AIAA 2003-3979

Algorithmic Enhancements to the VULCAN Navier-Stokes Solver

D. K. Litton, **J. R. Edwards**
North Carolina State University, Raleigh, NC
J. A. White
NASA Langley Research Center, Hampton, VA

16th AIAA Computational Fluid Dynamics Conference
23–26 June 2003
Orlando, Florida

Algorithmic Enhancements to the VULCAN Navier-Stokes Solver

D.K. Litton[*], J. R. Edwards[*], and J.A. White [+]

[*]Department of Mechanical and Aerospace Engineering
Campus Box 7910
North Carolina State University
Raleigh, NC 27695

[+]Hypersonic Airbreathing Propulsion Branch
Mail Stop 168
NASA Langley Research Center
Hampton, VA 23681-2199

VULCAN (Viscous Upwind aLgorithm for Complex flow ANalysis) is a cell centered, finite volume code used to solve high speed flows related to hypersonic vehicles. Two algorithms are presented for expanding the range of applications of the current Navier-Stokes solver implemented in VULCAN. The first addition is a highly implicit approach that uses subiterations to enhance block to block connectivity between adjacent subdomains. The addition of this scheme allows more efficient solution of viscous flows on highly-stretched meshes. The second algorithm addresses the shortcomings associated with density-based schemes by the addition of a time-derivative preconditioning strategy. High speed, compressible flows are typically solved with density based schemes, which show a high level of degradation in accuracy and convergence at low Mach numbers $(M \le 0.1)$. With the addition of preconditioning and associated modifications to the numerical discretization scheme, the eigenvalues will scale with the local velocity, and the above problems will be eliminated. With these additions, VULCAN now has improved convergence behavior for multi-block, highly-stretched meshes and also can solve the Navier-Stokes equations for very low Mach numbers.

Introduction

The VULCAN (Viscous Upwind aLgorithm for Complex flow ANalysis) Navier-Stokes solver is considered the NASA standard in simulating reacting internal flows characteristic of high-speed propulsion devices. [1] VULCAN can solve the Navier-Stokes or Parabolized Navier-Stokes equations using a variety of discretizations, integration algorithms, turbulence models, and chemistry models and is applicable to general structured grids on multi-block domains.

MPI message-passing is used to adapt VULCAN to parallel architectures. The baseline integration strategy for Navier-Stokes applications within VULCAN is a diagonalized approximate factorization scheme. This scheme is rather efficient on a per-iteration basis but is sensitive to the near-wall grid spacing. Convergence rates can degrade rapidly for highly-stretched meshes. Furthermore, convergence rates can degrade when large numbers of blocks are used, due to the lack of strong coupling between adjacent blocks. VULCAN is also designed for higher Mach number applications, and like most compressible flow solvers, can experience convergence degradation and solution inaccuracy for very low Mach number flows. The present effort is concerned with improving the numerical efficiency of VULCAN for viscous flows on multi-block, highly-stretched meshes and for general low Mach number calculations. In the former context, planar relaxation based implicit methods are introduced, along with sub-iterative procedures that allow for a large degree of implicit coupling among blocks. In the latter context, a time-derivative preconditioning strategy based on the "all-speed" flux formulae of [2] is implemented and tested for laminar and turbulent flows of calorically- and thermally-perfect gas at low Mach numbers. The cases used for validation are the West and Korkegi double wedge

configuration [3], a 3-D channel flow, inviscid flow over a bump along a 2-D channel, a flat plate simulation, and finally, a simulation of subsonic-to-supersonic flow transition in a two-dimensional nozzle.

Governing Equations

VULCAN solves the three-dimensional Navier-Stokes equations, expanded to include separate transport equations for individual species and two-equation turbulence model components. Written in a generalized coordinate system, the Navier-Stokes set may be written as

$$\frac{dQ}{\partial t} + R(Q) = 0, \tag{1}$$

with the residual operator $R(Q)$ given as

$$R(Q) = \frac{\partial(E - E_v)}{\partial x} + \frac{\partial(F - F_v)}{\partial h} + \frac{\partial(G - G_v)}{\partial z} - S \tag{2}$$

The solution vector, Q, is given by:

$$Q = \frac{1}{J}\begin{bmatrix} rY_1 \\ \vdots \\ rY_{N_{CS}} \\ r \\ ru \\ rv \\ rw \\ E_t \\ rk \\ rw \end{bmatrix} \tag{3}$$

and the inviscid flux terms are defined as

$$E = \frac{|\nabla x|}{J}\begin{bmatrix} rY_1U \\ \vdots \\ rY_{N_{CS}}U \\ rU \\ rUu + p\bar{x}_x \\ rUv + p\bar{x}_y \\ rUw + p\bar{x}_q \\ (E_t + p)U \\ rkU \\ rwU \end{bmatrix} \quad F = \frac{|\nabla h|}{J}\begin{bmatrix} rY_1V \\ \vdots \\ rY_{N_{CS}}V \\ rV \\ ruV + \bar{h}_x p \\ rvV + \bar{h}_y p \\ rwV + \bar{h}_z p \\ (E_t + p)V \\ rkV \\ rwV \end{bmatrix}$$

$$G = \frac{|\nabla z|}{J}\begin{bmatrix} rY_1W \\ \vdots \\ rY_{N_{CS}}W \\ rW \\ ruW + \bar{z}_x p \\ rvW + \bar{z}_y p \\ rwW + \bar{z}_z p \\ (E_t + p)W \\ rkW \\ rwW \end{bmatrix} \tag{4}$$

The Jacobian, J, of the transformation is defined as

$$J = \frac{\partial(x,h,z)}{\partial(x,y,z)} \tag{5}$$

The components of the cell face unit normal and the contravariant velocities are

$$\bar{x}_x = \frac{x_x}{|\nabla x|} \quad \bar{h}_x = \frac{h_x}{|\nabla h|} \quad \bar{z}_x = \frac{z_x}{|\nabla z|} \tag{6}$$

$$|\nabla x| = \sqrt{x_x^2 + x_y^2 + x_z^2}$$

$$U = \bar{x}_x u + \bar{x}_y v + \bar{x}_z w \qquad V = \bar{h}_x u + \bar{h}_y v + \bar{h}_z w$$

$$W = \bar{z}_x u + \bar{z}_y v + \bar{z}_z w \tag{7}$$

Definitions for viscous flux vectors, E_v, F_v, and G_v and the source-term vector S can be found in [1].

In the above equations, Y_i is the mass fraction of the i^{th} chemical species and N_{CS} is the total number of chemical species.

For a calorically perfect gas, the pressure, total enthalpy, static enthalpy, and total energy are given by

$$p = r\,RT$$

$$H = h + \frac{1}{2}\left(u^2 + v^2 + w^2\right)$$

$$h = C_p T \tag{8}$$

$$E_t = rH - P$$

For a thermally perfect gas, the pressure, species enthalpy, and static enthalpy are formed by the expressions:

$$p = rR_u T \sum_{i=1}^{N_{CS}} \frac{Y_i}{m_i}$$

$$h_i = h_i^0 + \int_{T_0}^{T} C_{pi}\, dT \qquad (9)$$

$$h = \sum_{i-1}^{N_{CS}} h_i Y_i$$

where R_i (R_u/μ_i) is the i^{th} species gas constant. R_u is the universal gas constant and μ_i is the species molecular weight.

