Solver and Turbulence Model Upgrades to OVERFLOW 2 for Unsteady and High-Speed Applications

R. H. Nichols[*]
The University of Alabama at Birmingham, Birmingham, Alabama, 35294

R. W. Tramel[†]
Digital Fusion, Huntsville, Alabama, 35806

and

P. G. Buning[‡]
NASA Langley Research Center, Hampton, Virginia, 23681

An implicit unfactored SSOR algorithm has been added to the overset Navier-Stokes CFD code OVERFLOW 2 for unsteady and moving body applications. The HLLEM and HLLC third-order spatial upwind convective flux models have been added for high-speed flow applications. A generalized upwind transport equation has been added for solution of the two-equation turbulence models and the species equations. The generalized transport equation is solved using an unfactored SSOR implicit algorithm. Three hybrid RANS/DES turbulence models have been added for unsteady flow applications. Wall function boundary conditions that include compressibility and heat transfer effects have been also been added to OVERFLOW 2.

Nomenclature

A	=	linearized flux in the ξ direction
B	=	linearized flux in the η direction
C	=	linearized flux in the ζ direction
c	=	chord length
E	=	inviscid and viscous flux in the ξ direction
e_0	=	total energy
F	=	inviscid and viscous flux in the η direction
G	=	inviscid and viscous flux in the ζ direction
k	=	turbulent kinetic energy
mm	=	update level for SSOR
P_k	=	production of turbulent kinetic energy (k)
p	=	pressure
s	=	distance normal to a wall
T	=	temperature
t	=	time
u,v,w	=	velocity vector components
u_τ	=	friction velocity
$V\!\!\!-$	=	computational cell volume
x,y,z	=	coordinate directions

[*] Research Associate Professor, Department of Mechanical Engineering, AIAA Senior Member.
[†] Senior Aerospace Engineer, AIAA Senior Member.
[‡] Aerospace Engineer, Configuration Aerodynamics Branch, Mail Stop 499, AIAA Associate Fellow.

X	=	eigenvector
y^+	=	nondimensional wall distance ($\rho\,u_\tau\,s/\mu$)
ε	=	turbulent dissipation
Λ	=	eigenvalue
μ	=	molecular viscosity
μ_t	=	eddy viscosity
ρ	=	density
σ_L	=	laminar Schmidt number
σ_T	=	turbulent Schmidt number
τ	=	pseudo time variable for dual time stepping
ξ,η,ζ	=	curvilinear coordinates
Ω	=	relaxation parameter for SSOR
ω	=	specific dissipation

Subscripts

| ∞ | = | free stream or reference value |
| i | = | spatial increment counter |

Superscripts

| m | = | subiteration counter |
| n | = | time step iteration counter |

I.   Introduction

The OVERFLOW 2[1,2] Navier-Stokes computational fluid dynamics (CFD) code was developed by merging the OVERFLOW[3,4] flow solver with the 6-degee-of-freedom moving body capability of the OVERFLOW-D flow solver[5,6,7]. The OVERFLOW code is a structured grid overset[8] Navier-Stokes flow solver. The OVERFLOW flow solver originally incorporated the diagonal form of the implicit approximate factorization algorithm of Pulliam and Chaussee[9] and a second-order in space central difference approximation for the inviscid fluxes. Mixed second and fourth order smoothing was added to the explicit and implicit side of the equations to provide numerical stability. A number of upgrades have been made to the original code including:

1. Lower Upper-Symmetric Gauss Seidel (LU-SGS) implicit solution algorithm and a Roe upwind inviscid flux scheme[10]
2. Multigrid solution procedure, low-Mach preconditioning, and a central difference/matrix dissipation inviscid flux scheme[11]
3. Parallization with OPENMP and MPI[12]
4. AUSM inviscid flux scheme[13]
5. Dual time stepping implicit solution algorithm[14]

The inclusion of a moving body capability and the increase in unsteady flow applications has led to a search for more robust implicit time marching algorithms both for the flow solver and for the scalar transport equations (turbulence and species) contained in the code. More robust algorithms are also required for hypersonic applications. Hypersonic applications require low numerical diffusivity to accurately resolve the waves in the flow, yet also require robustness for the large discontinuities in the flow that can occur near shocks. This effort attempts to improve the current code by adding the following:

1. HLLEM[15] and HLLC[16] inviscid Riemann flux algorithms
2. An unfactored Successive Symmetric Over Relaxation (SSOR) implicit solution algorithm
3. A robust and accurate algorithm for solving the transport equations based on the unfactored SSOR algorithm
4. Three hybrid Reynolds Averaged Navier-Stokes/Large Eddy Simulation (RANS/LES) turbulence models
5. Wall functions for the transport equation turbulence models

II. Numerical Algorithms

The Navier-Stokes equations may be written in generalized coordinates as

$$\frac{\partial \vec{q}}{\partial t} + \frac{\partial \vec{E}}{\partial \xi} + \frac{\partial \vec{F}}{\partial \eta} + \frac{\partial \vec{G}}{\partial \zeta} = 0 \tag{1}$$

where $\vec{q}$ is the vector of conserved variables

$$\vec{q} = V \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ \rho w \\ \rho e_0 \end{bmatrix} \tag{2}$$

The traditional linearized implicit form of Eq. (1) including subiterations is given by

$$\left[I + \frac{\Delta t}{(1+\theta)\Delta\tau} + \frac{\Delta t}{1+\theta} \left(\partial_\xi A + \partial_\eta B + \partial_\zeta C \right) \right] \Delta q^{n+1,m+1} =$$

$$-\left[\left(q^{n+1,m} - q^n \right) - \frac{\theta}{1+\theta} \Delta q^n + \frac{\Delta t}{1+\theta} RHS^{n+1,m} \right] \tag{3}$$

Here $\theta = 0$ for first order time differencing, and $\theta = \frac{1}{2}$ for second order time differencing. An artificial time term $\left(\dfrac{\Delta t}{(1+\theta)\Delta\tau} \right)$ has been explicitly added. The pseudo time ($\Delta\tau$) may vary throughout the flow field when a local time step is employed. The artificial time term must converge at each physical time step (i.e. $\Delta q^{n+1,m+1} = 0$) to assure time accuracy. The explicit viscous and inviscid fluxes are included in the term *RHS* given by

$$RHS = \frac{\partial \vec{E}}{\partial \xi} + \frac{\partial \vec{F}}{\partial \eta} + \frac{\partial \vec{G}}{\partial \zeta} \tag{4}$$

Eq. (3) has the general matrix form **Ax=b**. The first bracketed term in Eq. (3) is the left hand side matrix **A**. The second bracketed term in Eq. (3) represents the vector **b**. $\Delta q^{n+1,m+1} = q^{n+1,m+1} - q^{n+1,m}$ contains the change in the solution vector at the latest time step ($n+1$) and subiteration ($m+1$) when subiterations are employed in the solution process. $\Delta q^{n+1,1} = q^{n+1,1} - q^{n,1}$ contains the change in the solution vector at the latest time step ($n+1$) if no subiterations are used. If the time step remains constant everywhere in the field then Eq. (3) represents a Newton subiteration. If the pseudo time step is allowed to vary throughout the field then Eq. (3) represents a dual time stepping algorithm. In both cases the subiteration is used to improve the accuracy of the solution at each global time step. The individual grids are solved implicitly, but the overset interpolated boundaries are updated explicitly at each subiteration. Hence the subiterations improve the global convergence at each time step by allowing a global exchange of information among the grids.

Solving the above system of discreet equations requires inversion of the **A** matrix. Direct inversion of the matrix **A** for three dimensional flows requires a large amount of computational time and memory. Various approximations have been made to expedite the procedure in the past. Eq. (3) can be factored[17] in space

$$\left[I + \frac{\Delta t}{1+\theta}\partial_\xi A\right]\left[I + \frac{\Delta t}{1+\theta}\partial_\eta B\right]\left[I + \frac{\Delta t}{1+\theta}\delta_\zeta C\right]\Delta q^{n+1,m+1} =$$

$$-\left[\left(q^{n+1,m} - q^n\right) - \frac{\theta}{1+\theta}\Delta q^n + \frac{\Delta t}{1+\theta}RHS^{n+1,m}\right] + Error \tag{5}$$

where the factorization error (*Error*) is given by

$$Error = -\left(\frac{\Delta t}{1+\theta}\right)^2\left(\delta_\xi A\delta_\eta B + \partial_\xi A\delta_\zeta C + \delta_\eta B\delta_\zeta C\right) - \left(\frac{\Delta t}{1+\theta}\right)^3\left(\partial_\xi A\delta_\eta B\delta_\zeta C\right) \tag{6}$$

The factorization error term in Eq. (5) is usually ignored resulting in an approximate factorization of Eq. (3). The factorization error is scaled by the time step squared and cubed for three dimensional calculations. The factorization error is only scaled by the time step and the time step squared for two dimensional and axisymmetric calculations. The factorization error can limit or prevent convergence for large time steps.

The approximate factorization in Eq. (5) is called a three factor alternating direction implicit (ADI) scheme. *A*, *B*, and *C* are block tridiagonal matrices for structured grids with central difference or first order spatial upwind implicit flux jacobians. The factored system can be solved efficiently by inverting the block tridiagonal matrices in each direction.

The *A*, *B*, and *C* matrices in Eq. (5) may be decomposed into eigenvalues (Λ) and eigenvectors (*X*) as

$$A = X_A\Lambda_A X_A^{-1}$$
$$B = X_B\Lambda_B X_B^{-1} \tag{7}$$
$$C = X_C\Lambda_C X_C^{-1}$$

Pulliam and Chaussee[9] suggested pulling out the eigenvector matrices from Eq. (5), producing the following system of equations

$$X_A\left[I + \frac{\Delta t}{1+\theta}\delta_\xi\Lambda_A\right]X_A^{-1}X_B\left[1 + \frac{\Delta t}{1+\theta}\delta_\eta\Lambda_B\right]X_B^{-1}X_C\left[I + \frac{\Delta t}{1+\theta}\delta_\zeta\Lambda_C\right]X_C^{-1}\Delta q^{n+1,m+1} =$$

$$-\left[\left(q^{n+1,m} - q^n\right) - \frac{\theta}{1+\theta}\Delta q^n + \frac{\Delta t}{1+\theta}RHS^{n+1,m}\right] + Error \tag{8}$$

Eq. (8) results in a scalar pentadiagonal matrix form in each factored direction when mixed second and fourth order smoothing is included on the implicit side of the equation. The approximations used to derive Eq. (8) do affect time accuracy for CFL numbers greater than one. The inversion of a scalar pentadiagonal matrix at each point can be done very efficiently, making this algorithm extremely fast in terms of time/iteration/point. The diagonal schemes implemented in OVERFLOW 2 have made the code one the fastest available for obtaining steady state solutions.

The Pulliam-Chaussee[9], Beam-Warming[17], and diagonally dominate ADI (DDADI)[18] implicit algorithms in OVERFLOW 2 are three factor ADI schemes. The F3D[19] algorithm is a partially flux-split two factor ADI scheme that uses Steger-Warming flux vector splitting. The LUSGS[10] algorithm currently in OVERFLOW 2 is an unfactored scheme, but does not iterate the matrix solution at each subiteration. Dual time stepping and local time stepping methods have been implemented to improve convergence and stability for these algorithms. For many unsteady applications, the time step required for numerical stability is too small to be practical. The three factor ADI schemes have been the preferred solution algorithm in the past because of their low memory requirements and because they are relatively fast in terms of time/point/iteration. For a grid of dimensions (jd, kd, ld) the memory required to store the flux jacobian matrices during the solution process is 19*max(jd*kd, jd*ld, kd*ld) for the scalar

pentadiagonal algorithm and 80*max(jd*kd, jd*ld, kd*ld) for the Beam-Warming[18] algorithm since the matrices may be inverted a plane at a time in each computational direction.

A. SSOR Solution Algorithm

Several flow solvers have been developed to solve the unfactored system of equations (Eq. (3)) using relaxation procedures. This approach eliminates the factorization error at the expense of more computational work per time step and more computational memory since the entire implicit flux jacobian matrix (**A**) must be stored for the solution process. NXAIR[20] uses an SSOR procedure. UNCLE[21], TURBO[22], and BEGGAR[23] use a Gauss-Seidel solution procedure to solve the unfactored system. All four of these codes have been used for a wide range of unsteady and moving body simulations. The unfactored approach from NXAIR was implemented in OVERFLOW 2. First order upwind Steger-Warming[19] inviscid flux Jacobians and central differenced thin-layer viscous Jacobians are used. The NXAIR SSOR procedure can be written as

$$
\begin{aligned}
\Delta q_{j,k,l}^{mm+1} &= (1-\Omega)\Delta q_{j,k,l}^{mm} + \Omega(\overline{RHS} - \overline{A}_L \Delta q_{j-1,k,l}^{mm-1} - \overline{A}_R \Delta q_{j+1,k,l}^{mm-1} \\
&- \overline{B}_L \Delta q_{j,k-1,l}^{mk1} - \overline{B}_R \Delta q_{j,k+1,l}^{mk2} - \overline{C}_L \Delta q_{j,k,l-1}^{ml1} - \overline{C}_R \Delta q_{j,k,l+1}^{ml2})
\end{aligned}
\tag{9}
$$

The subscripts L and R denote the left and right blocks respectively of the tridiagonal matrices. The overbar indicates a premultiply by the inverse of the diagonal matrix $A_D+B_D+C_D$ where the subscript D denotes the diagonal block of the tridiagonal matrices. The update level of Δq during the iterative matrix solution procedure is given by mm. The scheme uses a forward (Jacobi) sweep in j and symmetric (Gauss-Seidel) sweeps in k and l. For a forward sweep in k and l, the update levels are defined as

$$
mk1 = mm+1,\ mk2 = mm,\ ml1 = mm+1,\ ml2 = mm
\tag{10}
$$

For a backward sweep in k and l, the update levels are defined as

$$
mk1 = mm,\ mk2 = mm+1,\ ml1 = mm,\ ml2 = mm+1
\tag{11}
$$

A symmetric sweep consists of a forward and a backward sweep. Multiple symmetric sweeps (normally 10) are performed at each subiteration. The relaxation parameter (Ω) is normally set to 0.9. The memory required for storing the flux jacobians for a grid of dimensions (jd, kd, ld) is 175*jd*kd*ld since the entire flux jacobian is required for the inversion process. Thus the SSOR algorithm requires 2.2*min(jd, kd, ld) times more memory to store the implicit flux jacobians than a tridiagonal ADI solver and 19*min(jd, kd, ld) times more memory than a diagonal ADI solver.

A three factor ADI factored Beam and Warming[17] algorithm based on the upwind Steger-Warming[19] inviscid flux jacobians was also added to OVERFLOW 2 for use in problems with smaller time step requirements. The upwind Steger-Warming inviscid flux jacobians are more compatible with upwind Riemann solvers than are central differenced flux jacobians coupled with second order implicit smoothing currently in the code.

B. HLLEM and HLLC Flux Algorithms

Several upwind flux methods have been added to OVERFLOW[10,11,13] to improve the numerical accuracy of the code for high speed applications. Improvements to the implicit algorithm will help to increase the numerical stability of these flux methods, but it was also felt that a more robust upwind flux algorithm was required. The HLL family of upwind approximate Riemann flux algorithms was derived by splitting the flow at a node into two pressure waves and a contact discontinuity. The NXAIR code was developed around the HLLEM[15] Riemann algorithm. The HLLEM algorithm neglects the contact surface and adds anti-diffusion terms to reduce the numerical diffusion that results from neglecting the contact surface. The HLLEM algorithm compares with the Roe schemes for accuracy, but has proven to be more robust than the Roe scheme. The HLLC[16] Riemann algorithm has recently emerged and shows promise for high-speed applications. The HLLC algorithm is also based on two pressure waves at a node, but includes a methodology for handling the contact discontinuity. The HLLC algorithm was found to have the following properties:

1. Exact preservation of isolated contact and shear waves.
2. Positivity-preserving for scalar quantities.

3. Enforcement of the entropy condition.

These properties are highly desirable for high-speed flow simulation and eliminate the need for an entropy fix to prevent carbuncles as required with the Roe algorithm. Both the HLLEM and HLLC algorithms have been added to OVERFLOW 2. Both of these algorithms use Monotone Upstream-centered Schemes for Conservation Laws (MUSCL) extrapolation to reach third order spatial accuracy. The van Albada, minmod, and Koren[24] flux limiters are available for both algorithms. A pressure and temperature switch of the form

$$Switch = \max\left(\frac{p_{i+1} - 2p_i + p_{i-1}}{p_{i+1} + 2p_i + p_{i-1}}, \frac{T_{i+1} - 2T_i + T_{i-1}}{T_{i+1} + 2T_i + T_{i-1}} \right) \tag{12}$$

can be used to force the HLLEM and HLLC algorithms towards first order in space in regions of strong gradients with the van Albada and minmod flux limiters. This can be useful during the convergence process when strong shocks are present or during the initial solution transient.

C. New Turbulence Models
Turbulence models are essential for high Reynolds number simulations. OVERFLOW 2 currently includes the following turbulence models:

1. Baldwin-Lomax (BL) algebraic model[25]
2. Baldwin-Barth (BB) one-equation transport model[26]
3. Spalart-Allmaras (SA) one-equation transport model[27]
4. k-ω two-equation transport model[28]
5. SST two-equation transport model[29]

All of these are low-Reynolds number models requiring integration to the wall. The Spalart-Allmaras model has been the most popular in applications because of its accuracy and numerical stability. The two-equation transport models as implemented in the code have suffered from numerical stability issues in large three dimensional applications and have been little used. All of the currently implemented transport models rely on the diagonally dominant ADI (DDADI)[30] solution algorithm and are solved loosely coupled with the mean flow equations within the Newton or dual-time subiteration loop.

