

## **Abstract**

LITTON, DANIEL Algorithmic Enhancements to the VULCAN Navier-Stokes Solver.  
(Under the direction of Dr. Jack R. Edwards.)

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 ( $M \leq 0.1$ ). These additions and associated modifications to the numerical discretization scheme have enabled the eigenvalues to scale with the local velocity, eliminating the above mentioned problems. With these additions, VULCAN now has improved convergence behavior for multi-block, highly-stretched meshes and also can accurately solve the Navier-Stokes equations for very low Mach numbers.

# **Algorithmic Enhancements to the VULCAN Navier-Stokes Solver**

by

**Daniel Litton**

A thesis submitted to the Graduate Faculty of  
North Carolina State University  
in partial fulfillment of the  
requirements for the Degree of  
Master of Science

**Aerospace Engineering**

Raleigh, North Carolina  
2003

**Approved By:**

---

**Dr. D. S. McRae, Member**

---

**Dr. A. Gopalarathnam, Member**

---

**Dr. J. R. Edwards, Chair**

## **Biography**

Daniel Litton was born in Chapel Hill, North Carolina on April 3, 1979. He grew up in Durham, North Carolina and graduated from C.E. Jordan High School in 1997. From there he went to North Carolina State University, where he received his Bachelor's in Aerospace Engineering in May of 2001. On June 21, 2003, Daniel married Katharine Lee Kapfer. After completion of his Master's degree, Daniel and his family will move to Southern Maryland. He will work for NAVAIR at Pax River Naval Base.

## **Acknowledgments**

I would like to thank my parents who have been there for me every step of the way through my whole education, always pushing me to go farther. I would love to thank my loving wife, who has been very understanding and supportive throughout every stage of my education at North Carolina State University.

I would like to thank my advisor, Dr. Jack R. Edwards, for his guidance and support throughout my masters. Through him I was able to gain a great deal of knowledge concerning CFD and advice for the future. The instruction from my other committee members, Dr. D. Scott McRae and Dr. Ashok Gopalarathnam has been instrumental in my success as a graduate student. I would like to also thank them for their serving on my committee.

I definitely would like to thank everyone in Broughton 4216 for their help and useful discussions over the past two years.

I would like to thank NASA Langley for supporting this research under grant NAG-1-02052. I would like to thank Jeff White for his support and guidance while working at NASA Langley.

Last, but not least, I want to thank the North Carolina Supercomputing Center, both for the computing time and for their expert assistance when problems arose.

# Table of Contents

List of Figures .....	v
List of Tables .....	vii
List of Symbols .....	viii
1 Introduction.....	1
2 Governing Equations .....	8
2.1 Navier-Stokes equation set .....	8
2.2 Equations for a calorically-perfect gas .....	14
2.3 Equations for a thermally-perfect gas .....	15
2.4 Boundary Conditions .....	16
3 Planar Relaxation Implicit Flow Solvers for Multi-Block Domains .....	18
3.1 Algorithm.....	18
4 Time-Derivative Preconditioning .....	23
4.1 Algorithm.....	23
4.2 Numerical Discretization .....	26
4.3 Time-Stepping Scheme.....	29
5 Results.....	34
5.1 Planar Relaxation Results .....	34
5.2 Preconditioning Results .....	37
5.2.1 Inviscid flow over a bump in a channel .....	38
5.2.2 Two-dimensional flow over flat plate.....	39
5.2.3 Two-dimensional flow through a nozzle .....	42
5.2.4 Three-dimensional flow through intersecting wedges .....	43
5.2.5 Three-dimensional flow through a channel .....	44
6 Conclusions.....	46
7 Future Work.....	47
References.....	48
Appendix.....	50
A Modal Matrices .....	50

## List of Figures

Figure 1: Example of decomposition of flat-plate .....	52
Figure 2: Computational grid for West-Korkegi intersecting-wedges .....	52
Figure 3: Computational grid for channel-flow .....	53
Figure 4: Convergence of planar relaxation scheme.....	53
Figure 5: Effect of CFL number on convergence .....	54
Figure 6: Effect of sweep direction on convergence .....	54
Figure 7: Effect of subiteration number on convergence.....	55
Figure 8: CPU time required versus number of subiterations.....	55
Figure 9: Effect of subiteration on subsonic channel-flow convergence.....	56
Figure 10: Condition number versus Mach number .....	56
Figure 11: Computational grid for flow over a bump.....	57
Figure 12: Convergence history for various Mach numbers for flow over a bump .....	57
Figure 13: Computational grid for flow over a flat-plate .....	58
Figure 14: Convergence history for various Mach numbers for laminar flow over a flat-plate .....	58
Figure 15: Convergence history for various Mach numbers for a turbulent, calorically perfect flow over a flat-plate.....	59
Figure 16: Convergence history for various Mach numbers for a turbulent, two-species air flow over a flat-plate.....	59
Figure 17: Convergence history for various Mach numbers for laminar flow over a flat-plate at a Reynolds number of $1.11 \times 10^7$ .....	60
Figure 18: Convergence history for various Mach numbers for a turbulent, calorically perfect flow over a flat-plate at a Reynolds number of $1.11 \times 10^7$ .....	60
Figure 19: Convergence history for various Mach numbers for a turbulent, two-species air flow over a flat-plate at a Reynolds number of $1.11 \times 10^7$ .....	61
Figure 20: Computational grid for UTRC nozzle flow.....	61
Figure 21: Mach number contours for UTRC nozzle flow.....	62
Figure 22: Normalized eddy viscosity for UTRC nozzle flow .....	62
Figure 23: Convergence histories: elliptic part of UTRC nozzle calculation.....	63
Figure 24: Convergence history for various Mach numbers for laminar flow through West-Korkegi intersecting-wedges .....	63
Figure 25: Convergence history for various Mach numbers for a turbulent, calorically perfect flow through the West-Korkegi intersecting-wedges.....	64
Figure 26: Convergence history for various Mach numbers for a turbulent, two-species air flow through the West-Korkegi	

	intersecting-wedges.....	64
Figure 27:	Convergence history for various Mach numbers for laminar flow through West-Korkegi intersecting-wedges at a Reynolds number of $1.11 \times 10^7$ .....	65
Figure 28:	Convergence history for various Mach numbers for laminar flow through a channel.....	65
Figure 29:	Convergence history for various Mach numbers for a turbulent, calorically perfect gas flow through a channel.....	66
Figure 30:	Convergence history for various Mach numbers for a turbulent, two-species air flow through a channel .....	66
Figure 31:	Comparison of preconditioned system with non-preconditioned system for supersonic flow .....	67