Planar Relaxation Implicit Flow Solvers for Multi-Block Domains.

Algorithm

One of the major problems encountered when solving three-dimensional problems on large numbers of blocks is a reduction in the overall convergence rate as the number of blocks increases. Typical domain-decomposition strategies used for finite-volume discretizations only allow one or two mesh cells of overlap between adjacent domains. Typical implicit solvers, when formulated for multi-block arrangements, do not consider matrix elements that would multiply corrections generated in adjacent domains. Therefore, subdomain coupling is only achieved in an explicit manner, through the residual evaluation at cells adjacent to block interfaces.

The RLX3D option in VULCAN is built around a planar relaxation scheme for solving the sub-domain implicit problem, and is designed to solve the complete (not parabolized) Navier-Stokes equations. The chosen sweep direction may be block-specific, and the crossflow plane linear system is approximately solved using an incomplete LU decomposition procedure. This approach alleviates much of the numerical stiffness associated with highly-stretched mesh cells, provided that the crossflow plane is oriented so that to encompass the coordinate direction(s) with the largest degree of mesh stretching. Techniques such as Jacobian freezing are used to reduce the overall CPU load, and implicit boundary conditions are incorporated to further enhance stability. The RLX3D option has been tested for supersonic turbulent flow past a cylinder, laminar flow between two intersecting wedges [3] and 1-D unsteady flow using a dual-time-stepping approach.

An improved implicit algorithm has been developed with better block-to-block coupling. M represents the planar relaxation implicit operator as applied over a subdomain, with its action upon a residual vector, R, denoted by the operation $M^{-1}R$. The actual Jacobian matrix is denoted as A. Note that the factorization of M is defined only over the interior grid points within a particular subdomain. In contrast, the Jacobian matrix A contains elements that may multiply corrections that are obtained from the solution of the linear system in adjacent subdomains. Thus we may split the matrix A into $M+N+E$, where N contains elements of A that would multiply corrections in adjacent subdomains and E is the factorization error. Given this, a general iterative scheme for improving the solution of the linear problem at a particular subdomain may be defined as:

$$M(\Delta Q^{n+1,l+1} - \Delta Q^{n+1,l}) = -R^n - A\Delta Q^{n+1,l}$$
$$\text{for } \Delta Q^{n+1,l+1} \qquad (10)$$

Here, the index n denotes a particular iteration level for the solution of the nonlinear problem (for unsteady flows, this could be part of another subiteration), and the index l denotes a particular iteration level for the iterative improvement of the solution of the linear problem. With this basic strategy in place, one can define an algorithm for improving block-to-block coupling:

Solve: $\Delta Q^{n+1,1} = -M^{-1}R^n$

For l=1, $l_{\max}$:

1: Pass appropriate $\Delta Q^{n+1,l}$ elements to ghost cells of adjacent blocks (parallel MPI send / receive)

2: Solve: $\Delta Q^{n+1,l+1} = \Delta Q^{n+1,l} + M^{-1}(-R^n - A\Delta Q^{n+1,l})$

End loop

Update: $Q^{n+1} = Q^n + \Delta Q^{n+1,l\max+1}$

This algorithm requires that an extra block diagonal matrix, corresponding to the block diagonal of A, which is normally over-written by the planar ILU factorization, be stored in addition to M itself. The only change to the VULCAN input deck necessary to implement this algorithm is a flag indicating the number of subiterations performed, $l_{\max}$. If this is set to zero, then no subiterations are performed and the planar relaxation scheme alone is used to advance the solution. Test cases shown in the Results section provide indications of the degree of improvement in convergence rates offered by this approach.

Time-Derivative Preconditioning Algorithm

The time-derivative preconditioning strategy currently implemented into VULCAN combines the rank-one preconditioning matrix of Weiss and Smith [4] with the "all-speed" version of the low diffusion flux-splitting scheme (LDFSS) of Edwards [5], developed according to a methodology presented in Edwards and Liou [2]. The preconditioning method can currently be used with Runge-Kutta explicit time integration and diagonalized approximate factorization implicit time integration.

The preconditioned Navier-Stokes equation set is given by:

$$P\frac{dQ}{\partial t} + R(Q) = 0, \tag{11}$$

where the preconditioning matrix, P, is defined as

$$P = I + q\,\vec{u}\vec{v}^{\,T} \tag{12}$$

with

$$\vec{u}^{\,T} = \begin{bmatrix} Y_1 & \ldots & Y_{N_{CS}} & 1 & u & v & w & H & k & w \end{bmatrix}$$

$$\vec{v}^{\,T} = \begin{bmatrix} \frac{\partial P}{\partial Q_1} & \frac{\partial P}{\partial Q_2} & \frac{\partial P}{\partial Q_3} & \ldots & \frac{\partial P}{\partial Q_{Neq}} \end{bmatrix} \tag{13}$$

$$q = \frac{1}{V_{ref}^2} - \frac{1}{a^2}$$

where N_{eq} represents the total number of equations.

The reference velocity, V_{ref}, is responsible for scaling the eigenvalues of the equation set at low speeds to be of the same order. V_{ref} is defined as

$$V_{ref}^2 = \min\left(a^2, \max\left(\left|\vec{V}\right|^2, KV_\infty^2\right)\right), \tag{14}$$

where a is the sound speed and $\left|\vec{V}\right|^2$ is the velocity magnitude. V_∞ in the above equation acts as a cutoff velocity to prevent singularities in the proximity of stagnation regions. In the VULCAN implementation, the constant K scaling the cutoff velocity is a user input, and V_∞ is set to the inputted free-stream velocity.

The eigenvalues of $P^{-1}\frac{\partial E}{\partial Q}$ are:

$$U, U, U$$

$$\frac{1}{2}\left[\left(1+M_{ref}^2\right)U \pm \sqrt{U^2\left(1-M_{ref}^2\right)^2 + 4V_{ref}^2}\right]$$

$$M_{ref}^2 = \frac{V_{ref}^2}{a^2} \tag{14}$$

As the incompressible limit is approached, the eigenvalues become

$$U, U, U$$

$$\frac{1}{2}\left[U \pm \sqrt{U^2 + 4V_{ref}^2}\right] \tag{15}$$

whereas the eigenvalues revert to their traditional values U, U, U, and $U \pm a$ as V_{ref}^2 ? a^2.