A common scalar transport equation solution algorithm was added for the two-equation turbulence models and species equations. The transport equation of a scalar s in generalized coordinates has the form

$$\frac{\partial \rho s}{\partial t} + \frac{\partial \rho Us}{\partial \xi} + \frac{\partial \rho Vs}{\partial \eta} + \frac{\partial \rho Ws}{\partial \zeta} = \frac{\partial}{\partial \xi}\left[\left(\frac{\mu}{\sigma_L} + \frac{\mu_t}{\sigma_T}\right)\left(\xi_x^2 + \xi_y^2 + \xi_z^2\right)\frac{\partial s}{\partial \xi}\right] +$$

$$\frac{\partial}{\partial \eta}\left[\left(\frac{\mu}{\sigma_L} + \frac{\mu_t}{\sigma_T}\right)\left(\eta_x^2 + \eta_y^2 + \eta_z^2\right)\frac{\partial s}{\partial \eta}\right] + \frac{\partial}{\partial \zeta}\left[\left(\frac{\mu}{\sigma_L} + \frac{\mu_t}{\sigma_T}\right)\left(\zeta_x^2 + \zeta_y^2 + \zeta_z^2\right)\frac{\partial s}{\partial \zeta}\right] +$$

$$\frac{\partial}{\partial \xi}\left[\left(\frac{\mu}{\sigma_L} + \frac{\mu_t}{\sigma_T}\right)\left(\xi_x\eta_x + \xi_y\eta_y + \xi_z\eta_z\right)\frac{\partial s}{\partial \eta}\right] + \frac{\partial}{\partial \xi}\left[\left(\frac{\mu}{\sigma_L} + \frac{\mu_t}{\sigma_T}\right)\left(\xi_x\zeta_x + \xi_y\zeta_y + \xi_z\zeta_z\right)\frac{\partial s}{\partial \zeta}\right] + \tag{13}$$

$$\frac{\partial}{\partial \eta}\left[\left(\frac{\mu}{\sigma_L} + \frac{\mu_t}{\sigma_T}\right)\left(\xi_x\eta_x + \xi_y\eta_y + \xi_z\eta_z\right)\frac{\partial s}{\partial \xi}\right] + \frac{\partial}{\partial \eta}\left[\left(\frac{\mu}{\sigma_L} + \frac{\mu_t}{\sigma_T}\right)\left(\eta_x\zeta_x + \eta_y\zeta_y + \eta_z\zeta_z\right)\frac{\partial s}{\partial \zeta}\right] +$$

$$\frac{\partial}{\partial \zeta}\left[\left(\frac{\mu}{\sigma_L} + \frac{\mu_t}{\sigma_T}\right)\left(\xi_x\zeta_x + \xi_y\zeta_y + \xi_z\zeta_z\right)\frac{\partial s}{\partial \xi}\right] + \frac{\partial}{\partial \zeta}\left[\left(\frac{\mu}{\sigma_L} + \frac{\mu_t}{\sigma_T}\right)\left(\eta_x\zeta_x + \eta_y\zeta_y + \eta_z\zeta_z\right)\frac{\partial s}{\partial \eta}\right] +$$

$$+ Source\ Terms$$

where *U*, *V*, and *W* are contravarient velocities. The full diffusion terms are used in this formulation. Eq. (13) can be discretized into the form of Eq. (3). The convective terms are solved using upwind fluxes. MUSCL extrapolation of primitive variables (ρ, *u, v, w*) and the conservative scalar variable (ρs) is used to reach third order in space using an HLL based Riemann scheme. The convective fluxes for the turbulence equations are limited to second order in space. The equations are solved using an SSOR algorithm and are loosely coupled from the other scalar equations and the mean flow equations through the Newton or dual time subiteration loop. The following turbulence models that utilize the new solution algorithm have been added to OVERFLOW 2

1. k-ω two-equation transport model[28]
2. SST two equation transport model[29] including a compressibility correction[32]
3. SST DES hybrid model[33]
4. SST Multi-Scale (MS) hybrid model[34]

The Spalart-Allmaras Detached Eddy Simulation (DES) hybrid RANS/LES turbulence model[31] has also been added utilizing the existing Spalart-Allmaras model in OVERFLOW 2.

The new two-equation models include stability enhancing features in the treatment of the implicit and explicit source terms. The turbulent production in the turbulent kinetic energy equation is limited as suggested by Menter[35] by

$$P_k = \min(P_k, 20\varepsilon) \tag{14}$$

This approximation is justified by the fact that these turbulence models were developed for near equilibrium turbulence (production = dissipation). As the turbulence moves farther from equilibrium, the models are not valid and need limiting to avoid numerical problems. This modification removes the problems often encountered in the stagnation regions of blunt bodies with two-equation models. The turbulent kinetic energy equation (TKE) implicit source term is given by

$$\text{TKE source term Jacobian} = -2\frac{P_k + \varepsilon}{k} \tag{15}$$

This helps to maintain diagonal dominance for the implicit matrix and improves the numerical stability of the model. A user specified limit on the maximum eddy viscosity has been added to the namelist input for each grid. The default limit on eddy viscosity (μ_t/μ_∞) is set at 200,000. The new two equation models have robustness similar to the existing Spalart-Allmaras one-equation model.

D. Wall Functions

Wall function boundary conditions[36] were added to OVERFLOW 2. Wall functions allow the first grid point from the wall to be placed at $y^+<100$. This can reduce the number of grid points required for a simulation, improve numerical stability, and expand the range of Reynolds numbers for which a given grid can be used. The wall function formulation in Ref. 36 replaces the wall shear stress and heat transfer in the viscous terms of the Navier-Stokes code with values from boundary layer empirical relationships. The wall functions use a single function to span the boundary layer sub-layer and log-layer. The turbulence transport variables and the eddy viscosity are replaced with empirically derived values at the first point off the wall. The wall function formulation added to OVERFLOW 2 includes the effects of compressibility and heat transfer. The wall functions automatically turn themselves off when the first point from the wall is below $y^+=10$. This automatic feature can be used to improve convergence during grid sequencing and multigrid cycles since the wall functions provide a better estimate of the wall shear stress and heat transfer as the distance between the first point and the wall is increased.

The wall functions work with all of the existing and new transport turbulence models in OVERFLOW 2. All of the inviscid flux options within OVERFLOW 2 have been modified to be compatible with wall functions. The fourth order smoothing is turned off at the first point off the wall for the central difference inviscid flux algorithms and the symmetric Yee flux algorithm[37] as recommended in Ref. 36. The Koren flux limiter[24] used by the AUSM, Roe, HLLEM, and HLLC flux algorithms is turned also off at the first point off the wall. The minmod and van

Albada limiters used by the HLLEM and HLLC flux algorithms require no modification. The forces calculated within OVERFLOW 2 have also been modified to use the wall function derived shear stress where appropriate.

E. Species Equations

The original species transport equations formulation in OVERFLOW 2 included convection terms only. Central difference with fourth order smoothing and first, second, or third order upwind fluxes were available for the convective terms. The DDADI scheme was used to provide the implicit solution. A new solution option for the species transport equations based on the generalized transport equation (Eq. (13)) was added to OVERFLOW 2. The HLL based upwind convective fluxes use MUSCL extrapolation to obtain second or third order spatial accuracy. The minmod flux limiter is used in the extrapolation process. Diffusion terms are also included for viscous calculations. The equations are solved using the unfactored SSOR approach. The species equations are solved loosely coupled with each other and the mean flow and turbulence equations through the Newton or dual time subiteration loop.

III.   Results

A.   Timing Study

Timings of the algorithms in OVERFLOW 2 were obtained for a 50x50x50 and a 100x100x100 Cartesian grid in terms of time/iteration/point. The timings included calculation of the full viscous terms and the new SST two-equation turbulence model. The grids were run on a Linux workstation with a 2.8 GHz Intel Xeon processor and 512 kb of cache. The code was compiled with the Intel 8.0 FORTRAN compiler with vectorization. The results are shown in Tables 1 and 2. IRHS and ILHS represent the OVERFLOW 2 input variables for the inviscid flux and the solution algorithm respectively. The diagonalized solution algorithms (ILHS=2, 3, and 4) are the fastest per iteration for both grids. This speed advantage is often offset by slower convergence and less robustness than the slower Beam and Warming algorithms (ILHS=0 and 5). The SSOR algorithm (ILHS=6) is the slowest, but it is also the most robust for many applications. The difference in timings (10%-30%) between the two grids is probably due to the small cache size of the test machine.

B. Inviscid Vortex Convection

The ability to conserve the vortex shape and strength is important in many unsteady cases in which a shed vortex interacts with bodies well downstream of the vortex origin. This case can also be used to examine the level of numerical dissipation and dispersion for a given flux algorithm. A vortex of strength $\Gamma=5$ is centered on an 81x81 uniform grid in the x-z plane and allowed to convect downstream at M=0.5 with a non-dimensional time step of 0.1. The grid spacing was set to 0.25 in both the x and z directions. This grid spacing is somewhat coarse for this application, but was chosen because it allows the limitations on the numerical schemes to be clearly seen. The vortex is given by

$$\omega = \frac{\Gamma(2-\overline{R})}{2\pi}\exp\left[0.5\left(1-\overline{R}\right)\right] \tag{16}$$

where $\overline{R}^2 = \left(x-x_0\right)^2 + \left(z-z_0\right)^2$ and x_0 and z_0 represent the location of the vortex center. The grid is given periodic boundary conditions in the flow direction. This allows the vortex to convect out of and back into the computational domain. The vortex should complete one cycle on the grid (i.e. return to its initial location) every 200 time steps. The vortex was allowed to convect for a non-dimensional length of 100 (five cycles through the grid). All of the calculations were run with second order time and three Newton subiterations. The Roe and HLLC algorithms were run third order in space. The fourth order central difference algorithm was run with a fourth order smoothing coefficient of 0.001 and a second order smoothing coefficient of 0.0.

Vorticity contours for several flux algorithms and the initial vortex are shown in Fig. 1. The vorticity profiles and the vortex location are well preserved by the Roe and HLLC algorithms using the Koren and minmod flux limiters after five cycles through the grid. The vortex core is under-resolved on this computational grid. The fourth order central difference algorithm maintains the vortex shape and position, but also tends to underpredict the minimum vorticity in the vortex core. The HLLC algorithm with the van Albada flux limiter both dissipates and distorts the vortex.

The minimum pressure in the vortex as a function of time is shown in Fig. 2 for several of the flux algorithms provided by OVERFLOW 2. In theory this pressure should be preserved for all time. The performance of the various algorithms is similar to that shown in Fig. 1. The Roe and HLLC algorithms yield similar results for the Koren flux limiter or when no limiter is employed.

C. NASA Ames Axisymmetric Bump

The NASA Ames transonic axisymmetric bump experiment[38] provides a good test of the ability of a code to predict separated flow. The geometry is shown in Fig. 3. The model consisted of a sharp-lipped hollow cylinder with a 15.2 cm outer surface diameter. The bump was a circular arc 20.3 cm long and 1.9 cm high that begins 60.3 cm downstream of the cylinder leading edge. The upstream intersection of the bump and cylinder was faired with a circular arc. The test was run at a Mach number of 0.875 and a chord Reynolds number of 2.67×10^6. Solutions were performed on a grid with an initial wall spacing corresponding to a $y^+ = 1$ and $y^+ = 50$. The stretching ratio for the grid was held to 1.2 away from the wall.

The surface pressure coefficient for various inviscid flux models using the new SST turbulence model is shown in Fig. 4 for an initial wall spacing of $y^+ = 1$. The velocity distribution at x/c=1 is shown in Fig. 5. The solutions are quite similar for all of the inviscid flux models. The surface pressure coefficient using wall functions in conjunction with the new SST model is shown for the various inviscid flux routines in OVERFLOW 2 for a wall spacing of $y^+ = 50$ are shown in Fig. 6. The velocity distribution for x/c=1 is shown in Fig. 7. There is some small variation in shock location prediction for the various flux routines, but overall the solutions are similar to the $y^+ = 1$ results. The surface pressure coefficient for the three transport turbulence models (SA, k-ω, and SST) are shown for both grid spacings in Fig. 8. The velocity profile at x/c=1 for the three models are shown in Fig. 9. The solutions for each individual model are similar for both wall spacings. This indicates that the wall functions are performing as expected.

D. Multi-Species

The OVERFLOW 2 three gas test case was run to demonstrate the species transport equation solver. A jet of a gas of one species is allowed to blow through a gas of a second species. Both gases have the same physical properties. All of the fluid is given a free stream Mach number of 0.2 and the flow is inclined at 30 degrees relative to the grid. No viscous terms are included in this test case. The 61x61 grid is uniform with a spacing of 0.1. This case demonstrates the level of numerical diffusion in a given algorithm. Fig. 10 shows contours of the jet mass fraction for the transport model solved using third order spatial convection terms and the minmod flux limiter. A vertical slice through the jet for the three species convective algorithms in OVERFLOW 2 is shown in Fig. 11. The original third order in space upwind and the third order in space transport model have low diffusion for this relatively coarse mesh and produce similar results. The central difference algorithm has greater numerical diffusion.

E. Supersonic Shear Layer

The need for a turbulence model compressibility correction can be seen in predictions of the supersonic axisymmetric jet shear layer of Eggers[39]. The jet exit Mach number was 2.22 and the jet exit static pressure was matched to the quiescent outer air. The solutions were run using the HLLC inviscid flux and the SSOR solution algorithm. A nondimensional time step of 0.5 (based on U_{inf}) was used without local time stepping. Fig. 12 shows the predicted and measured axial velocity on the jet centerline for the transport turbulence models in OVERFLOW 2. Fig. 13 contains velocity comparisons at $x/R_{noz} = 100$ downstream locations in the jet. The compressibility corrected SST model has the best comparison with the data. The uncorrected SST, k-ω, and Spalart-Allmaras turbulence models produce mixing consistent with each other. These models overpredict the mixing resulting in a shear layer that is too diffusive. The Baldwin-Barth model underpredicts the mixing in the shear layer and produces a somewhat unphysical profile shape.

F. WICS L/D=4.5 Bay

Unsteady computations for the Weapons Internal Carriage and Separation (WICS)[40] L/D=4.5 bay were performed for M=0.95 and Re=2.5×10^6/ft. The weapons bay was 18 in. long, 4 in. wide, and 4 in. deep. The bay was located behind a 15 in. flat plate in the experimental configuration. The computational geometry was a flat plate that extended 15 in. upstream of the bay to match the experimental geometry and 25 in. downstream of the bay. The sides of the computational grid extended 50 in. on either side of the bay centerline. The full bay geometry was modeled using wall functions. The wall spacing was chosen as 0.0075 in., which corresponds to a y^+ of 50 on the upstream plate. The wall spacing inside the bay was set to 0.075 in. The larger wall spacing may be used inside the

bay since the wall shear stress is much lower there. The HLLC inviscid fluxes were used with the SSOR solution algorithm and the SST-MS[33] hybrid RANS/LES turbulence model. The calculations were run 12,000 iterations and the final 8192 time steps were statistically analyzed. The entire computational grid had 1.1×10^6 points, and 1.2×10^5 points were used to discretize the bay. All calculations were performed using the 1.6×10^{-5} second time step that was shown to be adequate for time accuracy in Ref. 41.

Instantaneous Mach number contours on the bay centerline are shown in Fig. 14. The flow in the bay is highly unsteady. The time-averaged pressure coefficient on the WICS bay ceiling is shown in Fig. 15. The hybrid turbulence model results are in good agreement with the data. The overall sound pressure level (OASPL) on the ceiling is shown in Fig. 16. Again the hybrid model result is in reasonable agreement with the data. Spectral results for the K18 transducer locations are shown in Fig. 17. K18 is located on the bay back wall centerline 0.725 inches from the bay opening (Fig. 14). The K18 transducer is in a dynamic region of the flow because of its proximity to the shear layer above the bay. Seven data windows of 2048 samples were averaged to produce the spectra that are presented. The error[41] in the OASPL (defined as the difference of the individual window result and the averaged result divided by the averaged result) was less than one percent. The hybrid turbulence model results are in good agreement with the data for the first three spectral peaks.

G. Oscillating Airfoil

The low angle-of-attack NACA0015 oscillating airfoil case of Ref. 42 is a simple two-dimensional moving body problem for turbulent flows. This case was used as a validation case for unsteady flow for both NXAIR and OVERFLOW 2 in Ref. 43. The flow remains fully attached to the airfoil for most of the cycle, allowing two-dimensional simulations to adequately resolve the physics of the problem. Simulations at higher angle-of-attack would require three-dimensional simulations to capture the large-scale flow separation. The case chosen for this simulation used the angle-of-attack (α) variation given by

$$\alpha = 4^o + 4.2^o \sin\left(2\pi f t\right) \tag{17}$$

where f is the frequency of oscillation (10 cps). The conditions of the simulation were a Mach number of 0.29 and a chord Reynolds number of 1.95×10^6.

