Eddy Simulation for Analysis of Transonic Cavity Flows," a compressible form of the Smagorinsky sub grid scale model was implemented in the hybrid version of Splitflow. In this version of Splitflow, near wall boundary layers are resolved using triangular prisms generated in layers by marching a triangulated surface mesh away from the no-slip wall boundaries. The outer layer of cell faces from the prismatic mesh interface with the octree based Cartesian Splitflow grid. The Cartesian grid is limited to cubic cells, and grid refinement requires subdivision of a cell into eight "child" cells. This approach has several shortcomings. It has proven difficult to march the prismatic mesh far from the surface for complex geometries. As a result, resolution of the outer part of the boundary layer requires the use of large numbers of cubic cells. In addition, it is not possible to refine the prismatic mesh to local flow gradients because the location and density of the triangular surface mesh was fixed on the boundary surface prior to running Splitflow. As are result, automatic refinement capability, a key feature of Splitflow, was lost for the prismatic mesh. The compressible flow solver was used to simulate subsonic, essentially incompressible channel and backward facing step flows. Reasonable agreement between experimental data and simulations were obtained. Because it does not include parallel processing, the code took a large amount of wall time to complete the computations.

Between the completion of the LES for Analysis of Transonic Cavity Flows program and the beginning of the current program, Splitflow LES capability was significantly improved. The LES model was implemented in a parallel version of Splitflow with extensive new capabilities. The parallel capability allows multiple processors to be applied to a single problem, greatly reducing wall time required for solutions. The code can accept an externally generated, body conforming grid generated by the "Omnigrid" grid generator. Grids generated with Omnigrid can be refined in a manner similar to the traditional, "cut grid" version of Splitflow. This version of Splitflow can also generate its own Cartesian mesh. Unlike the hybrid version of Splitflow, the Cartesian mesh is not restricted to cubic cells. Cells can be split in any of the 3 Cartesian directions, creating either two, four or eight children in a refinement process, resulting in non-cubic cells. Aircraft geometries and flowfield gradients can be resolved with far fewer cells than was required with the oct-tree version of Splitflow.

Section 2: Methodology

In order to present a coherent picture of the methodology applied to the store separation and acoustic problems, methods developed prior to the current program as well as methods developed as part of the current program are described in Section 2. Section 2.6, LES Modeling and Implementation, describes methods developed under the "Large Eddy Simulation for the Analysis of Transonic Cavity Flows" program and implemented in the parallel version of Splitflow under an LM Aero IR&D program. Section 2.1, Parallel Splitflow and Section 2.2, Splitflow Grid Schemes describe methods developed and implemented under IR&D prior to the beginning of the current program. The balance of Section 2 describes methods developed as part of the current program.

2.1 Parallel Splitflow

Parallel processing is achieved using the Parallel Virtual Machine (PVM) message passing protocol. One process of each parallel job is designated as the "mother" process, where grid generation, post processing and the other parallel processes are controlled. The grid encompassing the entire flow domain is decomposed into a number of subdomains, each of which allocated to a process. The subdomains are defined automatically, and the number of cells and number of boundary triangular facets can be balanced in order to obtain efficient use of multiple processors.

A pointwise implicit solution algorithm is usually used for parallel processing in Splitflow. The pointwise scheme iteration history is independent of the number of processors applied. For the pointwise scheme, multiple subiterations are performed at each iteration level to obtain a time accurate solution. The basic numerical form of the implicit equation is:

$$\left[\frac{\Omega}{\Delta t}I + \sum\frac{\partial F}{\partial Q}\right](Q^{s+1} - Q^{s}) = -[\sum F^{s} \bullet \bar{A} + (Q^{s} - Q^{n})] , \qquad (1)$$

where the superscript *n* indicates the previous time step. The superscript *s* designates the most updated estimate of time level *n+1*. When the solution at sub-iteration *s+1* is sufficiently equal to that at sub-iteration *s*, then the sub-iterations are complete, and the solution is advanced to *n+1*.

The flux Jacobians include the inviscid Jacobians consistent with Roe's scheme, computed using first order data. The algorithm requires a block inversion of a 5×5matrix for each cell. The inverted matrix is computed during the first sub-iteration and stored for use in subsequent sub-iterations. Although the Jacobians are frozen, the fluxes must be updated with each sub-iteration. Upon convergence of the subiterations, the linearization errors are eliminated because the inter-face fluxes are evaluated directly from the most up-to-date solution at the centroids. Typically, 10 to 15 sub-iterations are adequate to sufficiently converge the implicit equations for a time dependent solution. More sub-iterations are generally required for as the time step is increased. A more detailed description of Splitflow solution algorithms is contained in Domel and Karman.[3]

2.2 Splitflow Grid Schemes

Cut Grid Method: In the cut grid scheme, cell faces are oriented with respect to Cartesian axes. Boundary surfaces, including no-slip walls, are represented by triangulated surface meshes. Where the triangulated surface mesh intersects Cartesian cells, the Cartesian cells are cut into irregular shapes. The cut cell surfaces are cut into triangular boundary facets as shown in Figure-Equation 2.1. Boundary conditions are applied at the boundary facets, and appropriate fluxes are evaluated at each boundary cut cell. Because of this approach, the Splitflow simulation can commence once a triangulated surface mesh is defined.

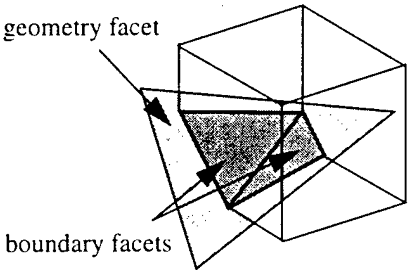

Figure 2.1 Splitflow automated cut grid method allows rapid problem set-up

Grid refinement is achieved by cell sub-division. Cells can be divided in any of three coordinate directions. Figure 2.15 shows the cell division process. Cells are divided in an initial grid generation process to meet geometry resolution constraints. Typically, 10 - 20 levels of cell division are present in an initial grid. Once the flow solver is applied to the initial grid, addition cells are refined in order to resolve flowfield gradients.

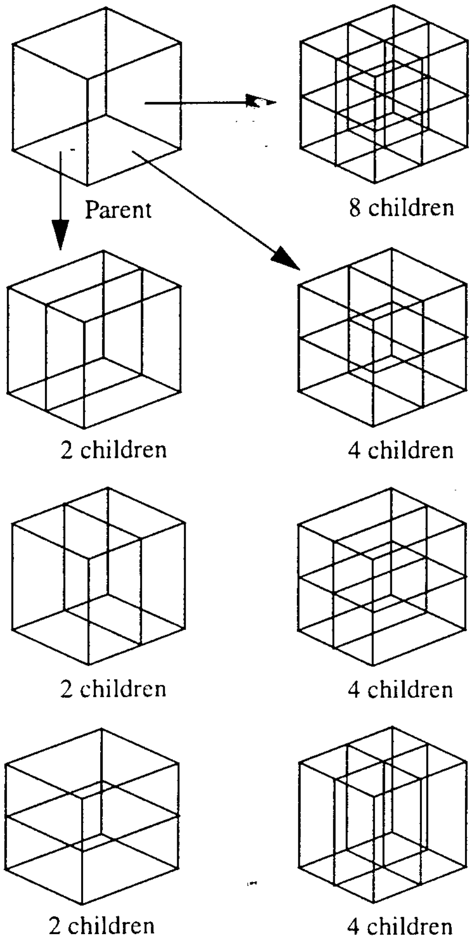

Figure 2.2 Splitflow grid cell refinement options

"Omnigrid" Body Fitted Grid Method: Omnigrid is used to develop body conforming grids with sufficient smoothness and near wall resolution for efficient viscous flow simulations. The omnigrid process begins with generation of a conventional oct-tree based Cartesian Splitflow mesh around a geometry represented by a triangulated surface facets. Next, cells intersected by or in close proximity to the geometry are removed. New grid lines are projected from the nodes on the exposed faces of the remaining cells to the surface. This produces a body fitted grid. High resolution grids for viscous analysis are generated by refining the grid, particularly in the layer of cells between the body surface and the Cartesian volume mesh. The resulting mesh can be refined to resolve other flow field gradients using user selected adaptation functions.

Resolution of sharp features in the geometry can be critical for accurate simulation. Omnigrid can capture sharp features in the geometry by "snapping" node points to edge locations. To preserve sharp edges in the geometry, quadrilateral faces on the surface of the geometry in the vicinity of the sharp edge are identified. Using these faces, the mesh nodes closest to the sharp edge are labeled and snapped to sharp edges. This approach allows the computational mesh to capture geometric details without excessive numbers of nodes.

The omnigrid generation process in general, and the snapping process in particular, can create regions of tangled mesh with grid crossing. To correct these regions and improve the quality of the grid, an elliptic smoothing scheme is used for interior points with a spring analogy scheme for boundary nodes. The current version of the elliptic smoothing solves the Laplace equations. An example of the smoothing scheme is shown in Figure 2.3 for the cross section of a chined forebody.

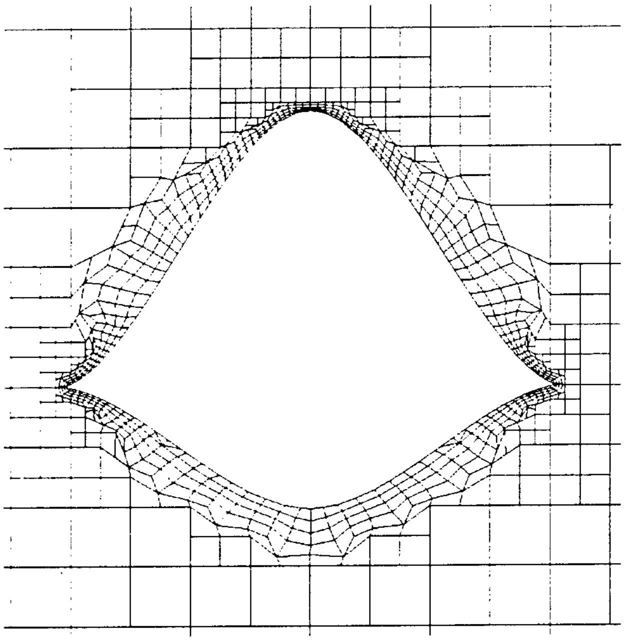

Figure 2.3 Example of a grid around a chined nozzle generated with Omnigrid

2.3 Moving Grid Methods

The addition of rigid body motion complicates the time-accurate CFD simulation by introducing grid speed terms and changing cell volumes. Investigators generally approach this complexity in one (or some combination) of three ways, each having advantages and disadvantages. The first method involves a single deforming grid which preserves the identity of each grid cell. Each cell is allowed to stretch and deform in order to fill the computational domain around the moving body. This works well for small displacements, but large displacements often result in highly skewed, compressed or stretched grid cells which necessitate occasional regridding and redistributing cells.

The second method, overset grids, allows rigid grids to be defined, thereby maintaining the initial grid resolution. An overset grid moves with a body over a stationary background grid. The background grid has a "hole" which is larger than the moving body, but smaller than the moving grid surrounding that body. The boundaries of the "hole" interface with the solution in the moving grid. Similarly, the outer boundaries of the moving grid interface with the background grid. Cells in the background grid are turned off and on as needed in order to allow the hole to follow the moving body. The extent of the overlap between the background grid and the moving grid must contain the entire motion of the body from one time step to the next, thereby ensuring that each point in the computational region is defined in at least one of the grids. Grids with large overlaps allow large displacements between time steps. However, care must be taken as the moving body nears the boundary of the background grid because the moving grid may extend into undefined territory outside the computational domain.

The third approach is to redefine boundary cells every iteration as the body moves. This approach allows for the majority of the cells to remain fixed from one time step to the next, with only boundary cells changing with time. This approach allows for very extreme motion of the body within the computational domain, but the appearance and disappearance of cells from within the computational domain must be addressed. This third method was adopted for Splitflow because of its automatic grid generation capability, and its ability to modify a pre-existing grid to contain a new geometry via the cell cutting process.

Splitflow models the motion of the geometry by updating the position of the store at every iteration, and recutting the Cartesian grid in order to accommodate the updated geometry. Although the underlying Cartesian mesh does not move with time, some Cartesian cells may undergo refinement in order to resolve the new geometry position. After recutting the Cartesian cells to fit around the new geometry position, Splitflow assigns grid speeds to the appropriate boundary facets. The grid speeds of these boundary facets determine the rate of change of the volumes of the boundary cells adjacent to the moving body. Only cells adjacent to a moving boundary have time-variant volumes.

This method of recutting the grid around a moving body causes new cells to appear inside the computational domain. Similarly, cells may disappear from the computational domain due to the motion of the body. The proper way to strictly enforce conservation in such cells is to merge each one with neighboring cells such that each aggregate merged cell simply changes volume, rather than appearing or disappearing completely. However, the algorithm for determining such cell combinations may get quite complicated if the moving boundary is allowed to traverse several cells in one iteration. For this program, cells are allowed to appear and disappear. While this approach is not strictly conservative, it provides adequate accuracy and reduces computational solution times. Cells are allowed to disappear without attention. However, attention must be focused upon the initialization of the solution in new cells which appear in the computational domain.

A method was developed for Splitflow which approximates the solution from an overset grid moving with the body. Because an actual overset grid does not exist, Splitflow estimates various parts of the solution as needed. Splitflow calculates the position of each uninitialized cell relative to the moving body. Then Splitflow interpolates from the appropriate position in the solution at the previous geometry location, taking into account the translation and rotation since the previous time level. Figure 2.4 illustrates the method.

Although the described method provides good accuracy for the simulation, the computational time required for searching and interpolating warranted a compromise for faster initialization from ancestor cell information. Splitflow accomplishes this by recognizing that an "appearing" cell is actually a pre-existing cell which switches its state from "off" to "on." Therefore, Splitflow determines a solution for each "off" cell before it switches state. Before the body position is updated for an iteration, Splitflow assigns each parent cell a solution based upon the volume weighted average of its computational children. Splitflow systematically works its way up to the root cell, assigning a solution to every parent in the computational domain. Then Splitflow reverses direction and assigns solutions to each cell outside the computational domain based upon its nearest computational ancestor. Therefore, all cells outside the computational domain may be initialized prior to moving the body for any iteration.

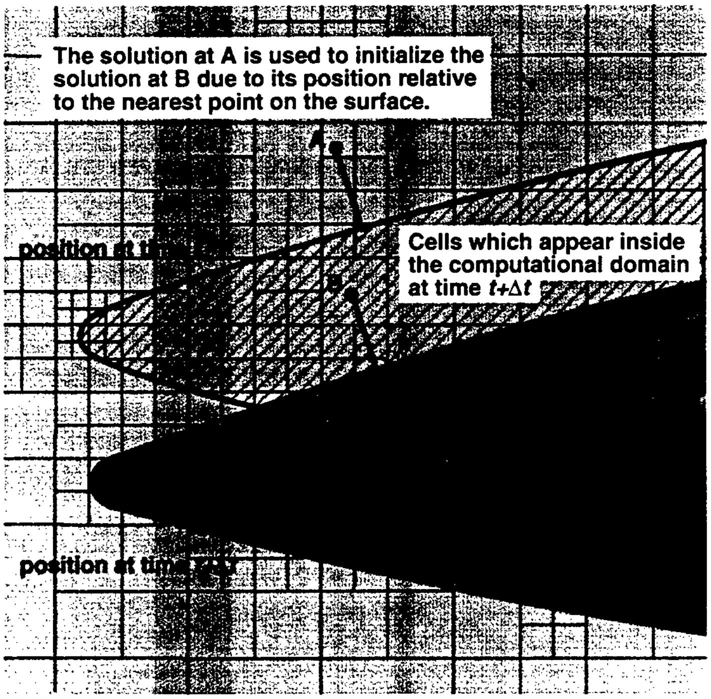
The solution at A is used to initialize the solution at B due to its position relative to the nearest point on the surface.
A
position at ti
Cells which appear inside the computational domain at time $t+\Delta t$
B
position at tim

Figure 2.4 Method for initializing cells which appear due to geometry motion

One other option included with the moving body computations in Splitflow is the capability to freeze the body position (and grid speeds and volume rates) for a few iterations, and then compensate when the body position is actually updated. This reduces frequency of body motion and grid cutting, which can significantly reduce solution times. Some sacrifice in solution accuracy is possible because the body position is updated less frequently, resulting in greater error in store positioning. However, the robustness corresponds to that of the smaller time step between iterations.