Numerical Discretization

To ensure accuracy at all flow speeds, it is necessary that the numerical discretization of the inviscid flux terms reflect the preconditioned eigensystem. There are several approaches for doing this, with the most rigorous being the use of characteristic analysis to derive preconditioned analogues of matrix-dissipation methods. In the VULCAN implementation, we instead implement a preconditioned variant of the low diffusion flux-splitting scheme (LDFSS) of Edwards [5], developed according to a methodology presented in Edwards and Liou [2]. The interface flux E_I in LDFSS is split into convective and pressure contributions as follows:

$$E_I = a_{\frac{1}{2}}\left[r_L C^+ E_L^C - r_R C^- E_L^C\right] + E^p\left[D_L^+ p_L + D_R^- p_R\right] \tag{16}$$

The vector E^C is the same as $\vec{u}$ in Eq. (13), while

$$E^p = \begin{bmatrix} 0 & \ldots & 0 & 0 & \bar{x}_x & \bar{x}_y & \bar{x}_z & 0 & 0 & 0 \end{bmatrix} \tag{17}$$

The "preconditioned" interface sound speed $a_{\frac{1}{2}}$ is defined as

$$a_{\frac{1}{2}} = \frac{\left[\sqrt{U^2\left(1-M_{ref}^2\right)^2 + 4V_{ref}^2}\right]_{1/2}}{\left(1+M_{ref}^2\right)_{1/2}} \tag{18}$$

where the subscript ½ represents evaluation of the quantity using flowfield information arithmetically-averaged to the cell interface. The quantities C^+, C^-, D^+, and D^- are functions of left-and right-state Mach numbers, specially defined in terms of the interface sound speed and the reference Mach number as follows [2]:

$$\tilde{M}_L = \frac{1}{2}\left[(1+M^2_{ref,1/2})M_L + (1-M^2_{ref,1/2})M_R\right] \quad (19)$$

$$\tilde{M}_R = \frac{1}{2}\left[(1+M^2_{ref,1/2})M_R + (1-M^2_{ref,1/2})M_L\right] \quad (20)$$

with

$$M_{L,R} = \frac{U_{L/R}}{a_{\frac{1}{2}}} \quad (21)$$

For gas-dynamic flows, the use of the modified Mach number definitions in conjunction with the "preconditioned" sound speed enables the numerical dissipation mechanism of LDFSS to scale properly at all speeds. Exceptions to this are the definitions for C^+ and C^-, which contain a pressure-dissipation term proportional to $p_L - p_R$. As shown in [2], this term acts to provide pressure-velocity coupling at low speeds, and to ensure that this effect scales properly, the term must be multiplied by $1/M^2_{ref,1/2}$. Precise definitions of the components of LDFSS may be found in [5], and a more recent extension valid for general fluids may be found in [6].

Time-Stepping Scheme

The solution is advanced in time by a preconditioned, diagonalized approximate factorization (DAF) scheme. The preconditioned version of the DAF scheme is written as follows:

$$\left[I - J\Delta t S_Q\right]\Delta Q^*_{i,j,k} = -J\Delta t P^{-1} R^n$$

$$\begin{aligned} T_x^p\left[I + J\Delta t d_x\left(l^p{}_{x,c} - l_{x,v}\right)\right](T_x^p)^{-1}\Delta Q^{**}_{i,j,k} &= \Delta Q^*_{i,j,k} \\ T_h^p\left[I + J\Delta t d_h\left(l^p{}_{h,c} - l_{h,v}\right)\right](T_h^p)^{-1}\Delta Q^{***}_{i,j,k} &= \Delta Q^{**}_{i,j,k} \\ T_z^p\left[I + J\Delta t d_z\left(l^p{}_{z,c} - l_{z,v}\right)\right](T_z^p)^{-1}\Delta Q_{i,j,k} &= \Delta Q^{***}_{i,j,k} \end{aligned} \quad (22)$$

$$Q^{n+1}_{i,j,k} = Q^n_{i,j,k} + \Delta Q_{i,j,k} \quad (23)$$

In this, the modal matrices T_x^p, $(T_x^p)^{-1}$, etc. are constructed from diagonalization transformations of the forms

$$T_x^p\, l^p{}_{x,c}(T_x^p)^{-1} = P^{-1}\frac{\partial E}{\partial Q}, \quad (24)$$

where $l^p{}_{x,c}$ is a diagonal matrix containing the eigenvalues of the preconditioned Euler system. Note that the first step of the DAF procedure is an approximation of the more exact expression

$$\left[P - J\Delta t S_Q\right]\Delta Q^*_{i,j,k} = -J\Delta t R^n \quad (25)$$

This approximation allows the use of the Sherman-Morrison theorem to compute the action of P^{-1} on the residual vector in O(n) operations. The action of all other source Jacobian elements (those corresponding to chemistry and turbulence source terms) on the residual vector is computed in a separate step, involving the use of Householder transformations to ensure good numerical stability.

Other additions to VULCAN required for preconditioning include the use of characteristic inflow boundary conditions based on the preconditioned equations and the use of local time steps based on the eigenvalues of the preconditioned system.

Results

Planar Relaxation Results

The testing of the algorithm has been focused on the West-Korkegi intersecting-wedge geometry [3] and a channel-flow analogue formed by eliminating the wedges and the clustering to the leading edge. The clustering in the Y and Z directions remains the same, as do the length, width, and height of the (now) rectangular geometry. The free-stream conditions for the intersecting-wedge simulations are: M_∞ =3, Re/m =2.11e6, T_∞ =105 K. The free-stream conditions for the channel-flow simulations are: M_∞ =0.5, Re/m =3.52e5, T_∞ =105 K. In both cases, the grid size is 65x125x125 and laminar flow is assumed. Unless otherwise mentioned, all cases were performed in parallel on the North Carolina Supercomputing Center IBM-SP2 using a 16-block load-balanced decomposition of the baseline grid.

Figure 1 presents baseline results for the RLX3D planar relaxation method ($l_{\max}$ =0). The positive effect of using implicit boundary conditions is clearly indicated, as is the fact that the planar relaxation procedure allows a much higher CFL that the baseline DAF scheme. This translates in a significant CPU savings, as the DAF scheme at a CFL of 3.5 takes 2.9 hours to run on the NCSC IBM-SP2 (16 processors) while the planar relaxation scheme at a CFL of 150 requires only 1.95 hours.

Figure 2 presents results from a CFL-ramping exercise performed for the supersonic West-Korkegi case. The final CFL number is reached by ramping from 0.1 to 20 over the first 500 iterations, from 20 to 150 over the next 500, and from 150 to the final value over the next

250 iterations. In this case and in most subsequent ones, the first 500 iterations are performed on a coarse mesh using a first-order accurate inviscid flux discretization. Jacobian freezing is initiated after 250 iterations, with re-evaluation and factorization of the matrices performed every 5 iterations past this point. The controlling parameter l_{max} is set to one for this study. Figure 2 shows that there is little advantage to choosing a CFL higher than 150 for this case. Figure 3 shows the effect of the choice of sweep direction on the performance of the iteratively improved planar relaxation algorithm with l_{max} = 0 and l_{max} =1. Sweeping in the "i" direction (the direction of the dominant movement of the supersonic flow) is clearly preferable to sweeping in the "j" direction. Performing one subiteration to improve the solution of the linear system improves the performance in both cases, at least in terms of the number of iterations.

Figure 4 illustrates the effect of varying l_{max} on the number of iterations required for convergence. As shown, the number of iterations required for convergence drops as the number of subiterations performed increases. Also shown for comparison is a calculation performed on the single-block grid using l_{max} = 0. This calculation was performed on a Compaq ES-40, which has enough shared memory to store all of the matrix elements in core. Interestingly, the single-block grid performance at l_{max} = 0 is slightly worse than the 16-block performance. Otherwise, the trends are what one might expect. As the work increases significantly with the increase in the number of subiterations, it is instructive to examine wall clock time. Figure 5 shows that for this predominately supersonic flow, there is little benefit to performing the subiterations, with only about a 15% improvement in overall execution time for the best case of l_{max} = 2.