A 441x71 "C" grid was used to discretize the NACA0015 airfoil. The grid included 241 points along the airfoil and was packed at both the leading and trailing edges. The wall spacing was set for a $y^+=1$ along the airfoil. The grid is the same as used in the Ref. 43 study. The time step corresponds to 512 (1/5120 sec.) steps per pitch cycle. The calculation was performed using third order HLLC inviscid fluxes, the SSOR solution algorithm, and the Spalart-Allmaras turbulence model. Ten Newton subiterations were used per time step. Local time stepping was not employed. The solution was run using second order time.

Force and moment coefficients as a function of airfoil angle-of-attack are shown in Figs. 18-20. The two-dimensional simulation is in good agreement with the data and with the results from Ref. 43. The initial transient from the steady state solution disappears after about 0.75 pitch cycles.

H. Wing/Pylon/Store

The wing/pylon/store configuration of Lijewski and Suhs[44] represents a more complicated three dimensional test case. The full-scale wing has a mean chord of 14.16667 ft. and the metric store length is 9.902 ft. The wind tunnel model was 1/20th scale. The free stream Mach number was 0.95 and the Reynolds number (per foot) was 1.2×10^5. The trajectory was performed for a simulated altitude of 26,000 ft. The overset grid system has 3.4×10^6 points in 58 grids (20 near-body grids and 38 off-body grids). The grid wall spacing corresponds to a $y^+=1$. The solution was run using the third order spatial HLLC algorithm and the SSOR solution algorithm. The new SST turbulence model was used in this simulation. The nondimensional global time step was 0.0005 seconds. Three levels of grid sequencing (coarse, medium, and fine) were used for 150 time steps each on the coarse and medium grids. The solution on the fine grid was run 1000 steps to obtain the carriage loads using three Newton subiterations per time step with second order time. No local time stepping was used. The moving body was simulated for 0.5 sec. (1000 time steps). Table 3 contains the mass and ejector properties for the store.

The pressure coefficient distribution on the surface for the carriage location is shown in Fig. 21. The force and moment convergence history on the fine mesh is shown in Figures 22 and 23 respectively. The loads converge in about 500 iterations on the fine grid. The predicted and experimental location of the store center-of-gravity is shown in Fig. 24. There is excellent agreement for the full trajectory. The predicted and experimental orientation of

the store is shown in Fig. 25. The CFD predictions and the experiment are in good agreement for the first 0.2 seconds of the trajectory. Experimental and predicted force and moment coefficients are shown in Figures 26 and 27 respectively. There is good agreement between CFD and experiment for the first 0.3 seconds of the trajectory.

I. Hypersonic Cylinder Bow Shock

The hypersonic bow shock experiment data of Holden, et. al[45] provides a good test of the ability of the code to predict heat transfer at high speeds. The experiment was run at a free stream Mach number of 16.01, a Reynolds number (based on cylinder diameter) of 9.11×10^4, a free stream temperature of 77.8^o R, and a wall temperature of 540° R. The test medium can still be considered a perfect gas with a ratio of specific heats of 1.4 at these conditions. The flow over the cylinder is laminar. This case could not be run with the original algorithms in OVERFLOW 2. The new SSOR solution algorithm and third order HLLC inviscid fluxes was used for this application. The time step was ramped to a nondimensional value (based on free stream velocity) of 0.005. Local time stepping was not used. Three Newton iterations were used per time step. Grid wall spacings corresponding to y^+ values of 0.4, 0.2, and 0.1 were run using an 81x81 grid.

Pressure contours are shown in Fig. 28. The shock is captured cleanly in four points. This can also be seen in the centerline Mach number distribution in Fig. 29. The convergence of the axial force coefficient is shown in Fig. 30. The solution required about 5500 iterations to reach convergence. The surface pressure coefficient is shown in Fig. 31. The prediction compares well with modified Newtonian theory. Predicted heat transfer is compared to data in Fig. 32. The code does an excellent job in predicting the heat transfer for this case. The change in the predicted stagnation heat transfer with initial wall spacing is shown in Fig. 33. The predicted heat transfer changes by less then five percent over the range shown and varies linearly with initial wall spacing.

J. Hypersonic Double Cone

The hypersonic double cone experiments of Holden, et. al[46] provides a very demanding test of the ability of the code to predict the heat transfer at high speeds. The geometry for this case is shown in Fig. 34, while the details of the flow field are shown in Fig 35. The attached shock from the first cone interacts strongly with the detached shock associated with the second cone. This shock/shock interaction produces a transmitted shock which in turn impinges on the second cone. This impingement produces very high surface heat transfer rates and pressures. In addition, the high pressures which result in the cone-cone junction cause the flow to separate in this region. This separation bubble in turn interacts with the inviscid flow field, which impacts the strength of the transmitted shock, etc. and the entire flow field is very dependent on the strengths of the relative interactions. Computational predictions of this type of flow are very sensitive to numerics, grid resolution, etc. (Druget, et. al[47]).

The OVERFLOW 2 computational results are compared to the experimental results of Holden's Run 35. The experiment was run at a free stream Mach number of 12.06, a Reynolds number of 6.79×10^4 per foot, a free stream temperature of 182° R and a wall temperature of 533° R. The test medium was pure nitrogen (N_2) and is considered a perfect gas at these conditions. The flow conditions were chosen such that the flow over the entire length of the cone remains laminar. This case could not be run with the original upwind algorithms in OVERFLOW 2. The new SSOR solution algorithm and third order HLLC inviscid flux was used for this application. The time step was ramped to a nondimensional value (based on free stream velocity) of 0.0032. Local time stepping was not used. Three Newton subiterations were used per time step. Computations were run on two grids of dimension 609x257x3 and 1217x513x3. Results obtained on these grids are referred to as the medium and fine grid solutions respectively. The solution required about 5000 iterations to reach convergence.

Mach number contours for the entire flow field are shown in Fig. 36, and a blow up of the contours in the region of the cone/cone intersection is shown in Fig 37. Heat transfer predictions for both the medium and fine grids are shown in Fig 38. The code does an excellent job in predicting the heat transfer for this case, and the solution is reasonably well grid converged.

IV.   Conclusion

An implicit unfactored SSOR algorithm has been added to OVERFLOW 2 for unsteady and moving body applications. The HLLEM and HLLC convective flux models have been added for high-speed flow applications. A generalized transport equation has been added for solution of the two-equation transport turbulence models and the species equations. The generalized transport equation is solved using an unfactored SSOR algorithm. Three hybrid RANS/DES turbulence models have been added for unsteady flow applications. Wall function boundary conditions that include compressibility and heat transfer effects have been also been added to OVERFLOW 2.

Steady flow, unsteady flow, moving body, and high speed flow test cases have been provided to demonstrate the accuracy, speed, and robustness of the new algorithms. The unfactored SSOR algorithm can normally be run without local time stepping and is more robust than the factored implicit algorithms during the initial transient of a calculation. The HLLC flux scheme was shown to produce good results for a wide range of Mach numbers. The wall functions were shown to produce reasonable results even for separated flows.

The new algorithms provide a good starting point for extending OVERFLOW 2 to chemically reacting flows. The transport equation solver is robust and accurate. The HLLC scheme can handle extremely large gradients in the flow. Source terms will have to be added for chemically reacting flows. The code will also require a general method for inputting fluid properties and reaction rates.

References

[1]Murphy, K. J., Buning, P. G., Pamadi, B. N., Scallion, W. I., and Jones, K. M., "Overview of Transonic to Hypersonic Stage Separation Tool Development for Multi-Stage-to-Orbit Concepts," AIAA-2004-2595, Jun. 2004.

[2]Buning, P. G., Gomez, R. J., and Scallion, W. I., "CFD Approaches for Simulation of Wing-Body Stage Separation," AIAA-2004-4838, Aug. 2004.

[3]Buning, P. G., Chiu, I. T., Obayashi, S., Rizk, Y. M., and Steger, J. L., "Numerical Simulation of the Integrated Space Shuttle Vehicle in Ascent," AIAA-88-4359, Aug. 1988.

[4]Renze, K. J., Buning, P. G., and Rajagopalan, "A Comparative Study of Turbulence Models for Overset Grids," AIAA-92-0437, Jan. 1992.

[5]Meakin, R. L., "Object X-Rays for Cutting Holes in Composite Overset Structured Grids," AIAA-2001-2537, Jun. 2001.

[6]Meakin, R. L., "Automatic Off-Body Grid Generation for Domains of Arbitrary Size," AIAA-2001-2536, Jun. 2001.

[7]Chan, W. M., Meakin, R. L., and Potsdam, M. A., "CHSSI Software for Geometrically Complex Unsteady Aerodynamic Applications," AIAA-2001-0539, Jan. 2001.

[8]Benek, J. A., Buning, P. G., and Steger, J. L., "A 3-D CHIMERA Grid Embedding Technique," AIAA-85-1523-CP, Jul. 1985.

[9]Pulliam, T. H., and Chaussee, D. S., "A Diagonalized Form of an Implicit Approximate Factorization Algorithm," *J. Comp. Physics*, Vol. 39, 1981.

[10]Kandula, M. and Buning, P. G., "Implementation of LU-SGS Algorithm and Roe Upwinding Scheme in Overflow Thin-Layer Navier-Stokes Code," AIAA-94-2357, Jun. 1994.

[11]Jespersen, D. C., Pulliam, T. H., and Buning, P. G., "Recent Enhancements to OVERFLOW (Navier-Stokes Code)," AIAA-97-0644, Jan. 1997.

[12]Jespersen, D. C., "Parallelism and OVERFLOW," NAS Technical Report NAS-98-013, NASA Ames Research Center, Moffett Field, CA, Oct. 1998.

[13]Liou, M. S., and Buning, P. G., "Contribution of the Recent AUSM Schemes to the OVERFLOW Code: Implementation and Validation," AIAA-2000-4404, Jun. 2000.

[14]Pandya, S. A., Venkateswaran, S., and Pulliam, T. H., "Implementation of Preconditioned Dual-Time Procedures in OVERFLOW", AIAA-2003-0072, Jan. 2003.

[15]Harten, A., Lax, P. D., and van Leer, B., "On Upstream Differencing and Gotlunov-type Scheme for Hyperbolic Conservation Laws," *SIAM Review*, Vol. 25, No. 1, 1983, pp 35-61.

[16]Toro, E. F., Spruce, M., and Speares, W., "Restoration of the Contact Surface in the HLL Riemann Solver," *Shock Waves*, Vol. 4, 1994, pp 25-34.

[17]Beam, R. and Warming, R. F., "An Implicit Finite-Difference Algorithm for Hyperbolic Systems in Conservation Law Form," *J. Comp. Physics*, Vol. 22, Sep. 1976, pp 87-110.

[18]Klopfer, G. H., Hung, C. M., Van der Wijngaart, R. F., and Onufer, J. T., "A Diagonalized Diagonal Dominant Alternating Direction Implicit (D3ADI) Scheme and Subiteration Correction," AIAA-98-2824, Jun. 1988.

[19]Steger, J. L. and Warming, R. F., "Flux-Vector Splitting of the Inviscid Gasdynamics Equations with Application to Finite Difference Methods," *J. Comp. Physics*, Vol. 40, No. 2, Apr. 1981, pp 263-293.

[20]Tramel, R. W. and Nichols, R. H., "A Highly Efficient Numerical Method for Overset-Mesh Moving–Body Problems," AIAA-97-2040, Jun. 1997.

[21]Arabshshi, A., Taylor, L. K., and Whitfield, D. L., "UNCLE: Toward a Comprehensive Time-Accurate Incompressible Navier-Stokes Flow Solver," AIAA-95-0050, Jan. 1995.

[22]Chen, J. P. and Briley, W. R., "A Parallel Flow Solver for Unsteady Multiple Blade Row Turbomachinery Simulations," Proceedings of ASME TURBO EXPO 2001, 2001-GT-0348, Jun. 2001.

[23]Rizk, M., Ellison, S., and Prewitt, N. C., "Beggar – A Store Separation Predictive Tool," AIAA-2002-3190, Jun. 2002.

[24]Koren, B., "Upwind Schemes, Multigrid and Defect Correction for the Steady Navier-Stokes Equations," Proceedings of the 11th International Conference in Numerical Methods in Fluid Dynamics, edited by D. L Dwoyer, M. Y. Hussani, and R. G. Voigt, Springer-Verlag, Berlin 1989.

[25]Baldwin, B. S. and Lomax, H., "Thin Layer Approximation and Algebraic Model for Separated Turbulent Flows," AIAA-78-0257, Jan. 1978.

[26]Baldwin B. S. and Barth, T. J., "A One-Equation Turbulence Transport Model for High Reynolds Number Wall-Bounded Flows," AIAA-91-0610, Jan. 1991.

[27]Spalart, P. R. and Allmaras, S. R., "A One-Equation Turbulence Model for Aerodynamic Flows," AIAA-92-0439, Jan. 1992.

[28]Wilcox, D. C., "Reassessment of the Scale-Determining Equation for advanced Turbulence Models, *AIAA Journal*, Vol. 26, No. 11, 1988, pp. 1299-1310.

[29]Menter, F. R. and Rumsey, C. L., "Assessment of Two-Equation Turbulence Models for Transonic Flows," AIAA-94-2343, Jun. 1994.

[30]Walsh, P. C. and Pulliam, T. H., "The Effect of Turbulence Model Solution on Viscous Flow Problems," AIAA-2001-1018, Jan. 2001.

[31]Spalart, P., Jou, W-H. Strelets, M., and Allmaras, S., "Comments on the Feasibility of LES for Wings and on a Hybrid RANS/LES Approach," First AFOSR Conference on DNS/LES, Aug. 1997, C. Liu and Z. Liu editors, Greyden Press, Columbus, Ohio.

[32]Suzen, Y. B. and Hoffmann, K. A., "Investigation of Supersonic Jet Exhaust Flow by One- and Two-Equation Turbulence Models," AIAA-98-0322, Jan. 1998.

[33]Strelets, M., "Detached Eddy Simulation of Massively Separated Flows," AIAA-2001-0879, Jan. 2001.

[34]Nichols, R. and Nelson, C., "Applications of Hybrid RANS/LES Turbulence Models," AIAA-2003-0083, Jan. 2003.

[35]Menter, F. R., "Zonal Two Equation k-ω Turbulence Models for Aerodynamic Flows," AIAA-93-2906, Jul. 1993.

[36]Nichols, R. H. and Nelson, C. C., "Wall Function Boundary Conditions Including Heat Transfer and Compressibility," *AIAA Journal*, Vol. 42, No. 6, pp 1107-1114, Jun. 2004.

[37]Yee, H. C., "A Class of High-Resolution Explicit and Implicit Shock-Capturing Methods," NASA TM 101088, Feb. 1989.

[38]Johnson, D. A., "Predictions of Transonic Separated Flow with an Eddy-Viscosity/Reynolds Shear Stress Closure Model," AIAA-85-1683, Jul. 1985.

[39]Eggers, J. "Velocity Profiles and Eddy Viscosity Distributions Downstream of a Mach 2.22 Nozzle Exhausting to Quiescent Air," NASA-TN D-3601, Sep. 1966.

[40]Dix, R. E. and Bauer, R. C., "Experimental and Theoretical Study of Cavity Acoustics," AEDC-TR-99-4, May 2000.

[41]Nichols, R. H., "Comparison of Hybrid Turbulence Models for a circular Cylinder and a Generic Cavity," accepted for publication in the *AIAA Journal*.

[42]Piziali, R. A., "2-D and 3-D Oscillating Wing Aerodynamics for a Range of Angles of Attack Including Stall," NASA TM 4632, Sep. 1994.

[43]Nichols, R. H. and Heikkinen, B. D., "Validation of Implicit Algorithms for Unsteady Flows Including Moving and Deforming Grids," AIAA-2005-0683, Jan. 2005.

[44]Lijewski, L. and Suhs, N. E., "Chimera-Eagle Store Separation," AIAA-92-4569, Aug. 1992.

[45]Holden, S. M., Kolly, J. M., and Martin, S. C., "Shock/Shock Interaction Heating in Laminar and Low-Density Hypersonic Flows," AIAA-96-1866, Jun. 1996.

[46]Holden, M. S., and Wadhams. T. P., "Code Validation Study of Laminar Shock/Boundary Layer and Shock/Shock Interactions in Hypersonic Flow. Part A: Experimental Measurements," AIAA-2002-1031, Jan. 2002.

[47]Druget, M.C., Candler, G. V., and Nompelis, I. N., "Simulations of Viscous Hypersonic Double-Cone Flows: Influence of Numerics," AIAA-2003-3548, Jun. 2003.

	Central[3] (IRHS=0)	Yee[37] (IRHS=2)	Roe[11] (IRHS=4)	HLLEM[15] (IRHS=5)	HLLC[16] (IRHS=6)
Beam and Warming[17] with central (ILHS=0)	1.93×10^{-5}	1.87×10^{-5}	1.91×10^{-5}	2.04×10^{-5}	2.03×10^{-5}
F3D[19] (ILHS=1)	2.88×10^{-5}	2.67×10^{-5}	2.91×10^{-5}	2.97×10^{-5}	2.91×10^{-5}
Pulliam and Chaussee Diagonalized[9] (ILHS=2)	7.60×10^{-6}	7.70×10^{-6}	8.10×10^{-6}	9.22×10^{-6}	8.74×10^{-6}
Jameson and Yoon LU-SGS[10] (ILHS=3)	7.23×10^{-6}	7.59×10^{-6}	7.82×10^{-6}	8.95×10^{-6}	8.48×10^{-6}
D3ADI[18] (ILHS=4)	7.81×10^{-6}	7.96×10^{-6}	8.40×10^{-6}	9.48×10^{-6}	9.11×10^{-6}
Beam and Warming[17] with Steger-Warming (ILHS=5)	1.92×10^{-5}	1.98×10^{-5}	2.06×10^{-5}	2.18×10^{-5}	2.13×10^{-5}
SSOR[20] (ILHS=6)	3.15×10^{-5}	3.17×10^{-5}	3.26×10^{-5}	3.34×10^{-5}	3.29×10^{-5}

Table 1. Timings (seconds/point/iteration) for OVERFLOW 2 for a 50x50x50 Cartesian grid.

