HYPERSONIC FLOW COMPUTATIONS ON UNSTRUCTURED MESHES

K. L. Bibb*
NASA Langley Research Center, Hampton, VA 23681
J. Peraire†
Massachusetts Institute of Technology, Cambridge, MA 02139
C. J. Riley‡
NASA Langley Research Center, Hampton, VA 23681

A method for computing inviscid hypersonic flow over complex configurations using unstructured meshes is presented. The unstructured grid solver uses an edge–based finite–volume formulation. Fluxes are computed using a flux vector splitting scheme that is capable of representing constant enthalpy solutions. Second–order accuracy in smooth flow regions is obtained by linearly reconstructing the solution, and stability near discontinuities is maintained by locally forcing the scheme to reduce to first–order accuracy. The implementation of the algorithm to parallel computers is described. Computations using the proposed method are presented for a sphere-cone configuration at Mach numbers of 5.25 and 10.6, and a complex hypersonic re-entry vehicle at Mach numbers of 4.5 and 9.8. Results are compared to experimental data and computations made with established structured grid methods. The use of the solver as a screening tool for rapid aerodynamic assessment of proposed vehicles is described.

Introduction

THE current NASA/Industry effort toward next generation launch vehicles, namely, the Reusable Launch Vehicle (RLV) Program, has reinforced the need for rapid computational analysis of hypersonic re-entry vehicles. This program is being conducted under the NASA "better, cheaper, faster" philosophy. The initial design phase of the X-33 program required definition of a vehicle that could safely fly through all speed regimes and was less than 18 months in duration.

During the X-33 initial design phase, most of the aerodynamic and aeroheating data required for the design was obtained from wind tunnel tests, where models were rapidly tested across the subsonic to hypersonic speed regimes. The time to develop a suitable computational grid from an initial computer-aided design (CAD) representation is typically several weeks, and small modifications to the geometry can require extensive re–working of the computational grids. As such, Euler and Navier-Stokes computational fluid dynamics (CFD) capabilities are generally utilized at later stages of the design process; i.e., CFD codes are used to provide aerodynamics and heating information on geometrically mature configurations.

A rapid computational capability that provides a timely analysis of control surface loads and design modifications to the vehicle outer mold lines (OML) is needed to parallel wind tunnel efforts. Reasonable estimates of the aerodynamic characteristics for a flight vehicle are necessary in the early design stages, with the required accuracy increasing as the design matures. Unstructured mesh methods are ideally suited for rapid analysis as they have the capability to greatly reduce the time associated with grid generation on complex configurations as compared to traditional block structured grid methods. Unstructured mesh systems which provide automatic mesh generation starting from a CAD definition and compute inviscid flow solutions have already been developed.[1–3] These systems can handle complex configurations such as complete aircraft, and can produce reliable data using Euler flow models. However, most of the development work in the field of unstructured methods has been carried out for transonic or low supersonic flows. As these methods are extended to higher speed flows, they typically become less robust and require more dissipation and limiting to maintain stability.

In this paper, the application of the unstructured mesh system FELISA[4,5] to the solution of flows about hypersonic re-entry vehicles is presented. The mathematical formulation of the solver is presented, and the issues unique to maintaining stable hypersonic computations are discussed. The parallel implementation of the flow solution algorithm is also addressed.

Three applications are presented to illustrate the developed capability. The first application compares FELISA results with inviscid LAURA[6] (a structured

*Research Engineer, Aerothermodynamics Branch, Aero- and Gas Dynamics Division. Member, AIAA

†Associate Professor, Department of Aeronautics and Astronautics. Senior Member, AIAA

‡Research Engineer, Aerothermodynamics Branch, Aero- and Gas Dynamics Division. Member, AIAA

grid code) calculations and experimental data for a sphere-cone configuration. In the second example, calculations are performed for a complex configuration and the resolution of flow features is compared with solutions calculated with viscous LAURA and inviscid DPLUR.[7] Lastly, the use of FELISA as a screening tool in preliminary design phases for rapid and accurate estimation of aerodynamic coefficients is demonstrated. Additional related applications have also been presented by Prabhu.[8]

Unstructured Mesh Generation

The discretization of the three–dimensional computational domain into tetrahedra is carried out by the mesh generator within the FELISA system. This requires that the geometry of the domain be defined in terms of an assembly of composite bi–cubic surface patches, which intersect along lines defined as composite cubic curves. For realistic geometries generated using CAD systems, the data files are sufficiently complex that tools for their efficient generation become necessary.

The software package GridTool[9] allows the user to import geometries in a wide variety of formats, interactively perform the necessary geometry manipulations and output a FELISA compatible data file. These geometry manipulations may include geometry clean–up, definition of intersection curves, re–parametrization of body surfaces, and definition of the far field domain boundaries. The grid density is controlled by means of a background mesh together with a system of point, line, and triangular sources.

The first step in the mesh generation is the triangulation of the boundary surfaces using an advancing front method. The domain interior is then discretized using a constrained Delaunay algorithm which incorporates exact integer arithmetic to resolve situations which are ambiguous when performed with standard floating point arithmetic.[10]

The use of the mesh generator in combination with GridTool has proven to be very reliable and easy to use. The speed at which elements are generated is above three million tetrahedra per hour using a single processor high–end workstation such as an SGI–R10000. The user–interactive part of mesh generation process is confined to defining the geometry and defining the grid density. It is seen that the unstructured grid approach can give large reductions in grid generation time as compared to structured grid methods, with typical times required for the generation of initial grids on the order of one to two days. The generation of multiple grids on the same or similar geometries can also be accomplished efficiently.

Flow Solution

The FELISA unstructured grid hypersonic flow solver uses a finite volume formulation. The basic scheme is implemented using an efficient edge data structure[5,11,12] and several flux vector/difference splitting options for computing the edge fluxes. The most successful and reliable option is the flux vector splitting proposed by Hänel et al.,[13] which leads to a scheme that produces steady numerical solutions with constant total enthalpy (when the inflow conditions have constant total enthalpy). Second–order accuracy in smooth flow regions is obtained by linearly reconstructing the solution following MUSCL concepts.[14,15] Stability near discontinuities is maintained by locally forcing the scheme to reduce to first–order accuracy using the Local Extremum Diminishing criteria.[16]

Governing Equations

The three–dimensional time–dependent Euler equations written in integral form, over an arbitrary control volume Ω with boundary $\partial\Omega$, are expressed as

$$\frac{\partial}{\partial t}\int_{\Omega} \mathbf{U}\ d\Omega + \int_{\partial\Omega} \mathbf{F}_k\ dS_k = \mathbf{0}, \tag{1}$$

where t denotes time, S_k is the k-th Cartesian component of the outward area vector to $\partial\Omega$, and summation over repeated indices is assumed. The vector of unknowns $\mathbf{U}$ and the vector of inviscid fluxes $\mathbf{F}_k$ are given by

$$\mathbf{U} = \begin{pmatrix} \rho \\ \rho u_1 \\ \rho u_2 \\ \rho u_3 \\ \rho E \end{pmatrix} \qquad \mathbf{F}_k(\mathbf{U}) = \begin{pmatrix} \rho u_k \\ \rho u_1 u_k + p\delta_{1k} \\ \rho u_2 u_k + p\delta_{2k} \\ \rho u_3 u_k + p\delta_{3k} \\ \rho H u_k \end{pmatrix}. \tag{2}$$

The quantities ρ, p, u_k, E, and H represent the density, pressure, k-th velocity component, total specific internal energy, and enthalpy, respectively, and δ_{jk} denotes the Kronecker delta.