The next test case corresponds to Mach 0.5 flow through a channel similar in dimension to the West-Korkegi geometry. The single-block 65x125x125 grid is decomposed into 16 blocks along the "i" coordinate. The CFL is ramped from a starting value of 0.1 to a final value of 20 for the planar relaxation variants, and again, 500 iterations are performed to first-order spatial accuracy on the coarser mesh before interpolating the solution to the finest mesh. Figure 6 portrays convergence histories for four runs: the first-order planar relaxation scheme with l_{max} = 0 and l_{max} = 2, and the second-order planar relaxation scheme with l_{max} = 0 and l_{max} = 2. As shown, more than a three-fold improvement in the number of iterations required for convergence is evidenced for the first–order discretization. CPU times for the first order discretizations are 107.30 minutes for l_{max} = 0 and 33.41 minutes for l_{max} =2. These times are also nearly a three-fold improvement. It is possible that the system load may have been different for each of these runs, as the fact that the CPU speedup is nearly in accord with the iteration count is somewhat surprising, given the extra expense of the subiteration procedure. It is noteworthy that the use of subiterations stabilizes the second-order case to the point that its convergence rate is very similar to the first-order case. Otherwise, as evidenced by the results, the calculation eventually diverges. In comparison with the supersonic West and Korkegi case, these results indicate that the benefits of subiterative improvement of the linear system solution may be much larger for subsonic flows. The technique appears to aid in damping and/or expelling pressure disturbances that otherwise tend to reflect from physical and interface boundaries.

Preconditioning Results

The four test cases for the preconditioned system are flow over a flat-plate, flow through a two-dimensional UTRC nozzle, flow between intersecting wedges, and inviscid flow over a bump in a channel. These correspond to variations on standard test cases included in the VULCAN package. In all cases the maximum CFL is set to 2.5, and most cases involve both turbulent and laminar flow as well as multi-component gases. In all turbulent cases, the Wilcox (1998) k-ω model is used, while for all multi-component cases, a mixture of nitrogen and oxygen is used.

Two-dimensional flow over flat plate

Figure 7 shows the 65x129 grid for the flat-plate simulation. The first run was to compare the results of ramping the Mach number down from Mach 0.5 to Mach 0.005. The free-stream conditions for the Mach 0.5 simulation are: Re/m = `1.11e7`, T_∞ = 300K. In each successive case the only parameter changed was the Mach number, which decreased the Reynolds number by a factor of 10 for each succeeding run. Figure 8 presents convergence histories for preconditioned and non-preconditioned cases at each Mach number. Strikingly, it can be seen that the convergence of the non-preconditioned system is not altered by lowering the Mach number, whereas the preconditioned system converges faster for only Mach 0.5 and 0.05. A closer investigation into the solution produced by the non-preconditioned system verified that the solution was in

fact incorrect. This was verified when comparing with the Blasius solution at distances of 0.3, 0.40, and 0.45 meters from the front edge of the flat plate. By using the Blasius solution for flow over a flat plate, the boundary layer thickness can be calculated as:

$$d(x) = \frac{5.0x}{\sqrt{\mathrm{Re}_x}} \qquad (26)$$

Table 1 shows how the non-preconditioned system behaves in comparison to the preconditioned system. From this table it can be inferred that the non-preconditioned system is in fact converging to an incorrect solution. The preconditioned system has a more realistic value for the thickness of the boundary layer. For the Mach 0.05 run, the error for the preconditioned system is no larger than 6.7%. For the Mach 0.005 simulation, the error is less than 2.3%. Both of these numbers are in stark contrast to the 85-95% error found in the results obtained without preconditioning.

Figures 9 and 10 show the results for a turbulent, calorically perfect and a turbulent, two-species air flow over a flat plate, respectively. The convergences for each case are very similar to the laminar flow in Figure 9. In all three examples, the preconditioned Mach 0.5 case showed a marked improvement in convergence over the non-preconditioned system. This is somewhat in contrast with the results from the eigenvalue analysis, which indicate that the non-preconditioned system should have a better overall condition number at this Mach number. One reason may be the presence, in the preconditioned flux-splitting, of pressure-diffusion terms that tend to smooth out variations in the pressure field. As will be seen in the next few examples, this result is consistently independent of geometry. The convergence degradation indicated for the Mach 0.005 calculations could be associated with the decrease in Reynolds number. Stiffness due to low Reynolds numbers will not be alleviated by the inviscid preconditioning techniques currently employed in VULCAN.

Two-dimensional flow through a nozzle

The next case considered is flow through the two-dimensional UTRC nozzle. This is a standard test case for the VULCAN solver that involves decomposing the nozzle flow into two regions: an elliptic region upstream of the nozzle throat and a parabolic region downstream of the throat. The elliptic region is solved using the DAF scheme (preconditioned and non-preconditioned), while the parabolic region is solved using space-marching once the elliptic solution has been obtained. Figure 11 shows the grid used in the calculation, while Figures 12 and 13 show contours of Mach number and eddy-viscosity ratio (referenced to the laminar value), respectively. In the other calculations presented in this paper, the constant K scaling the cutoff velocity in Eq. (14) is set to one, since the velocity everywhere is near the free-stream velocity. In the nozzle calculation, however, the free-stream sound speed is chosen as the reference velocity. Thus, a choice of K=1 will not activate preconditioning in the elliptic region. Figure 14 illustrates the effect of lowering K (equal to "qctoff" in the figure) on the convergence rates. The best results (for a four order-of-magnitude reduction) are obtained for K~0.1. Lower values result in a significant degradation in convergence rate, though the slope appears to be more consistent. These results indicate that the proper reference-velocity choice for strongly-mixed subsonic / supersonic flowfields may not be obvious, and that trial-and-error procedures may have to be used to obtain the best results. Even with this ambiguity, the use of preconditioning results in a factor of 2 savings in iteration count. This translates into nearly a factor of 2 savings in CPU time, as the modifications necessary for implementing preconditioning require very little additional CPU time.

Three-dimensional flow through intersecting wedges

Figures 15-18 present results from simulations of subsonic viscous flow through the West-Korkegi intersecting wedge geometry (shown in Figure 15) Free-stream conditions are chosen to be the same as in the two-dimensional flat-plate case. Figure 16 illustrates the convergence histories for laminar flow through the wedges. The three non-preconditioned solutions do not converge at the same rate and show worse convergences than the preconditioned scheme at all Mach numbers, except Mach 0.005. As the Mach number decreases in magnitude beyond the Mach 0.5 case, the convergence rate is shown to decrease quickly. The preconditioned system contains a few oscillations but maintains its downward trend towards convergence. Figure 17 shows results for the turbulent, calorically perfect case. This case shows trends that are almost identical to the laminar flow.

The disparity in the results given by the non-preconditioned system is magnified when running the two species, turbulent simulation. As can be seen in Figure 18, the non-preconditioned residual at Mach 0.005 does not go down, but rather oscillates wildly around 10^{-1}. Although its preconditioned counterpart does not display this behavior, the convergence rate is noticeably slowed down. The residual of the preconditioned system continues to go down towards convergence with minimal oscillations (in comparison to non-preconditioned system).