2.4 Interface Grid Methods

Flow field data transfer between domains was accomplished through an interfacing system developed at LM Aero. This flexible interfacing method uses a object-oriented file structure to pass flow field data and, in some cases, flux data between otherwise unconnected domains. It can also be used to pass information between separate Splitflow instances, or processes, and allow for simultaneous solving on mixed grid types. A description of the interfacing method and how it was used for this program follows.

Method

Two types of interfaces have been developed. The type 1 interface is used to communicate between two Splitflow cut-type grids. Interface type 2 can communicate between a Splitflow cut grid and a body-fitted grid (Figure 2.5). The body-fitted grid could theoretically be any structured or unstructured grid. Both interface types share flow field data across the interface. Flow field data includes density, three momentum components, and energy and will hereafter be referred to as "Q" data. Flow field data are obtained from the solutions on each side of the interface. The Roe scheme is used to evaluate the boundary fluxes that are applied to both solutions. The resulting data transfer conserves mass, momentum, and energy.

Identical geometry triangulation is required for both domains so that a common set of areas and orderings can be assumed. The resolution or topology of this triangulation is not restricted for

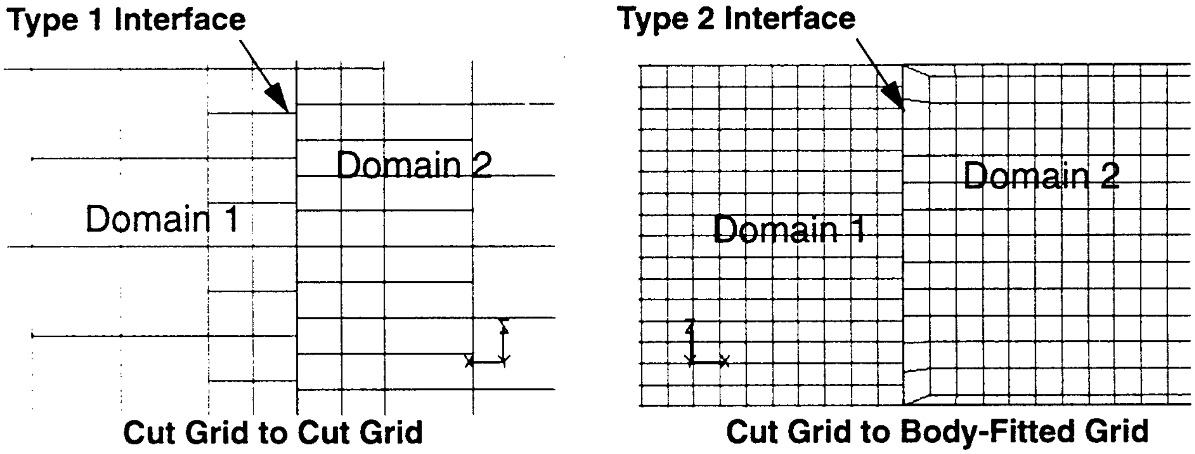

Figure 2.5 Two interface types are available in Splitflow

the type 1 interface but resolution should be on the same order as the associated boundary cells. For a type 2 interface, the triangulation is generated by connecting opposite corners of the body-fitted grid quadrilateral elements and using that on the cut grid side. Interfacing with a tetrahedral grid could be done without modifying the surface mesh.

The sequence for a type 1 interface on Domain 1 (Figure 2.6) is as follows:

1. Domain 1 collects local boundary solution to the geometry facets.
2. Local geometry facet solutions are exported to Domain 2.
3. 3) Geometry facet data is imported from Domain 2.
4. The Roe scheme is applied to each geometry facet using the solutions from both sides of the interface.
5. The resulting fluxes are applied to both domains.

The same sequence takes place on Domain 2 so that the same Q and flux information is applied to both sides of the boundary.

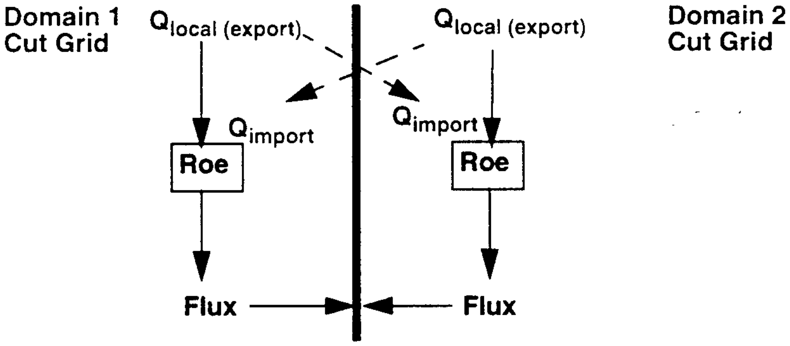

Figure 2.6 Type 1 interface computation and transfer sequence

The sequence for a type 2 interface (Figure 2.7) is as follows:

1. Domains 1 and 2 collect local boundary solution to the geometry facets.
2. Domain 2 exports local boundary solution to Domain 1.
3. Domain 1 averages the solutions (Q_{bound}) and applies Roe scheme to obtain fluxes.
4. Boundary Q data and fluxes are applied in Domain 1

5. Boundary Q and flux data are transferred to Domain 2 ($Q_{bound,}$ Flux)

6. Boundary Q and flux data are applied in Domain 2

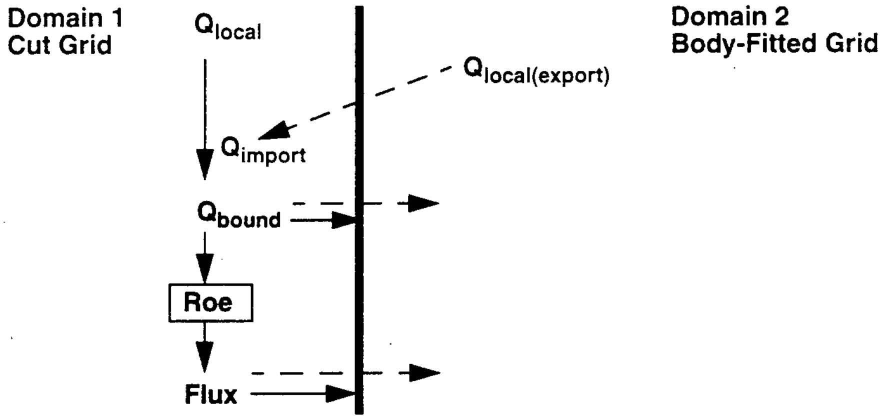

Figure 2.7 Type 2 interface computation and transfer sequence

Data can be passed in one direction as well. This amounts to skipping step 2 and using Q_1 data on both sides of the Roe scheme. A similar method can also be used to get data from Domain 1 at some point in time to be used as a non-uniform fixed boundary input for Domain 2. The "outflowing" domain can also apply another boundary condition in conjunction with the interface. For example, an extrapolation boundary condition could be applied and those Q data would also be transmitted, through the interface, to another domain.

Synchronization can be used to keep both solutions advancing in time at the same rate. This is done on the Iteration level for local time stepping (time warped) and on the sub-iteration level for global time stepping (time accurate). Rotations can also be applied to the momentum components at the interface to account for arbitrary relative grid orientations. This might be used for a case where one grid is moving (with rotations) through another.

The communication path for these interfaces is a file that is accessible to both domains. A portable, self-describing, object-oriented file format is used which was developed by LM Aero. Such an object-oriented approach allows for a theoretically infinite number of interfaces, although file input/output (I/O) times, etc. impose a practical limit. On start-up, Splitflow calculates interface sizes and allocates the necessary memory. The first time communication is attempted, existence of the interface file is checked. If no file exists, one is created with an (I/O) blocking object and interface objects are added as needed. The I/O blocking object includes a flag which gets turned on whenever a process is writing data into the file. This allows simultaneous reading but not simultaneous writing or reading while another process is writing, since the latter two situations could result in corrupted data. In the future a message passing protocol could be used in place of the interface file method.

Simple Cavity Application

For the simple cavity calculation, two interfaces were used. One interface was used at the outflow of the channel domain to "copy" data to a transition domain. The other linked the transition domain to the cavity domain (Figure 2.8). Since the channel and transition domains resided in the same grid and were run as one instance of the Splitflow solver, no synchronization was necessary. The synchronization feature was used for the type 2 interface to keep the two solutions at the same time level.

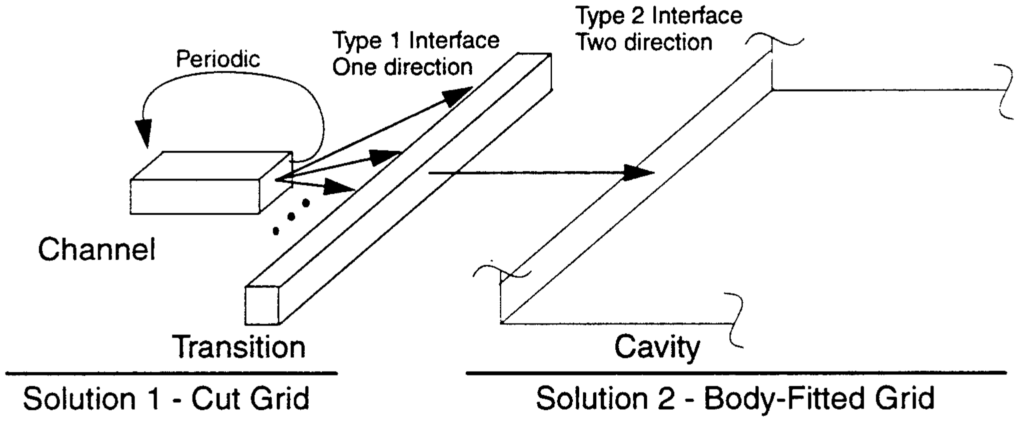

Figure 2.8 Simple cavity grid and interface topology

F-22 Cavity Application

To produce an appropriate incoming boundary layer for the F-22 main weapons bay, a body-fitted grid was generated for the part of the F-22 forebody. This forebody grid communicated with a cut grid defining the rest of the aircraft through an irregularly-shaped interface boundary. Figure 2.9 shows the aircraft surface boundaries included in each of the domains - gray in the cut grid and cyan in the body-fitted grid. In Figure 2.10 the interface boundary is shown. A

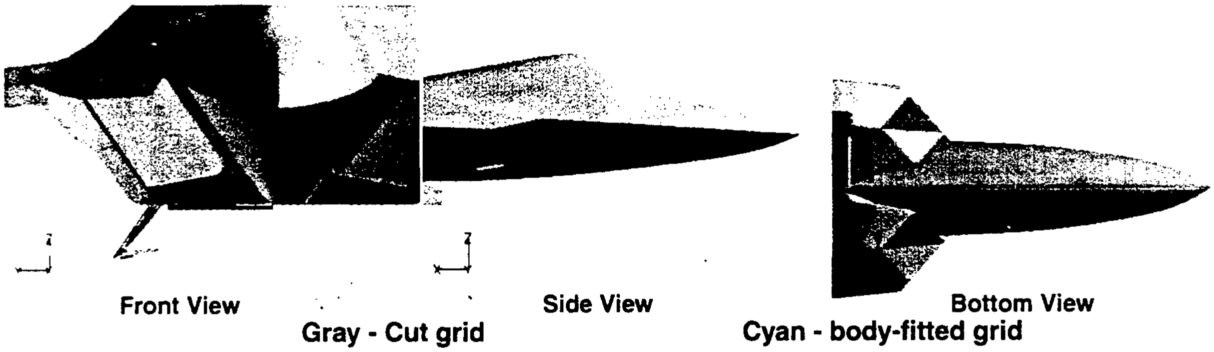

Figure 2.9 Aircraft surfaces solved in each domain

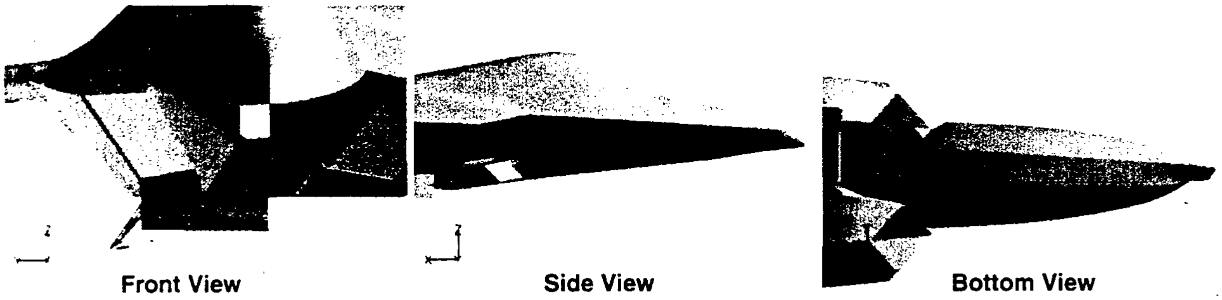

Figure 2.10 The forebody interface

two-direction, type 2 interface was applied on this boundary to pass Q and flux data between the grids (Figure 2.11 and Figure 2.12). The two grids were run as separate instances of the Splitflow solver and synchronized on the sub-iteration level.

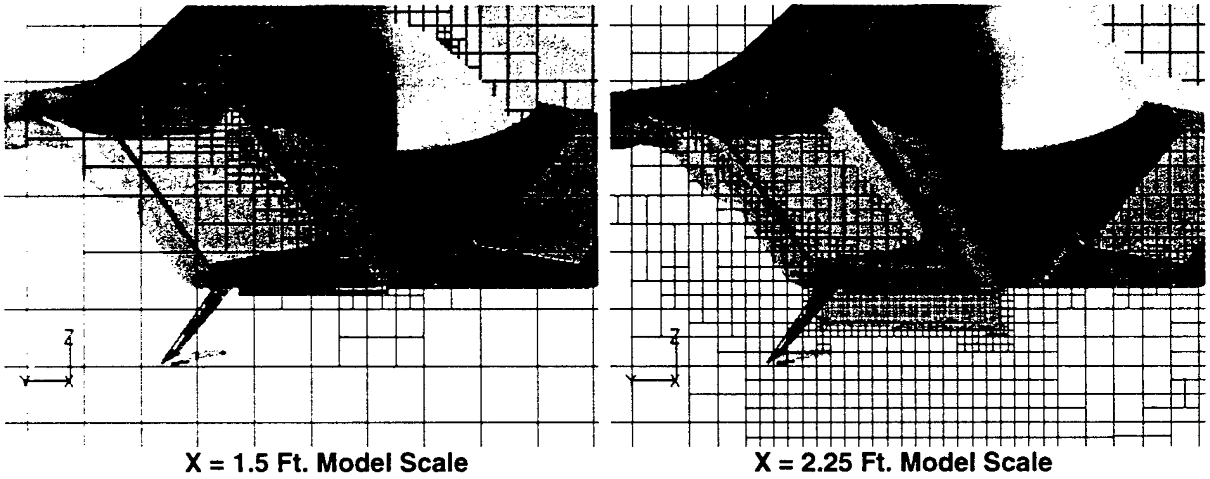

Figure 2.11 Fuselage station cuts through both grids

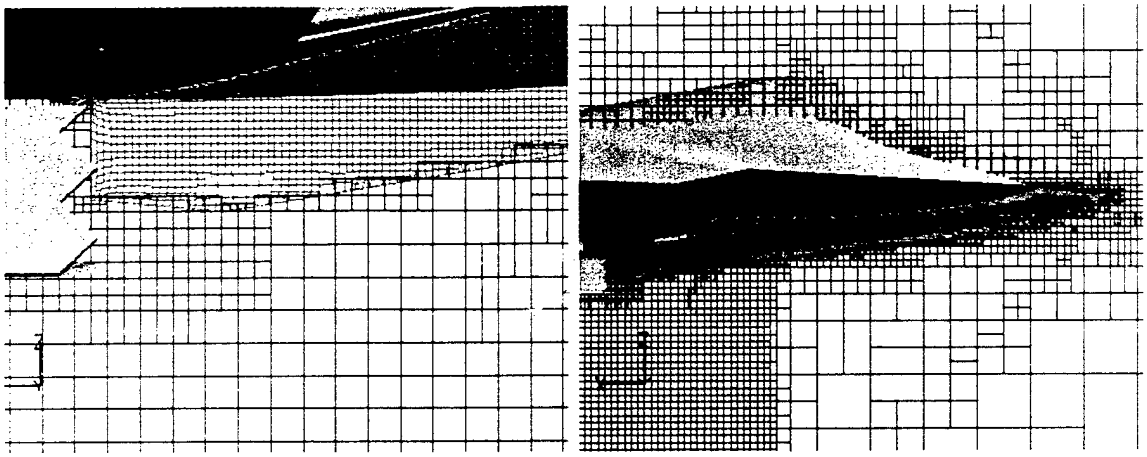

Figure 2.12 Butt line cut through both grids with close-up side view

2.5 Six Degree of Freedom Methods

Following the addition of time-accurate body motion in Splitflow, six degree-of-freedom (6DOF) capability was implemented. Constant-rate motion was first introduced and tested to show that the grid speed terms were properly implemented. Having accomplished this, body motion capability was expanded to include arbitrary motion. Two types of arbitrary motion are allowed for: 1) prescribed and 2) 6DOF. A restraint module is also available in the 6DOF routines which can be used to reduce degrees of freedom as desired.