Spatial Discretization

For each vertex i, a fixed control volume Ω_i, consisting of all tetrahedra sharing that vertex is defined. Considering equation (1) over each Ω_i, and approximating the integral over each triangular facet in $\partial\Omega_i$ by the value of the integrand at the centroid times the area we obtain

$$\frac{\partial}{\partial t}\int_{\Omega_i} \mathbf{U} d\Omega + \sum_{f\in\mathcal{F}_i} \frac{1}{3}(\mathbf{F}_k^l + \mathbf{F}_k^m + \mathbf{F}_k^n) S_k^f = \mathbf{0} \tag{3}$$

where $\mathcal{F}_i$ denotes the set of triangular facets which form $\partial\Omega_i$; l, m and n, are the three nodes of a typical facet f; and $\mathbf{F}_k^l = \mathbf{F}_k(\mathbf{U}^l)$ where $\mathbf{U}^l$ denotes the value of the unknown vector $\mathbf{U}$ at node l. By further approximating the volume integral by $\mathbf{U}^i$ times V^i, the volume of Ω_i, we can write the following system of coupled ordinary differential equations for the evolution of the unknown vector at each node

$$\frac{d\mathbf{U}^i}{dt} + \frac{1}{V^i}\sum_{f\in\mathcal{F}_i}\frac{1}{3}(\mathbf{F}_k^l + \mathbf{F}_k^m + \mathbf{F}_k^n)S_k^f = \mathbf{0}. \qquad (4)$$

In the above expression, the flux corresponding to a neighboring node appears as many times as facets share that node. For computational efficiency, it is convenient to re–arrange these contributions in such a way that instead of summing over the facets bounding Ω_i, summation is performed over the mesh edges connecting i and its neighboring nodes. Thus,

$$\frac{d\mathbf{U}^i}{dt} + \frac{1}{V^i}\sum_{e\in\mathcal{E}_i}\mathbf{F}_k^l S_k^e = \mathbf{0} \qquad (5)$$

where $\mathcal{E}_i$ denotes the set of mesh edges containing node i, e is a typical interior edge joining nodes i and l (Figure 1), and S_k^e is the k-th component of the area vector associated to edge e, which in turn is calculated as

$$S_k^e = \frac{1}{3}\sum_{f\in\mathcal{F}_{il}} S_k^f. \qquad (6)$$

In the above expression $\mathcal{F}_{il}$ is the subset of facets in $\mathcal{F}_i$ that contain node l. Noting that the sum of the area vectors over all edges belonging to $\mathcal{E}_i$ must equal zero, expression (5) can be re–written for an interior node as

$$\frac{d\mathbf{U}^i}{dt} + \frac{1}{V^i}\sum_{e\in\mathcal{E}_i}(\mathbf{F}_n^i + \mathbf{F}_n^l)S^e = \mathbf{0} \qquad (7)$$

where S^e is the modulus of the area vector associated to edge e, $\mathbf{F}_n^i = \mathbf{F}_k^i \hat{S}_k^e$, and $\hat{S}_k^e = S_k^e/S^e$. For interior edges, only one area vector needs to be stored for each edge. That is, the area vector required to form the equation for node l can be obtained from that required to form the equation for node i with a simple sign change. For boundary edges however, a special treatment is required to ensure that all the contributions defining the volumes associated with boundary nodes are considered. Details of this process are given in Peraire et al.,[17] and in practice it amounts to performing an additional loop over all the boundary facets.

Flux Vector Splitting

The above scheme uses a centered approximation to the flux terms and is, therefore, non–dissipative. In order to produce a practical scheme which introduces sufficient damping to allow for strong shocks to be captured, we follow a flux vector splitting approach and replace equation (7) by

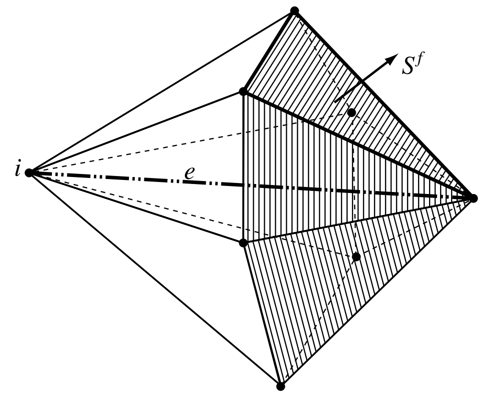


Fig. 1 Set of triangular facets used to evaluate the area vector associated with an interior edge e joining vertices i and l.

$$\frac{d\mathbf{U}^i}{dt} + \frac{1}{V^i}\sum_{e\in\mathcal{E}_i}(\mathbf{F}_n^+(\mathbf{U}^i) + \mathbf{F}_n^-(\mathbf{U}^l))S^e = \mathbf{0}. \qquad (8)$$

Several choices for the numerical flux functions $\mathbf{F}_n^\pm$ are available in the literature. We have adopted the form proposed by Hänel et al.,[13] since our experience indicates that it provides one of the most competitive options when robustness, accuracy and cost are taken into account. In contrast, the unstructured mesh implementation of the flux difference splitting of Roe,[18] as used in the LAURA code, was found to be much less stable and in flows exhibiting strong normal shocks, it often exhibited the so–called "carbuncle phenomenon."[19]

The flux functions used here are

$$\mathbf{F}_n^\pm(\mathbf{U}) = \rho u_n^\pm \begin{pmatrix} 1 \\ u_1 \\ u_2 \\ u_3 \\ H \end{pmatrix} + p^\pm \begin{pmatrix} 0 \\ \hat{S}_1^e \\ \hat{S}_2^e \\ \hat{S}_3^e \\ 0 \end{pmatrix}. \qquad (9)$$

The definition for $u_n^\pm$ and $p^\pm$ is given by

$$u_n^\pm = \begin{cases} \pm(u_n \pm c)^2/4c, & \text{if} \quad |u_n| \le c \\ \frac{1}{2}(u_n \pm |u_n|), & \text{otherwise} \end{cases} \qquad (10)$$

and

$$p^\pm = \begin{cases} p(u_n/c \pm 1)^2(2 \mp u_n/c), & \text{if} \quad |u_n| \le c \\ p(u_n \pm |u_n|)/2u_n, & \text{otherwise} \end{cases} \qquad (11)$$

respectively, where $u_n = u_k \hat{S}_k^e$, and the sound speed c is given by $c^2 = \gamma p/\rho$. An attractive feature of this form of flux vector splitting is that, whenever the enthalpy of incoming flow is uniform, the converged discrete solution also has constant enthalpy.

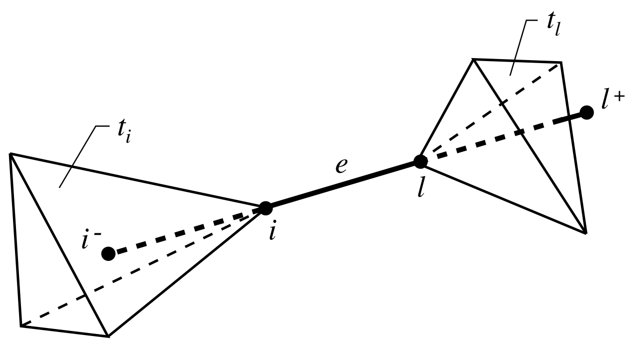


Fig. 2 Dummy vertices and elements used in the solution reconstruction process.

Reconstruction

The dissipation added using the above flux vector splitting is sufficient to make the overall scheme stable and allows for shocks of arbitrary strength to be captured over just two or three cells. However, in smooth flow regions the scheme has poor accuracy, and requires unrealistically fine meshes to compute useful solutions. To obtain a scheme which retains good shock capturing properties, a geometric reconstruction scheme is employed.