## List of Tables

Table 1: Reference conditions at Mach 0.5 .....	68
Table 2: Boundary layer thickness for laminar flow over flat-plate .....	68
Table 3: Boundary layer thickness for laminar flow through a channel.....	69



## List of Symbols

$a$	Speed of sound
$a_{\frac{I}{2}}$	Preconditioned interface sound speed
$C$	Condition number
$C_p$	Specific heat at constant pressure
$D_i$	Diffusion coefficient of species i
$D_{\phi x_j}$	Modeled turbulent diffusion
$E^C$	Convective contribution to the flux
$E_I$	Interface flux
$E^P$	Pressure contribution to the flux
$E_t$	Total energy per unit volume
$E, F, G$	Flux vector
$E_v, F_v, G_v$	Viscous flux vector
$e_v, f_v, g_v$	Components of the viscous energy flux
$h$	Enthalpy per unit mass
$H$	Total enthalpy
$i, j, k$	Grid locations representing x, y, and z directions
$J$	Jacobian
$K$	Cutoff scalar
$k$	Turbulent kinetic energy
$l$	Iteration number
$M$	Mach number
$\tilde{M}_L, \tilde{M}_R$	Left and right Mach numbers
$M_{ref}$	Reference Mach number
$M_i$	Molecular weight of species i
$n$	Time level
$P_{L,R}^{\pm}$	Subsonic pressure splitting
$Pr_l, Pr_T$	Laminar and turbulent Prandtl number
$p$	Pressure
$q_x, q_y, q_z$	Total heat flux vector
$Q$	Solution vector
$R$	Residual
$R_i$	Species gas constant
$R_u$	Universal gas constant
$Re$	Reynolds number

$Re_x$	Reynolds number at distance x from leading edge
$S$	Source term
$S_c, S_{c_T}$	Laminar and turbulent Schmidt number
$T$	Temperature
$T_k^p, (T_k^p)^{-1}$	Preconditioned modal matrix
$t$	Time
$\Delta t$	Time step
$u, v, w$	Cartesian velocity components
$u^d, v^d, w^d$	Cartesian diffusion velocity component
$U, V, W$	Contravariant velocity components
$U^d, V^d, W^d$	Contravariant diffusion velocity components
$U_L, U_R$	Left and right state at cell interface
$V_\infty$	Freestream velocity
$V_{ref}$	Reference velocity
$x, y, z$	Cartesian coordinates
$Y_i$	Mass fraction of species i
$\xi, \eta, \zeta$	Transformed coordinates
$\mu, \mu_t$	Laminar and turbulent viscosity
$\lambda$	Eigenvalue
$\lambda^p_{\xi,c}$	Eigenvalue of the preconditioned Euler system
$\lambda^p_{\xi,v}$	Viscous eigenvalue of the preconditioned Navier-Stokes system
$\kappa$	Parameter to choose higher order schemes
$\Phi$	Parameter to choose first order scheme
$\psi$	Limiter
$\chi$	Thermal conductivity
$\omega$	Turbulence frequency
$\rho$	Density
$\tau_{ij}$	Viscous stresses
$\theta$	Scaling parameter

# 1 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. VULCAN solves the Navier-Stokes using conservative variables, rather than primitive. The purpose of this research is to extend the capabilities of VULCAN in two ways. First an implicit algorithm will be used so that VULCAN can more efficiently solve multiblock problems by using an algorithm that has better block-to-block coupling. The second enhancement involves an improvement to the compressible flow solver already in VULCAN. The addition of time derivative preconditioning to the compressible solver would extend VULCAN's range to very low Mach numbers. Without preconditioning the current compressible flow solver can not accurately or efficiently solve for low speed ( $M < 0.1$ ) flows. The goal of both of these enhancements is to ensure that the VULCAN code will be able to solve for any flow, over any domain, without facing the stiffness problems associated with widely-differing characteristic speeds, widely-varying grid spacing, and large numbers of connected grid blocks.

The baseline integration strategy for Navier-Stokes applications within VULCAN is a diagonalized approximate factorization (DAF) scheme. This scheme is rather efficient on a per-iteration basis; however, it is sensitive to the near-wall grid

spacing. Convergence rates can degrade rapidly for highly-stretched meshes. Due to the lack of strong coupling between adjacent blocks, convergence rates are further degraded when large numbers of blocks are used.

Part of the present effort is concerned with improving the numerical efficiency of VULCAN for viscous flows on multi-block, highly-stretched meshes. Planar relaxation based implicit methods are introduced, along with sub-iterative procedures that allow for a large degree of implicit coupling among blocks. The implicit algorithm is an important aspect of eliminating convergence problems created by using multiple blocks. In order to accomplish this, the algorithm takes advantage of the MPI message-passing implemented in VULCAN. MPI message-passing is used to adapt VULCAN to parallel architectures by allowing necessary information to be passed between processors. With this new algorithm in place, it is expected that the convergence of most, if not all problems, will be greatly improved. It will be proven that the benefits of the new algorithm will outweigh the extra calculation time required in implementing this algorithm, at least for the problems tested.

The other issue being addressed by this research is VULCAN's capability to solve low speed flows. VULCAN is designed for higher Mach number applications, and like most compressible flow solvers, it can experience convergence degradation and solution inaccuracy for very low Mach number flows. This problem is due to the stiffness of the Navier-Stokes equations. Preconditioning of the compressible solver is an essential element in eliminating this problem and in being able to solve low speed, nearly incompressible flows.

Due to the stiffness of the Navier-Stokes equations, it was once necessary to solve low speed flows using techniques different from those used to solve for compressible flows. One of the major issues when solving the Navier-Stokes and Euler equations for low speed, near incompressible flows is that the continuity equation no longer contains a time derivative term of density due to this term approaching zero as the incompressible limit is reached. This leaves no physical way to advance the solution of the pressure field. Due to this issue, low speed, incompressible flows were typically solved by a pressure-based scheme and compressible flows by a density based scheme. The pressure-based methods typically solve the Navier-Stokes equations in a decoupled approach. A typical pressure-based code initially solves the momentum equation for a provisional velocity field based on an initial guess for the pressure distribution. Since the momentum equations are uncoupled from the continuity equation, the continuity equation itself will not be satisfied. As a result, the pressure will need to be improved in order to satisfy the continuity equation. This improvement is typically done by a relaxation scheme to solve the resulting Poisson equation for pressure. As a result of the change in pressure, the solution to the momentum equations will change, leaving a new velocity field. These steps are repeated until convergence is reached based on the continuity equation or other such criteria. In contrast, compressible flows are typically solved through a time advancement scheme, and the equations are advanced in tandem with one another.