	Central[3] (IRHS=0)	Yee[37] (IRHS=2)	Roe[11] (IRHS=4)	HLLEM[15] (IRHS=5)	HLLC[16] (IRHS=6)
Beam and Warming[17] with central (ILHS=0)	2.51×10^{-5}	2.48×10^{-5}	2.51×10^{-5}	2.62×10^{-5}	2.57×10^{-5}
F3D[19] (ILHS=1)	3.67×10^{-5}	3.60×10^{-5}	3.76×10^{-5}	3.77×10^{-5}	3.71×10^{-5}
Pulliam and Chaussee Diagonalized[9] (ILHS=2)	9.24×10^{-6}	9.07×10^{-6}	1.00×10^{-5}	1.11×10^{-5}	1.06×10^{-5}
Jameson and Yoon LU-SGS[10] (ILHS=3)	9.03×10^{-6}	9.46×10^{-6}	9.78×10^{-6}	1.10×10^{-5}	1.05×10^{-5}
D3ADI[18] (ILHS=4)	9.53×10^{-6}	9.95×10^{-6}	1.01×10^{-5}	1.13×10^{-5}	1.08×10^{-5}
Beam and Warming[17] with Steger-Warming (ILHS=5)	2.58×10^{-5}	2.63×10^{-5}	2.68×10^{-5}	2.77×10^{-5}	2.72×10^{-5}
SSOR[20] (ILHS=6)	3.58×10^{-5}	3.56×10^{-5}	3.58×10^{-5}	3.70×10^{-5}	3.64×10^{-5}

Table 2. Timings (seconds/point/iteration) for OVERFLOW 2 for a 100x100x100 Cartesian grid.

Weight: 2,000 lb
Center of gravity: 4.65 ft aft of the store nose
Roll Moment of Inertia: 20 slug-ft^2
Pitch Moment of Inertia: 360 slug-ft^2
Yaw Moment of Inertia: 360 slug-ft^2
Forward Ejector Location: 4.06 ft aft of store nose
Forward Ejector Force: 2,400 lb
Aft Ejector Location: 5.73 ft aft of store nose
Aft Ejector Force: 9,600 lb

Table 3. Store mass and ejector properties for the wing/pylon/store.
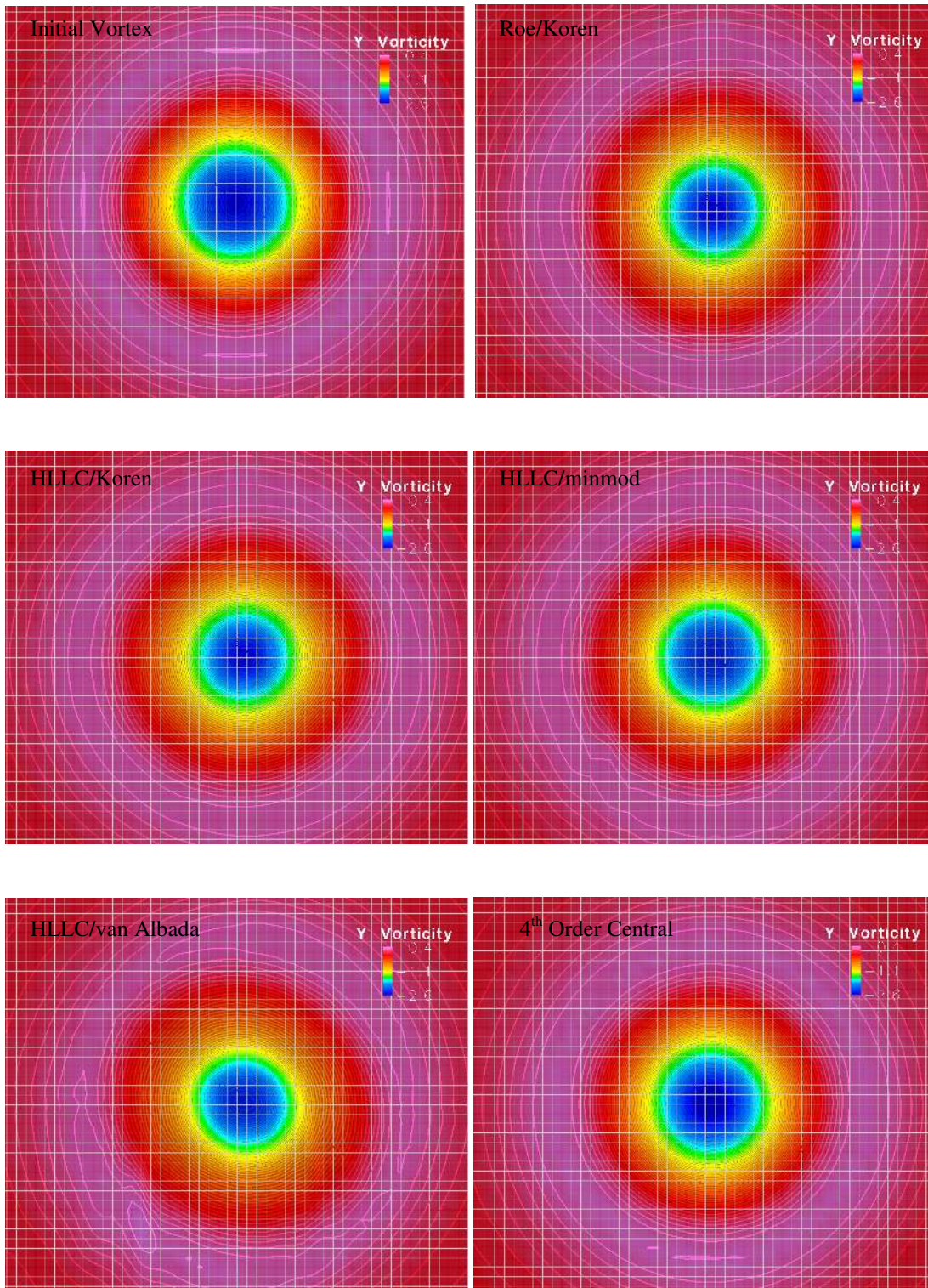
Figure 1. Vorticity contours for a convected inviscid vortex.
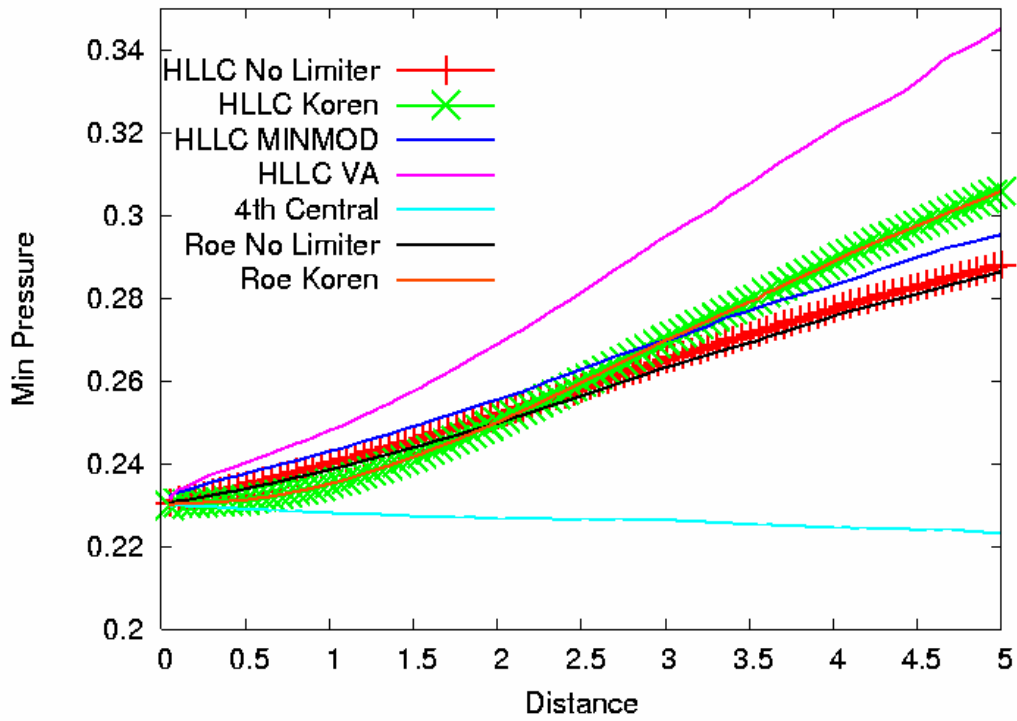
Figure 2. Minimum pressure for a convected inviscid vortex.
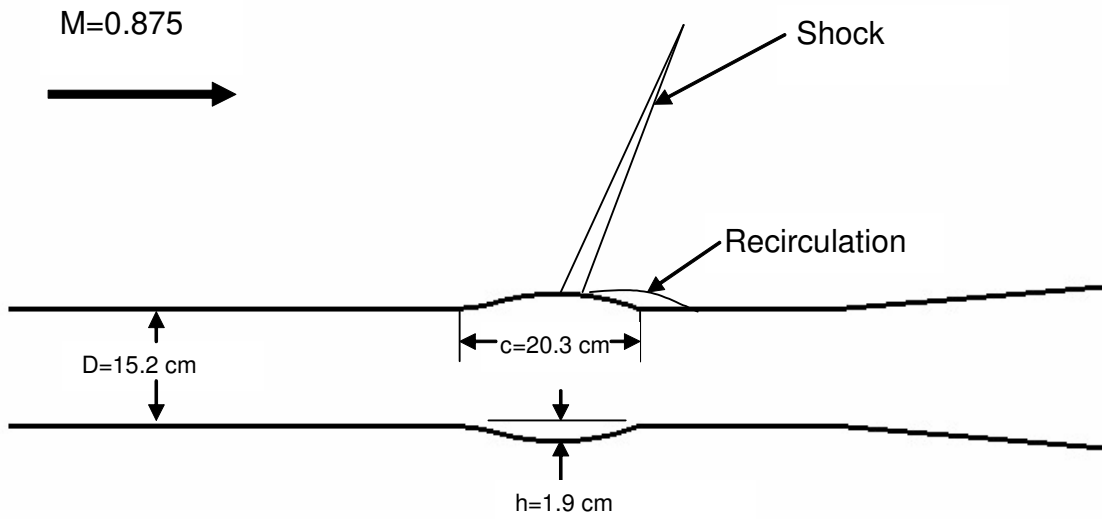
Figure 3. Geometry for the transonic axisymmetric bump.
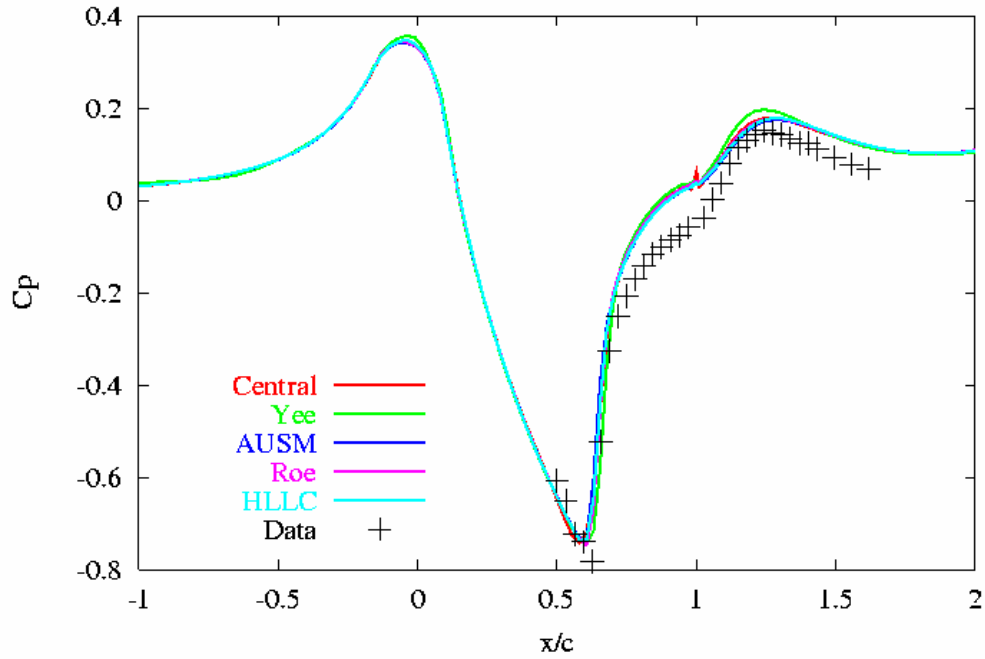
Figure 4. Pressure coefficient for the axisymmetric bump with an initial grid spacing of y+=1.
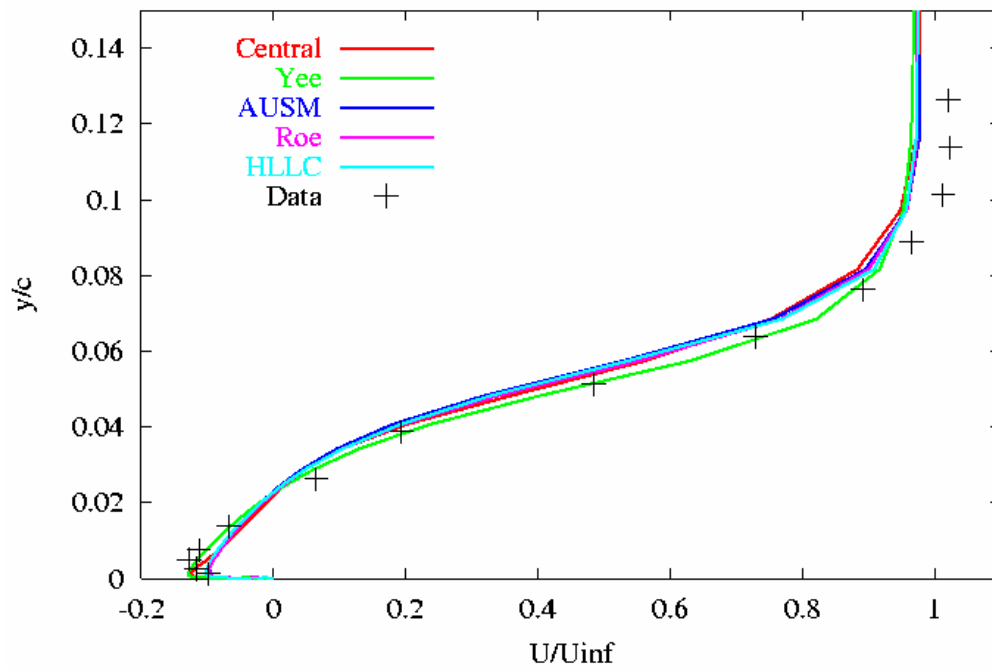
Figure 5. Velocity at x/c=1 for the axisymmetric bump with an initial grid spacing of y+=1.
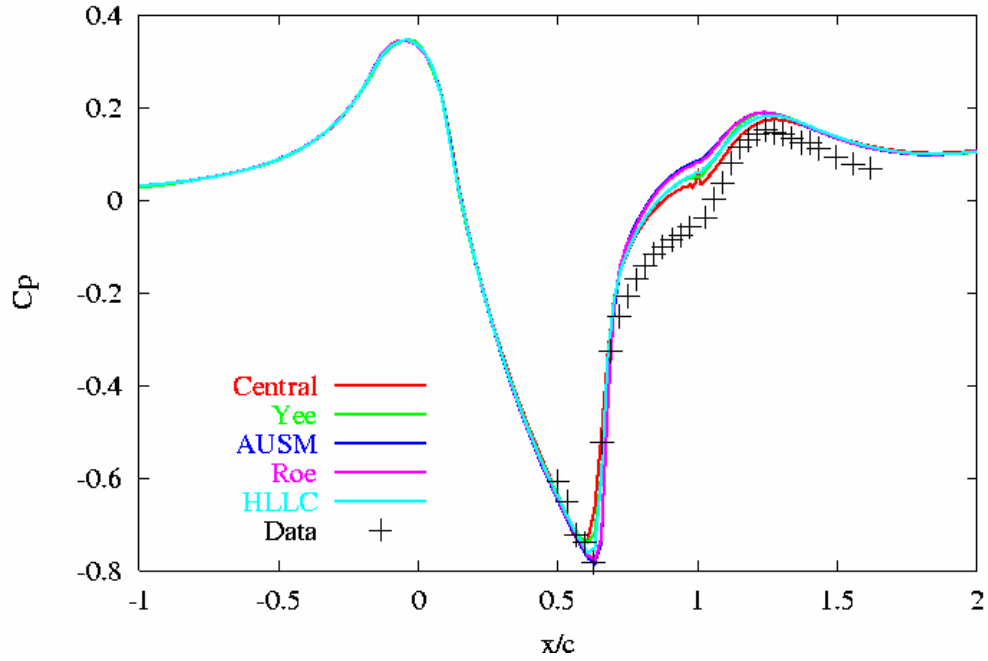
Figure 6. Pressure coefficient for the axisymmetric bump with an initial grid spacing of y+=50.
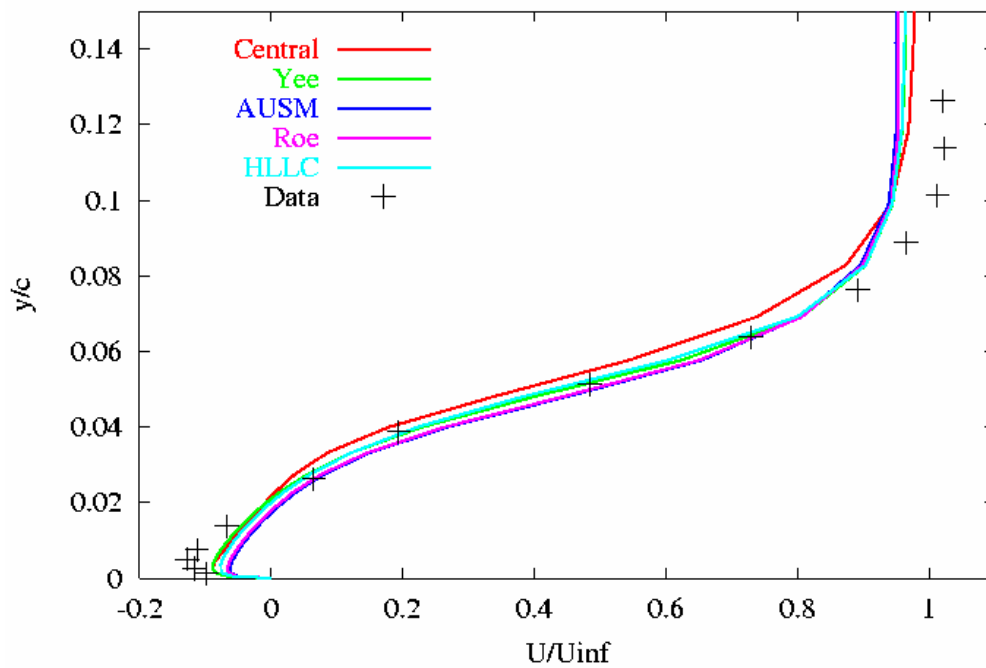
Figure 7. Velocity at x/c=1 for the axisymmetric bump with an initial grid spacing of y+=50.
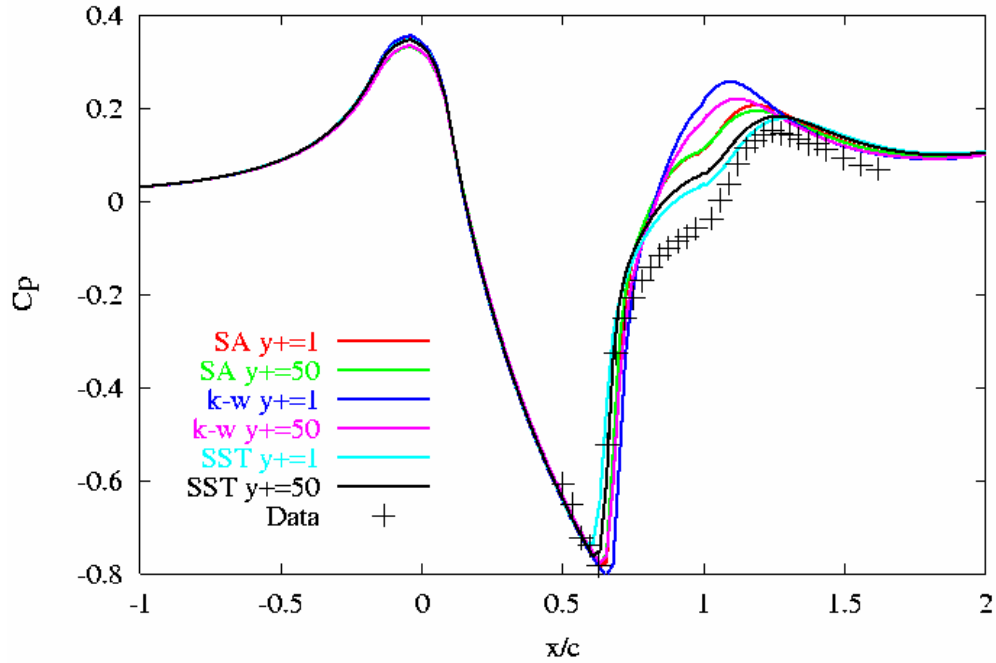
Figure 8. Pressure coefficient for the axisymmetric bump.
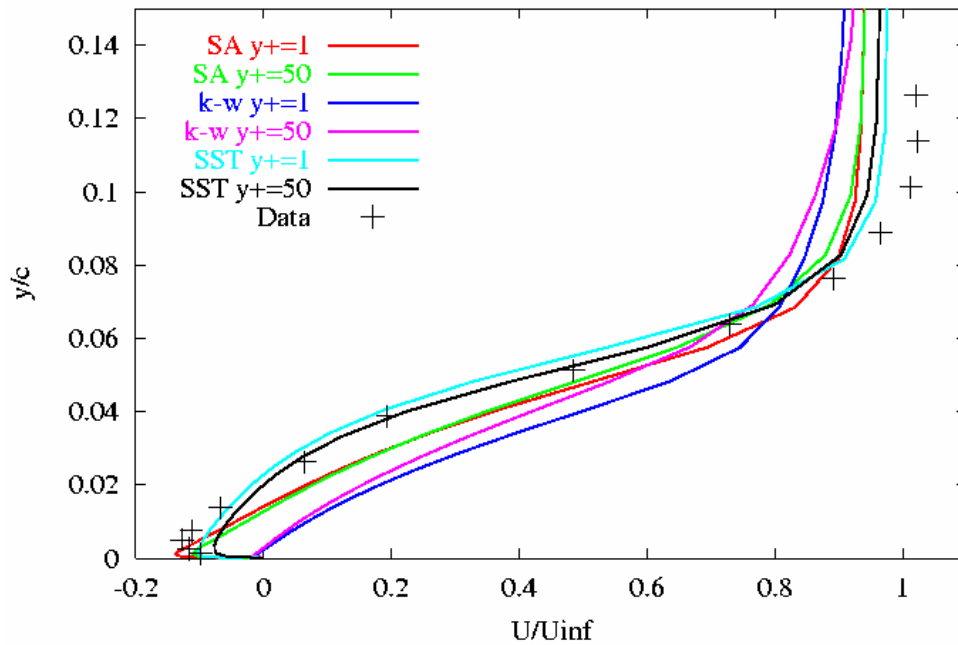
Figure 9. Velocity at x/c=1 for the axisymmetric bump.