To verify that the results produced by the preconditioning formulation were physically consistent, calculations of laminar flow through the channel analogue of the West-Korkegi geometry were also performed. At locations far enough away from the corner, it was anticipated that the boundary layer would develop according to the Blasius scaling shown in Eq. (26). Table 2 compares predictions from the preconditioned and non-preconditioned formulations versus the Blasius result. The boundary layer thickness obtained from the non-preconditioned formulation turned out to have an error of no less than 85%, while the preconditioned formulation provided results within a reasonable 6% of the theoretical values. Convergence trends were similar to those corresponding to the intersecting wedges and are thus not shown.

Inviscid flow over a bump in a channel

The last example of the validation of the preconditioning strategy is inviscid flow over a bump in a channel. For this particular case, the Euler equations are solved, thus Reynolds-number effects illustrated in the earlier calculations are not present. Figure 19 portrays the grid used for the calculations, while Figure 20 shows the convergence histories for Mach numbers of 0.5, 0.05, and 0.005. These calculations reveal the expected trend of (nearly) Mach-number independent convergence rates when using the preconditioning technique. In contrast, the non-preconditioned formulation displays a significant degradation in convergence rate as the Mach number is lowered.

Finally, an important property of the preconditioned DAF scheme is verified in Figure 21. The preconditioned scheme is designed to revert back to the compressible Navier Stokes equations for higher Mach-number flows. It is apparent from this figure that the two schemes are in fact identical for supersonic flow over the bump, thus demonstrating the validity of the preconditioned diagonalized approximate factorization scheme for all flow speeds.

Conclusions

Two algorithms for enhancing the capabilities of the VULCAN Navier-Stokes solver have been presented. The purpose of the first algorithm is to improve convergence rates for viscous flows on highly-stretched, multi-block meshes. The algorithm decreases not only the number of iterations to convergence, but also the time to convergence, an important factor in weighing the importance of this new addition. So far, this addition is specialized for calorically-perfect gases. The second algorithm is a time-derivative preconditioning strategy that is intended to expand the range of applicability of VULCAN toward low-speed, nearly incompressible flows. This addition is valid for calorically and thermally-perfect gas and is designed for use with the baseline diagonalized approximate factorization algorithm in VULCAN. Test cases show that the preconditioning framework greatly improves solution accuracy for low Mach numbers. Stiffness due to low-Reynolds number effects, is not, however, eliminated in the present formulation, leading to some convergence degradation for low-speed, low Reynolds number flows. This may require the use of "viscous" preconditioning strategies or a more implicit treatment of the viscous terms. Future work will focus on combining the time-derivative preconditioning techniques with the fully-implicit formulations to arrive at a framework capable of alleviating most sources of numerical stiffness present in large-scale flow calculations.

Acknowledgments

This research was supported by NASA Langley under grant NAG-1-02052. IBM SP-2 computer time was obtained from a grant from the North Carolina Supercomputing Center.

References

[1] White, J.A. and Morrison, J.H., "A Psuedo-Temporal Multi-Grid Relaxation Scheme for Solving the Parabolized Navier-Stokes Equations," AIAA 99-3360, July, 1999.

[2] Edwards, J.R. and Liou, M.-S. "Low-Diffusion Flux-Splitting Methods for Flows at all Speeds", *AIAA Journal*, Vol. 36, No. 9, 1998, pp. 1610-1617.

[3] West, J.E., and Korkegi, R.H., "Supersonic Interaction in the Corner of Intersecting Wedges at High Reynolds Numbers," *AIAA Journal*, Vol. 10, No. 5, 1972.

[4]Weiss, J.M. and Smith, W.A., "Preconditioning Applied to Variable and Constant Density Flows," *AIAA Journal*, Vol. 33, 1995, p. 2050.

[5] Edwards, J.R. "A Low-Diffusion Flux-Splitting Scheme for Navier-Stokes Calculations," *Computers & Fluids*, Vol. 26, No. 6, 1997, pp. 635-637.

[6] Edwards, J.R., "Towards Unified CFD Simulations of Real Fluid Flows," (invited), AIAA 2001-2524CP, 2001.