To illustrate this reconstruction procedure, consider (Figure 2) a typical interior edge e of length δ_e, joining nodes i and l, and define the *dummy* nodes i^- and l^+ by extending the edge e a distance δ_e at either side of nodes i and l, respectively. Let t_i denote the tetrahedron containing node i, such that the solid angle defined by its three triangular facets containing node i includes the segment, or part of the segment, joining nodes i and i^-. In a similar manner, we define the tetrahedron t_l such that it contains node l and includes the segment, or at least part of the segment, joining nodes l and l^+. The linear variations defined by the values of (ρ, u_k, H) at the four vertices of t_i and t_l are used to determine the value of (ρ, u_k, H) at the dummy nodes i^- and l^+, respectively.

The modified algorithm, using reconstruction, is thus obtained by replacing (8) with

$$\frac{d\mathbf{U}^i}{dt} + \frac{1}{V^i}\sum_{e\in\mathcal{E}_i}(\mathbf{F}_n^+(\mathbf{U}^+) + \mathbf{F}_n^-(\mathbf{U}^-))S^e = \mathbf{0}. \quad (12)$$

The unknown vectors $\mathbf{U}^+$ and $\mathbf{U}^-$ are computed by first considering the vector of unknowns $\mathbf{V} = (\rho, u_1, u_2, u_3, H)^T$ and then evaluating $\mathbf{U}^+ = \mathbf{U}(\mathbf{V}^+)$ and $\mathbf{U}^- = \mathbf{U}(\mathbf{V}^-)$, where $\mathbf{V}^+$ and $\mathbf{V}^-$ are given as follows

$$\begin{aligned}
&\mathbf{V}^+ = \mathbf{V}^i + \tfrac{1}{2}\Delta\mathbf{V}\\
&\mathbf{V}^- = \mathbf{V}^l - \tfrac{1}{2}\Delta\mathbf{V}\\
&\Delta\mathbf{V} = minmod(\mathbf{V}^{l^+} - \mathbf{V}^l, \mathbf{V}^l - \mathbf{V}^i, \mathbf{V}^i - \mathbf{V}^{i^-}),
\end{aligned} \quad (13)$$

and where the *minmod* function returns the argument with smallest absolute value when all the arguments are of the same sign and zero otherwise.

It is seen that by introducing the auxiliary vector $\mathbf{V}$, the resulting limited scheme retains the ability to represent discrete constant enthalpy solutions. In addition, it is observed that in smooth regions of the flow, the limited scheme can be interpreted as a central difference approximation (7) with higher order artificial viscosity.

Special care needs to be exercised near boundaries to appropriately choose the tetrahedron to be used in the determination of $\mathbf{V}$ at the dummy nodes. In this case a tetrahedron meeting the requirements specified above may not exist. A reasonable compromise that we have found to work well in practice is to choose the tetrahedron that more closely meets these requirements.

Time Integration

The equation set (12) is discretized in time using a forward Euler explicit time stepping scheme with a local time step determined from linearized stability analysis. For problems involving strong shocks, it is found that this form of determining the time step may lead to limit cycle oscillations that prevent the solution from converging to machine precision. A remedy to that situation[20] consists of supplementing the local time step calculation with an additional check that enforces that the local monotonicity of the solution be preserved.[16] This additional check is found to be active for a very small number of points, but it is sufficeint to eliminate the limit cycle behavior.

Boundary Conditions

At the solid wall boundaries, the only contribution to the flux is through the pressure term. In addition, when the normal at the wall is well defined, the velocity vector is forced to be tangent to the surface. Along trailing edges and other locations where a unique normal is not defined, the velocity vector is not explicity modified.

At the far field, boundary conditons are imposed using the flux splitting procedure described above and evaluating the appropriate flux function $\mathbf{F}^{\pm}$ using far–field values.

Parallel Implementation

The above described explicit algorithm is easily parallelized by decomposing the domain into subdomains which are then assigned to different processors and dealt with in a concurrent manner. The necessary communication between subdomains is accomplished using explicit message passing. The code can use either of the standard message passing libraries, PVM[21] or MPI,[22] and runs on a number of platforms ranging from clusters of heterogeneous local workstations to mainframe parallel computers such as the IBM SP2 or the Cray T3D.

Currently, the mesh generation is carried out in a serial mode. The mesh partitioning strategy is such that every edge in the mesh belongs to a single subdomain, whereas the nodal points on the subdomain boundaries are shared between two or more subdomains. The domain is decomposed using a simple coordinate bisection algorithm. Within each subdomain, edges are divided into two groups. The first group is made up of those edges which have at least one node that requires communication (i.e., a node belonging to more than one subdomain). The second group is made up of the remaining edges. Typically, a loop over the edges in the first group is performed first. The values accummulated at the boundary nodes are then broadcast to the neigboring subdomains. The interior edges are considered next, and then the information from the neigboring subdomains is received. This strategy allows for a considerable amount of communication and computation to take place concurrently, thus minimizing latencies. The parallel efficiency of the code using 16 nodes is higher than 95% for typical computations of the size reported in this paper.

Results

The current effort will demonstrate that the FELISA grid generation system and hypersonic Euler solver, FELISA_HYP, give reasonable and consistent predictions of aerodynamic performance and can be used effectively in rapid analysis of complex configurations. Predictions with FELISA_HYP for a sphere-cone configuration and a preliminary RLV/X–33 lifting body configuration are compared with experimental data and with calculations made using a structured grid Navier-Stokes code, LAURA,[6] and an inviscid structured grid code, DPLUR.[7] LAURA has been verified with Shuttle orbiter flight data[23,24] and a wide spectrum of configurations for supersonic to hypersonic flow conditions.[25] The parallel implementation of DPLUR has allowed it to be used as an aerodynamic screening tool for single-block configurations.

The FELISA_HYP solver has both scalar and parallel implementations; however, all FELISA calculations presented herein were made using a single processor of an SGI-R10000 Onyx machine, as were the LAURA computations for the cone cases. The LAURA computations on the RLV/X-33 lifting body were performed on the NAS Cray C-90. All DPLUR calculations were made on the IBM SP2 at LaRC.

The strategy used in developing a block structured grid to perform a series of analyses on a given configuration using LAURA is to provide for the 'best' grid that will be used. This means that each grid has spacing for viscous computations, and sufficient resolution of the geometry to accurately capture the heating rates on the windside surfaces. LAURA utilizes an algorithm that adjusts the outer domain and spacing normal to the body to capture the shape of the outer bow shock for each case to be run, thus minimizing the number of complete grids that must be generated to analyze a particular configuration. For a typical grid on an X-33 configuration, the grid generation time is 4 weeks (6 weeks were required on the first grid; lessons learned made subsequent discretizations faster.).

The flexibility of the unstructured mesh approach offers a significant time advantage over the block structured grid approach, and thus allows for a different strategy in developing meshes for series of computations. The initial geometry definition for a FELISA mesh is done once for all grids, and produces a mesh that has the surface geometry adequately resolved, and an initial attempt at resolving the flow features of the first case. For typical X–33 geometries, this takes about 4 days; the first X–33 configuration that was analyzed required about 1.5 weeks. There are several strategies available for generation and adaption to a particular case. The 'brute force' method was used in the examples in this paper; coarse mesh solutions are calculated, and the spacing distribution is revised with significant user intervention to capture the bow shock properly. Mesh enrichment, movement, and remeshing strategies have been developed for FELISA, but the results have not been satisfactory when the dominant flow feature is a strong bow shock. Solution adaptive strategies are under development which will allow for automatic resolution of the bow shock simultaneous with resolution of the flow features surrounding the body.

Spherically Blunted Cone

The first case presented is for flow over a 15° half–angle spherically blunted cone with a nose radius of 1.0 inch. FELISA solutions were calculated at freestream Mach numbers of 5.25 and 10.6 and angles of attack of 0° and 15°. These results are compared to the experimental results of Cleary[26] and computations made using DPLUR and LAURA in an inviscid mode.