American Institute of Aeronautics and Astronautics
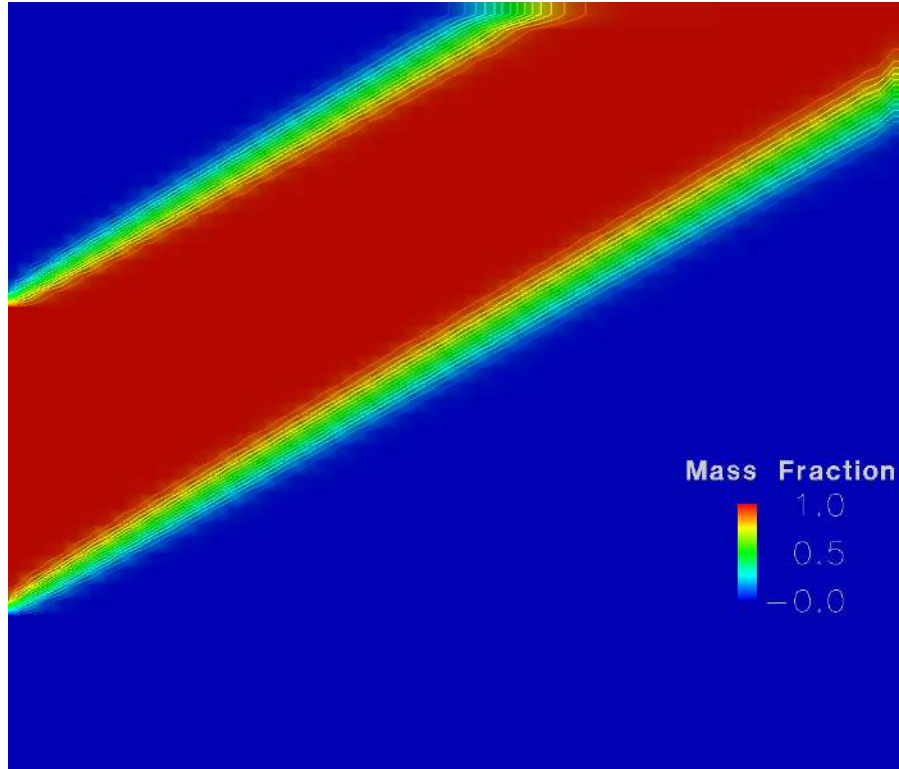
Figure 10. Jet mass fraction contours for the transport equation for the three gas test case.
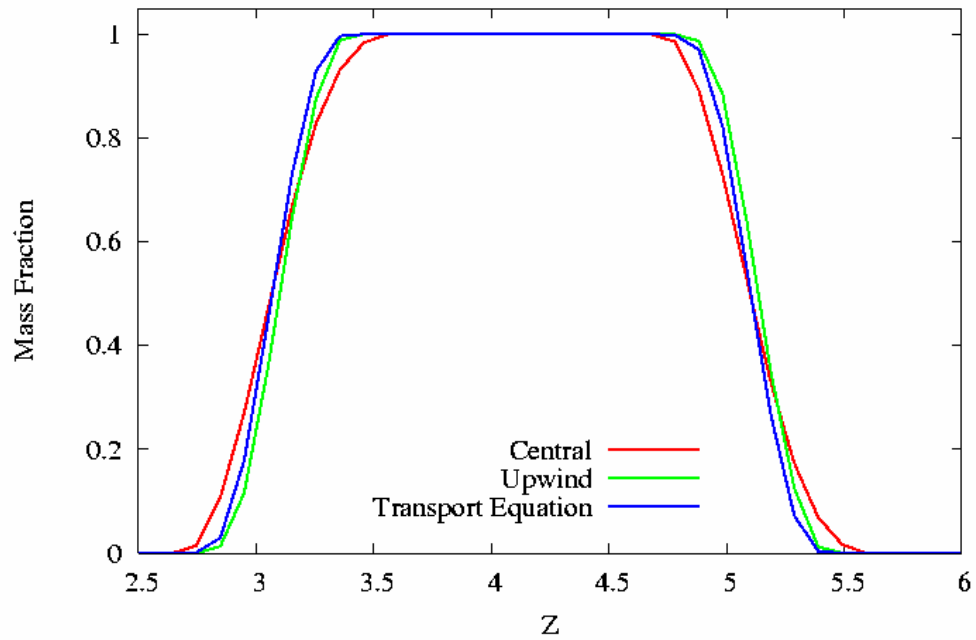
Figure 11. Vertical cut of the jet species for the three gas test case.
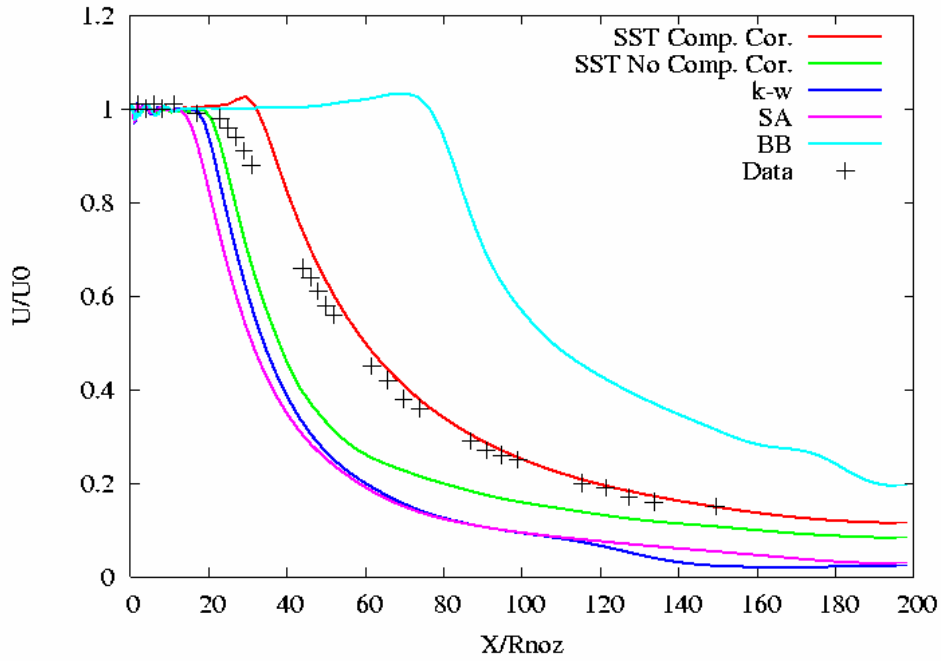
Figure 12. Centerline velocity distribution for a M=2.2 axisymmetric jet.
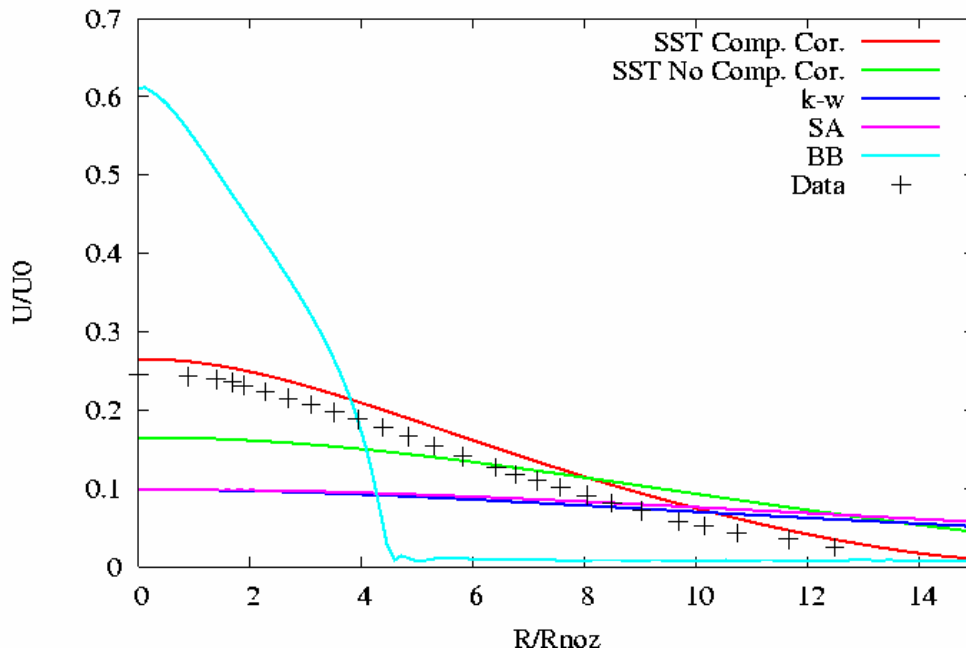
Figure 13. Velocity distribution at x/R_{noz}=100 for the M=2.2 axisymmetric jet.
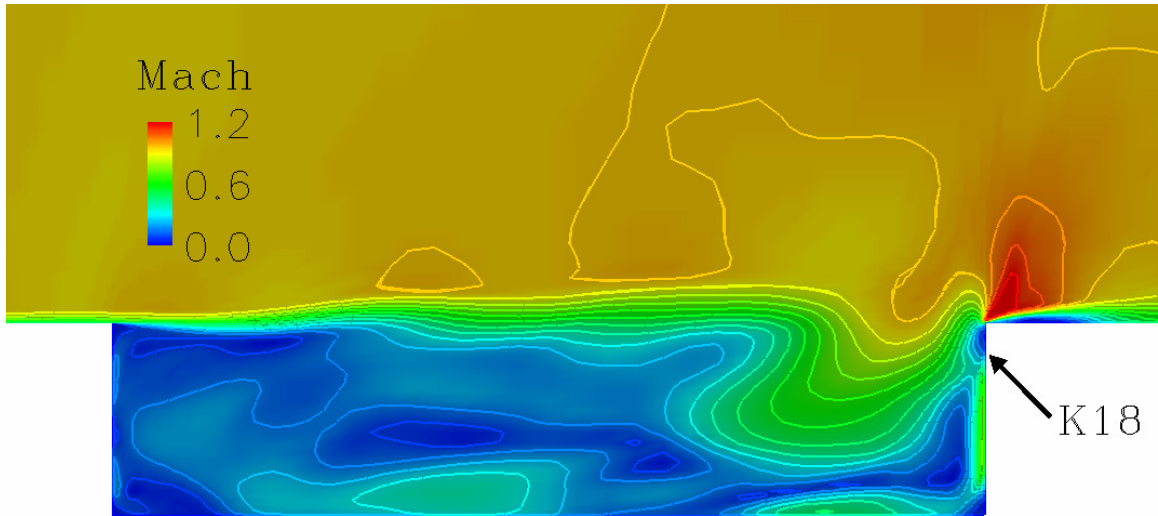
Figure 14. Instantaneous Mach number contours in the WICS L/D=4.5 bay centerline.
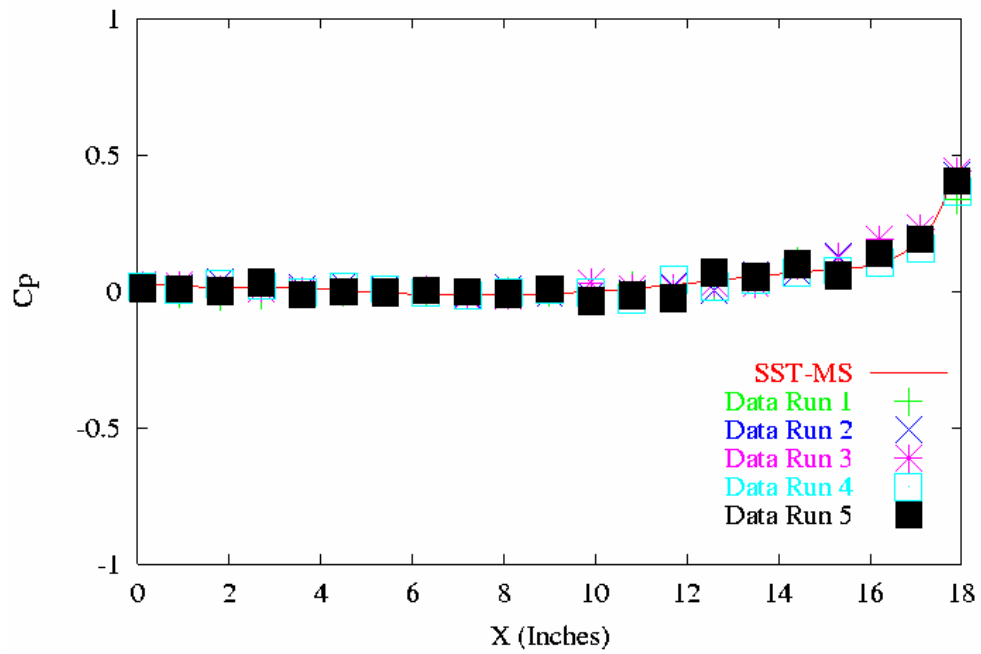
Figure 15. Time averaged pressure coefficient on the WICS L/D=4.5 bay ceiling.