Since many flows contain regions of very low Mach number flow embedded in regions of high speed flow, it can become cumbersome and inefficient to have different solvers for different regions of the flow. An example of this problem would be an

airfoil flying in the transonic region that contains an area of low speed flow at the stagnation point.

To eliminate the need to have a different solver for low and high speed flows, it has become important to find one solver that would combine the best features of the “pressure-based” and “density-based” codes. The major issue that needs to be addressed is the disparity in the eigenvalues at low speeds. For a low speed flow, the magnitude of the convection speed is much lower than the acoustic speed. Due to the magnitude difference of the convection and acoustic speeds, the eigenvalues of the Euler equations  $(u, u + a, u - a)$  vary greatly from one another. This disparity in the eigenvalues is what causes the system to become very stiff. There are currently many solutions available to address this issue for compressible and incompressible flow solvers of all types. One of the first attempts [2] introduced an artificial time derivative of pressure into the continuity equation. With this artificial time derivative in place, it is now possible to advance the solution of the pressure field. Also, this term contains a parameter that serves to condition the eigenvalues to the same magnitude as the convection speed. This term is designed to vanish when the flow reaches steady state. This method, termed the Artificial Compressibility Method, was first introduced by Alexandre Chorin in 1967. [2] This artificial time derivative is placed only in the continuity equation and is given by

$$\frac{1}{V_{ref}^2} \frac{\partial p}{\partial t} + \rho \left( \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} \right) = 0 \quad (1)$$

for 2-D, incompressible flows. The term  $\frac{1}{V_{ref}^2} \frac{\partial p}{\partial t}$  is the artificial time derivative that will vanish at steady state. The reference velocity,  $V_{ref}$ , is typically set to a multiple of the maximum flow velocity. Since the initial equation set is formulated for the incompressible Navier-Stokes equations, this derivation will not work for compressible flows without additions.

Although Chorin's preconditioner only works for low speed, nearly incompressible flows, it can be extended to work for compressible flows. Currently, there are many different methods available to extend compressible solvers to low speed flows. [3] [4] [5] The preconditioner in VULCAN is that of Weiss and Smith [5]; however, it is formulated for conservative variables, rather than primitive variables. Their formulation has been proven to work for flows of all speeds.

According to Weiss and Smith, the artificial time derivative needs one major change in order to work for compressible flows. The equation set needs to have the ability to revert back to the compressible equations as the compressible limit is reached. This is accomplished by the addition of a parameter similar to Chorin's to all equations in the compressible Navier-Stokes equation set. As before, this parameter will help the system to maintain good conditioning of the eigenvalues for low speed flows. The continuity equation with the addition of this parameter is defined as

$$\left( \frac{1}{V_{ref}^2} - \frac{1}{a^2} \right) \frac{\partial p}{\partial t} + \frac{\partial \rho}{\partial t} + \frac{\partial(\rho u)}{\partial x} + \frac{\partial(\rho v)}{\partial y} = 0 \quad (2)$$

The parameter  $\left( \frac{1}{V_{ref}^2} - \frac{1}{a^2} \right) \frac{\partial p}{\partial t}$  is the additional term added to the continuity equation.

This term is derived as follows:

1. Chorin's artificial time derivative,  $\frac{1}{V_{ref}^2} \frac{\partial p}{\partial t}$ , is added to the continuity equation
2. The time derivative of density,  $\frac{\partial \rho}{\partial t}$  is then subtracted from the continuity equation in order to counter the effect of Chorin's artificial time derivative
3. This time derivative of density is then converted to a time derivative of pressure by means of the isentropic flow relations to produce  $\frac{1}{a^2} \frac{\partial p}{\partial t}$

Once these changes are implemented, the altered continuity Equation 2 is formed. At low speeds ( $V_{ref}^2 \ll a^2$ ), Chorin's artificial time derivative dominates, and the eigenvalues become well conditioned. When  $V_{ref}^2$  is equal to  $a^2$ , the whole term vanishes, and the original compressible equations are solved. Further derivation and discussion can be located in Section 4 after the necessary equations have been defined.

The remainder of this thesis is outlined as follows: Section 2 will introduce the Navier-Stokes equation set and the necessary equations to close them. Section 3 will explain the planar-relaxation method and the algorithm to improve block-to-block coupling. Section 4 will describe the algorithm used for preconditioning, the numerical discretization, and the time stepping scheme. Finally, Section 5 will present the results



for the inter-block coupling, as well as the results for the preconditioned system, while Sections 6 and 7 will present conclusions and future work.

## 2 Governing Equations

### 2.1 Navier-Stokes equation set

VULCAN solves the three-dimensional Navier-Stokes equations, expanded to include separate transport equations for individual species and two-equation turbulence model components. The Navier Stokes Equations are written in a generalized coordinate system by the transformation:

$$\begin{aligned}\xi &= \xi(x, y, z) \\ \eta &= \eta(x, y, z) \\ \zeta &= \zeta(x, y, z)\end{aligned}\tag{3}$$

Written in this generalized coordinate system, the Navier-Stokes equations are given by

$$\frac{dQ}{dt} + R(Q) = 0\tag{4}$$

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

$$R(Q) = \frac{\partial(E - E_v)}{\partial\xi} + \frac{\partial(F - F_v)}{\partial\eta} + \frac{\partial(G - G_v)}{\partial\zeta} - S\tag{5}$$

The Navier-Stokes equation set is Favre-averaged, but the notation is left out for simplicity. The solution vector,  $Q$ , is given by:

$$Q = \frac{1}{J} \begin{bmatrix} \rho Y_1 \\ \vdots \\ \rho Y_{N_{CS}} \\ \rho \\ \rho u \\ \rho v \\ \rho w \\ E_t \\ \rho k \\ \rho \omega \end{bmatrix} \quad (6)$$

and the inviscid flux terms are defined as

$$E = \frac{|\nabla \xi|}{J} \begin{bmatrix} \rho Y_1 U \\ \vdots \\ \rho Y_{N_{CS}} U \\ \rho U \\ \rho U u + p \bar{\xi}_x \\ \rho U v + p \bar{\xi}_y \\ \rho U w + p \bar{\xi}_q \\ (E_t + p)U \\ \rho k U \\ \rho \omega U \end{bmatrix} \quad F = \frac{|\nabla \eta|}{J} \begin{bmatrix} \rho Y_1 V \\ \vdots \\ \rho Y_{N_{CS}} V \\ \rho V \\ \rho u V + \bar{\eta}_x p \\ \rho v V + \bar{\eta}_y p \\ \rho w V + \bar{\eta}_z p \\ (E_t + p)V \\ \rho k V \\ \rho \omega V \end{bmatrix} \quad G = \frac{|\nabla \zeta|}{J} \begin{bmatrix} \rho Y_1 W \\ \vdots \\ \rho Y_{N_{CS}} W \\ \rho W \\ \rho u W + \bar{\zeta}_x p \\ \rho v W + \bar{\zeta}_y p \\ \rho w W + \bar{\zeta}_z p \\ (E_t + p)W \\ \rho k W \\ \rho \omega W \end{bmatrix} \quad (7)$$