For the 0° cases, the FELISA solution was calculated on a mesh of 267,459 points. The LAURA solution was calculated on a 61×33 axisymmetric structured grid. For the FELISA computations at $M_\infty = 5.25$ and $\alpha = 15°$, an initial mesh of 186,276 points and a fine mesh of 693,910 points were used. For the $M_\infty = 10.6$ and $\alpha = 15°$ case, the solution was calculated on a medium mesh of 328,527 points and the fine mesh used for the $M_\infty = 5.25$, $\alpha = 15°$ case. The LAURA and DPLUR calculations utilized a $61 \times 19 \times 33$ grid, with the outer boundary adapted at each Mach number to the bow shock using the LAURA shock alignment routine.

The theoretical value of C_p at the stagnation point is 1.81 for $M_\infty = 5.25$, and 1.83 for $M_\infty = 10.6$. All of the solutions presented here slightly underpredict the

Case	Stag C_p $M_\infty = 5.25$	Stag C_p $M_\infty = 10.6$
Theory	1.81	1.83
LAURA, 0°	1.81	1.82
FELISA, 0°	1.80	1.82
LAURA, 15°	1.80	1.82
DPLUR, 15°	1.79	—
FELISA, 15°initial	1.78	—
FELISA, 15°medium	—	1.82
FELISA, 15°fine	1.80	1.81

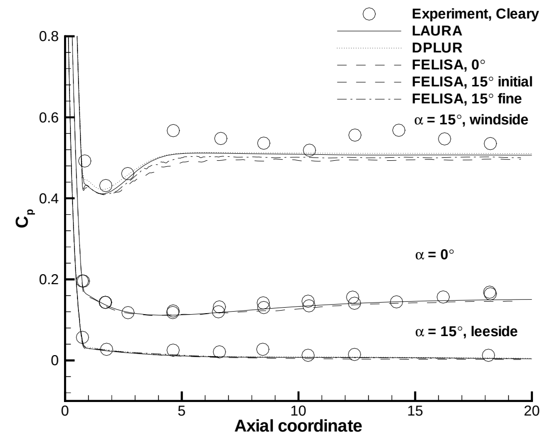
Table 1 Stagnation $C_p$values for 15° half–angle spherically blunted cone.

stagnation point C_p, with the LAURA results at 0° being closest to the theoretical value. Table 1 shows the stagnation values of C_p for all of the cases presented here.

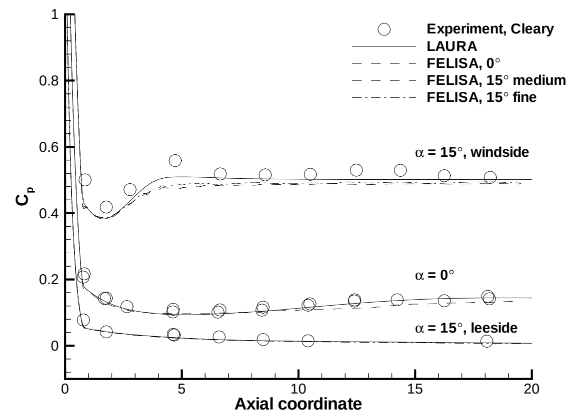
The comparisons of centerline pressure coefficients, shown in Figure 3, indicate good agreement for the computational cases. For the $M_\infty = 5.25$ solutions, the largest differences in the C_p values for the fine grid FELISA solutions as compared to the LAURA solutions are on the order of 2%, and are in the recompression region of the windside of the cone at 15° angle of attack. In most regions, the difference is less than .5%. The solution on the initial mesh of 186K nodes gives a poorer comparison, due to inadequate resolution of the bow shock. Experience has shown that a mesh density of approximately 15 points is needed between the bowshock and the body to resolve the shock adequately; the initial FELISA mesh had approximately 10 points, while the fine mesh had approximately 18. The comparisons for the $M_\infty = 10.6$ cases show differences between the LAURA and the FELISA solutions on the order of 6% in the recompression region, 2% elsewhere. The agreement (of all of the computations) with experiment is reasonable except for the 15° cases on the windside.

RLV/X–33 Lifting Body Configuration

A preliminary Lockheed-Martin RLV/X-33 lifting body configuration is used in the remainder of the comparisons. The full vehicle is shown in Figure 4. It has twin vertical tails and fins; the engines are modeled by the box-shaped structure on the base. This configuration was evaluated during Phase I of the RLV/X-33 program with wind tunnel testing and computational analysis to predict aerodynamics and aeroheating for the full trajectory. The LAURA and DPLUR computations and the Mach 4.5 FELISA computations presented in this paper were run as part of the Aerothermodynamics Branch's analysis program. The Mach 9.8 FELISA solution on this configuration was run specifically to compare with the LAURA and DPLUR results; the grid density was similar to a computation made with FELISA on an earlier con-



a) $M_\infty = 5.25$



b) $M_\infty = 10.6$

Fig. 3 $C_p$distributions for a 15° half–angle spherically blunted cone. Experiment, LAURA, DPLUR, and FELISA_HYP.

figuration.

RLV/X-33 Mach 9.8 Computations

In this example, FELISA results are compared with viscous LAURA and inviscid DPLUR calculations for $M_\infty = 9.8$ and $\alpha = 40°$. The base flow of the vehicle was not calculated.

The computational grids for the FELISA and LAURA calculations are shown in Figure 5. The unstructured FELISA grid has 82K surface nodes, 716K volume nodes, and 4.2M tetrahedra. The LAURA viscous solution was computed on a 24-block structured grid system with a total of 1.85M grid points. The forebody portion of the grid was $65 \times 65 \times 65$, and the 12-block aftsection (including wings and vertical tail) was $101 \times 248 \times 65$. The unusual shape of the exit

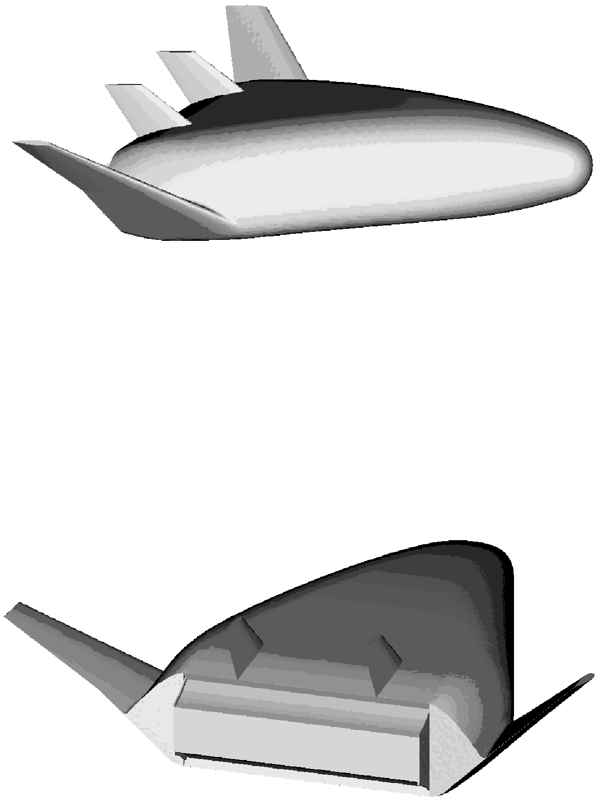
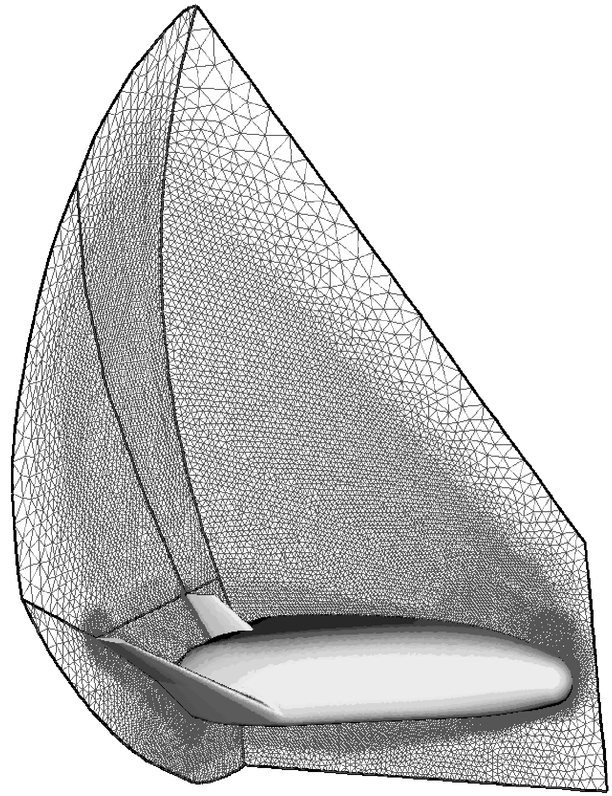