Assumptions and Definitions

The 6DOF routines in Splitflow use the Euler equations of motion as described in *Dynamics of Flight: Stability and Control*, second edition, by Bernard Etkin, P. 88.[4] From inertial properties, forces and moments, and initial linear and rotational velocities, linear and rotational accelerations can be computed. An integration in time produces linear and rotational velocities. Another integration in time produces position and orientation.

Store Separation

The 6DOF routines implemented in Splitflow were originally developed for store separation simulation. As a result, many of the conventions used are those which are common to store separation analyses. Axis system conventions (described in a following section) are one of the more important outcomes of this approach in that it becomes important for the user to understand them to properly set up a simulation. The inclusion of aircraft maneuver (restricted to the pitch plane, i.e. pull-up and push over) and ejector modeling are other features useful for store separation that may not be of value in more generalized use.

Axis System Convention

Aircraft geometry is typically defined in a fuselage station, butt line, water line system (upper left in Figure 2.13). The convention commonly used for store separation analysis is called a body axis system. This system, along with the corresponding force and moment coefficient definitions, is shown on the right in Figure 2.13. Transforming from a fuselage station, butt line, water line system to a body system is simply a matter of rotating 180 degrees about the Y axis. Origins of either of these systems are typically located forward and below the nose of the aircraft on centerline or at the aircraft center of gravity (CG). A pylon axis system (lower left in Figure 2.13) is defined by a store body axis system at the carriage position. Most trajectory data are provided in a pylon system.

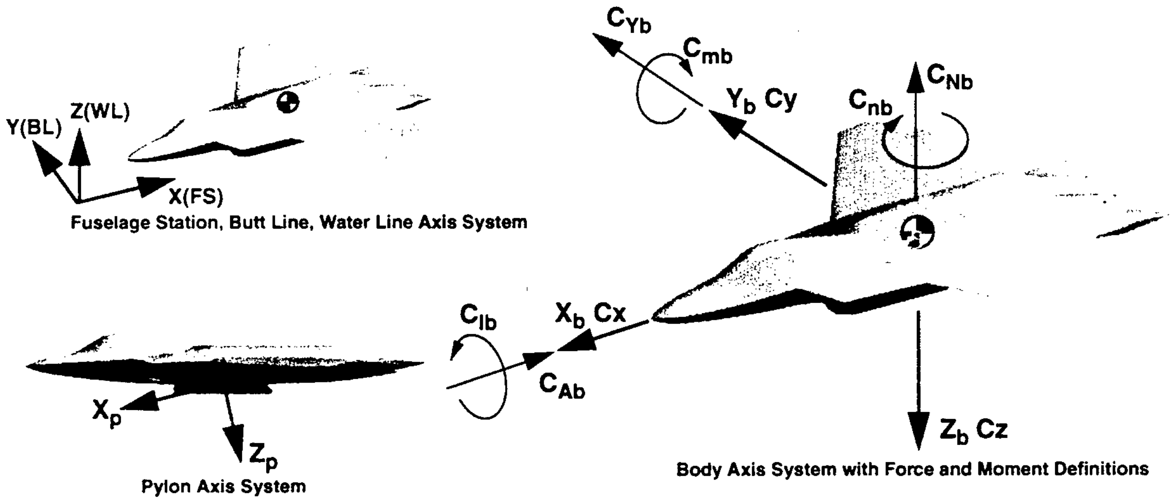

Figure 2.13 6DOF axis system conventions

Integration Scheme

A trapezoidal scheme is used to integrate accelerations from the Euler equations of motion to obtain velocities. Another integration is performed to produce position and orientation. Higher order integration schemes have been investigated in the past as part of an effort to reduce trajectory error. The result of the study was that, since these routines are not computationally intensive, simply reducing the integration time step was the easiest and most effective means of improving accuracy. As implemented in Splitflow, the integration time step is independent of the solution time step so an integration time step one to two orders of magnitude smaller than the solution time step is recommended (something on the order of $1.0\text{x}10^{-6}$ or 1.0x10-7 seconds per integration step)

Models and Initial Conditions

Aircraft maneuver modeling is limited to the pitch plane (pull-up and push-over). Angular velocity of the aircraft is computed from:

$$Q = \frac{(n-1)g}{V} \tag{2}$$

where Q is angular velocity in the Y-direction, n is the load factor, g is acceleration due to gravity, and V is aircraft velocity tangent to the maneuver radius. A load factor of one corresponds to straight and level flight (no maneuver), a load factor greater than one corresponds to a pull-up maneuver, and a load factor less than one corresponds to a push-over maneuver.

Two store ejectors can be modeled by supplying ejector positions, end of stroke (EOS) length, and a force-time history of the ejector. Typical ejectors produce a somewhat "sawtooth" force-time history over about 60 milliseconds with enough force to accelerate the store to ~30 feet per second at EOS. Often the forward ejector force is higher to produce a nose-down angular velocity. An alternative to ejector modeling is setting initial conditions, such as EOS linear and angular velocities.

6DOF Code Flow Chart

```
SOLVE
  Call MOVE_BODIES
    •Solve flow field for current iteration
    •Sum aerodynamic loads on the body
    •Append current body position, orientation, and aerodynamic loads to trajectory history file
```

```
END

MOVE_BODIES
        •Collect boundaries acting as one arbitrarily moving body
        •Sum aerodynamic forces and moments on the body

    Call ARB_MOTION

        •Initialize newly uncovered cells
        •Move geometry
        •Recut grid
        •Update moment center

RETURN

ARB_MOTION
        •Read input file
        •Attempt to open trajectory history file and read trajectory data
        •Scale body loads
        •Transform body loads into the inertial axis system
        •Transform body loads into the body axis system
        •Establish end time

    Call SIXDOF_INTEG

        •Scale new body position, orientation, and rotational rates

RETURN

SIXDOF_INTEG
        •Set up aircraft motion terms (maneuver in pitch plane)
        •Set up store motion terms as a result of aircraft maneuver
        •Start integration loop
                ••Interpolate store loads in Z
                ••Interpolate other inputs in time (ejector, etc.)
```

••Formulate acceleration terms

••Integrate to get velocities

••Integrate to get position and orientation

••Transform results into inertial axis system

••Record results to output files

••Check for stop condition

RETURN

2.6 LES Modeling and Implementation

LES Model: The model is a compressible form of the Smagorinsky model.[7] By applying a "filter function" to the Navier-Stokes equations, the LES form of the flow equations can be derived. The filter is a grid dependent integral of the flow variables that eliminates variations in the flow quantities that are smaller than can be resolved by the computational mesh. A filtered variable, $\bar{f}$, is the spatial average using a filter function, G, whose integral at any **x** is unity. Favre averaging is utilized inside the filtering operation to account for density variations, which gives the mass averaged, filtered variable $\tilde{f}$.

$$\tilde{f}(\boldsymbol{x}) = \frac{1}{\bar{\rho}}\int \rho f(\boldsymbol{x}) G(\boldsymbol{x} - \boldsymbol{y}) d\boldsymbol{y} \tag{3}$$

For finite volume applications, the "top hat" filter is typically used, where:

$$G(x_i) = \begin{cases} 1/(\Delta x_i) & if(|x_i| < \Delta x_i/2 \\ 0 & if(|x_i| > \Delta x_i/2 \end{cases} \tag{4}$$

and x_i is the component of the grid spacing in computational space, and Δx_i is the cell spacing in the i-direction. Applying the filter function to the continuity, momentum and energy equations gives:

$$\frac{\partial \bar{\rho}}{\partial t} + \frac{\partial}{\partial x_i}(\bar{\rho}\tilde{u}_i) = 0 \tag{5}$$

$$\frac{\partial\bar{\rho}\tilde{u}_i}{\partial t}+\frac{\partial}{\partial x_j}(\bar{\rho}\tilde{u}_i\tilde{u}_j)+\frac{\partial}{\partial x_i}P-\frac{\partial\tilde{\tau}_{ij}}{\partial x_j}+\frac{\partial\sigma_{ij}}{\partial x_j} = 0 \qquad (6)$$

The subgrid scale momentum stress is $\sigma_{ij} = \widetilde{\rho u_i u_j}-\bar{\rho}\tilde{u}_i\tilde{u}_j$

$$\frac{\partial\bar{\rho}\tilde{E}_t}{\partial t}+\frac{\partial}{\partial x_i}(\bar{\rho}\tilde{E}_t\tilde{u}_i+P\tilde{u}_i)-\frac{\partial\tilde{u}_i\tilde{\tau}_{ij}}{\partial x_j}+\frac{\partial}{\partial x_i}\bar{q}_i+\left(\frac{\partial\varphi_i}{\partial x_i}+\frac{\partial\psi_i}{\partial x_i}+\frac{\partial\theta_i}{\partial x_i}\right) = 0 \qquad (7)$$

The subgrid scale diffusion terms are:

$$\begin{aligned}\varphi_i &= \widetilde{u_i\tau_{ij}}-\tilde{u}_i\tilde{\tau}_{ij}\\ \psi_i &= \frac{1}{2}(\widetilde{\rho u_i u_k u_k}-\bar{\rho}\tilde{u}_i\tilde{u}_k\tilde{u}_k)\\ \theta_i &= \widetilde{(\rho e+P)u_i}-(\bar{\rho}\tilde{e}+\tilde{P})\tilde{u}_i\end{aligned} \qquad (8)$$

The subgrid scale stress term is modeled as in Smagorinsky with a simple Bousinesq approximation.

$$\sigma_{ij} = -\mu_t\left(2S_{ij}-\frac{2}{3}S_{kk}\delta_{ij}\right) \qquad (9)$$

where

$$S_{ij} = \frac{1}{2}\left(\frac{\partial\tilde{u}_j}{\partial x_i}+\frac{\partial\tilde{u}_i}{\partial x_j}\right) \qquad (10)$$

The eddy viscosity is:

$$\mu_t = \bar{\rho}l^2|S| \qquad |S| = \sqrt{2S_{ij}S_{ij}} \qquad (11)$$

Away from no slip walls the mixing length, l, is simply a function of the grid scale, Δ, and the Smagorinsky constant, C_s, $l = C_s\Delta$. The constant C_s is set to 0.1 for this work. A simple model is also used for the subgrid scale heat flux.

$$\theta_i = \frac{-\mu_t c_p}{Pr_t} \frac{\partial}{\partial x_i} \tilde{T} \qquad (12)$$

Where c_p is the specific heat at constant pressure and we set the turbulent Prandtl number, Pr_t = 0.9. The final model term for the subgrid scale viscous work in the energy equation is:

$$\psi_i + \varphi_i = \tilde{u}_j \sigma_{ij} \qquad (13)$$

Given the value of the filter length scale, l, the continuity, momentum and energy equations, Equations 5, 6 and 7 are closed with the model terms, Equations 9, 12 and 13.

Filter Length for Splitflow Grid LES modeling requires gradual changes in the filter length throughout the flowfield in order to be able to simulate large turbulent structures. In LES, the filter length is generally determined as a function of the local cell volume:

$$\Delta = (cell\ volume)^{1/3} \qquad (14)$$

Unlike a structured grid, the a Splitflow mesh can have neighboring cells with large disparities in cell size due to cell cutting by geometry surfaces or grid refinement. The volumes of two uncut cells sharing a common face will never differ by more than a factor of eight. Rapid changes in filter length are inconsistent with sub grid scale approximation. The small scales of turbulence will not be produced or simulated well in a limited region of the flow field with a fine grid and small filter lengths surrounded by coarser grids and larger filter lengths. Only larger scale structures will be convected through the region of reduced scales; there is insufficient time for the smaller scale turbulence to develop. If the filter length is not smoothed, the turbulent viscosity will be much smaller in the fine grid region. This can cause "kinks" in mean velocity profiles obtained from averaged LES solutions. Conversely, if the filter length in local regions of high grid refinement is much larger than the local grid scale, but roughly the same magnitude as adjacent regions with coarser grids, the numerical accuracy of the solution is improved in the region of highly refined grid, and the turbulent viscosity varies smoothly throughout this region. For this reason a smoothing scheme that increases the filter length smoothly across regions with rapid changes in cell volume was devised.

For each sweep of the smoothing function, the maxima of the filter length of each cell and the filter length of its neighboring cells are averaged. For each smoothing sweep, the filter length for cell at c, Δ_c is:

$$\Delta_c^{p+1} = \frac{1}{N}\sum_{n=1}^{N} maximum(\Delta_c^p, \Delta_n^p) \tag{15}$$

where, N is the number of neighbors. For all of the results presented here, three smoothing passes were employed. With this approach a reasonably smooth filter length distribution was obtained.

The Smagorinsky constant is reduced near no slip wall boundaries using van Driest damping. In addition, the filter length is not allowed to exceed the product of the distance from the wall and von Karman constant, κ. Most LES simulations with Smagorinsky's model do not make this restriction, resulting in poor near wall results when large spanwise and streamwise grid spacings cause the filter length to exceed κy outside the viscous sublayer. The length scale is:

$$l = min(C_s\Delta, \kappa Dy) \qquad D = 1 - exp\left(\frac{-yu_s\rho}{\mu A^+}\right) \tag{16}$$

The constant A^+ is set to 25.0. The friction velocity, u_s is calculated from a combination of local conditions and wall conditions.

$$u_s = \left\{\max\left[\frac{|\tau_w|}{\rho_w}, \left(\frac{\mu}{\rho} + l_{n-1}{}^2|S|\right)|S|\right]\right\}^{1/2} \tag{17}$$

The nearest boundary facet is identified for each cell. The wall shear stress associated with this boundary facet is applied in Equation 17. The length scale, l_{n-1} is lagged from the previous iteration. The second term in Equation 17 prevents the damping term from going to zero away from the wall when the instantaneous wall shear stress is zero. This is reasonable since we expect the near wall damping to reduce turbulent viscosity when the size of sub grid scale turbulent structures is greater than the distance from the wall. When the wall shear stress is zero, velocity gradients and subgrid scale stresses away from the wall are still significant and should be damped.

Wall function boundary condition: The wall function boundary condition for the LES model implemented in Splitflow solves for a velocity profile that matches the velocity at the cell center one point away from the wall and the shear stress at the face of the grid cell one grid cell height away from the wall. This profile is used to obtain the wall shear stress, which is used in the viscous flux routine. The wall function greatly relaxes near wall grid resolution requirements. The LES wall function is more sophisticated than the law of the wall because the velocity profile conforms to the instantaneous shear stress profile between the wall and the first grid point off of the wall, while the law of the wall assumes constant shear stress.