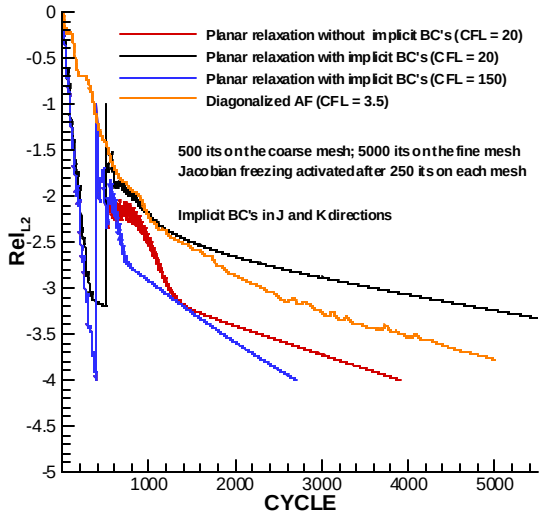

Figure 1: Convergence of planar relaxation scheme

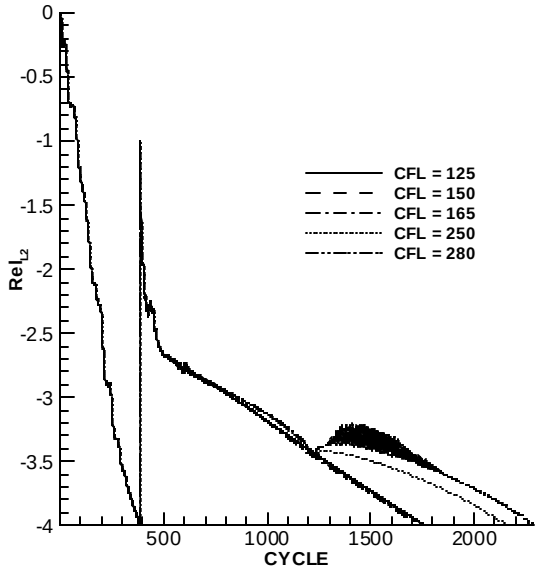

Figure 2: Effect of CFL number on convergence

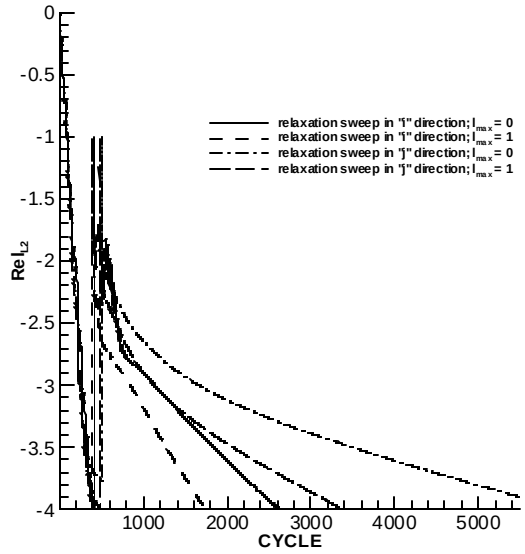

Figure 3: Effect of sweep direction on convergence

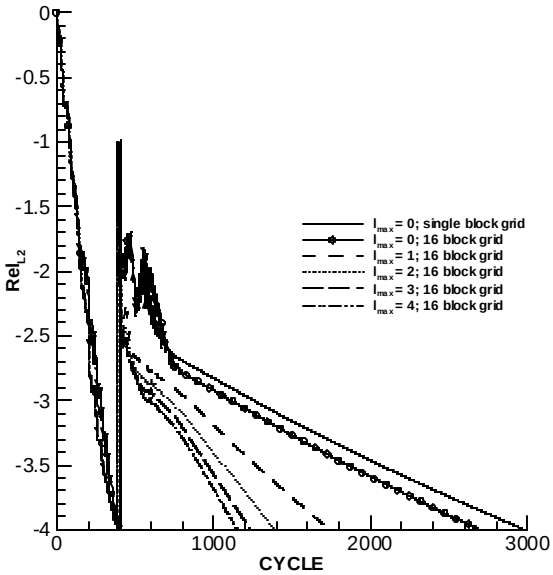

Figure 4: Effect of subiteration number on convergence

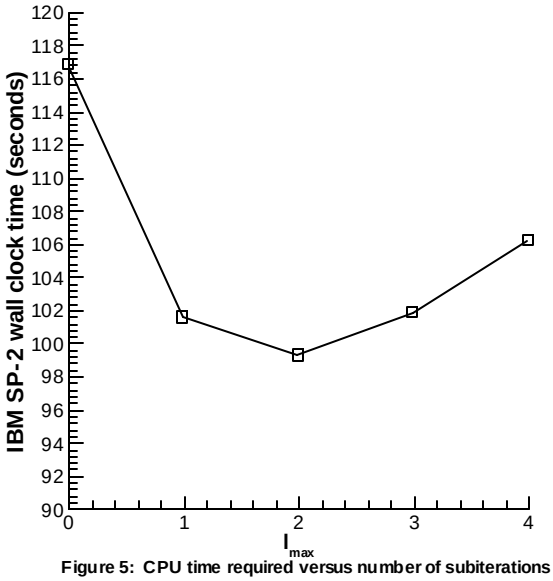

Figure 5: CPU time required versus number of subiterations

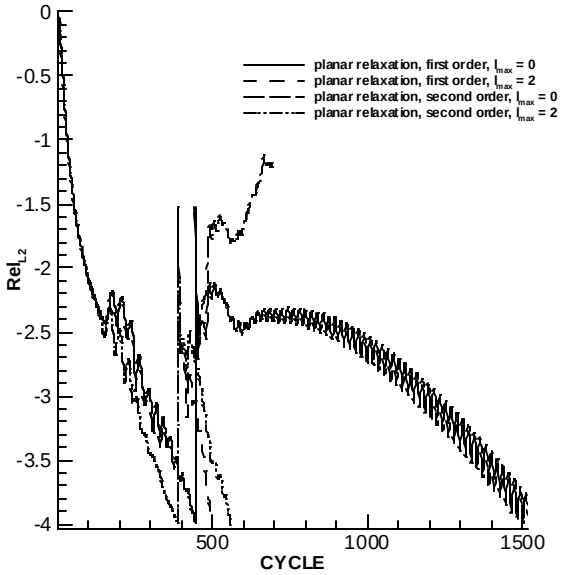

Figure 6: Effect of subiterations on subsonic channel-flow convergence

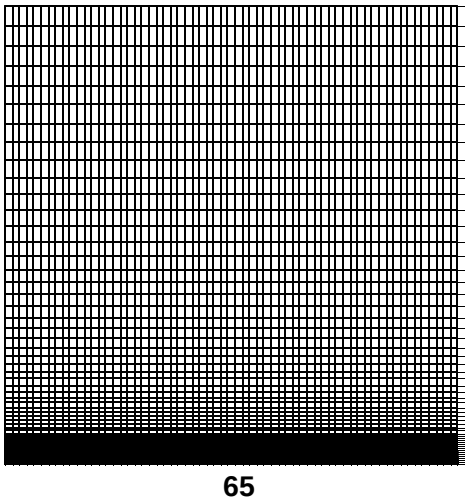

Figure 7: Computational Grid for flow over a Flat-Plate

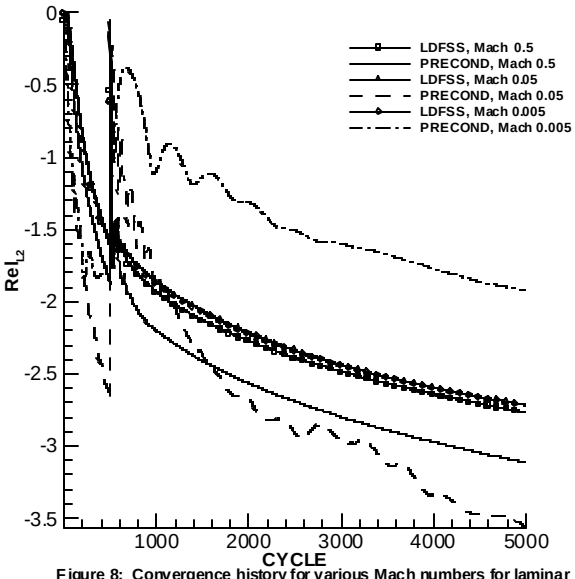

Figure 8: Convergence history for various Mach numbers for laminar flow over a Flat-Plate

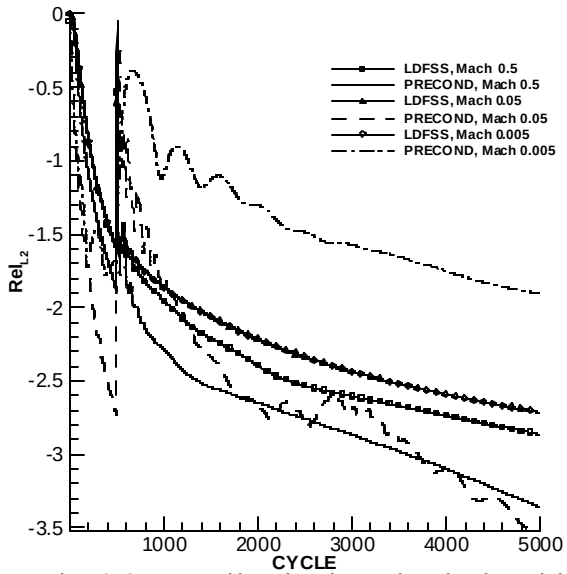

Figure 9: Convergence history for various Mach numbers for a turbulent, calorically perfect flow over a Flat-Plate