Fig. 4 Preliminary configuration of the Lockheed-Martin RLV/X-33.

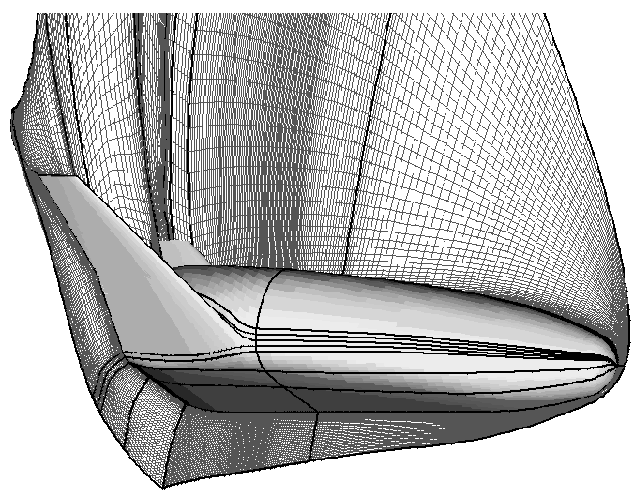
plane was generated using the shock alignment routine within the LAURA code, and was copied for the FELISA case. The DPLUR data is for the forebody only, and used a single block, $51 \times 65 \times 41$ grid. When compared to the structured computational grid, the FELISA mesh provided comparable resolution of the windside, much better resolution of the leeside of the vehicle, and somewhat coarser resolution on the wing. Also, the structured grid was a viscous grid and, as such, provided for finer normal spacing of grid points, and coarser axial spacing.

Figure 6 shows the centerline pressure coefficients for the LAURA, DPLUR, and FELISA calculations. Overall, the agreement is very good, with small local differences in the recompression region of the forebody. Stagnation C_p for all three cases is 1.81; the theoretical value is 1.83.

A comparsion of the windside C_p contours is given in Figure 7. The flow features of the FELISA solution compare very well to both the viscous LAURA and the



a) FELISA mesh



b) LAURA mesh

Fig. 5 Computational grids for $M_\infty = 9.8$, $\alpha = 40°$ calculations on preliminary Lockheed-Martin RLV/X-33 configuration.

inviscid DPLUR. The slight differences on the wing surface (Figure 7(a)) are attributable to grid density differences. The LAURA grid is more resolved in the wing region than this FELISA grid, due to the original purpose for each of the calculations. The LAURA calculation was made to resolve the heating levels on

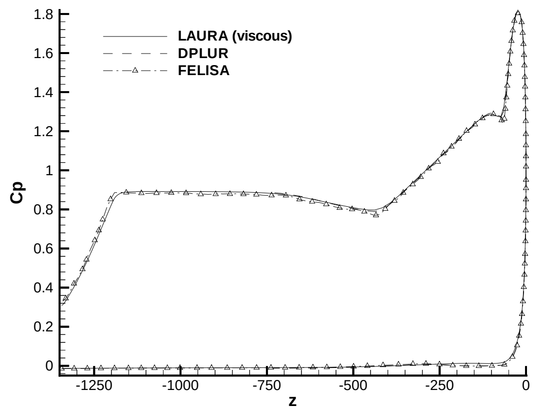


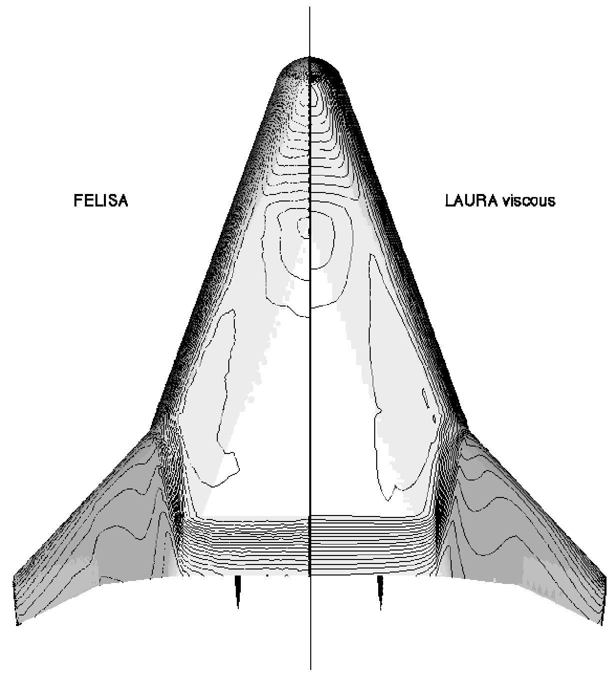
Fig. 6 $C_p$distribution on the symmetry plane of preliminary Lockheed–Martin RLV/X-33 configuration, $M_\infty = 9.8$ and $\alpha = 40°$.

the vehicle, while the purpose of the FELISA solutions was to generate first cut aerodynamic design data.

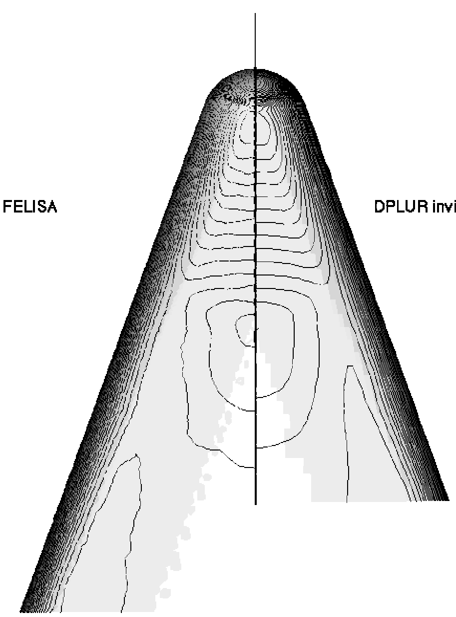
The C_p distribution on the leeside of the vehicle is shown in an expanded view (compared to the scale of Figure 6) in Figure 8. A significant variation between viscous LAURA and FELISA in the recompression region is observed, due to the flow separation predicted by the viscous solution. The DPLUR inviscid and the FELISA compare very well up to an axial location of about 250. The FELISA grid is much denser in this region, and predicts a higher compression. The agreement with LAURA further down the body is completely fortuitous; the actual flow fields are very different due to the flow separation on the leeside.