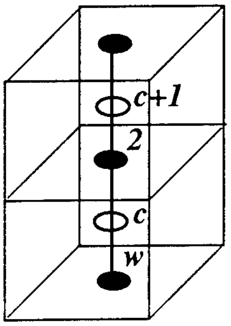

Figure 2.14 Notation for wall function grid locations.

A diagram of the wall function locations is shown in Figure 2.14. The velocity profile is integrated from the relation:

$$\frac{du}{dy} = \frac{1}{\mu_t}\tau \qquad \tau = \tau_w + y\frac{\tau_2 - \tau_w}{\Delta y} \tag{18}$$

The velocity, u, is the local velocity direction tangential to the wall, and y is normal to the wall. The turbulent viscosity is an algebraic function of the local shear stress and the length scale.

$$\mu_t = \kappa y \rho |\tau|^{1/2} D \tag{19}$$

The near wall damping D, is a function of a friction velocity obtained from the local shear stress, Equation 18, $u_s = |\tau/\rho|^{1/2}$, and the damping function of Equation 16.

The wall function is evaluated using the secant approximation to Newton's method to solve for τ_w. The iteration continues at each boundary point until a wall shear stress found that gives a velocity at the cell center that matches the current cell center velocity in the finite volume solution. At each finite volume iteration one or two secant iterations are normally required for acceptable convergence. Each secant iteration step requires numerical integration of Equation 18, which is performed on a one-dimensional mesh with about 20 mesh points. Use of the wall function iteration reduces the near wall spacing requirements by roughly an order of magnitude in an LES solution. In addition, a coarser near wall mesh may allow larger computational time steps. However, because the wall function does not resolve the fine scale eddies in near the wall, it should not be used where a high fidelity LES boundary layer solution is required.

2.7 Numerical Diffusion and Limiting

LES predictions are extremely sensitive to numerical diffusion. The implicit numerical diffusion in lower order flux schemes can be of the same order as the sub grid scale diffusion from the LES equations. The weapons bay problem is particularly challenging for LES because of the strong moving shocks responsible for the acoustic phenomena in the bay. Numerical stability for flows with strong shocks typically requires the application of upwind schemes with flux limiters. Unfortunately, the upwind schemes typically include significant numerical diffusion. For this program, several approaches were investigated for accurate compressible flow simulations. A third order scheme was implemented in parallel Splitflow. Two hybrid approaches for reducing the implicit diffusion of Roe's scheme were developed and tested. Implementation of an Essentially Non-Oscillatory method was investigated for Splitflow

Third order scheme: A third order accurate, upwind discretization of the convective flux terms was implemented in Splitflow. This scheme includes a flux limiter that makes the scheme Total Variation Diminishing (TVD). While the limiter helps to prevent spurious oscillations, particularly in flows with shock waves, it adds implicit diffusion to the solution. Fluxes are constructed at each grid face in the flow field by extrapolating the primitive variables, (p, u, v, w, T) from cell centers to cell faces to obtain right and left extrapolated values at each face. For regions of the flowfield where the mesh is regularly spaced, the Splitflow and structured mesh results are identical. Notation for the primitive variables and faces is shown in Figure 2.15. For an unlimited flux, the third order scheme is:

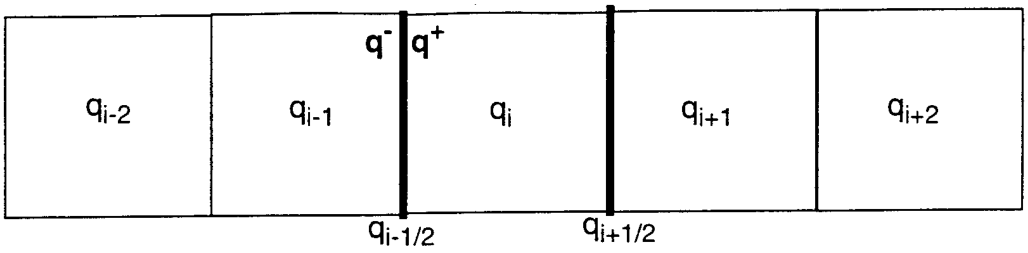

Figure 2.15 Cell notation for flux descriptions.

$$
\begin{aligned}
q^-_{i-\frac{1}{2}} &= q_{i-1} + \frac{1}{3}(q_{i-1} - q_{i-2}) + \frac{1}{6}(q_i - q_{i-1}) \\
q^+_{i-\frac{1}{2}} &= q_i - \frac{1}{3}(q_{i+1} - q_i) - \frac{1}{6}(q_i - q_{i-1})
\end{aligned}
\tag{20}
$$

Limiting is included in this scheme with a compression parameter, *c*:

$$
\begin{aligned}
q^-_{i-\frac{1}{2}} &= q_{i-1} + \frac{1}{3} minmod[(q_{i-1} - q_{i-2}), c(q_i - q_{i-1})] \\
&+ \frac{1}{6} minmod[c(q_{i-1} - q_{i-2}), (q_i - q_{i-1})]
\end{aligned}
\tag{21}
$$

The compression factor should be set between 1 and 4. Setting the factor to 4 is the most compressive option that maintains the TVD property, resulting in the smallest truncation error in the solution. Setting the compression factor to 1 reduces this scheme to a second order "minmod" scheme. The minmod function is the standard definition, where

$$
minmod(a, b) = \begin{array}{c} a, if(|a| \geq |b|, ab > 0) \\ b, if(|a| < |b|, ab > 0) \\ 0, if(ab < 0) \end{array}
\tag{22}
$$

Construction of the fluxes on the i+1/2 face is similar. Construction of the fluxes is not as simple in a Splitflow mesh as it is in a structured grid mesh because the cell face for one cell can be split into multiple faces for neighboring cells, as shown in Figure 2.16. For a one dimensional structured mesh, evaluation of 3rd order upwind flux at a face requires a four point stencil with two cells on either side of each face. Since the center of a Splitflow cell is no longer aligned on a nor-

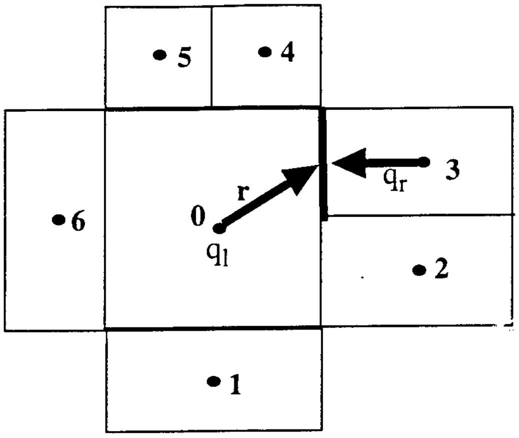

Figure 2.16 Primitive variable extrapolation scheme considers tangential components

mal to the face center, gradients in direction parallel to cell faces must be accounted for in flux extrapolation. Cell center gradients of the primitive variables are evaluated using Gauss theorem,

$$\nabla q = \frac{1}{\Omega}\sum_{m} A(q_{0-m})_{interp} \tag{23}$$

where A is the outward facing normal vector, Ω is the cell volume, and $(q_{0\text{-}m})_{interp}$ is the value of q interpolated between the cell "0" and the cell m. The sum of the dot product of the vector **r** and ∇q gives a second order accurate value of the primitive variable at the cell face. To use the upwind formulation of Equation 21, an extrapolation representative of the difference in q between cells 0 and 6, and an interpolation representative of the difference in q between cells 0 and 3 must be determined. By subtracting the contribution of the interpolated difference between cells 0 and 3 from the cell centered gradient term, the extrapolated difference is derived. Equation 21 is then applied to these differences to obtain the right and left primitive variables at the face. This method gives third order accuracy for regions of the grid with equally spaced cells. For the portions of the mesh with varying spacing, the formal accuracy reverts to second order. The truncation error in regions of the grid with variations in the cell size has a smaller truncation error than either a fully upwind extrapolation or a central difference extrapolation.

Roe Scheme Limiting. Fluxes at cell faces are obtained from the left and right extrapolations of the primitive variables.

$$F_i(Q) = \frac{1}{2}[F_i(Q_L) + F_i(Q_R) - \omega|\tilde{A}|(Q_R - Q_L)] \tag{24}$$

where $\mathbf{Q}_L$ and $\mathbf{Q}_R$ are the flow variables from the cells to the left and the right of the face, and $A = (\partial F/\partial Q)$. The Roe averaged state, $\tilde{A}$ is obtained by evaluating A using Roe averaged values from the left and right states.[5] The matrix $|\tilde{A}|$ is obtained by pre- and post-multiplying the absolute value of the diagonal matrix of eigenvalues of A by the diagonalizing matrices obtained from the eigenvectors of A. The coefficient ω, the "Roe Factor", controls the amount of Roe scheme based diffusion to be included in the simulation. For Roe scheme, the factor $\omega = 1$, while for central differencing, $\omega = 0$. In the "Large Eddy Simulation for Cavity Flow Analysis" program, the Roe flux difference splitting technique was found excessively diffusive for a low speed, Mach 0.2, channel flow simulation. The matrix $|\tilde{A}|$ represents a matrix of smoothing coefficients for the Euler equations. Because some of the coefficients are functions of the speed of sound, these coefficients can dominate the diffusion for low speed flow. The goal for the current program was to assess the effectiveness of Roe scheme for a higher Mach number flow, and modify Roe scheme as required for transonic and supersonic applications.

Higher speed flow simulations should be less sensitive to Roe scheme because the damping coefficients in the Roe matrix proportional to the speed of sound are similar in magnitude to the flow velocity in this case. A Mach 1.2 channel flow simulation was performed for this program. From multiple tests, it was found that while some unsteady turbulent structures could be obtained with $\omega = 1$, higher, more realistic levels of resolved turbulence were obtained for $\omega = 0.2$. In addition, use of the flux limiter proved necessary to obtaining a stable flow. The compression factor was set to 4.0 for the channel flow simulation, the most compressive setting. In contrast, the low speed solution required $\omega = 0.1$ to obtain reasonable turbulence levels, and the flux limiter was turned off. Table 2.1 shows the differences in the settings between the low speed solution obtained with the hybrid version of Splitflow and the higher speed solution obtained with parallel Splitflow.

Table 2.1 Flux scheme coefficient settings for low speed and transonic simulations

Simulation Mach Number	Roe Factor	Limiting Compression Factor
0.2	0.1	No limiting
1.2	0.2	4.0

As expected, the higher speed solutions could tolerate, and in fact required, larger Roe coefficients and flux limiting than the low speed solutions.

An alternative approach for Roe Scheme limiting was developed. In this scheme, instead of limiting the entire Roe Scheme diffusion matrix, only one speed of sound related term was limited. The product of the Roe diffusion matrix is and the primitive variable differences, $\begin{bmatrix}\Delta\rho & \Delta u & \Delta v & \Delta w & \Delta p\end{bmatrix}$ is:

$$|\Delta F_{2,3,4}| = |\tilde{U}_c|\left\{\left(\Delta\rho - \frac{\Delta p}{\tilde{a}^2}\right)\begin{bmatrix} 1 \\ \tilde{u} \\ \tilde{v} \\ \tilde{w} \\ \dfrac{\tilde{u}^2 + \tilde{v}^2 + \tilde{w}^2}{2} \end{bmatrix} + \tilde{\rho}\begin{bmatrix} 0 \\ \Delta u - \eta_x \Delta U_c \\ \Delta v - \eta_y \Delta U_c \\ \Delta w - \eta_z \Delta U_c \\ \tilde{u}\Delta u + \tilde{v}\Delta v + \tilde{w}\Delta w - \tilde{U}_c \Delta U_c \end{bmatrix}\right\}, \qquad (25)$$

$$|\Delta F_{1,5}| = |\tilde{U}_c \pm \tilde{a}| \frac{[\Delta p \pm \tilde{\rho}\tilde{a}(\eta_x \Delta u + \eta_y \Delta v + \eta_z \Delta w)]}{2\tilde{a}^2}\begin{bmatrix} 1 \\ \tilde{u} \pm \beta\eta_x\tilde{a} \\ \tilde{v} \pm \beta\eta_y\tilde{a} \\ \tilde{w} \pm \beta\eta_z\tilde{a} \\ \tilde{h}_t \pm \tilde{U}_c\tilde{a} \end{bmatrix}$$

where $U_c = u\eta_x + v\eta_y + w\eta_z$ is the contravarient component of velocity. For the standard Roe scheme, $\beta = 1$. The term for which β is a coefficient is proportional to the speed of sound. By limiting this coefficient, Roe's scheme can be less diffusive for LES while reducing spurious flow oscillations at shock-waves.

Roe's scheme is based on the Euler equations in one dimension. When the flow direction is close to, but not exactly parallel with a cell face and a shear strain is present in the direction normal to the face, the $|\Delta F_{1,5}|$ term of the Roe diffusion vector can have a significant effect on the shear stress. For LES simulations, this term adds excessive damping to shear and vortical regions of the flow domain. Conversely, when the flow direction is nearly normal to the cell face, differences in the velocity across the face is primarily an inviscid phenomenon and the Roe scheme based on the Euler equations is appropriate. For the modified scheme, β is set to a function that reduces the diffusion magnitude when the flow direction is not aligned with the face normal.

$$\beta = maximum\left(\sigma, \frac{|U_c|}{\sqrt{u^2 + v^2 + w^2}}\right) \tag{26}$$

The constant σ was iound to be necessary to maintain some of the effects of this smoothing coefficient even when the flow nearly parallel to the cell face. Typically, we set $\sigma = 0.1$. Results obtained with this model will be discussed in Section 3.

ENO scheme evaluation: Recently, Essentially Non-Oscillatory (ENO) and Weighted ENO (WENO) schemes[6] have been used for finite difference and finite volume LES applications. ENO schemes provide a procedure for obtaining high order accuracy for piecewise smooth problems with discontinuities. An ENO scheme could be used in place of the extrapolation and limiting procedures specified in Equations 20, 21, and 22. A Roe flux difference split method can be used to determine the flux. The ENO procedure can provide higher order expressions for unevenly spaced structured grids. As part of this program, the implementation of a third order ENO scheme in Splitflow was investigated.

A third order, upwind ENO scheme requires information from three cells on each side of a flux face. By comparison, the Splitflow third order scheme requires two cells on each side of a face. For an unstructured grid with split cells such as Splitflow, expansion of the flux stencil adds greatly to the computational complexity of the scheme. A method to implement an ENO scheme in Splitflow was developed. This scheme uses the cell center gradients at two cells on either side of the cell face to construct the ENO differences. In the logical structure of Splitflow only cells touching a cell face can be readily identified. For the current third order scheme, flux evaluation is a two step process, where cell center gradients are determined using one pass through all cell faces, and the extrapolations and fluxes are obtained by a second pass through all cell faces in the flow domain. While a method for identification of the additional cell center information required for ENO was developed, the method was unwieldy for parallel implementation. The difficulties arose from the need to pass excessive amounts of information large numbers of times across boundaries between the flow domains on each Splitflow parallel node. A practical method for implementation of and ENO scheme in Splitflow would require a total rewrite of the Splitflow data structure. This is a task that would require many man months to complete, so this approach to improve numerical accuracy was not pursued further. However, it does appear that implementation of an ENO scheme is feasible, and results from the literature indicate that implementation of an ENO scheme could benefit accuracy and stability of Splitflow for LES applications.

2.8 Periodic Boundary Conditions

For channel and cavity flow studies, periodic boundary conditions allow small computational domains to be used for the efficient development of turbulent boundary layers. Periodic boundary conditions allow the turbulent structures to be passed across the boundaries of the flow domain. Two types of periodic boundaries were required for this research, a lateral periodic boundary condition and an inflow - outflow periodic condition. The inflow - outflow periodic boundary condition used in this research underwent significant modifications compared to the boundary condition used in the "LES for Cavity Flow Analysis" program. In the earlier research, the boundary condition was applied to a Mach 0.2 channel flow. This formulation did not work well for the Mach 1.2 channel flow in the current research. The new formulation worked well.