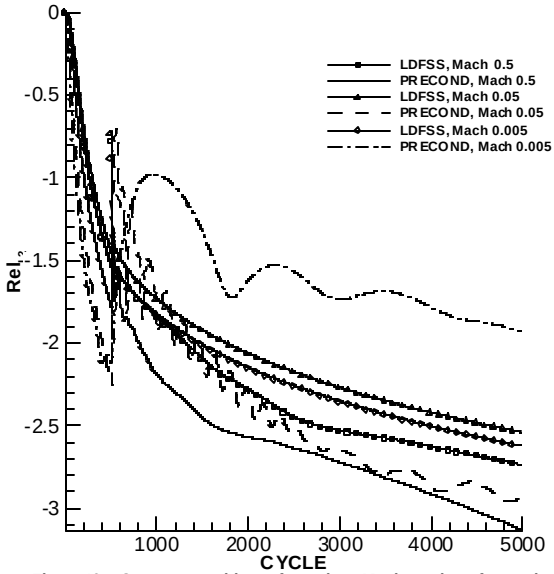

Figure 10 : Convergence history for various Mach numbers for a turbulent, two-species air flow over a Flat-Plate

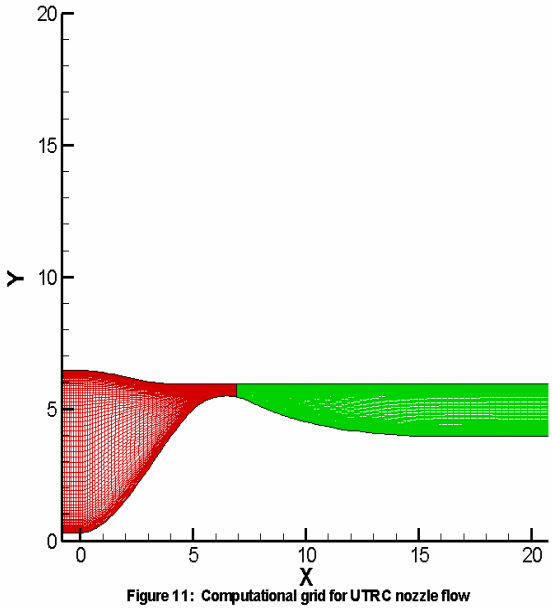

Figure 11: Computational grid for UTRC nozzle flow

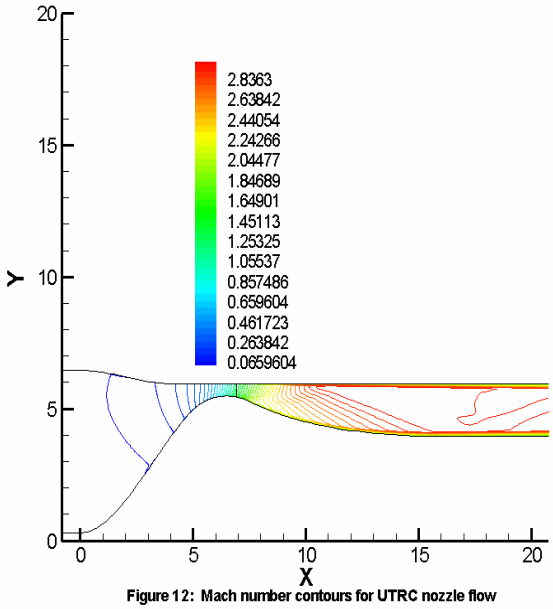

Figure 12: Mach number contours for UTRC nozzle flow

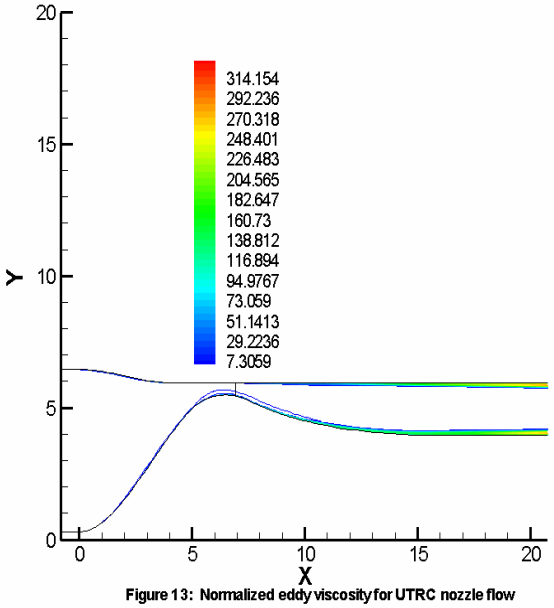

Figure 13: Normalized eddy viscosity for UTRC nozzle flow

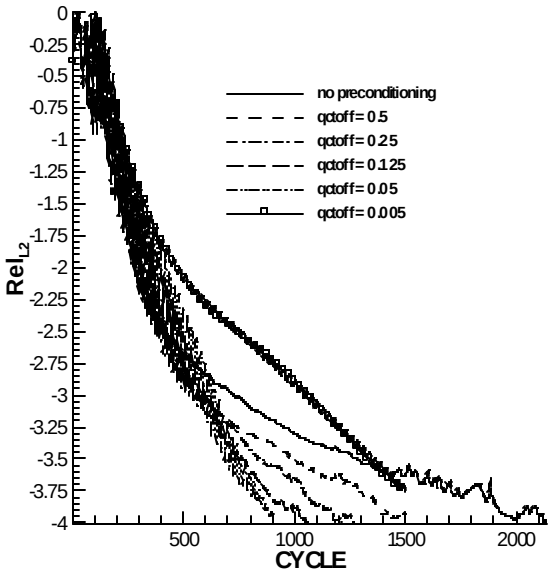

Figure 14: Convergence histories: elliptic part of UTRC nozzle calculation

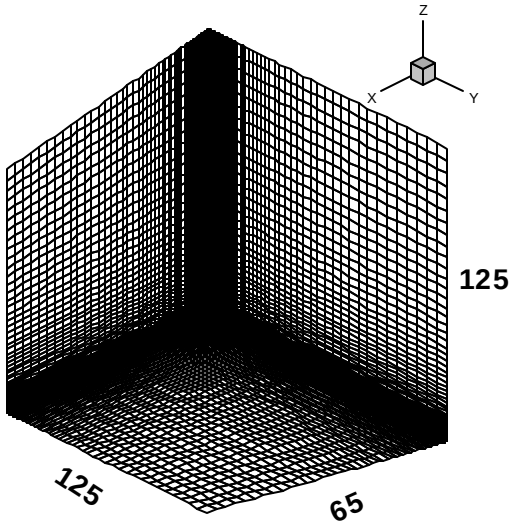

Figure 15: Computational Grid for West-Korkegi intersecting-wedges flow

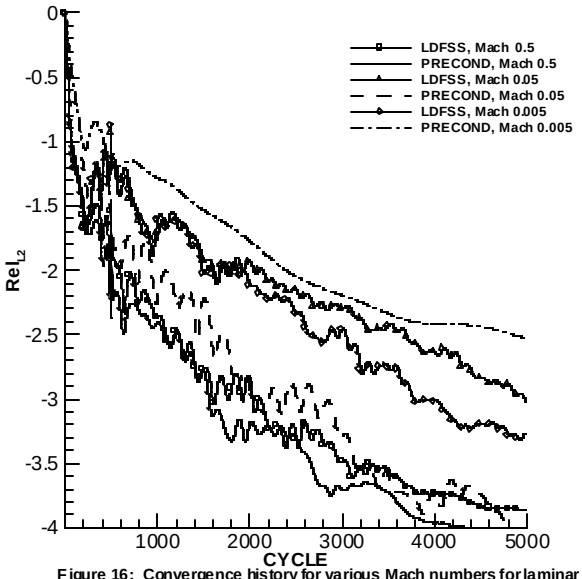

Figure 16: Convergence history for various Mach numbers for laminar flow through West-Korkegi intersecting-wedges