American Institute of Aeronautics and Astronautics
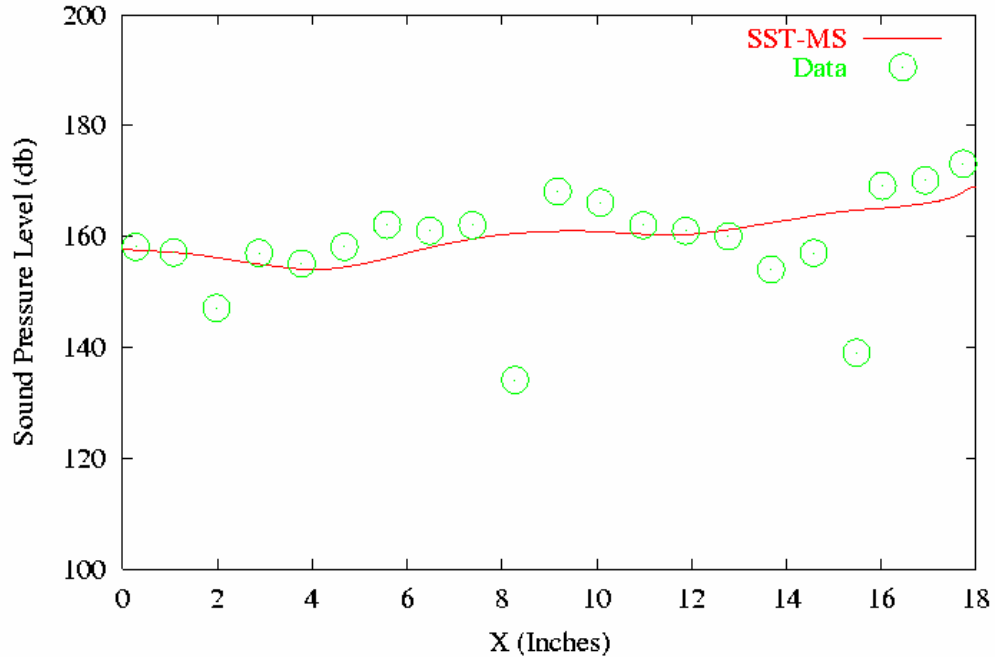
Figure 16. Overall sound pressure level on the WICS L/D=4.5 bay ceiling.
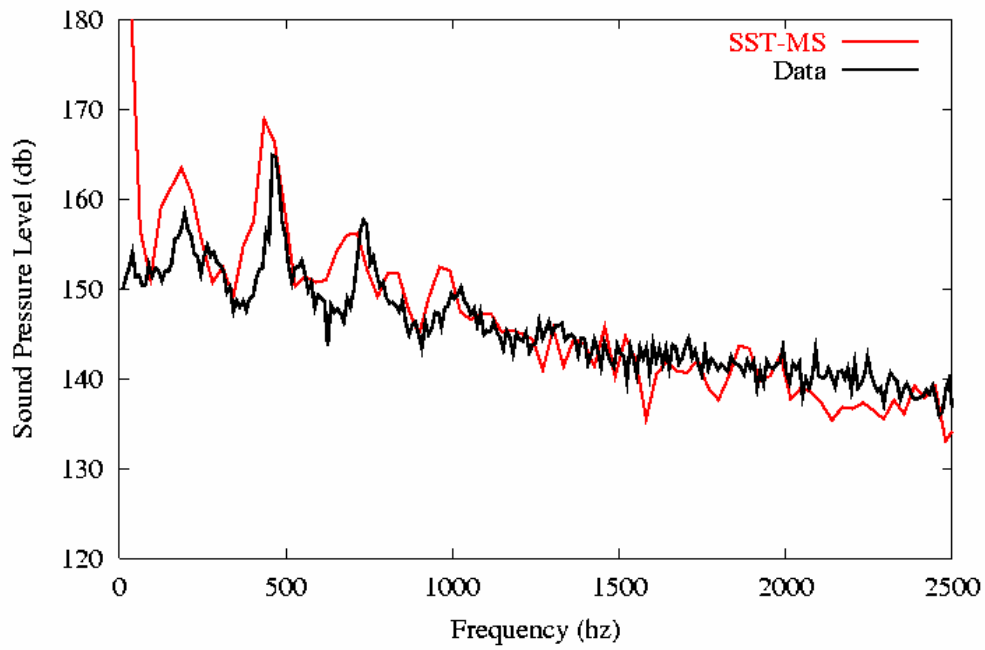
Figure 17. Sound pressure level spectrum for the WICS L/D=4.5 bay.
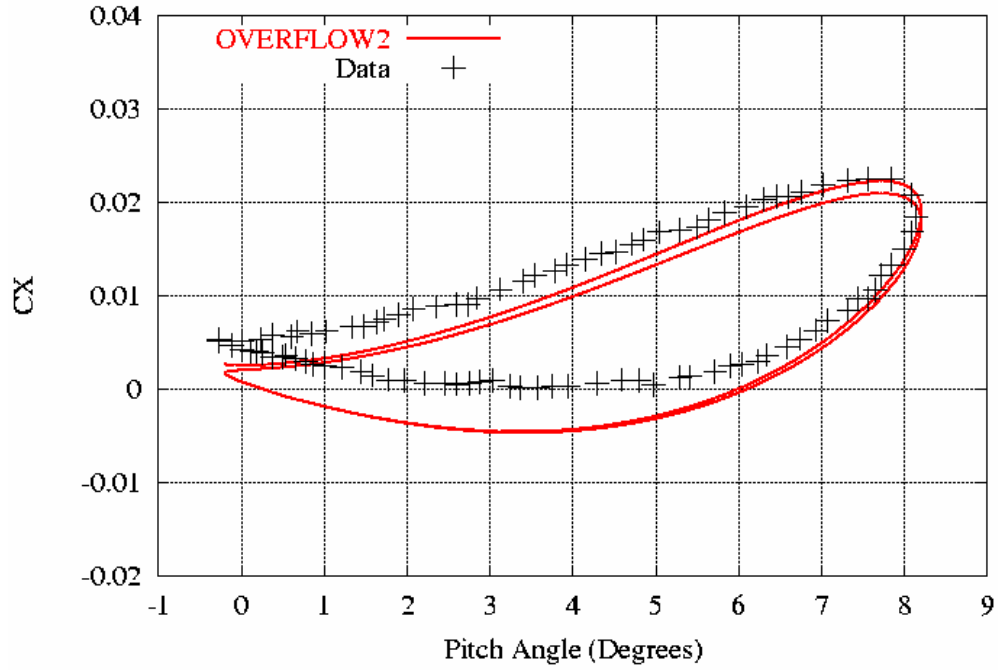
Figure 18. Axial force coefficient for the pitching NACA0015 airfoil.
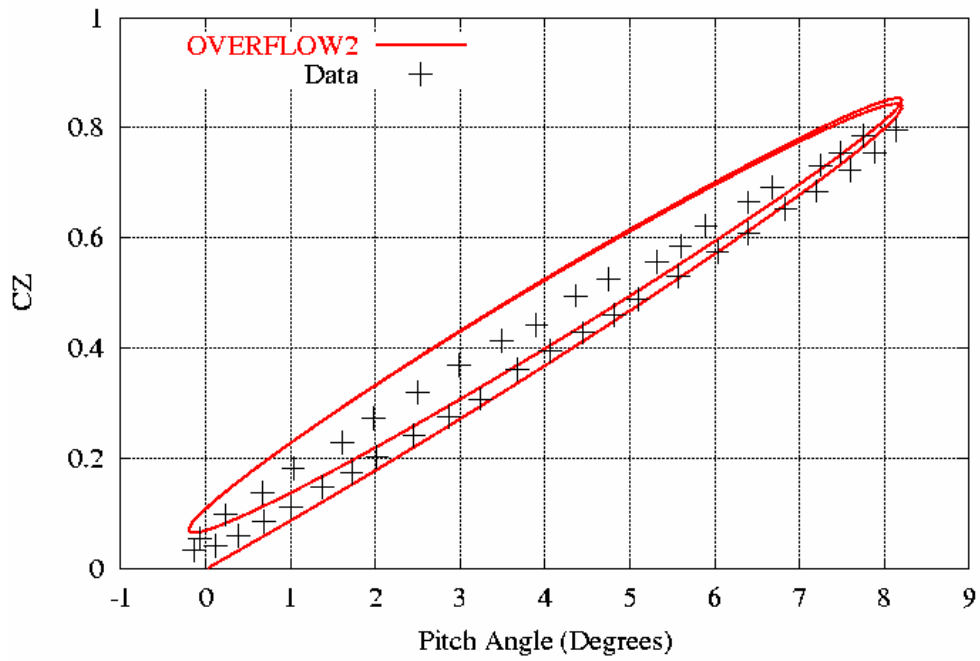
Figure 19. Normal force coefficient for the pitching NACA0015 airfoil.
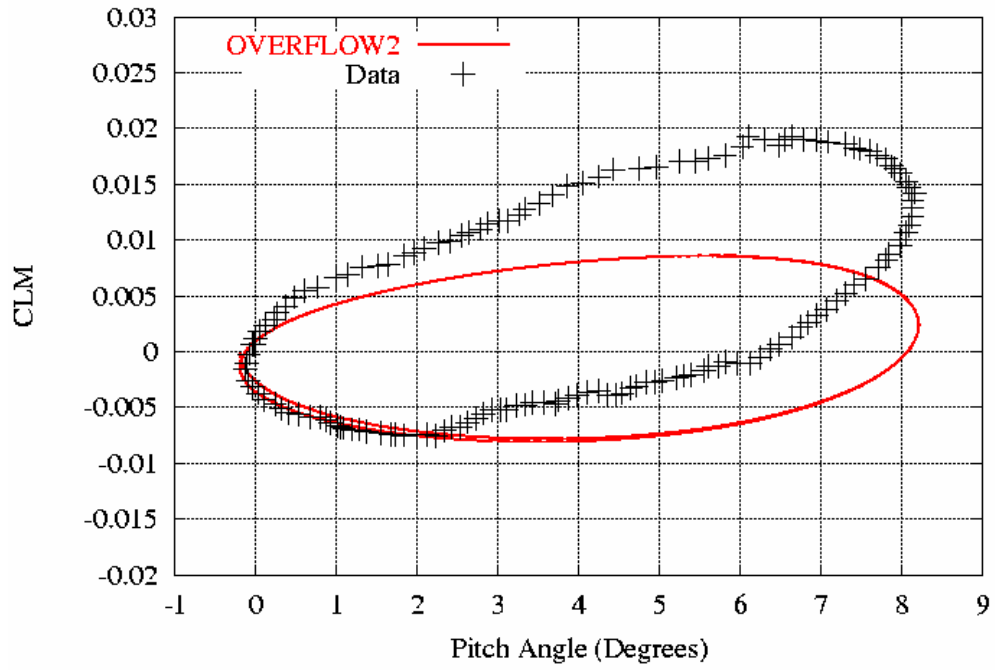
Figure 20. Pitching moment coefficient for the pitching NACA0015 airfoil.
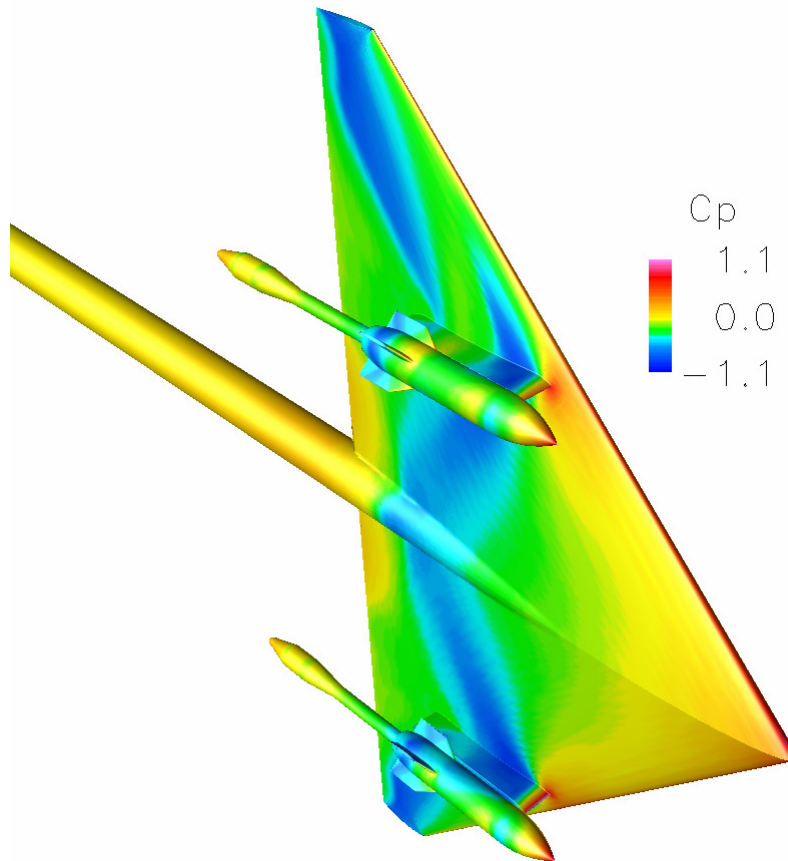
Figure 21. Pressure coefficient on the wing/pylon/store.
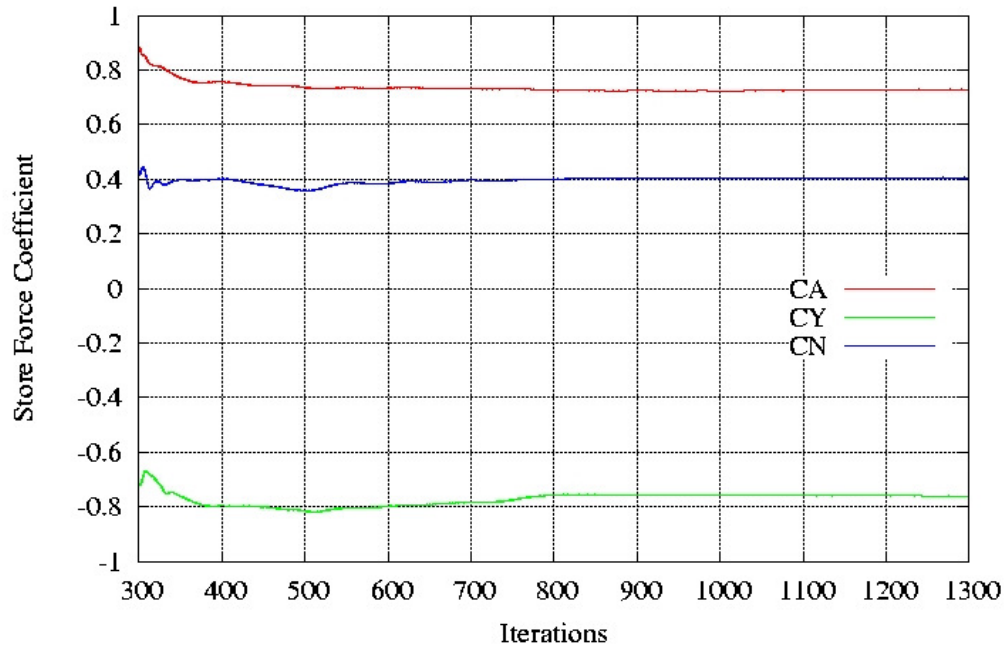
Figure 22. Force convergence history for the wing/pylon/store.
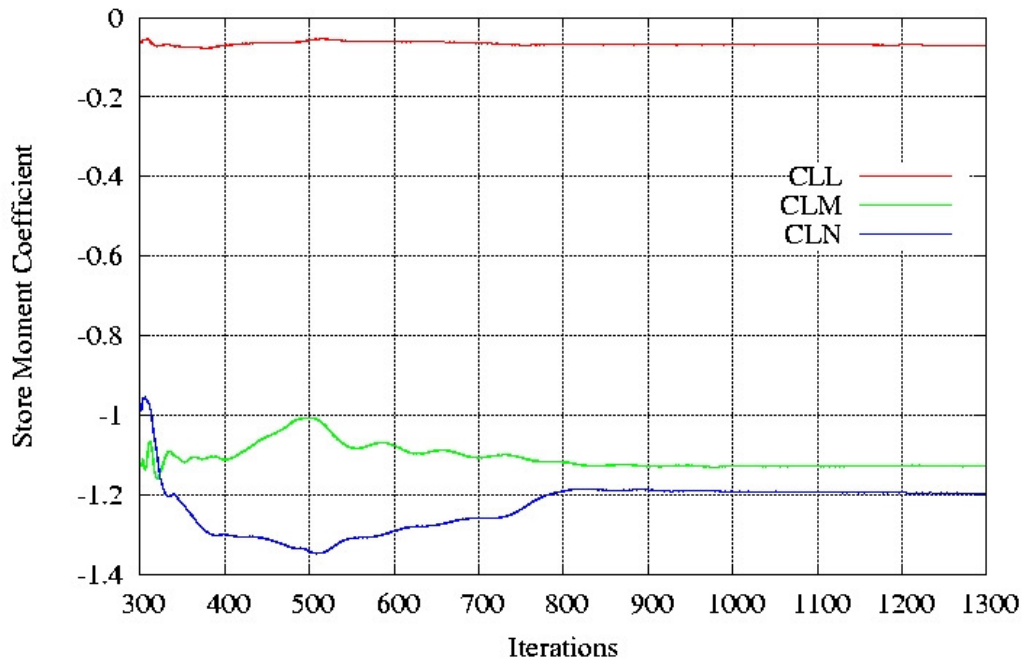
Figure 23. Moment convergence history for the wing/pylon/store.
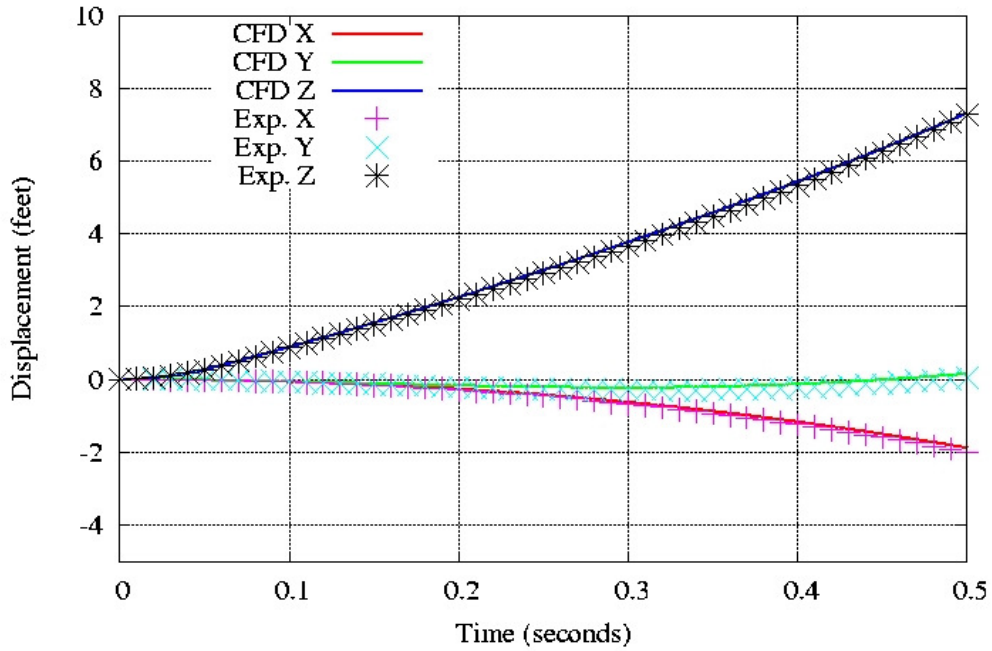
Figure 24. Store trajectory for the wing/pylon/store.
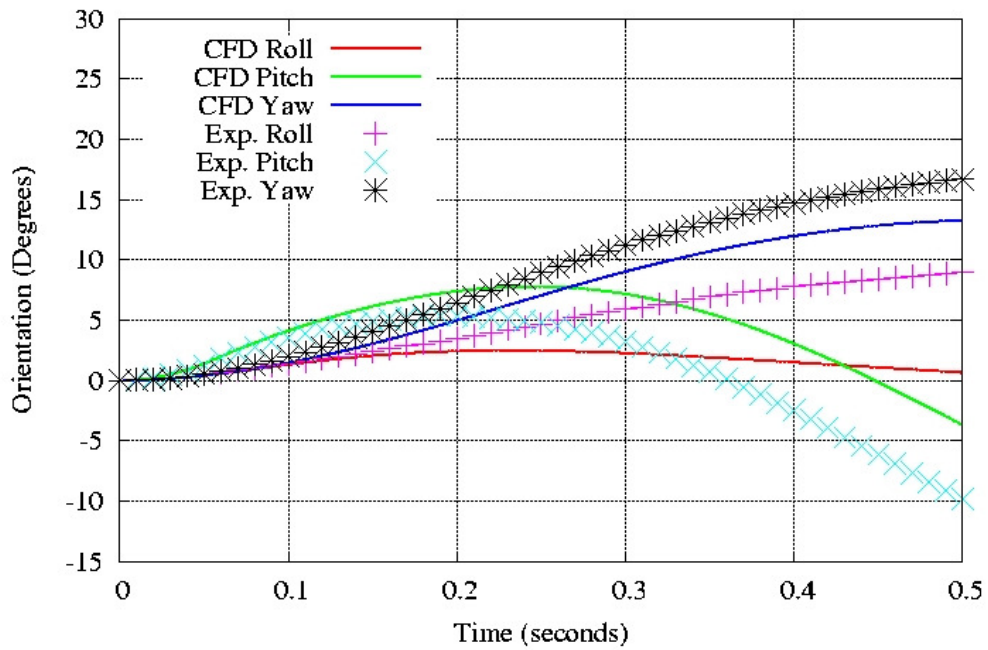
Figure 25. Store orientation for the wing/pylon/store.
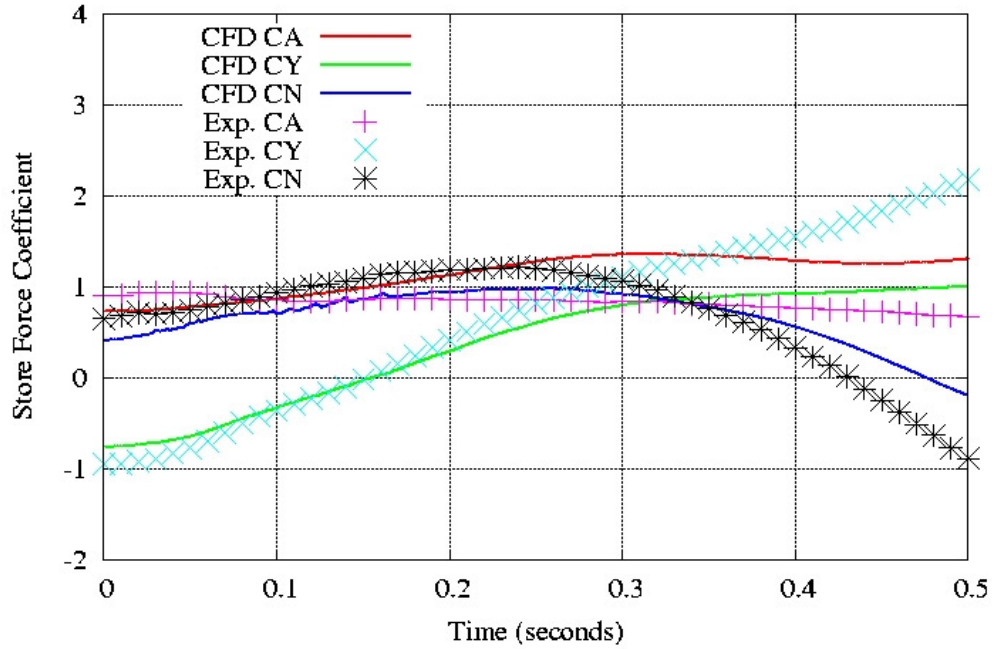
Figure 26. Store forces during separation for the wing/pylon/store.
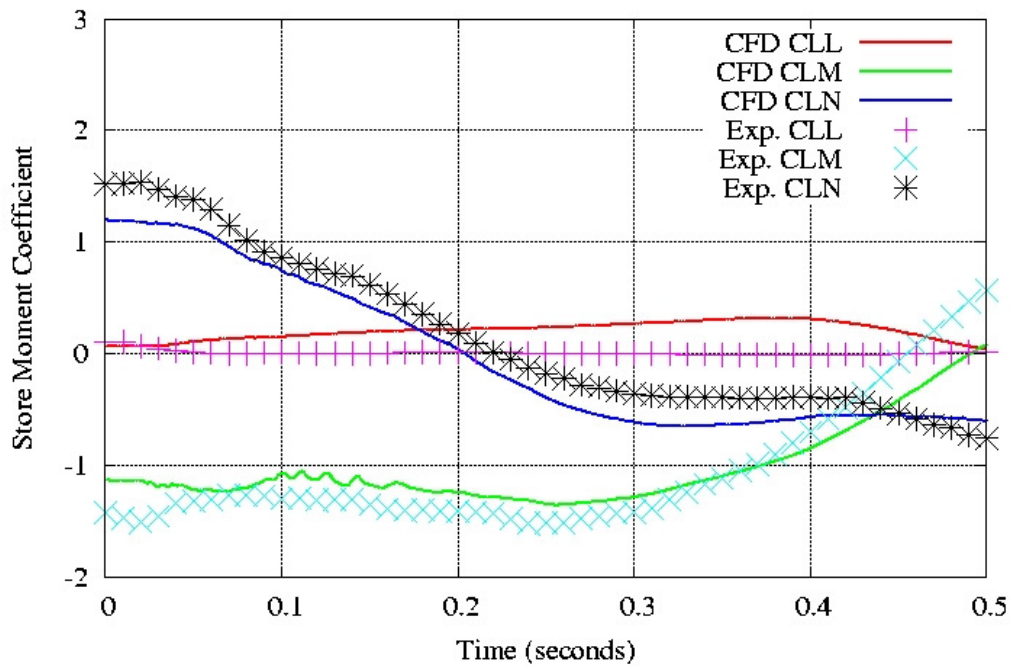
Figure 27. Store moments during separation for the wing/pylon/store.
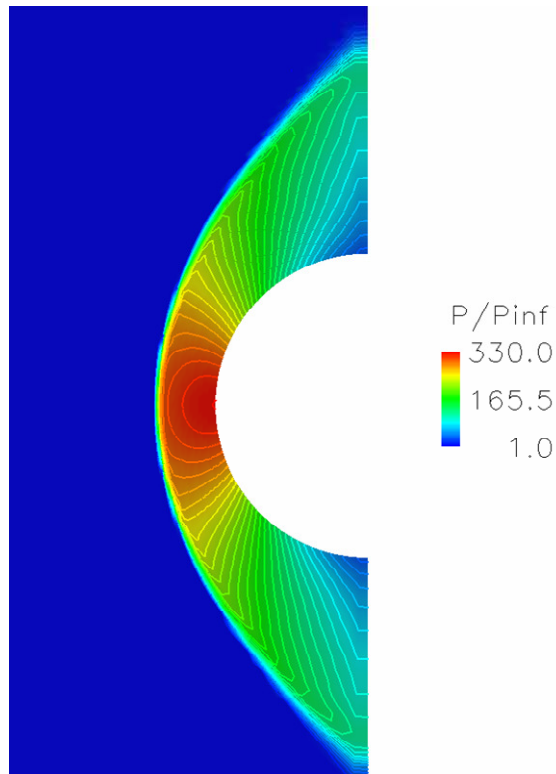
Figure 28. Pressure contours for the hypersonic cylinder bow shock
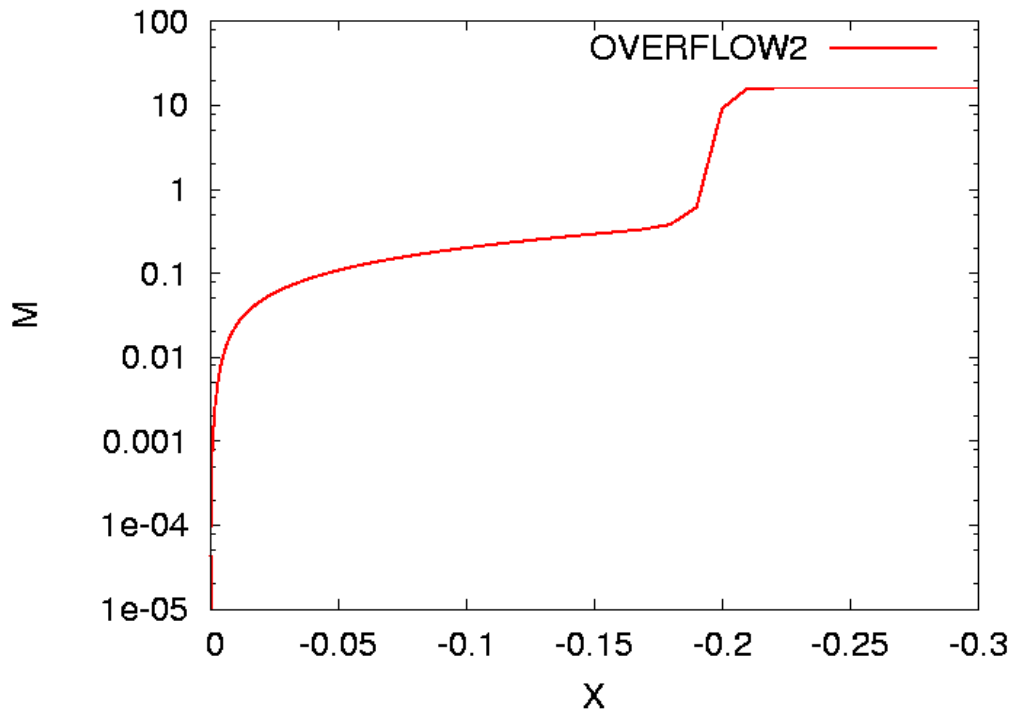
Figure 29. Mach number on the centerline of the hypersonic cylinder bow shock.
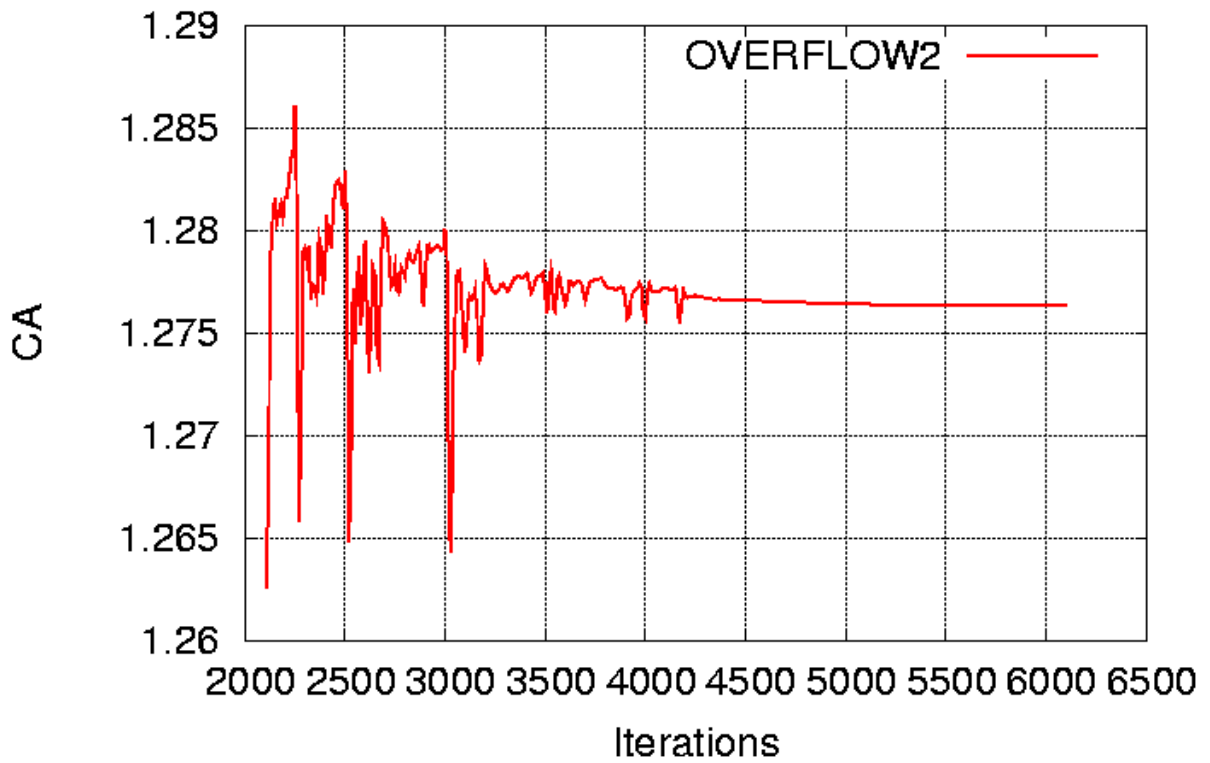
Figure 30. Axial force convergence for the hypersonic cylinder bow shock.
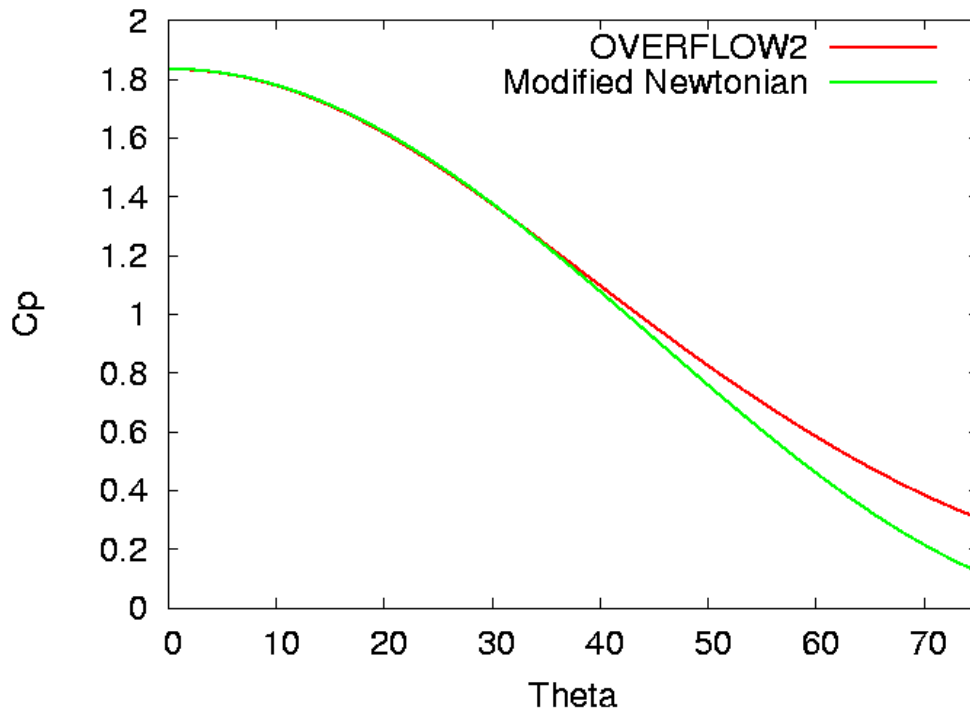
Figure 31. Pressure coefficient for the hypersonic cylinder bow shock.
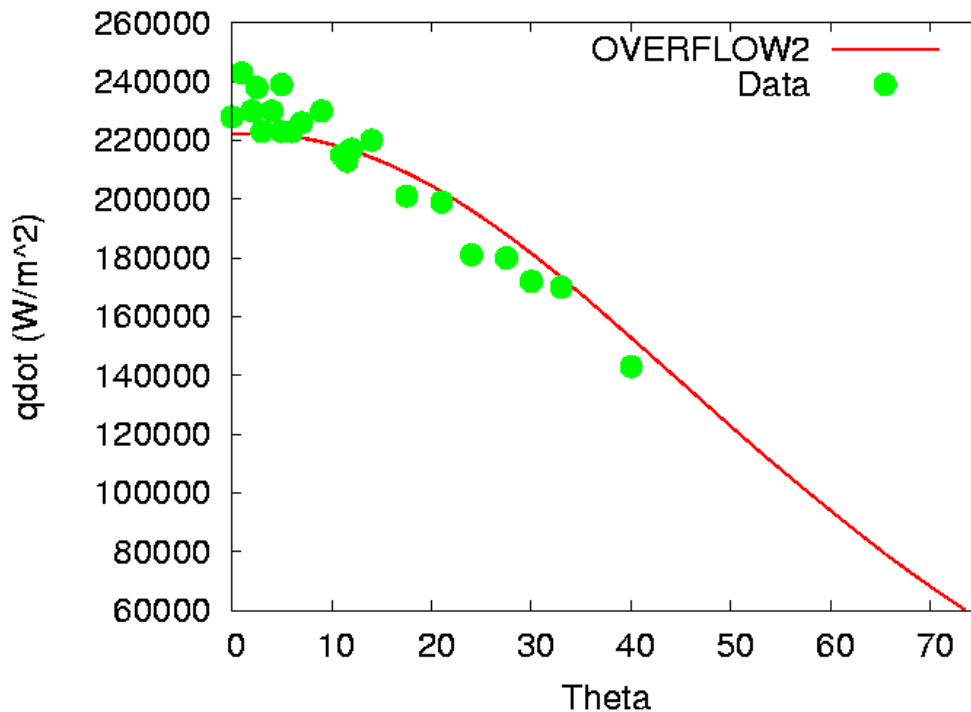
Figure 32. Heat transfer for the hypersonic cylinder bow shock.
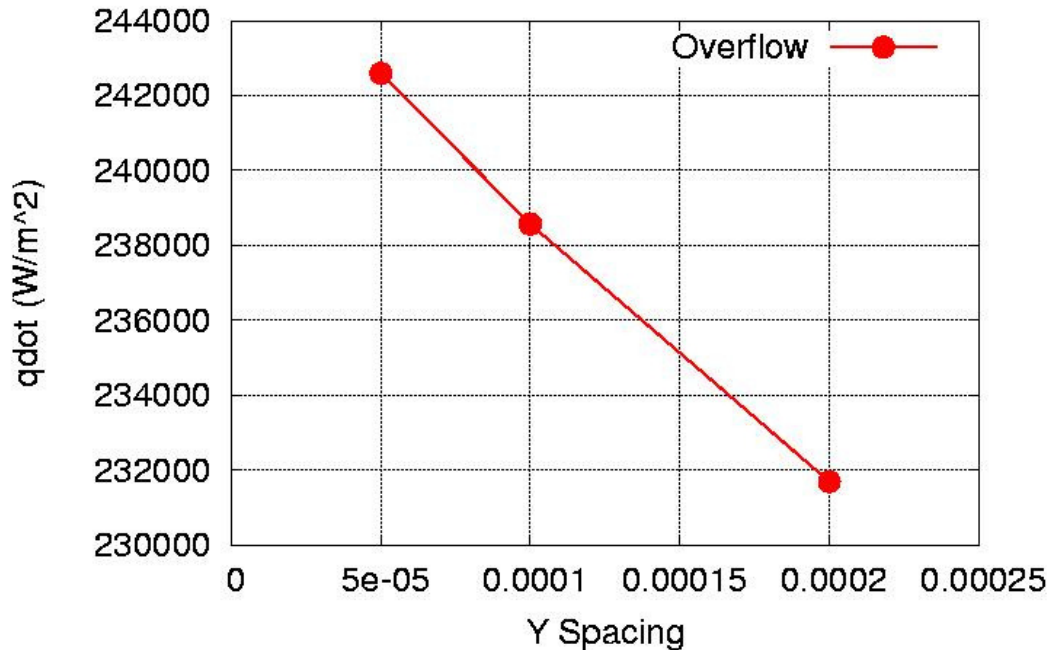
Figure 33. Change in stagnation heat transfer with initial wall spacing for the hypersonic cylinder bow shock.
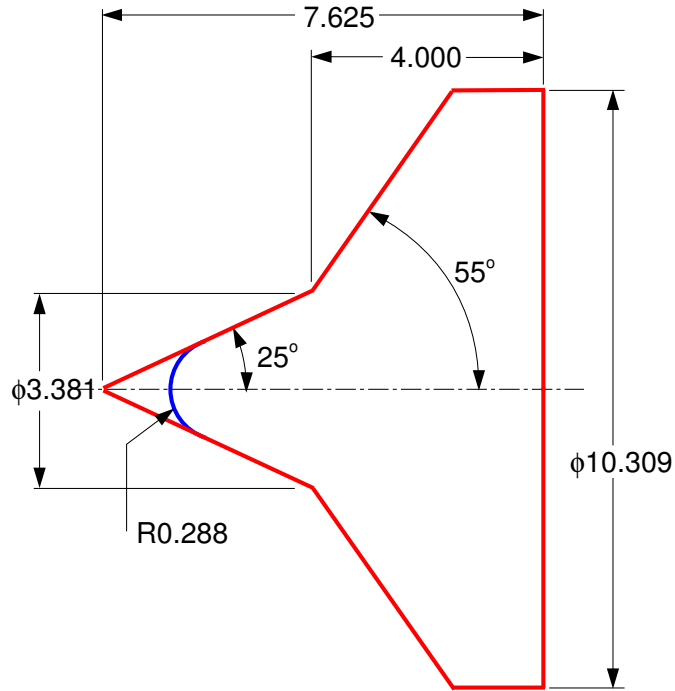
Figure 34. Geometry for the double cone experiments (all dimensions in inches).
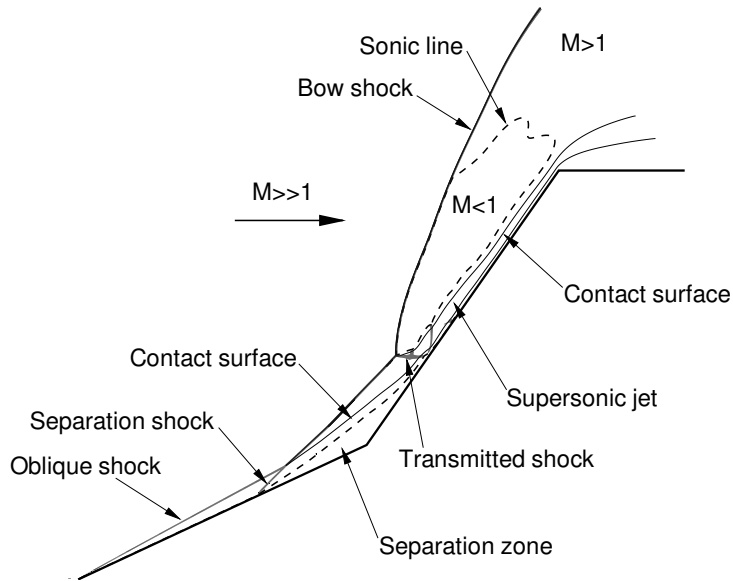
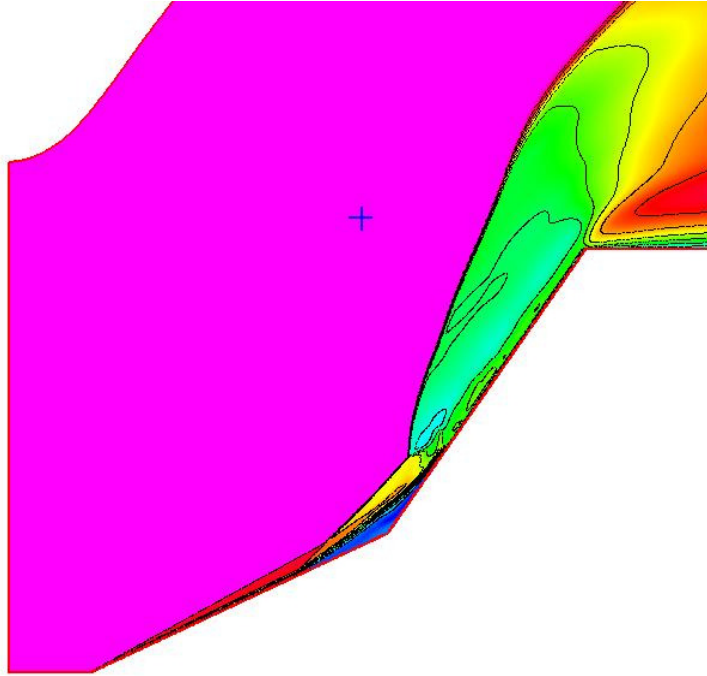Figure 35. Flow field features for the double cone experiment.