Periodic Boundary Condition Flux Calculation

For cut grids, periodic boundaries are defined by the triangular "geometry facets". The geometry facets on each side of the periodic boundary should match exactly in size, shape and orientation. The geometry facets are cut on each side into "boundary facets" by the Cartesian cell faces. Flow variables are accumulated using an area weighted procedure from the boundary facets to their corresponding geometry facets from each side of the flow domain. Fluxes are then calculated on the geometry facets. The fluxes from the geometry facets are then distributed to the appropriate cells using the boundary facets. Figure 2.17 shows the relationship between geometry facets, cell faces and boundary facets. The accuracy of the periodic boundary condition depends on the resolution of the geometry facets.

For body fitted Omnigrids, geometry facets are generated by triangulating the quadrilateral faces on the periodic boundary. As a result, each cell face is associated with two geometry facets

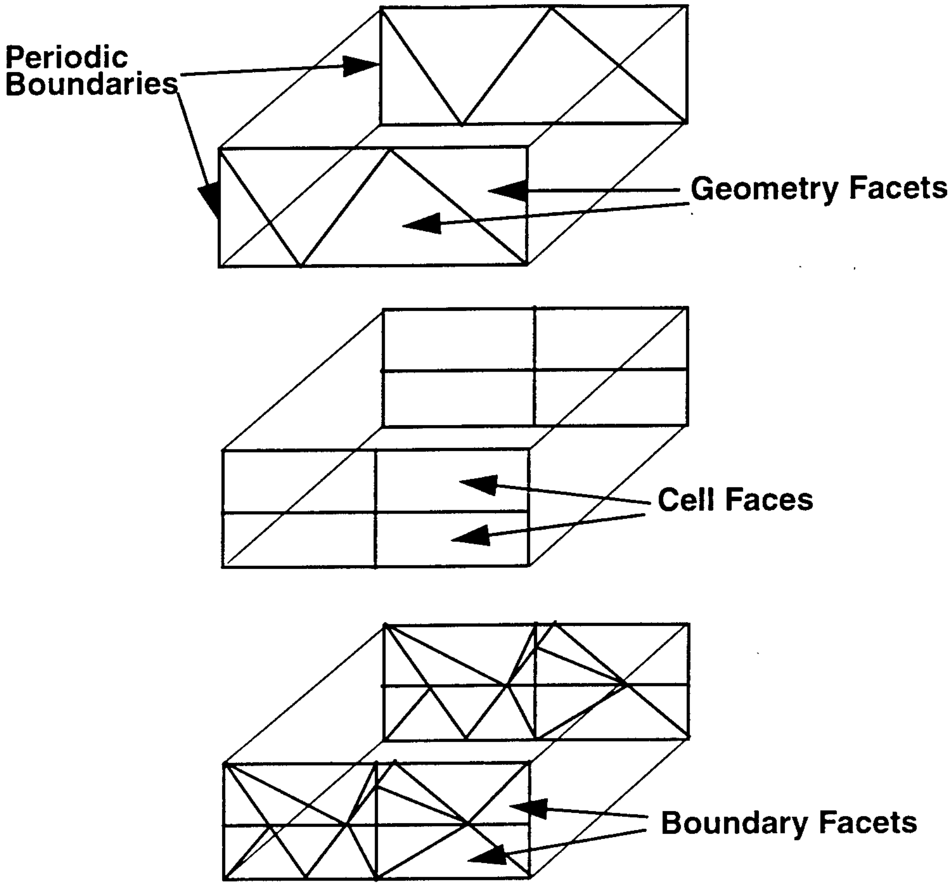

Figure 2.17 Periodic Boundary Facet Definitions

that fit perfectly within its face. As a result, the periodic boundary condition is more accurate for Omnigrid calculations.

Compressible Inflow - Outflow Periodic Boundary Conditions

For fully developed channel flow, jump conditions must be included between the periodic inflow and outflow boundaries. We hold mass flux and total energy flux constant at the inflow, and the average static pressure constant at the outflow. The flow variables from the outflow must be adjusted to be used at the inflow and the inflow must be adjusted for use at the outflow. A schematic that shows the notation for the periodic boundary condition derivation is shown in Figure 2.18. Surface integrals of inflow and outflow pressure and outflow mass flux and total energy flux are evaluated. The average pressure is:

$$I_0 = P_0 \qquad I_1 = \iint P_1 dy\, dz \qquad I_2 = \iint P_2 dy\, dz \tag{27}$$

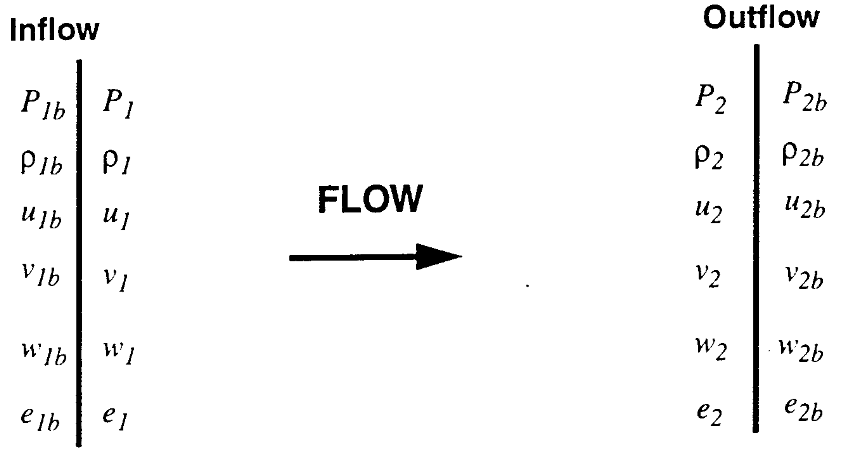

Figure 2.18 Diagram of notation for periodic inflow/outflow boundary condition.

The mass flux integrals are defined as:

$$J_0 = \rho_\infty U_\infty \qquad J_2 = \iint \rho_2 u_2 dy\, dz \tag{28}$$

The total energy integrals are defined as

$$K_0 = \rho_\infty u_\infty E_{T\infty} \qquad K_2 = \iint \rho_2 u_2 E_{T2} dy\, dz \tag{29}$$

The total energy, E_T, is the sum of the internal energy, e, and $\frac{1}{2}(u^2 + v^2 + w^2)$. For the LES for Weapons Bay Analysis program, total enthalpy was used instead of total energy. For the inflow conditions at each geometry facet, assuming the perfect gas equation of state, we have five unknowns, $\rho_{1b,}$, density, u_{1b}, v_{1b}, and w_{1b}, velocity components, and e_{1b}, so five equations in terms of the outflow variables and integrals are required. We assume $v_{1b} = v_2(u_{1b}/u_2)$ and $w_{1b} = w_2(u_{1b}/u_2)$. Then from the pressure, mass and total enthalpy equations, we obtain three additional equations,

$$\begin{aligned} P_{1b} &= P_2 + I_0 - I_2 \\ \rho_{1b} u_{1b} &= \rho_2 u_2 \frac{J_0}{J_2} \\ u_{1b}\left(\rho_{1b} e_{1b} + P_{1b} + \frac{1}{2}\rho_{1b} V_{1b}^2\right) &= u_2\left(\rho_2 e_2 + P_2 + \frac{1}{2}\rho_2 V_2^2\right)\frac{K_0}{K_2} \end{aligned} \tag{30}$$

where $V_{1b} = u_{1b}^2 + v_{1b}^2 + w_{1b}^2$ and $V_2 = u_2^2 + v_2^2 + w_2^2$. Substituting the first two equations from Equation 30 into the third equation, a quadratic equation for u_{1b} is obtained

$$\frac{\rho_2 V_2^2 J_0}{2u_2 J_2} u_{1b}^2 + \frac{\gamma}{\gamma - 1}(P_2 + I_0 - I_1)u_{1b} - \rho_2 u_2 H_{T2} \frac{K_0}{K_2} = 0 \tag{31}$$

After solving for u_{1b} we can then solve for the other variables.

A similar approach is used for the outflow periodic boundary. We assume $v_{2b} = v_1(u_{2b}/u_1)$ and $w_{2b} = w_1(u_{2b}/u_1)$, and obtain for the pressure, mass and total enthalpy equations:

$$\begin{aligned} P_{2b} &= P_1 + I_2 - I_1 \\ \rho_{2b} u_{2b} &= \rho_1 u_1 \frac{J_2}{J_0} \\ u_{2b}\left(\rho_{2b} e_{2b} + P_{2b} + \frac{1}{2}\rho_{2b} V_{2b}^2\right) &= u_1\left(\rho_1 e_1 + P_1 + \frac{1}{2}\rho_1 V_1^2\right)\frac{K_2}{K_0} \end{aligned} \tag{32}$$

As for the inflow case, we now obtain a quadratic equation for u_{2b}.

$$\frac{\rho_1 V_1^2 J_2}{2u_1 J_0} u_{2b}^2 + \frac{\gamma}{\gamma - 1}(P_1 + I_2 - I_1)u_{2b} - \rho_1 u_1 H_{T1} \frac{K_2}{K_0} = 0 \tag{33}$$

The pressure relations for P_{1b} and P_{2b} in Equations 30 and 32 are different than the relation used in the LES for Cavity Flow Analysis program. The old method tended to over-specify the pressure. The inflow - outflow periodic boundary condition succeeded in passing turbulent flow structures with limited distortion from the outflow into the inflow of the channel flows.

Section 3: Demonstration Cases

Three Splitflow demonstration cases were run as part of this program. A turbulent channel flow is used to provide the inflow turbulent boundary layer to a simple cavity geometry. An F-22 Store Release simulation was also performed. These cases represent first attempts at calculating these complex, compressible flows with Splitflow. The accuracy and computational robustness of these demonstrations was not as good as we had hoped they would be. In particular, the turbulent boundary layer results from the channel simulation are degraded by excessive numerical diffusion. The parallel scalability of the automatic gridding logic is worse than that of the solver, and its frequent use to update the geometry caused the overall scalability to degrade on large numbers of processors. As a result, simulations took excessive wall clock time. In addition, as the store moved through the grid, the grid cutting process occasionally resulted in problem cells which caused numerical instability problems. Due to these computational problems, the duration of the F-22 store release simulation was limited and the final position of the store in the simulation is not tremendously far from the initial position. While disappointing, these results provide valuable information on both the strengths and weaknesses of the current procedures and where additional development work should be focused.

3.1 Channel Flow

The channel flow provides a good test case for near wall compressible LES capability. In addition, the channel flow case was used to provide an inflow boundary layer for the simple cavity flow solution. The Mach number at the channel centerline 1.2. The physical domain is one inch wide, two inches long and 0.56 inches high. The channel height is twice the thickness of the incoming boundary layer for the cavity flow problem. The Reynolds number for the channel using one half the channel height as the characteristic length is $Re_H = 19000$.

The grid was generated using the cut grid option in Splitflow. The near wall grid spacing is wall normal, $\Delta z^+ = 1.7$, lateral, $\Delta y^+ = 29$ and streamwise, $\Delta x^+ = 58$. A side view of the grid is shown in Figure 3.1, while a cross section view is shown in Figure 3.2. Figure 3.3 is a zoomed-in view of the grid in the cross flow direction near the wall showing the additional levels of cell refinement employed near the wall. Similar refinement is employed in the streamwise direction

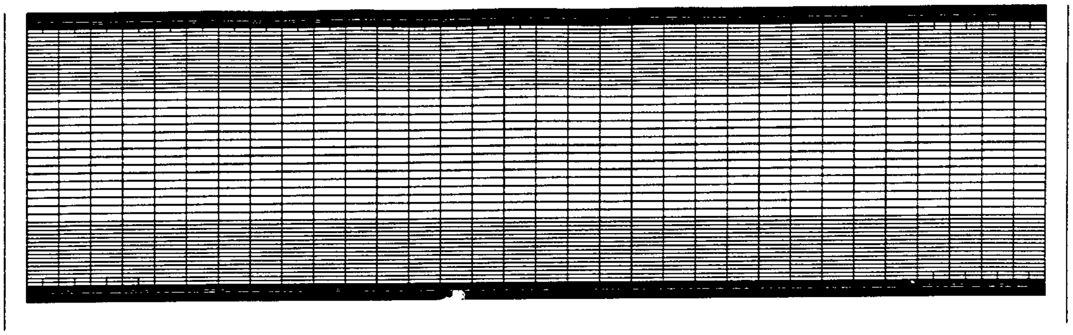

Figure 3.1 Streamwise view of channel grid

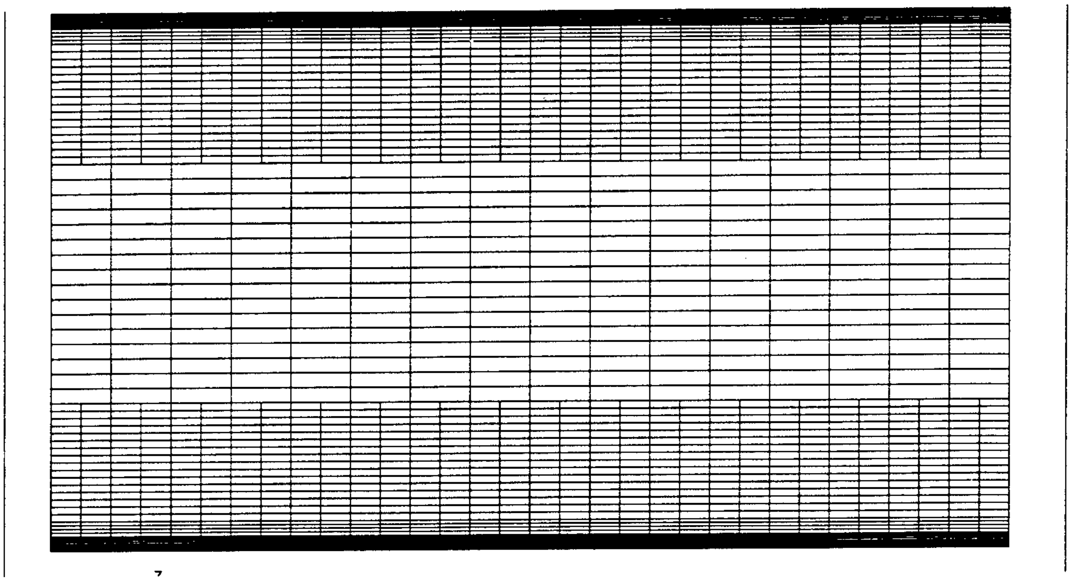

Figure 3.2 Cross section view of channel grid

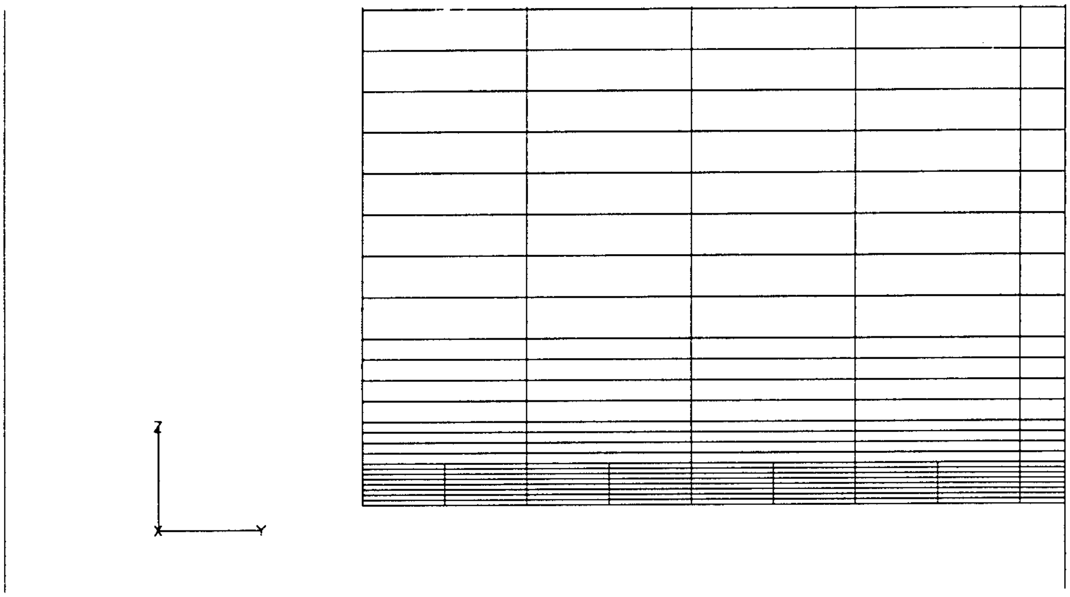

Figure 3.3 Zoomed in view of channel grid in cross flow direction.

near the wall. The total number of cells for the simulation is 138,240. If the grid had been generated with a structured mesh with the same near wall spacing would require 327,680 cells.