The viscous flux vectors,  $E_v$ ,  $F_v$ , and  $G_v$  are given as

$$\begin{aligned}
E_v = & \begin{bmatrix} \rho Y_1 U^d \bar{\xi}_{x_i} \\ \vdots \\ \rho Y_{N_{CS}} U^d \bar{\xi}_{x_i} \\ \rho U^d \bar{\xi}_{x_i} \\ \bar{\xi}_x \tau_{xx} + \bar{\xi}_y \tau_{xy} + \bar{\xi}_z \tau_{xz} - \bar{\xi}_{x_j} \overline{\overline{\rho u'' u''}}_j \\ \bar{\xi}_x \tau_{xy} + \bar{\xi}_y \tau_{yy} + \bar{\xi}_z \tau_{yz} - \bar{\xi}_{x_j} \overline{\overline{\rho v'' u''}}_j \\ \bar{\xi}_x \tau_{xz} + \bar{\xi}_y \tau_{yz} + \bar{\xi}_z \tau_{zz} - \bar{\xi}_{x_j} \overline{\overline{\rho w'' u''}}_j \\ \bar{\xi}_x e_v + \bar{\xi}_y f_v + \bar{\xi}_z g_v \\ \bar{\xi}_x D_{kx} + \bar{\xi}_y D_{ky} + \bar{\xi}_z D_{kz} \\ \bar{\xi}_x D_{\omega x} + \bar{\xi}_y D_{\omega y} + \bar{\xi}_z D_{\omega z} \end{bmatrix} \quad F_v = \begin{bmatrix} \rho Y_1 V^d \bar{\eta}_{x_i} \\ \vdots \\ \rho Y_{N_{CS}} V^d \bar{\eta}_{x_i} \\ \rho V^d \bar{\eta}_{x_i} \\ \bar{\eta}_x \tau_{xx} + \bar{\eta}_y \tau_{xy} + \bar{\eta}_z \tau_{xz} - \bar{\eta}_{x_j} \overline{\overline{\rho u'' u''}}_j \\ \bar{\eta}_x \tau_{xy} + \bar{\eta}_y \tau_{yy} + \bar{\eta}_z \tau_{yz} - \bar{\eta}_{x_j} \overline{\overline{\rho v'' u''}}_j \\ \bar{\eta}_x \tau_{xz} + \bar{\eta}_y \tau_{yz} + \bar{\eta}_z \tau_{zz} - \bar{\eta}_{x_j} \overline{\overline{\rho w'' u''}}_j \\ \bar{\eta}_x e_v + \bar{\eta}_y f_v + \bar{\eta}_z g_v \\ \bar{\eta}_x D_{kx} + \bar{\eta}_y D_{ky} + \bar{\eta}_z D_{kz} \\ \bar{\eta}_x D_{\omega x} + \bar{\eta}_y D_{\omega y} + \bar{\eta}_z D_{\omega z} \end{bmatrix} \quad (8)
\end{aligned}$$

$$\begin{aligned}
G_v = & \begin{bmatrix} \rho Y_1 W^d \bar{\zeta}_{x_i} \\ \vdots \\ \rho Y_{N_{CS}} W^d \bar{\zeta}_{x_i} \\ \rho W^d \bar{\zeta}_{x_i} \\ \bar{\zeta}_x \tau_{xx} + \bar{\zeta}_y \tau_{xy} + \bar{\zeta}_z \tau_{xz} - \bar{\zeta}_{x_j} \overline{\overline{\rho u'' u''}}_j \\ \bar{\zeta}_x \tau_{xy} + \bar{\zeta}_y \tau_{yy} + \bar{\zeta}_z \tau_{yz} - \bar{\zeta}_{x_j} \overline{\overline{\rho v'' u''}}_j \\ \bar{\zeta}_x \tau_{xz} + \bar{\zeta}_y \tau_{yz} + \bar{\zeta}_z \tau_{zz} - \bar{\zeta}_{x_j} \overline{\overline{\rho w'' u''}}_j \\ \bar{\zeta}_x e_v + \bar{\zeta}_y f_v + \bar{\zeta}_z g_v \\ \bar{\zeta}_x D_{kx} + \bar{\zeta}_y D_{ky} + \bar{\zeta}_z D_{kz} \\ \bar{\zeta}_x D_{\omega x} + \bar{\zeta}_y D_{\omega y} + \bar{\zeta}_z D_{\omega z} \end{bmatrix} \quad (9)
\end{aligned}$$

with the source-term vector is given as

$$S = \frac{1}{J} \begin{bmatrix} \dot{\omega}_i \\ \vdots \\ \dot{\omega}_{N_{CS}} \\ \dot{\omega} \\ 0 \\ 0 \\ 0 \\ 0 \\ S_k \\ S_\omega \end{bmatrix} \quad (10)$$

In the above equations,  $Y_i$  is the mass fraction of the  $i^{\text{th}}$  chemical species and  $N_{CS}$  is the total number of chemical species. Also,  $p$  is the pressure,  $\rho$  is the density,  $u$ ,  $v$  and  $w$  are the Cartesian velocity components,  $E_t$  is the total energy, and the metric derivatives are defined as  $\xi_k, \eta_k, \zeta_k$ . Finally, the turbulent terms,  $k$  and  $\omega$ , are defined as the turbulent kinetic energy and the turbulence frequency. In the viscous flux vectors,  $\tau_{ij}$  are the viscous stresses and  $e_v, f_v, g_v$  are the components of the viscous energy flux given by

$$\begin{aligned} \tau_{xx} &= \mu \left[ 2 \frac{\partial u}{\partial x} - \frac{2}{3} \left( \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} \right) \right] \\ \tau_{yy} &= \mu \left[ 2 \frac{\partial v}{\partial y} - \frac{2}{3} \left( \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} \right) \right] \\ \tau_{zz} &= \mu \left[ 2 \frac{\partial w}{\partial z} - \frac{2}{3} \left( \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} \right) \right] \end{aligned} \quad (11)$$