Comparisons of pressure coefficient contours on the leeside of the RLV/X-33 are shown in Figure 9. The difference in the crossflow shock structure between the FELISA inviscid and LAURA viscous solutions (Figure 9(a)) is clearly evident. Comparison between the FELISA and DPLUR inviscid solutions (Figure 9(b)) shows similar crossflow shock structures. The DPLUR shock is slightly more smeared, and located further off the centerline of the body. This is due to the finer spacing of the FELISA mesh. In practice, structured grids are generated such that the leeside flow is not as resolved as the windside, so as to focus computational resources on the regions that have the most impact on the heating and aerodynamic forces on the body.

Figure 10 shows pressure coefficient contours at a cut near the middle of the fuselage, comparing the FELISA inviscid results to the LAURA viscous and DPLUR inviscid. The LAURA solution clearly shows the separation as flow comes around the body. The inviscid FELISA and DPLUR solutions remain attached, and generate a crossflow shock to turn the flow downstream. The outer shock for the FELISA is more smeared; this is due to the the grid not being well



a) **FELISA and viscous LAURA solutions**



b) **FELISA and DPLUR solutions**

Fig. 7 $C_p$contours on the windside of preliminary Lockheed–Martin RLV/X-33 configuration, $M_\infty = 9.8$ and $\alpha = 40°$.

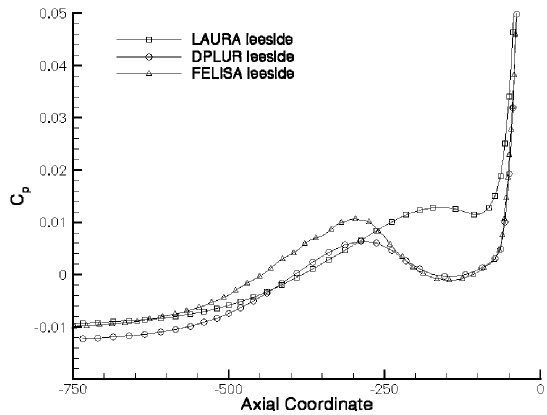


Fig. 8 Leeside $C_p$distribution on the symmetry plane of preliminary Lockheed–Martin RLV/X-33 configuration, $M_\infty = 9.8$ and $\alpha = 40°$.

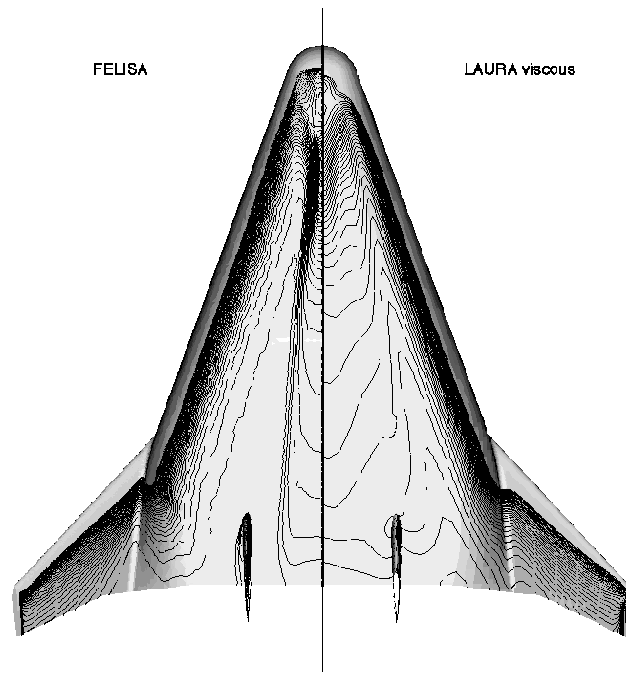
aligned with the shock and a coarse FELISA grid in the leeside bow shock region. Both of the structured grid solutions used LAURA's shock alignment routine to capture the outer bow shock more efficiently, and it would be very difficult to generate an unstructured grid that provides as much alignment with the shock. Improvements to the adaptive capabilities of FELISA are needed to better align the bow shock, particularly on the windside of the vehicle.

Figure 11 shows pressure coefficient contours for FELISA and LAURA viscous at a cut near the rear of the fuselage including the wing and vertical tail. The differences here are primarily in the region of the cross-flow shock and the outer bow shock. Qualitatively, the features compare very well, particularly in the flow around the wing.

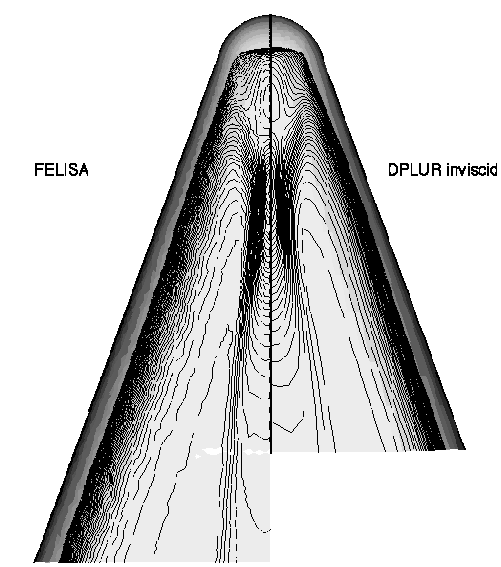
X-33 Aerodynamic Comparisons, Mach 4.5

The final example is a calculation of the aerodynamic forces over the complete X-33 configuration. The computations were performed at a Mach number of 4.5 in air, and at angles of attack of 0° and 10°. The experimental data shown for comparison is from a test performed in the LaRC Unitary Plan Wind Tunnel facility,[27] and the data is courtesy of NASA LaRC. The test was run at $M_\infty = 4.50$ and Re/ft of 2×10^6 on a 0.0105 scale model with transition artificially induced.

The shaded surface and symmetry plane meshes used in the aerodynamic calculations are shown in Figure 12. The FELISA grid used had a surface mesh of 55K points, and a volume mesh containing 439K points and 2.56M tetrahedra. The same mesh was used for calculations at both angles of attack. The mesh extended over the entire body, including the base region. The LAURA calculation used the same initial grid as the $M_\infty = 9.8$ calculations and was adapted to form the grid shown in Figure 12(b) for the $\alpha = 10°$ case and a similar grid for the $\alpha = 0°$ case. The base re-