Results using the two different types of Roe scheme limiting, described in Section 2.7, will be presented. For the first case, the Roe scheme factor $\omega = 0.2$, while for the second case, the modified Roe scheme based on Equation 26 is used with $\sigma = 0.1$. The channel flow velocities, shown in Figure 3.4 appear reasonable. However, the profiles do not agree well with the law of the wall. Velocity profiles from the two schemes are compared in Figure 3.5. The modified Roe scheme gives slightly more accurate results than simply limiting the Roe scheme diffusion with the Roe factor. Profiles were evaluated using both incompressible and compressible wall units. A comparison of incompressible and compressible wall units for the modified Roe scheme case is shown in Figure 3.6. Compressible wall units account for the effect of density variations in the boundary layer using the van Driest transformation.

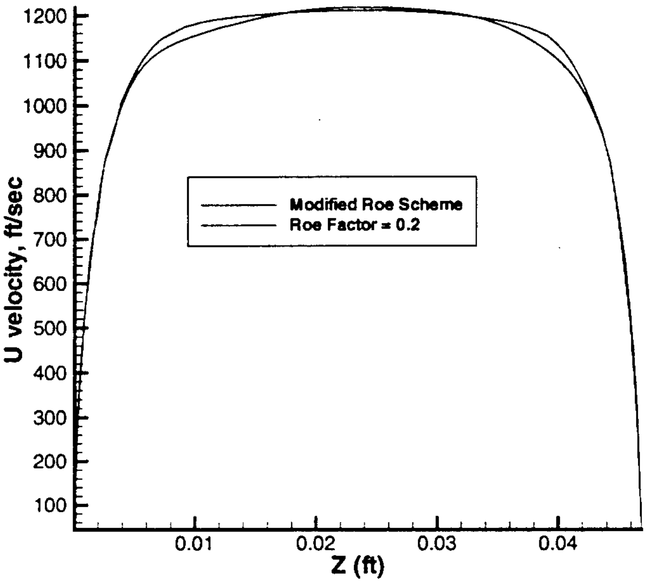

Figure 3.4 Channel flow velocity profiles for two versions of Roe sc'.eme.

The resolved turbulent energy was evaluated for the channel flow calculation. The turbulent intensities are shown in Figure 3.7. The peak level for the streamwise turbulent intensity is significantly lower than the peak turbulent intensity levels found experimentally. Based on the data of

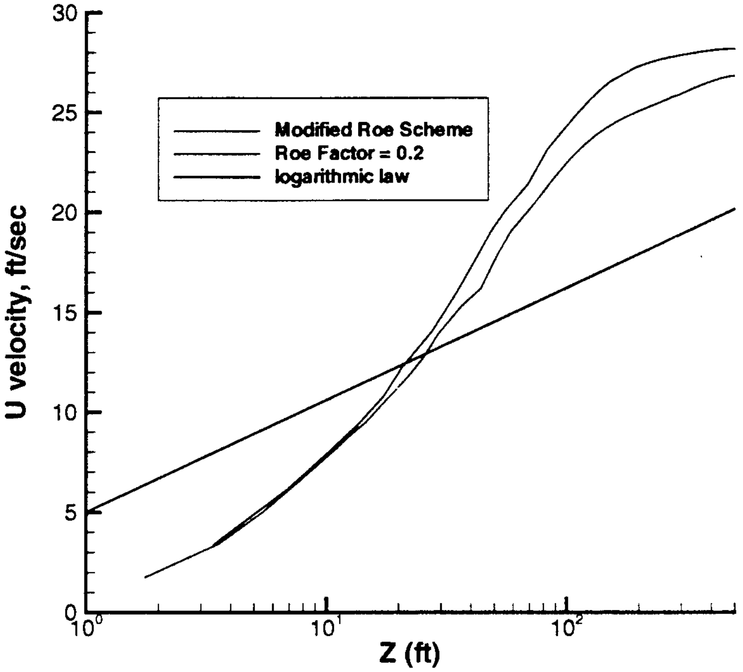

Figure 3.5 Velocity profiles for channel flow in wall units.

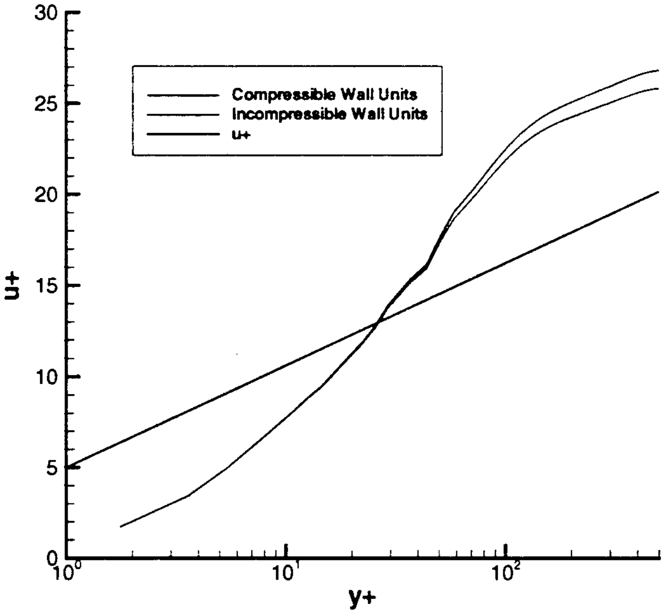

Figure 3.6 Comparison of compressible and incompressible wall units for modified Roe Scheme.

Wei and Willmarth,[8] the ratio of the streamwise peak intesity to the friction velocity is approximately 2.5, and the ratio of intensity normal to the wall to the friction velocity is approximately 1.0. Our resolved intensities are $u'/u_\tau = 1.4$ and $w'/u_\tau = 0.15$.

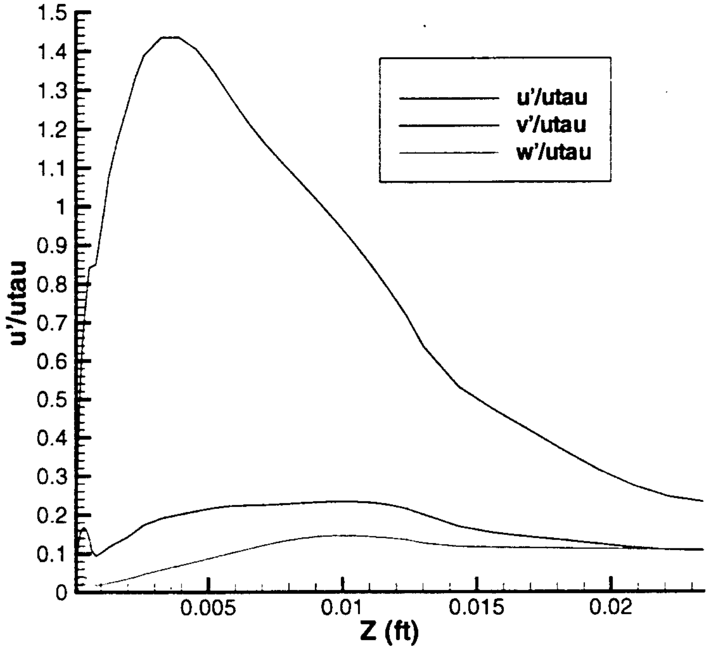

Figure 3.7 Turbulent intensities for channel flow

3.2 Cavity Flow

The cavity flow demonstration is useful for assessing the effectiveness of Splitflow with the LES model for acoustic load simulation. A flow case from the Weapons Internal Cavity and Separation (WICS) program was chosen for comparison.[9] The test model has a sharp leading edge and a 15 inch long, 16.5 inch wide flat plate in front of the cavity. The cavity L x W x D = 18" x 4" x 4". The flow condit:ons for the case selected are summarized in Table 3.1.

Table 3.1 Flow conditions for cavity simulation

Mach	1.2
U_∞	1208 ft/sec
T_∞	422 R
P_∞	188 psf
q_∞	189 psf
Re/ft	$1.0\text{x}10^6$

In order to obtain a good quality incoming turbulent boundary layer that includes unsteady structures, a small channel domain was defined and run with periodic boundary conditions. The methods for this simulation are described in Section 2.4 and Section 2.8 and the results are discussed in Section 3.1. The one inch wide channel domain was applied across the computational domain by duplicating the solution along the inflow. Because the channel flow solution uses periodic boundary condition in the lateral direction, the flow variables are continuous along the inflow plane. The distance between the center of the channel and the floor of the channel corresponds to the thickness of the incoming boundary layer.

The cavity mesh was generated using the omnigrid mesh generator. The cavity grid contained 614,000 cells. A centerline view of the mesh is shown in Figure 3.8. A cross section cut through the grid two inches downstream of the cavity leading edge is shown in Figure 3.9.

As expected, a highly unsteady flowfield was obtained in the cavity. In addition, significant asymmetry was observed. Contours of the streamwise velocity component at one instant of time in the solution along the cavity centerline are shown in Figure 3.10. Note the significant region of strong reverse flow present in the solution. Streamwise velocity contours at cross section cuts 2, 10 and 16 inches downstream of the cavity leading edge are shown in Figures 3.11, 3.12 and 3.13.

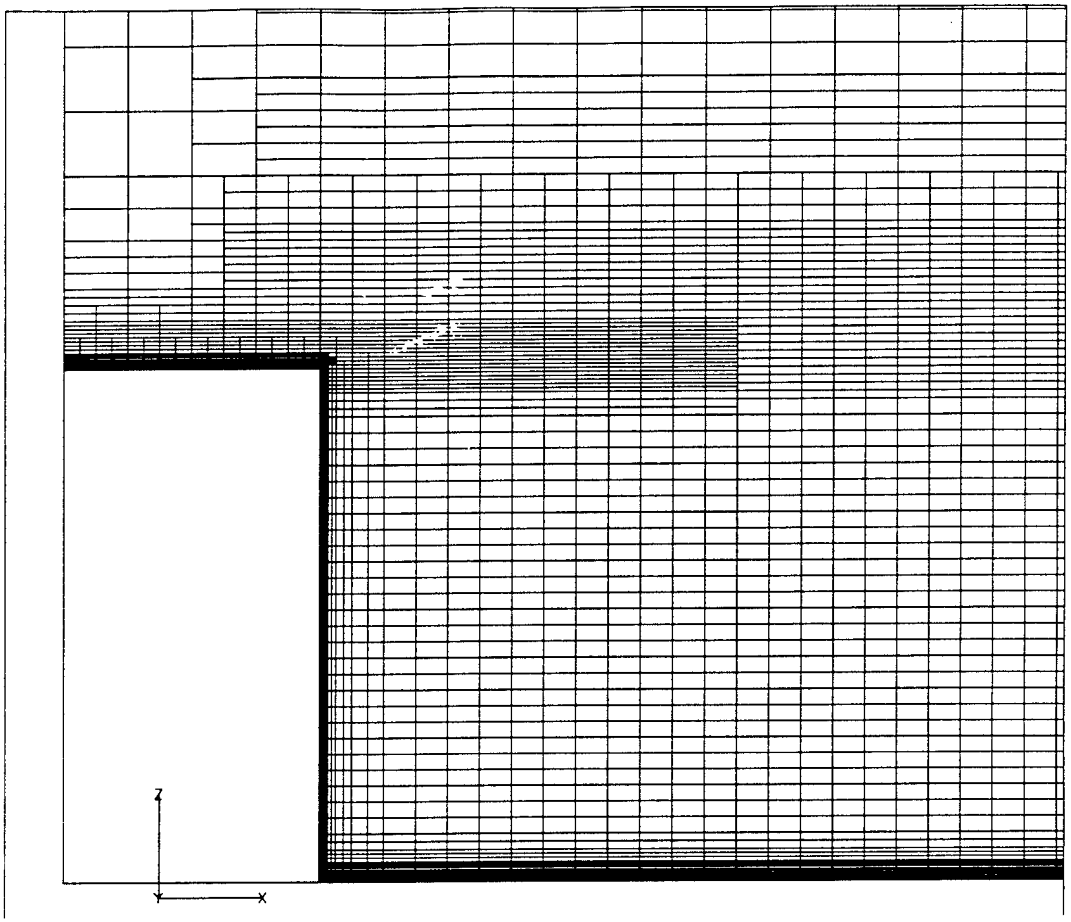

Figure 3.8 Centerline view of cavity grid generated using omnigrid.

The flow solution was computationally intensive. Stable, accurate solution of the incoming channel flow required use of a time step c.f 5×10^{-7} second. This is not unreasonably small for the channel flow solution, since it represents a $CFL = (U + a)\Delta t/(\Delta x) = 0.4$ in the streamwise direction. The CFL number in the direction normal to the wall, due to the speed of sound component a in the CFL definition and the fine grid in the wall normal direction, is several orders of magnitude larger. As a result, many thousands of iterations were required to obtain a Power Spectral Density (PSD) spectra. Power Spectral Density results were generated for four of the locations where high response pressure data were available. The spectra were based on 16384 samples averaged over two samples. This results in a bandwidth of 121 Hz. As expected, the highest acoustic levels are found near the back wall of the cavity. Cavity geometry and location of the pressure transducers are given in Figure 3.14. Acoustic spectra are shown in Figures 3.15, 3.16, 3.17 and 3.18.

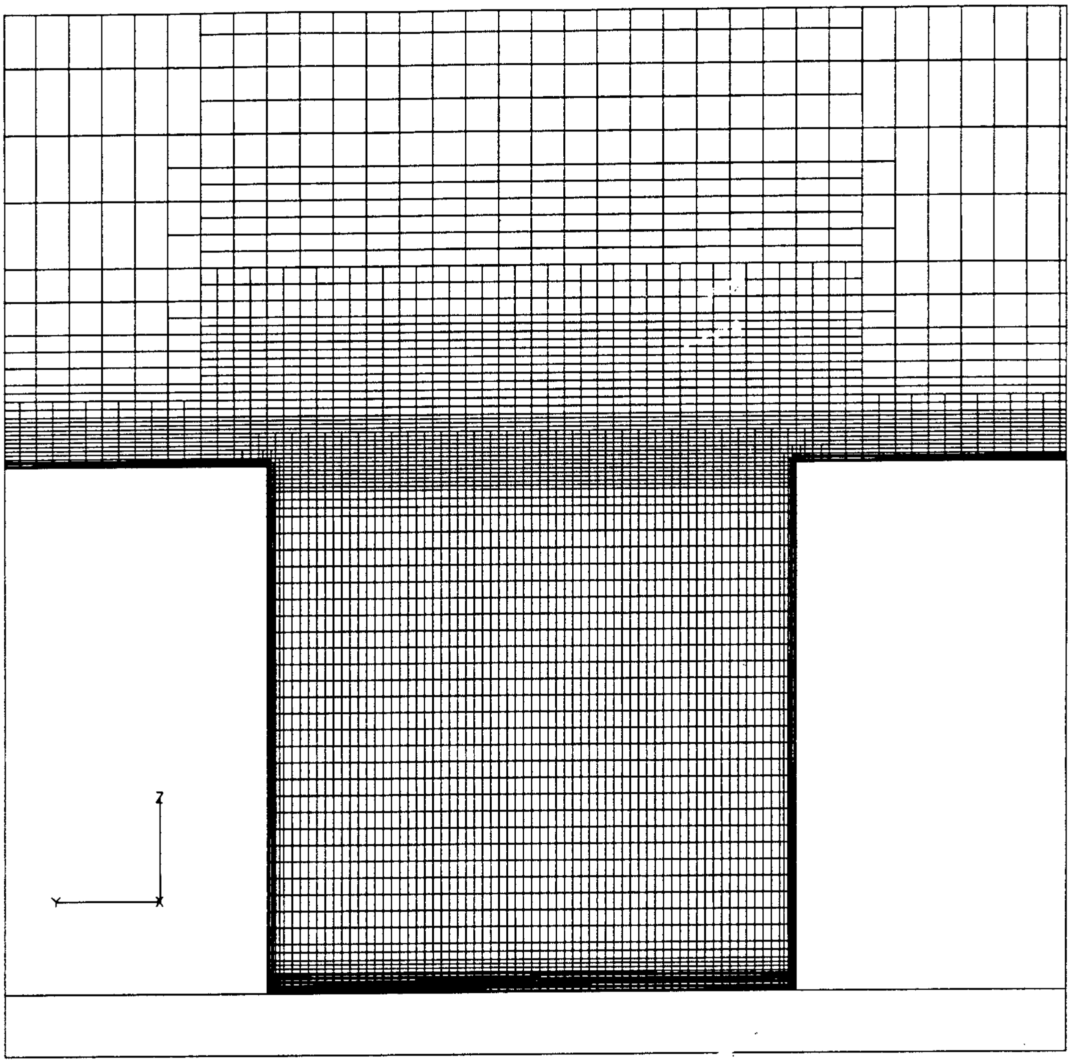

Figure 3.9 Cross section cut through the cavity grid.