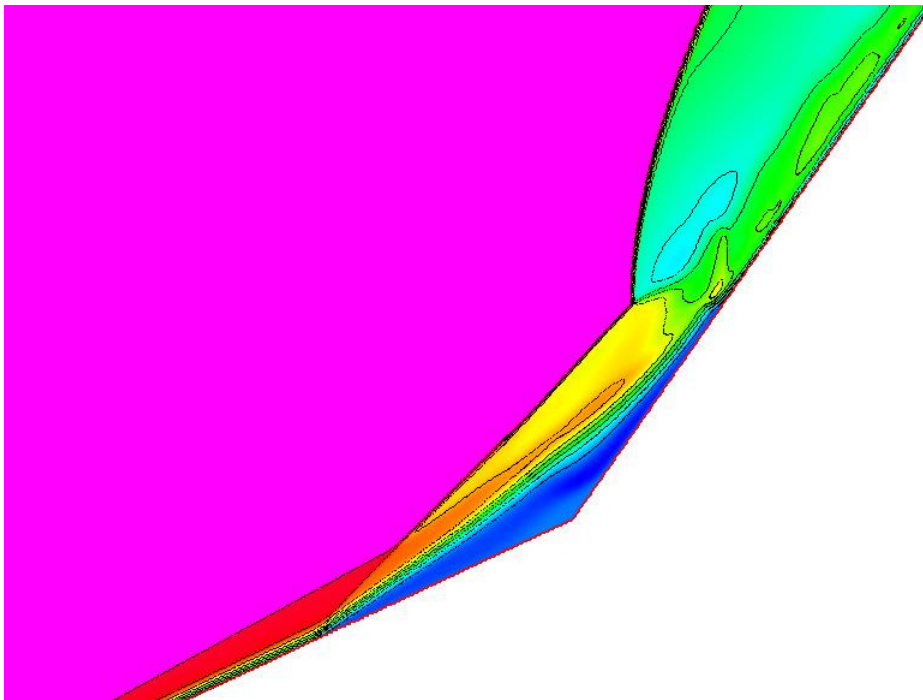Figure 36. Mach contours for the double cone experiment.