$$\begin{aligned}
\tau_{xy} &= \mu \left( \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \right) \\
\tau_{xz} &= \mu \left( \frac{\partial u}{\partial z} + \frac{\partial w}{\partial x} \right) \\
\tau_{yz} &= \mu \left( \frac{\partial v}{\partial z} + \frac{\partial w}{\partial y} \right)
\end{aligned} \tag{12}$$

and

$$\begin{aligned}
e_v &= u\tau_{xx} + v\tau_{xy} + w\tau_{xz} - q_x + D_{\phi x} \\
f_v &= u\tau_{xy} + v\tau_{yy} + w\tau_{yz} - q_y + D_{\phi y} \\
g_v &= u\tau_{xz} + v\tau_{yz} + w\tau_{zz} - q_z + D_{\phi z}
\end{aligned} \tag{13}$$

where the total heat flux and the modeled turbulent diffusion terms are defined as

$$q_{x_j} = - \left( \chi + \frac{\mu_t}{\text{Pr}_t} \right) \frac{\partial T}{\partial x_j} - \rho \left( \frac{\mu}{S_c} + \frac{\mu_t}{S_{ct}} \right) \sum_{i=1}^{N_{CS}} h_i Y_{ij} \tag{14}$$

$$D_{\phi x_j} = \left( \frac{\mu}{\text{Pr}_{l_\phi}} + \frac{\mu_t}{\text{Pr}_{t_\phi}} \right) \frac{\partial \phi}{\partial x_j} \tag{15}$$

The subscript  $j$  represents the coordinate direction x, y, or z and  $\phi$  represents either the  $k$  or  $\omega$  turbulent equation.  $\chi$  represents the thermal conductivity. The terms  $\mu$  and  $\mu_t$  are defined as the laminar and turbulent viscosities, while  $\text{Pr}_l$  and  $\text{Pr}_t$  are the laminar and turbulent Prandtl number.  $S_c$  and  $S_{ct}$  are the laminar and turbulent Schmidt numbers. Finally, the term  $\overline{\rho u_i'' u_j''}$  is defined as the Favre-averaged Reynolds-stress and is closed with the Boussinesq approximation

$$-\overline{\overline{\rho u_i'' u_j''}} = \mu_t \left[ \left( \frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) - \frac{2}{3} \frac{\partial u_k}{\partial x_k} \delta_{ij} \right] - \frac{2}{3} \rho k \delta_{ij} \quad (16)$$

where

$$\mu_t = \frac{k}{\omega} \quad (17)$$

The source terms  $S_k$  and  $S_\omega$  are represented by

$$\begin{aligned} S_k &= \overline{\overline{\rho u_i'' u_j''}} \frac{\partial U_i}{\partial x_j} - \beta^* k \omega \\ S_\omega &= \alpha \frac{\omega}{k} \overline{\overline{\rho u_i'' u_j''}} \frac{\partial U_i}{\partial x_j} - \beta \omega^2 \end{aligned} \quad (18)$$

where the relations in the above equation are given from Wilcox [6] as

$$\alpha = \frac{13}{25}, \quad \beta = \beta_0 f_\beta, \quad \beta^* = \beta_0^* f_{\beta^*}, \quad \beta_0 = \frac{9}{25}, \quad \beta_0^* = \frac{9}{100} \quad (19)$$

$$F_\beta = \frac{1+70\chi_\omega}{1+80\chi_\omega}, \quad \chi_\omega \equiv \left| \frac{\Omega_{ij} \Omega_{jk} S_{ki}}{(\beta_0^* \omega)^3} \right|, \quad \Omega_{ij} = \frac{1}{2} \left( \frac{\partial U_i}{\partial x_j} - \frac{\partial U_j}{\partial x_i} \right) \quad (20)$$

$$S_{ij} = \frac{1}{2} \left( \frac{\partial U_i}{\partial x_j} + \frac{\partial U_j}{\partial x_i} \right), \quad f_{\beta^*} = \begin{cases} 1, & \chi_k > 0 \\ \frac{1+680\chi_k^2}{1+400\chi_k^2}, & \chi_k > 0 \end{cases} \quad \chi_k \equiv \frac{1}{\omega^3} \frac{\partial k}{\partial x_j} \frac{\partial \omega}{\partial x_j} \quad (21)$$

In the above equations, the components of the cell face unit normal and the contravariant velocities are given as

$$\bar{\xi}_x = \frac{\xi_x}{|\nabla \xi|}, \quad \bar{\eta}_x = \frac{\eta_x}{|\nabla \eta|}, \quad \bar{\zeta}_x = \frac{\zeta_x}{|\nabla \zeta|} \quad (22)$$

$$|\nabla \xi| = \sqrt{\xi_x^2 + \xi_y^2 + \xi_z^2} \quad (23)$$

$$U = \bar{\xi}_x u + \bar{\xi}_y v + \bar{\xi}_z w, \quad V = \bar{\eta}_x u + \bar{\eta}_y v + \bar{\eta}_z w, \quad W = \bar{\zeta}_x u + \bar{\zeta}_y v + \bar{\zeta}_z w \quad (24)$$

The contravariant diffusion velocities are given by

$$U^d = \bar{\xi}_x u^d + \bar{\xi}_y v^d + \bar{\xi}_z w^d \quad V^d = \bar{\eta}_x u^d + \bar{\eta}_y v^d + \bar{\eta}_z w^d \quad (25)$$

$$W^d = \bar{\zeta}_x u^d + \bar{\zeta}_y v^d + \bar{\zeta}_z w^d$$

where  $u^d, v^d$ , and  $w^d$  are defined as the Cartesian diffusion velocities given by Fick's law:

$$u^d = D_i \frac{\partial Y_i}{\partial x} \frac{1}{Y_i}, \quad v^d = D_i \frac{\partial Y_i}{\partial y} \frac{1}{Y_i}, \quad w^d = D_i \frac{\partial Y_i}{\partial z} \frac{1}{Y_i} \quad (26)$$

with  $D_i$  representing the diffusion coefficient of species  $i$  and  $\frac{\partial Y_i}{\partial x}$  the concentration gradient.