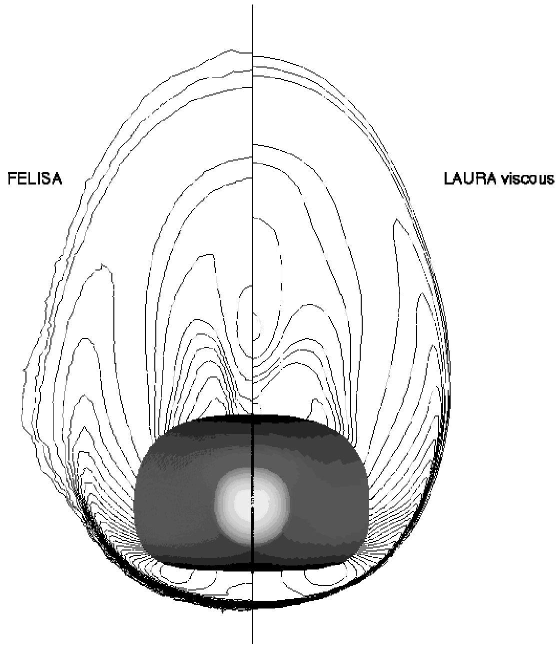


a) **FELISA and viscous LAURA solutions**

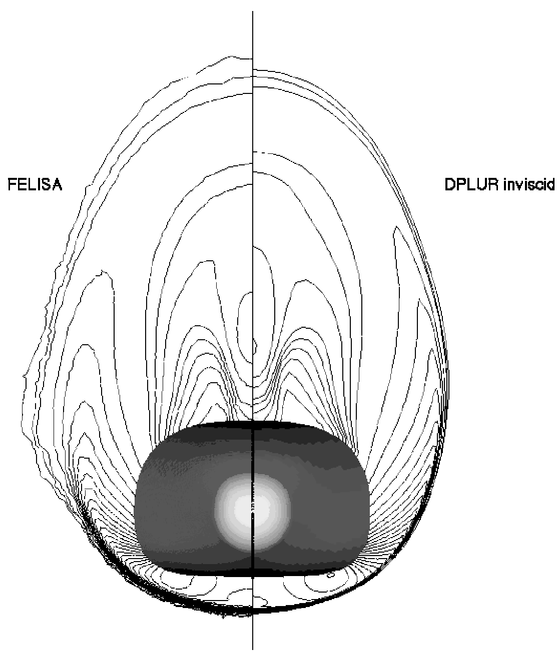


b) **FELISA and DPLUR solutions**

Fig. 9 $C_p$contours on the leeside of preliminary Lockheed–Martin RLV/X-33 configuration, $M_\infty = 9.8$ and $\alpha = 40°$.



a) **FELISA and viscous LAURA solutions**



b) **FELISA and DPLUR solutions**

Fig. 10 $C_p$contours on forebody cut of preliminary Lockheed–Martin RLV/X-33 configuration, $M_\infty = 9.8$ and $\alpha = 40°$.

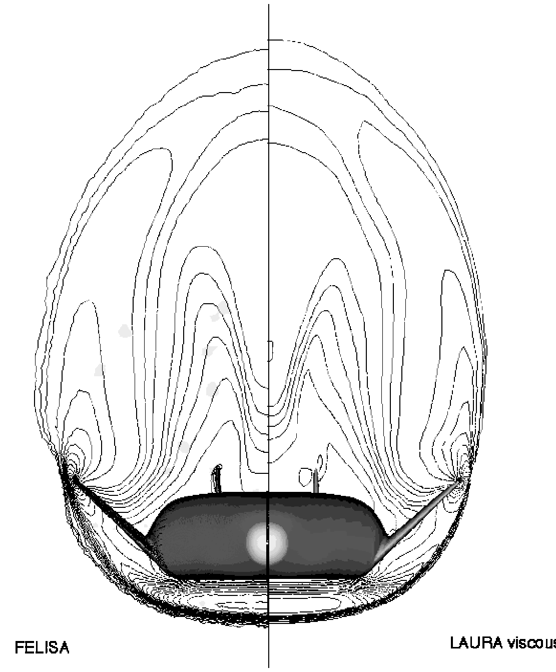


Fig. 11 $C_p$contours on rear fuselage cut of the preliminary Lockheed–Martin RLV/X-33 configuration, $M_\infty = 9.8$ and $\alpha = 40°$, FELISA and LAURA viscous solutions.

gion was not modeled in the LAURA computations.

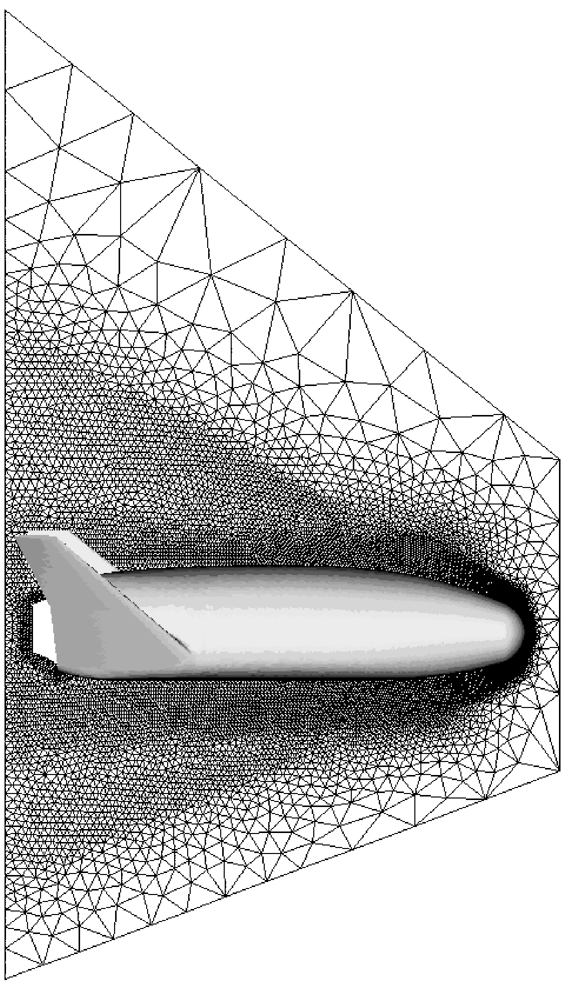
A plot of the normal force and moment coefficients vs. angle of attack is shown in Figure 13. The agreement of the FELISA normal force and moment coefficients with the experimental data and the viscous LAURA calculations is very good. For the axial force coefficients, the agreement is slightly degraded. This is expected, since the FELISA calculations do not include viscous effects.

Conclusions

A new algorithm for calculating inviscid hypersonic flow over re–entry vehicles using unstructured meshes has been developed. The flow solver uses a finite–volume formulation with an edge based data structure. Fluxes are calculated with the Hänel flux vector splitting formulation, and a MUSCL gradient reconstruction is implemented for higher–order accuracy. These algorithms for flux formulation provide more robust simulation capabilities for hypersonic flows as compared to earlier implementations in FELISA. The new algorithm has been implemented on both scalar and parallel architectures.

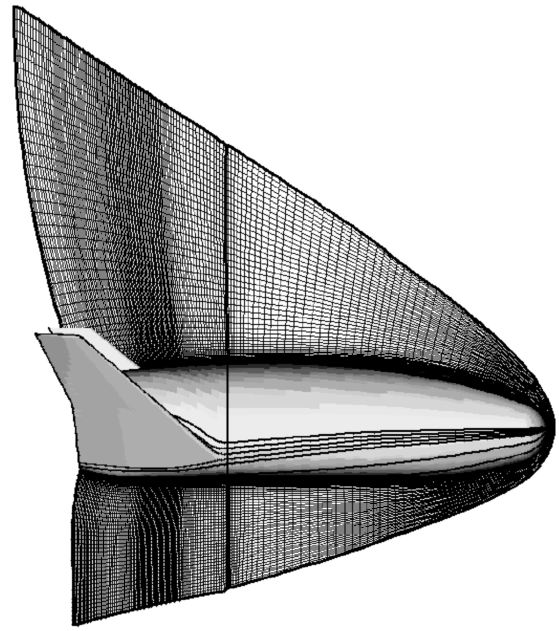
The FELISA_HYP flow solver has been shown to be accurate in predicting both integrated aerodynamic characteristics and resolving flow features for complex re–entry vehicles. Examples of flow solutions over an RLV/X-33 configuration were presented. Comparisons of these solutions to experimental data and computational solutions from established structured grid flow solvers provide the first, strong evidence for the validity of the FELISA_HYP algorithm in the code validation process. The use of unstructured meshes permits faster mesh generation than traditional structured grid methods for complex configurations using state-of-the-art technology and expertise.