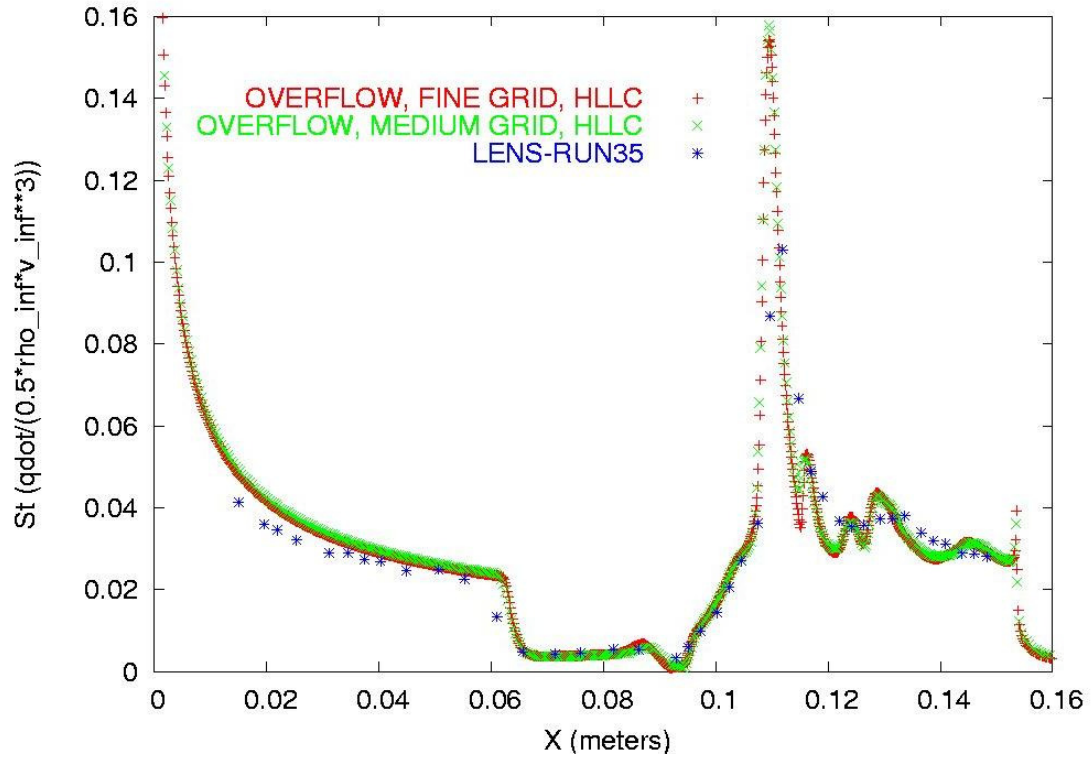Figure 37. Mach contours for the double cone experiment separation region.

Figure 38. Heat transfer predictions for the double cone experiment.