The Jacobian,  $J$ , of the transformation is defined as [7]

$$J = \frac{1}{J^{-1}} = \frac{1}{\frac{\partial(x, y, z)}{\partial(\xi, \eta, \zeta)}} = \frac{1}{\begin{vmatrix} x_\xi & x_\eta & x_\zeta \\ y_\xi & y_\eta & y_\zeta \\ z_\xi & z_\eta & z_\zeta \end{vmatrix}} \quad (27)$$

## 2.2 Equations for a calorically-perfect gas

VULCAN has the ability to model both a single component, calorically perfect and a multi-component, thermally perfect gas. For a calorically perfect gas, the



pressure, total enthalpy, static enthalpy, specific heat at constant pressure, and total energy are given by

$$p = \rho R T \quad (28)$$

$$H = h + \frac{1}{2}(u^2 + v^2 + w^2) \quad (29)$$

$$h = C_p T \quad (30)$$

$$C_p = \frac{\gamma R}{\gamma - 1} \quad (31)$$

$$E_t = \rho H - p \quad (32)$$

where  $R$  is defined to be the gas constant (287 J / Kg-K for air).

## 2.3 Equations for a thermally-perfect gas

The equations for a multi-component, thermally-perfect gas are different from a single component, calorically perfect gas. For a thermally-perfect gas, the pressure, species enthalpy, and static enthalpy are formed by the expressions:

$$p = \rho R_u T \sum_{i=1}^{N_{CS}} \frac{Y_i}{M_i} \quad (33)$$

$$h_i = h_i^0 + \int_{T_0}^T C_{p,i} dT \quad (34)$$

$$h = \sum_{i=1}^{N_{CS}} h_i Y_i \quad (35)$$

where  $R_u$  is the universal gas constant (8314 J / kmol-K) and  $M_i$  is the species molecular weight. The specific heat and enthalpy are modeled with a polynomial curve fit based on the temperature from McBride et al [8], [9].

## 2.4 Boundary Conditions

The boundary conditions are specified for four different sets of conditions. Three of these are expressed for the Navier Stokes equations. The final set of boundary conditions is written for the Euler equations.

The first set of boundary conditions are set up for solving a supersonic flow. Customary of supersonic inflow boundary conditions, all variables are fixed, equal to the freestream quantities. All outflow boundaries are set up as a zeroth order extrapolation of all variables. Finally, the no-slip, adiabatic wall condition is maintained at every wall.

The second set of boundary conditions entails solving for a laminar and turbulent flow of a calorically perfect, subsonic, gas. Based upon the sign of the incoming and outgoing waves traveling normal to the boundary, the Riemann invariants are used to solve for the inflow boundary conditions. The inflow boundary condition requires specifying the values of entropy,  $u$ ,  $v$ , and  $w$  velocities and solving for the density. The outflow boundary has a specified static pressure while a zeroth order extrapolation is used to determine all other variables. Again, the no-slip, adiabatic wall condition is used at all walls.

The third set of boundary conditions is required to solve for a laminar and turbulent flow of a thermally perfect, subsonic, gas. The main difference from the second set is that the characteristic equations are not coded for a thermally perfect gas. Rather, a zeroth-order extrapolation of the Riemann invariant is used to calculate the sound speed at the inflow, and the rest of the flow is solved from calorically perfect, isentropic relationships. As before, the variables are determined at the wall by ensuring that the no-slip, adiabatic wall condition is upheld.

Finally, the last set of boundary conditions is specified for solving the Euler equations. The only difference from the second set of boundary conditions is the wall boundary condition. Here, the wall is treated as a slip wall. An interior extrapolation of pressure is used to determine its value on the wall.

## **3 Planar Relaxation Implicit Flow Solvers for Multi-Block**

### **Domains**

#### **3.1 Algorithm**

Typical computational geometries are extremely complex, requiring many mesh nodes to resolve sufficiently for an accurate solution. Resolution is frequently enhanced by decomposing the geometry into smaller pieces, called blocks. An example of such a grid-decomposition can be seen in Figure 1. This is an example of the decomposition of a flat-plate into 16 blocks, where blocks 1 and 16 are denoted in the figure. One or more blocks can be assigned to a processor in such a manner that the memory and storage requirements are distributed evenly. Through MPI message passing, each processor then shares the necessary information between blocks/processors that are adjacent in the domain in order to complete the calculations.

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 for one or two mesh cells of overlap between adjacent domains. Problems associated with slower convergence are a direct result of having typical implicit solvers, formulated for multi-block arrangements, not considering 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 subdomain implicit problem and is designed to solve the complete (not parabolized) Navier-Stokes equations. The planar relaxation scheme is first set to solve the equations implicitly on each plane and then step to the next plane in the chosen direction and again solve the equations implicitly. The crossflow plane linear system is approximately solved using an incomplete LU decomposition procedure. Finally, the sweep, or step, may be block-specific in any of the desired directions:  $\xi, \eta, \zeta$ .

The planar relaxation scheme is based upon solving the planar block-pentadiagonal system for each domain. Let the block septadiagonal system,  $A$ , be defined as  $[a, b, c, d, f, g, h]$ , where  $a$  represents the linearization of the equation system at the point  $(i-1, j, k)$ ,  $b$  represents the linearization about point  $(i, j-1, k)$ ,  $c$  represents the linearization about point  $(i, j, k-1)$ ,  $d$  represents the block diagonal,  $f$  represents the linearization about point  $(i, j, k+1)$ ,  $g$  represents the linearization about point  $(i, j+1, k)$ , and finally,  $h$  represents the linearization about point  $(i+1, j, k)$ . This implicit operator is then factored as

$$M = (D + a)D^{-1}(D + h) \quad (36)$$

where

$$D = (\tilde{d} + b + c)\tilde{d}^{-1}(\tilde{d} + f + g) \quad (37)$$

with  $\tilde{d}$  recursively defined as  $\tilde{d}_{i,j,k} = d_{i,j,k} - b_{i,j,k}\tilde{d}_{i,j-1,k} - c_{i,j,k}\tilde{d}_{i,j,k-1}$

The system  $M\Delta Q^{n+1} = -R^n$  is then solved by inverting the implicit operator,  $M$ . The subdomain implicit operator,  $M$ , is inverted with a forward, then a backward sweep in the  $i$  direction as

$$\text{Forward Sweep solving } (D + a)\Delta Q^{**} = -R^n \quad (38)$$

$$1. \text{ solve - } \Delta Q_i^{**} = D^{-1}(-R_i - a\Delta Q_{i-1})$$

$$\text{Backward Sweep solving } (D + h)\Delta Q^{***} = D\Delta Q^{**} \quad (39)$$

$$1. \text{ solve - } \Delta Q_i^{***} = \Delta Q_i^{**} - D^{-1}h\Delta Q_{i+1}^{***}$$

This approach alleviates much of the numerical stiffness associated with highly-stretched mesh cells, provided that the crossflow plane contains the coordinate direction(s) with the largest degree of mesh stretching. Techniques such as Jacobian freezing and implicit boundary conditions are used to reduce the overall CPU load, and further enhance stability. Jacobian freezing is where the Jacobian matrices are reevaluated and re-factored after a certain number of iterations. In the cases presented, the Jacobian matrices are typically calculated, factored, and stored every iteration for the first 500 iterations, and then once every 5 iterations afterwards.