The algorithm presented here enables rapid screening for aerodynamics and pressure loads on space transportation vehicles in a time frame of two to fourteen days, including grid generation. The methodology presented in this work has been successfully applied to preliminary phases of both the X-33 and X-34 programs. The examples presented here are a small sampling of the work performed in the Aerothermodynamics Branch over the past two years. While the method described herein has been shown to be accurate and effective, there are several improvements in the grid generation and solution adaption areas that would further enhance the efficiency of the method.



a) FELISA mesh, $\alpha = 0°$ and $10°$

Fig. 12 Body and symmetry plane grid for aerodynamic calculations on preliminary Lockheed–Martin RLV/X-33 configuration, $M_\infty = 4.5$.



b) LAURA grid, $\alpha = 10°$ case

Fig. 12 Concluded.

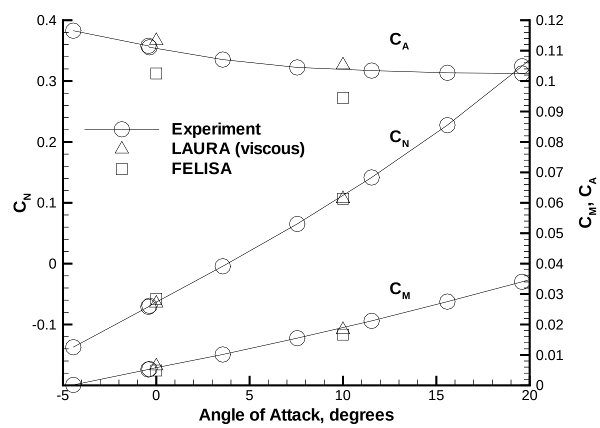


Fig. 13 Aerodynamic force coefficients for the Lockheed-Martin X-33 B1001A configuration at $M_\infty = 4.5$, $\alpha = 0°$ and $10°$.

These algorithm enhancements are the subject of ongoing research.

Acknowledgements

The authors would like to thank Mr. William Scallion and Mr. Matthew Rhode of NASA Langley Research Center for providing the X-33 aerodynamic data from their UPWT test. Ms. Norma Bean of GeoLab, CSC and Mr. Stephen Alter of LESC generated the structured X-33 grids used in the computations. Dr. Ramadas Prabhu of LESC provided invaluable assistance in running and improving this solver. The second author would like to thank the Aerothermodynamics Branch of NASA, LaRC for providing funding for development of the flow solver under grant NAG1-1587.

References

[1] Jameson, A. and Baker, T. J., "Improvements to the Aircraft Euler Method," AIAA Paper 87–0452, January 1987.

[2] Peraire, J., Morgan, K., and Peiro, J., "Unstructured Finite Element Mesh Generation and Adaptive Procedures for CFD," *AGARD Conference Proceedings No. 464, Applications of Mesh Generation to Complex 3D Configurations*, No. 18.1–18.12, 1990.

[3] Frink, N. T., Pirzadeh, S., and Parikh, P., "Grid Software System for Solving Complex Aerodynamic Problems," NASA CP 3291, May 1995.

[4] Peiro, J., Peraire, J., and Morgan, K., "FELISA System Reference Manual and User's Guide," Tech. rep., University College Swansea Report, 1993.

[5] Peraire, J., Peiro, J., and Morgan, K., "Multigrid solutions of the 3D Compressible Euler Equations on Unstructured Tetrahedral Grids," *International Journal for Numerical Methods in Engineering*, Vol. 36, 1993, pp. 1029–1044.

[6] Cheatwood, F. M. and Gnoffo, P. A., "User's Manual for the Langley Aerothermodynamic Upwind Relaxation Algorithm (LAURA)," NASA TM 4674, April 1996.

[7] Candler, G. V., Wright, M. J., and McDonald, J. D., "Data–Parallel Lower–Upper Relaxation Method for Reacting Flows," *AIAA Journal*, Vol. 32, No. 12, pp. 2380–2386.

[8] Prabhu, R. K., "Computational Study of a McDonnell Douglas Single–Stage–to–Orbit Vehicle Concept for Aerodynamic Analysis," NASA CR 201606, September 1996.

[9] Samereh, J., "GridTool: A Surface Modeling and Grid Generation Tool," NASA CP 3291, May 1995.

[10] Peraire, J. and Morgan, K., "Unstructured Mesh Generation Including Directional Refinement For Aerodynamic Flow Simulation," 1997, to appear in *International Journal for Engineering Analysis and Design*.

[11] Barth, T. J., "Numerical Aspects of Computing Viscous High Reynolds Number Flows on Unstructured Meshes," AIAA Paper 91–0721, 1991.

[12] Mavriplis, D. J., "Three Dimensional Unstructured Multigrid for the Euler Equations," AIAA Paper 91–1549, 1991.

[13] Hänel, D., Schwane, R., and Seider, G., "On the accuracy of Upwind Schemes for the Solution of the Navier–Stokes Equations," AIAA Paper 87–1005, 1987.

[14] van Leer, B., "Towards the Ultimate Conservation Scheme: A Second Order Sequel to Gudonov's Method," *Journal of Computational Physics*, Vol. 32, 1979, pp. 101–136.

[15] Arminjon, P. and Dervieux, A., "Construction of TVD–like Artificial Viscosities on 2-dimensional Arbitrary FEM grids," INRIA Report 1111, 1989.

[16] Jameson, A., Reuther, J., and Martinelli, L., "Computational Algorithms for Aerodynamic Analysis and Design TLSP: Annual Report, 1 Oct. 1993 – 30 Sep. 1994," Tech. Rep. AFOSR–95–0082TR, Princeton University, 1994.

[17] Peraire, J., Peiro, J., and Morgan, K., "Finite Element Multigrid Solution of Euler Flows Past Installed Aero–engines," *Computational Mechanics*, Vol. 11, No. 33, 1993, pp. 451.

[18] Roe, P. L., "Approximate Riemann Solvers, Parameter Vectors, and Difference Schemes," *Journal of Computational Physics*, Vol. 43, October 1981, pp. 357–372.

[19] Quirk, J. J., "A Contribution to the Great Riemann Solver Debate," ICASE Report 92–64, November 1992, (Also as NASA CR 191409.).

[20] Elliot, J. K. and Peraire, J., "Practical 3D Aerodynamic Design and Optimization using Unstructured Meshes," 1996, submitted to *AIAA Journal*.

[21] Geist, A., Beguelin, A., Dongerra, J., Jiang, W., Manchek, R., and Sunderam, V., "PVM 3 User's Guide and Reference Manual," ORNL/TM 12187, 1994.

[22] Message Passing Interface Forum, "MPI: A Message–Passing Interface Standard," Computer Science Department Technical Report CS-94-230, University of Tennessee, Knoxville, TN, 1994.

[23] Weilmuenster, K. J., Gnoffo, P. A., and Greene, F. A., "Navier-Stokes Simulations of Orbiter Aerodynamic Characteristics Including Pitch Trim and Bodyflap," *Journal of Spacecraft and Rockets*, Vol. 31, No. 3, May-June 1994, pp. 355–366.

[24] Gnoffo, P. A., Weilmuenster, K. J., and Alter, S. J., "Multiblock Analysis for Shuttle Orbiter Re-entry Heating from Mach 24 to Mach 12," *Journal of Spacecraft and Rockets*, Vol. 31, No. 3, May-June 1994, pp. 367–377.

[25] Gnoffo, P. A., "Code Calibration Program in Support of the Aeroassist Flight Experiment," *Journal of Spacecraft and Rockets*, Vol. 27, No. 2, March-April 1990, pp. 131–142.

[26] Cleary, J. W., "An Experimental and Theoretical Investigation of the Pressure Distribution and Flow Fields of Blunted Cones at Hypersonic Mach Numbers," NASA TN D–2969, 1965.

[27] Jackson, Jr., C. M., Corlett, W. A., and Monta, W. J., "Description and Calibration of the Langley Unitary Plan Wind Tunnel," NASA TP 1905, 1981.