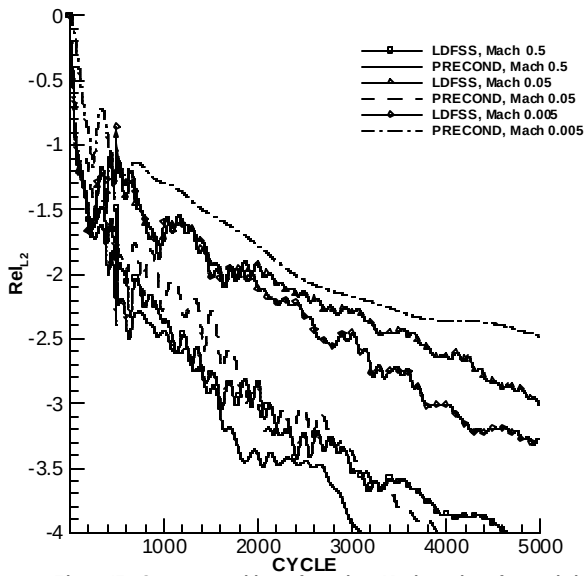

Figure 17: Convergence history for various Mach numbers for a turbulent, calorically perfect flow through the West-Korkegi intersecting-wedges

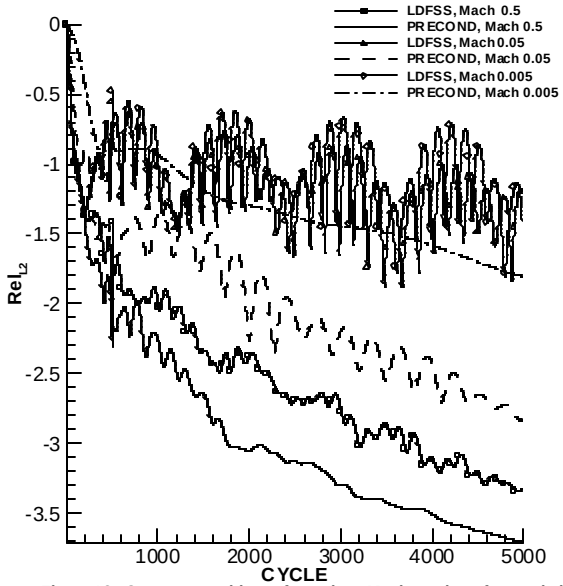

Figure 18: Convergence history for various Mach numbers for a turbulent, two-species air flow through the West-Korkegi intersecting-wedges

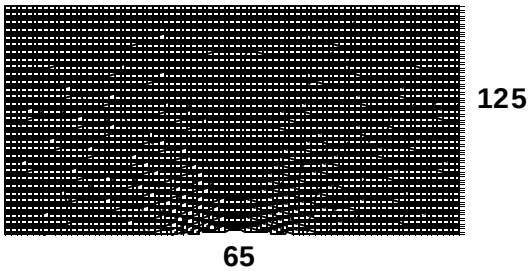

Figure 19: Computational Grid for flow over a Bump

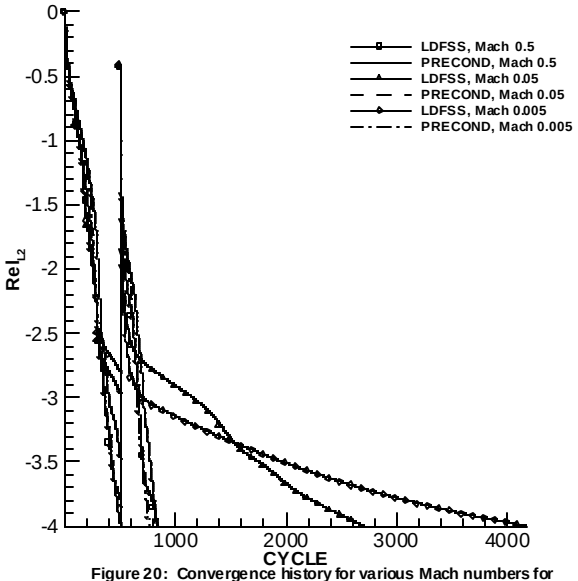

Figure 20: Convergence history for various Mach numbers for flow over a Bump

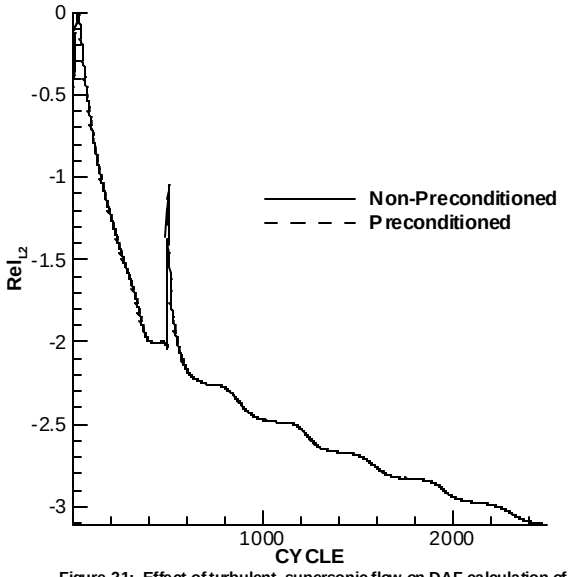

Figure 21: Effect of turbulent, supersonic flow on DAF calculation of channel flow

Table 1: Boundary layer thickness for laminar flow over Flat-Plate

X-location (meters)	d (meters) at Mach 0.05			d (meters) at Mach 0.005		
	Non-preconditioned	Preconditioned	Actual	Non-preconditioned	Preconditioned	Actual
0.3	3.68E-04	2.55E-03	2.60E-3	3.59E-04	8.15E-03	8.22E-3
% Error	85.8	2.0		95.6	0.9	
0.4	3.71E-04	2.83E-03	3.00E-3	3.63E-04	9.27E-03	9.49E-3
% Error	87.6	5.5		96.2	2.3	
0.45	3.85E-04	2.97E-03	3.18E-3	3.63E-04	9.86E-03	1.01E-2
% Error	87.9	6.7		96.4	2.1	

Table 2: Boundary layer thickness for laminar flow through Channel

X-location (meters)	d (meters) at Mach 0.05			d (meters) at Mach 0.005		
	Non-preconditioned	Preconditioned	Actual	Non-preconditioned	Preconditioned	Actual
0.3	3.72E-04	2.52E-03	2.60E-3	3.59E-04	8.16E-03	8.22E-3
% Error	85.7	2.9		95.6	0.6	
0.4	3.61E-04	2.84E-03	3.00E-3	3.79E-04	9.25E-03	9.49E-3
% Error	88.0	5.3		96.0	2.5	
0.45	3.60E-04	2.99E-03	3.18E-3	3.76E-04	9.84E-03	1.01E-2
% Error	88.7	6.1		96.3	2.2	