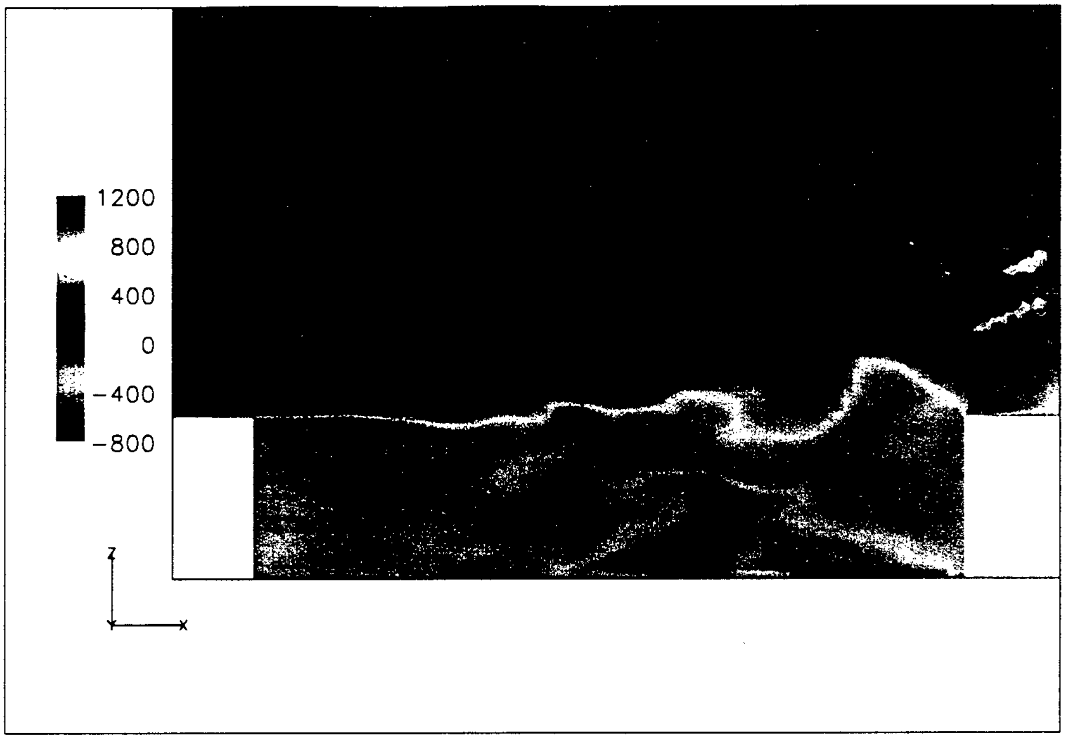

Figure 3.10 Velocity contours on cavity centerline.

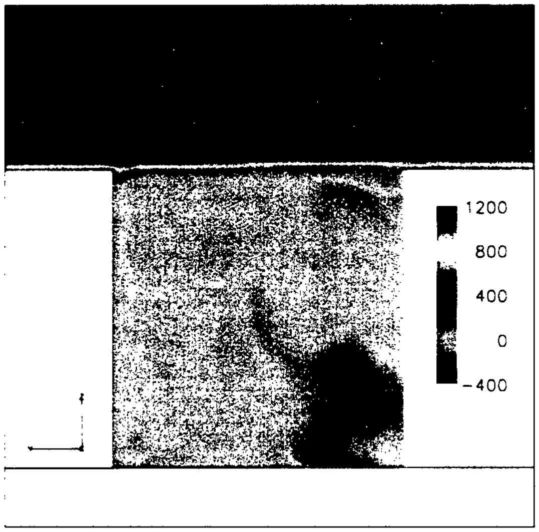

Figure 3.11 Velocity contours 2 inches from leading edge of cavity.

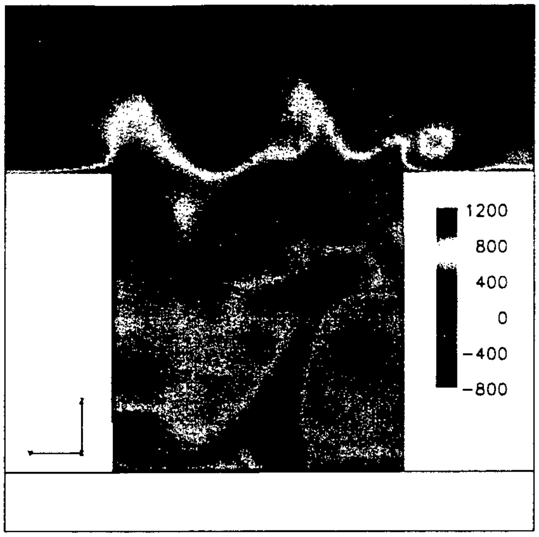

Figure 3.12 Velocity contours 10 inches from leading edge of cavity

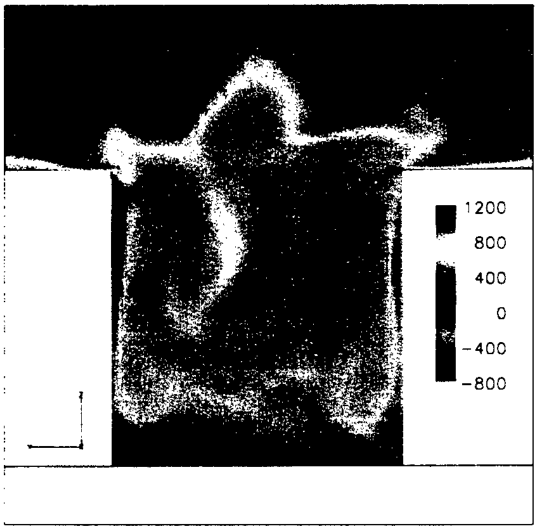

Figure 3.13 Velocity contours 16 inches from leading edge of cavity, 2 inches upstream of back wall.

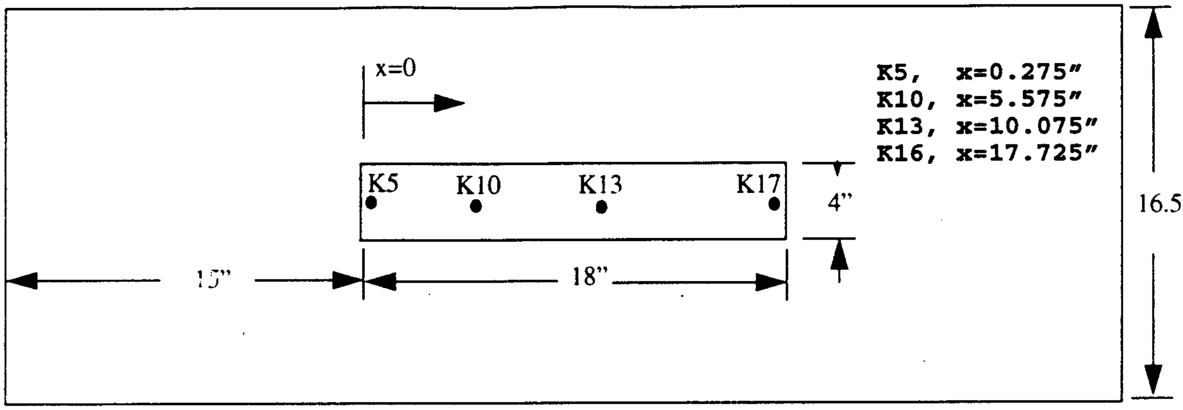

Figure 3.14 Diagram of WICS cavity and high response pressure transducer locations.

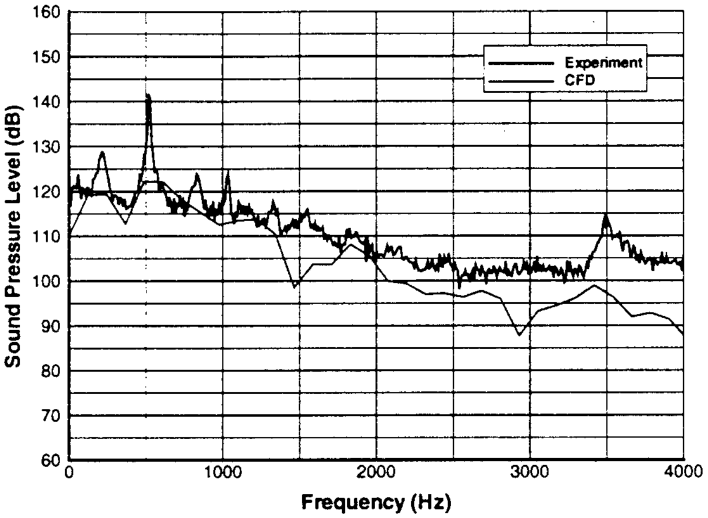

Figure 3.15 Acoustic Spectra for K5, at front wall of cavity

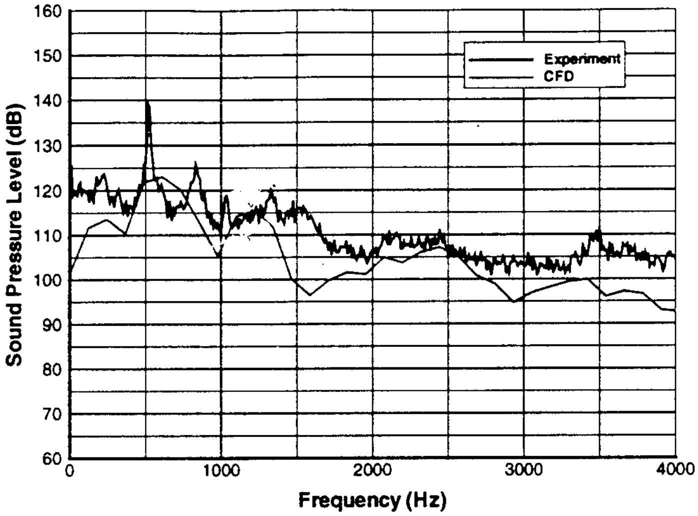

Figure 3.16 Acoustic Spectra for K10, 5.575" from front wall of cavity

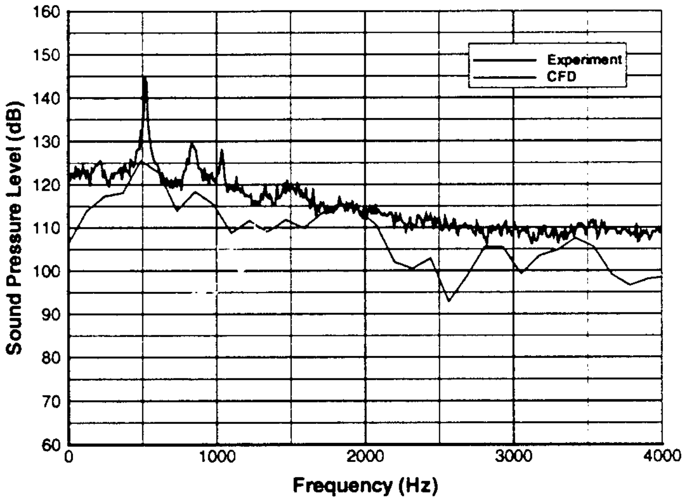

Figure 3.17 Acoustic Spectra for K13, 10.075" from front wall of cavity

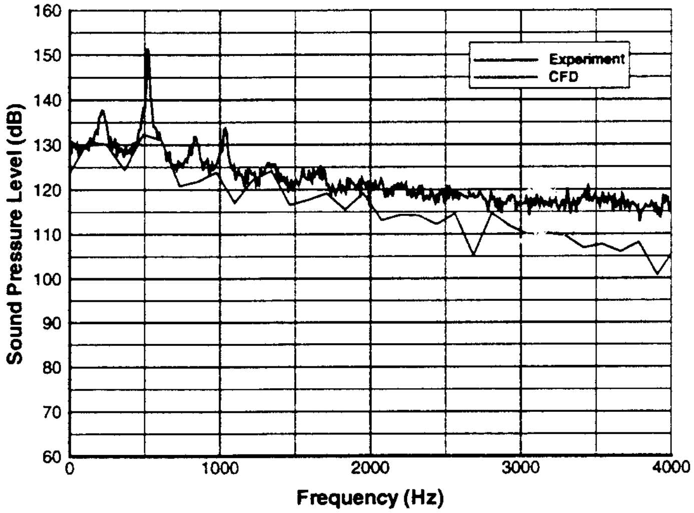

Figure 3.18 Acoustic Spectra for K16, at back wall of cavity

3.3 F-22 Store Release Simulation

Release Conditions

A release condition was chosen for this simulation from the F-22 program wind tunnel data base. The chosen case is interesting and challenging since the store shows significant response to the aircraft flow field. Configuration, flow and release conditions for the case are described in Table 3.2. The flow field and trajectory were simulated 1/15 scale (model scale). Nctə that the load factor (Nz) is less than 1.0 meaning that a push-over maneuver was modeled.

Table 3.2Trajectory simulation release conditions

Variable	Value
Test	4T-2
Run #	1756
Mach	1.6
Altitude	20,000 Ft
Nz	+0.5 g
a	1 deg
b	0 deg
V_{EOS}	28 Ft/sec
Q_{EOS}	-0.5 rad/sec
Station 7	Empty
Station 8	Empty
Station 9	AIM-120C

Grid Generation

Two grid domains were used for the store release simulation. A body conforming omnigrid was generated at the forebody of the aircraft in order to obtain accurate boundary layer profiles at the weapons bay leading edge. A cut grid surrounded this grid and the remainder of the aircraft. More details on the grid interface are contained in Section 2.4, Interface Grid Methods. A view of the F-22 geometry that was resolved and simulated is shown in Figure 3.19.