An implicit algorithm has been developed to improve 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:

$$M(\Delta Q^{n+1,l+1} - \Delta Q^{n+1,l}) = -R^n - A\Delta Q^{n+1,l} \quad (40)$$

for  $\Delta Q^{n+1,l+1}$

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:

$$\text{Solve: } \Delta Q^{n+1,1} = -M^{-1}R^n \quad (41)$$

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

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

$$2: \text{Solve: } \Delta Q^{n+1,l+1} = \Delta Q^{n+1,l} + M^{-1}(-R^n - A\Delta Q^{n+1,l}) \quad (42)$$

End loop

$$\text{Update: } Q^{n+1} = Q^n + \Delta Q^{n+1,l_{\max}+1} \quad (43)$$

This algorithm requires that an extra block diagonal matrix, corresponding to the block diagonal of  $A$ , which is normally over-written by the incomplete LU 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. Of course, the addition of the subiteration will cause the time per iteration to increase, making it necessary to investigate the time to convergence for each case. The effect of the subiterations on wall-clock time will also be presented in the Results section.



## 4 Time-Derivative Preconditioning

### 4.1 Algorithm

The time-derivative preconditioning strategy currently implemented in VULCAN combines the rank-one preconditioning matrix of Weiss and Smith [5] with the “all-speed” version of the low diffusion flux-splitting scheme (LDFSS) of Edwards [10]. The “all-speed” version was developed according to a methodology presented in Edwards and Liou [11]. In order to complete the derivation of the Weiss and Smith preconditioning matrix begun in the introduction, it is necessary to make certain additions to the Navier-Stokes equation set. The first step in creating a more complete

system is to add a variant of the preconditioning term,  $\left( \frac{1}{V_{ref}^2} - \frac{1}{a^2} \right) \frac{\partial p}{\partial t}$ , to each

equation in the Navier-Stokes equation set. The additions are of the form  $\theta u_j \frac{\partial p}{\partial t}$  for

the momentum equations and  $\theta h \frac{\partial p}{\partial t}$  for the energy equation, where  $\theta$  is the scaling

parameter,  $\frac{1}{V_{ref}^2} - \frac{1}{a^2}$ . The next step is to take advantage of the fact that the pressure,

$p$ , is a function of the conserved variables,  $Q$ , and can be expanded by the chain rule

as

$$\frac{\partial p}{\partial t} = \frac{\partial p}{\partial q_1} \frac{\partial q_1}{\partial t} + \frac{\partial p}{\partial q_2} \frac{\partial q_2}{\partial t} + \frac{\partial p}{\partial q_3} \frac{\partial q_3}{\partial t} + \dots + \frac{\partial p}{\partial q_{N_{eq}}} \frac{\partial q_{N_{eq}}}{\partial t} \quad (44)$$

$N_{eq}$  and  $q_i$  in Equation 44 represent the total number of equations and conserved variable, respectively. Once the expansion has been done, Equation 44 can be factored into a more suitable form:

$$\frac{\partial p}{\partial t} = \bar{v}^T \frac{\partial Q}{\partial t} \quad (45)$$

where

$$\bar{v}^T = \left[ \frac{\partial p}{\partial q_1} \quad \frac{\partial p}{\partial q_2} \quad \frac{\partial p}{\partial q_3} \quad \dots \quad \frac{\partial p}{\partial q_{N_{eq}}} \right] \quad (46)$$

With this addition, the two time derivative terms in Equation 2 (with the addition of the momentum, energy, and turbulence equations) can now be factored into one term as

$$\theta \bar{u} \bar{v}^T \frac{\partial Q}{\partial t} + \frac{\partial Q}{\partial t} = \left( I + \theta \bar{u} \bar{v}^T \right) \frac{\partial Q}{\partial t} \quad (47)$$

where  $\bar{u}^T$ , given by

$$\bar{u}^T = \left[ Y_1 \quad \dots \quad Y_{N_{CS}} \quad 1 \quad u \quad v \quad w \quad H \quad k \quad \omega \right], \quad (48)$$

represents the variants added to the momentum, energy, and turbulence equations.

Finally, the preconditioned Navier-Stokes equation set with the additions from Equations 46-48 is given by

$$P \frac{\partial Q}{\partial t} + R(Q) = 0 \quad (49)$$

where the preconditioning matrix,  $P$ , is defined as

$$P = I + \theta \bar{u} \bar{v}^T \quad (50)$$

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

$$V_{ref}^2 = \min\left(a^2, \max\left(|\vec{V}|^2, KV_\infty^2\right)\right), \quad (51)$$

where  $a$  is the sound speed and  $|\vec{V}|^2$  is the velocity magnitude.  $V_\infty$  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  $V_\infty$  is set to the inputted free-stream velocity.

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

$$U, U, U \quad (52)$$

$$U' \pm a' = \frac{1}{2} \left[ (1 + M_{ref}^2)U \pm \sqrt{U^2(1 - M_{ref}^2)^2 + 4V_{ref}^2} \right] \quad (53)$$

where

$$M_{ref}^2 = \frac{V_{ref}^2}{a^2} \quad (54)$$

In the above equations,  $U$  is the contravariant velocity. As the incompressible limit is approached ( $V_{ref}^2 \rightarrow 0$ ), the eigenvalues become

$$U, U, U \quad (55)$$

$$\frac{1}{2} \left[ U \pm \sqrt{U^2 + 4V_{ref}^2} \right] \quad (56)$$

whereas the eigenvalues revert to their traditional values  $U, U, U$ , and  $U \pm a$  as  $V_{ref}^2 \rightarrow a^2$ .  $U$  in the above equations represents the contravariant velocity in the  $\xi$  direction.