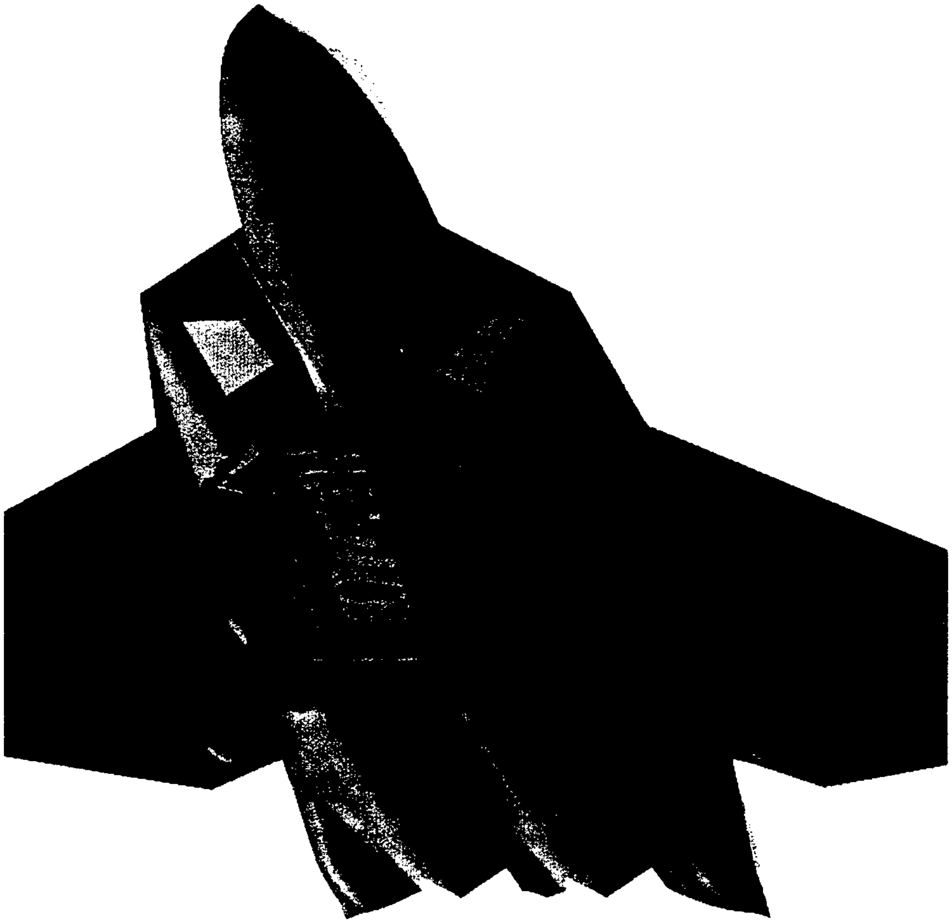

Figure 3.19 Full view of F-22 geometry simulated and Aim 120 store in initial position.

Initial Conditions

The EOS store CG location was 9 vertical inches (+Z in body axes) from the full scale carriage position. The store orientation was -1.67 degrees nose down in the body axis system. This resulted in an initial position of (533.787, 30.415, 59.210) inches full scale and (2.965, 0.1689, 0.3289) feet model scale. The store CG at the carriage position is (533.787, 30.415, 68.210) full scale inches and (2.965, 0.1689, 0.3789) feet model scale. The EOS linear and rotational velocities (Table 3.2) were applied as an initial condition for the store.

Moving Body Solutions

Figure 3.20 shows the grid in the region near the interface between the boundary fitted grid and the cut grid and in the region of the moving store for different time levels. The addition of grid cells is clearly visible as the store moves down and leaves behind a trail of small grid cells which were required to resolve previous geometry positions. Occasional cell deletion was programmed to eliminate cells as their number grew excessively. Figure 3.21 shows another view of the simulation and illustrates how the pressure coefficients on the store are changing as it encounters higher speed flow from the freestream as it moves downward in 4.76 milliseconds (model scale). The interface between the grids can be seen below the upstream edge of the weapons bay.

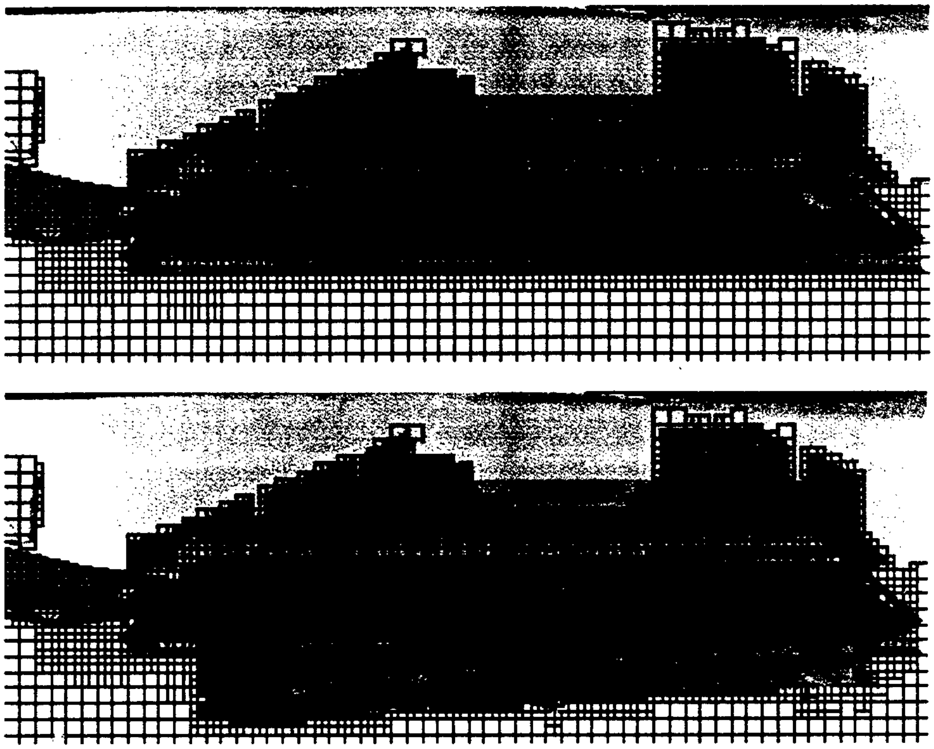

Figure 3.20 Side views of moving store. Both grids visible.

Trajectory Results

Although a full near-field trajectory was not obtained due to computational issues discussed above, initial results look quite good. Figure 3.22 shows sequences of the first 0.00476 seconds model scale of the trajectory (0.0714 seconds full scale), with flow field features. An area of low Mach number flow is observed above the missile revealing an influence of body velocity on the flow field. A plot of position versus time is shown in Figure 3.23. Position is in full scale feet and time is in model scale seconds for comparison to the above flow field figures.

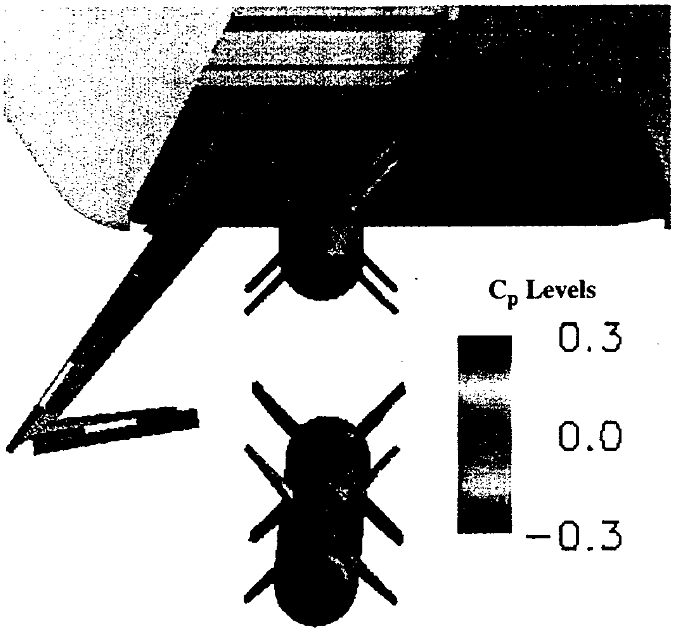

Figure 3.21 Two store positions indicate changes in aerodynamic forces.

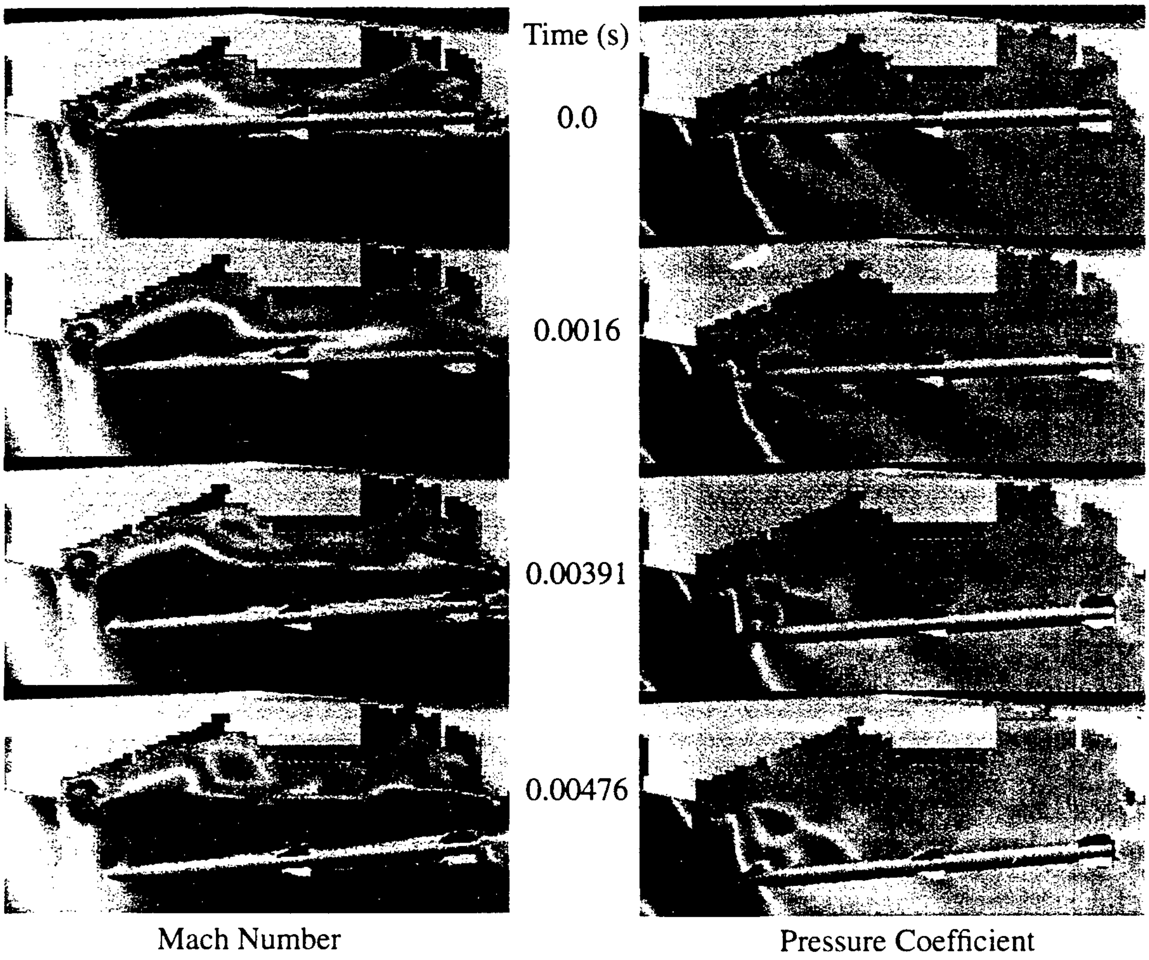

Figure 3.22 Store trajectory with mach and pressure contours in flow field.

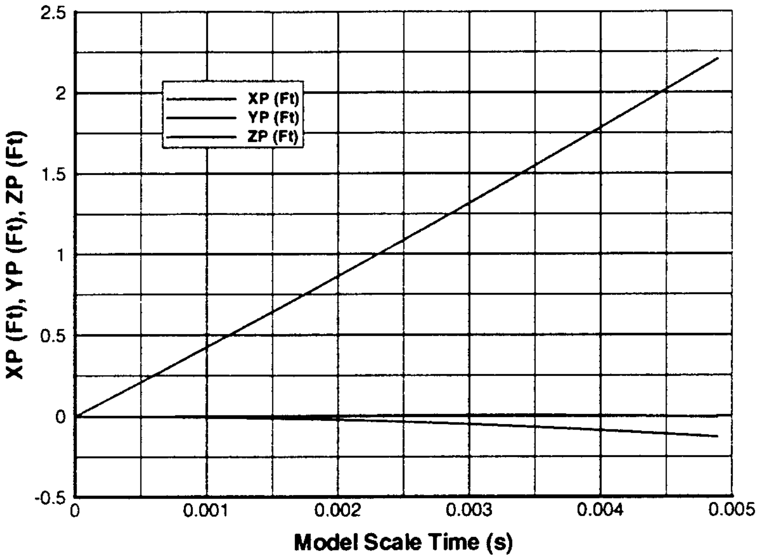

Figure 3.23 Trajectory of store from simulation. Full scale geometry, model scale time,

Section 4: Conclusions

Significant progress was made toward the development of a highly accurate method that integrates acoustic load prediction and store separation simulation using CFD. The program goals were extremely ambitious. While more work is needed improve the computational efficiency of this highly automated method, this program pioneered the development and application of this methodology. The main accomplishments of this program were:

1. Developed moving body capability in the highly automated, Splitflow parallel unstructured grid flow solver
2. Integrated six degree of freedom dynamics capability into the Splitflow solver
3. Developed a method to simultaneously solve multiple Splitflow solutions separated by faceted surface boundaries that pass fluxes conservatively between the solutions
4. Demonstrated the capability to simulate the release of a real store from an F-22
5. Predicted the unsteady acoustic loads and the PSD using Splitflow
6. Developed a methodology for simulation of compressible channel flows with periodic inflow-outflow boundary conditions

Much was learned during the methodology development and application efforts for this program. These lessons point to areas were additional work is required. The most important lessons from this study are:

1. Development of flux discretization schemes that are high order accurate, low diffusion for compressible flow analysis using unstructured parallel flow solvers remains a significant challenge. High accuracy is essential for accurate Large Eddy Simulation.
2. Improvements to the parallel scalability of Splitflow are needed for practical application of viscous unsteady flow problems around complex geometries and moving bodies.
3. Numerical approaches for initializing the flowfield for cells created at the boundaries of a moving surface undergoing large displacements require additional development to assure efficient and accurate results.

The capabilities developed in this research effort are important to the overall goal of reducing costs and design cycle times for aerospace programs. Additional development effort will be required before these methods will be ready for routine application. However, when considered in light of continued improvements in computer performance, reductions in aircraft program development budgets, and high cost of wind tunnel testing, the need for weapons bay simulation methods will persist, and the feasibility of their use will continue to improve.

Section 5: Contributors

The following LM Aero personnel contributed to the development effort:

Dr. Brian R. Smith - LES development, channel flow and cavity applications

Dr. Neal Domel - Moving body methods, overall Splitflow coordination and support

Tracy Welterlen - Interface grid methods, six degree of freedom implementation

Paul McClure - F-22 geometry definition and Omnigrid application

Dr. Steve Karman - Omnigrid refinements and cavity grid generation.

Section 6: References

1.Smith, B. R., Jordan, J. K., Bender, E. E., Rizk, S. N. and Shaw, L. L., "Computational Simulation of Active Control of Cavity Acoustics," AIAA Paper 2000-1927, Lahaina, HI, June, 2000.

2.Smith, B. R., Welterlen, T. J. and Karman, S. L., "Large Eddy Simulation for Analysis of Transonic Cavity Flows," AFOSR Final Report, August 19, 1998.

3.Domel, N. D. and Karman, S. L. "Splitflow: Progress in 3D CFD with Cartesian Omni-tree Grids for Complex Geometries,", AIAA Paper 2000-1006, Reno, NV, January, 2000.

4.Etkin, B. Dynamics of Flight: Stability and Control, Second Edition, John Wiley & Sons, New York 1982.

5.Roe, P. L., "Characteristic-Based Schemes for the Euler Equations," *Annual Review of Fluid Mechanics*, Vol. 18, 1986, pp. 337-365.

6.Shu, Chi-Wang, "Essentially Non-Oscillatory and Weighted Essentially Non-Oscillatory Schemes for Hyperbolic Conservation Laws," NASA/CR-97-206253, ICASE Report No. 97-65, November, 1997.

7.Smagorinsky, J., "General Circulation Experiments with the Primitive Equations," Mon. Weather Rev., (1963), Vol. 91. pp. 99-164.

8.Wei, T. and Willmarth, W. W., "Reynolds-number effect on the structure of a turbulent channel flow," *Journal of Fluid Mech.*, vol. 204, 1989, pp. 57-95.

9.Dix, R. E. and Dobson, T. W. Jr., "Database for Internal Store Carriage and Jettison," Final Report, AEDC-TR-90-23.