## 4.2 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. In the VULCAN implementation, a preconditioned variant of the low diffusion flux-splitting scheme (LDFSS) of Edwards [10], developed according to a methodology presented in Edwards and Liou [11]. The interface flux  $E_I$  in LDFSS is split into convective and pressure contributions as follows:

$$E_I = a_{\frac{1}{2}} \left[ \rho_L C^+ E_L^C + \rho_R C^- E_R^C \right] + E^p \left[ D_L^+ p_L + D_R^- p_R \right] \quad (57)$$

The subscripts  $L$  and  $R$  denote the left and right states at the cell interface. The vector  $E^C$  is the same as  $\vec{u}$  in Equation 57, while

$$E^p = \begin{bmatrix} 0 & \dots & 0 & 0 & \bar{\xi}_x & \bar{\xi}_y & \bar{\xi}_z & 0 & 0 & 0 \end{bmatrix} \quad (58)$$

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

$$a_{\frac{1}{2}} = \frac{\left[ \sqrt{U^2 (1 - M_{ref}^2)^2 + 4V_{ref}^2} \right]_{1/2}}{(1 + M_{ref}^2)_{1/2}} \quad (59)$$

where the subscript  $\frac{1}{2}$  represents evaluation of the quantity using flowfield information arithmetically-averaged to the cell interface. The quantities  $D^+$ ,  $D^-$ ,  $C^+$ , and  $C^-$  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 [11]:

$$\tilde{M}_L = \frac{1}{2} \left[ (1 + M_{ref, \frac{1}{2}}^2) M_L + (1 - M_{ref, \frac{1}{2}}^2) M_R \right] \quad (60)$$

$$\tilde{M}_R = \frac{1}{2} \left[ (1 + M_{ref, \frac{1}{2}}^2) M_R + (1 - M_{ref, \frac{1}{2}}^2) M_L \right] \quad (61)$$

with

$$M_{L,R} = \frac{U_{L/R}}{a_{\frac{1}{2}}} \quad (62)$$

The quantities  $D^+$ ,  $D^-$ ,  $C^+$ , and  $C^-$  are then given by

$$D_{L,R}^\pm = \alpha_{L,R}^\pm (1.0 + \beta_{L,R}) - \beta_{L,R} P_{L,R}^\pm \quad (63)$$

$$C^+ = \alpha_L^+ (1 + \beta_L) M_L - \beta_L M_L^+ - M_{\frac{1}{2}}^+ \quad (64a)$$

$$C^- = \alpha_R^- (1 + \beta_R) M_R - \beta_R M_R^- + M_{\frac{1}{2}}^- \quad (64b)$$

The definitions for the subsonic pressure splitting and the other functions are as follows

$$P_{L,R}^\pm = \frac{1}{4} (M_{L,R} \pm 1.0)^2 (2.0 \mp M_{L,R}) \quad (65)$$

$$\beta_{L,R} = -\max[0, 1.0 - \text{int}(|M_{L,R}|)] \quad (66)$$

$$\alpha_{L,R}^{\pm} = \frac{1}{2} [1.0 \pm \text{sgn}(M_{L,R})] \quad (67)$$

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  $p_L - p_R$ ,

$$\begin{aligned} M_{\frac{1}{2}}^+ &= M_{\frac{1}{2}} \left( 1 - \frac{p_L - p_R}{p_L + p_R} - \delta \frac{|p_L - p_R|}{p_L} \right) \\ M_{\frac{1}{2}}^- &= M_{\frac{1}{2}} \left( 1 + \frac{p_L - p_R}{p_L + p_R} - \delta \frac{|p_L - p_R|}{p_R} \right) \end{aligned} \quad (68)$$

where

$$M_{\frac{1}{2}} = \frac{1}{4} \beta_L \beta_R \left( \sqrt{\frac{1}{2} (\tilde{M}_L^2 + \tilde{M}_R^2)} - 1.0 \right)^2 \quad (69)$$

As shown in [11], 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  $1/M_{ref, \frac{1}{2}}^2$ .

The primitive variables are extrapolated to the cell interfaces by means of the variable extrapolation scheme, MUSCL (Monotone Upstream-centered Schemes for Conservation Laws), of van Leer [12], given by

$$\begin{aligned} Q_{Ri+\frac{1}{2},j} &= Q_{i+1,j} - \frac{\Phi}{4} [(1+\kappa)(Q_{i+1,j} - Q_{i,j}) + (1-\kappa)(Q_{i+2,j} - Q_{i+1,j})] \\ Q_{Li+\frac{1}{2},j} &= Q_{i,j} + \frac{\Phi}{4} [(1-\kappa)(Q_{i,j} - Q_{i-1,j}) + (1+\kappa)(Q_{i+1,j} - Q_{i,j})] \end{aligned} \quad (70)$$

For values of  $\Phi$  equal to 1, the MUSCL scheme is a higher order scheme, whereas  $\Phi$  equal to zero results in a first order scheme. The scheme is second order accurate when  $\kappa$  is equal to  $\frac{1}{3}$  and becomes Fromm's second order scheme when  $\kappa$  equals 0. All results in this paper are second order accurate, using  $\kappa$  equal to  $\frac{1}{3}$ . In order to ensure that no oscillations were present in the solution, the smooth limiter of Venkatakrishnan [13] was employed. Venkatakrishnan's limiter is given by

$$\psi(r) = \frac{1}{2}(r+1) \min \left[ \frac{4r(3r+1)}{11r^2+4r+1}, \frac{4(r+3)}{r^2+4r+11} \right] \quad (71)$$

After the addition of this limiter, the resulting scheme is

$$\begin{aligned} Q_{Ri+\frac{1}{2},j} &= Q_{i+1,j} - \frac{\Phi}{4} \left[ (1-\kappa)\psi\left(r_{i+\frac{3}{2}}^{-}\right) + (1+\kappa)\psi\left(\frac{1}{r_{i+\frac{3}{2}}^{-}}\right)r_{i+\frac{3}{2}}^{-} \right] (u_{i+2} - u_{i+1}) \\ Q_{Li+\frac{1}{2},j} &= Q_{i,j} + \frac{\Phi}{4} \left[ (1-\kappa)\psi\left(r_{i-\frac{1}{2}}^{+}\right) + (1+\kappa)\psi\left(\frac{1}{r_{i-\frac{1}{2}}^{+}}\right)r_{i-\frac{1}{2}}^{+} \right] (u_i - u_{i-1}) \end{aligned} \quad (72)$$

where

$$r_{i-\frac{1}{2}}^{+} = \frac{u_{i+1} - u_i}{u_i - u_{i-1}} \quad r_{i+\frac{3}{2}}^{-} = \frac{u_{i+1} - u_i}{u_{i+2} - u_{i+1}} \quad (73)$$

### 4.3 Time-Stepping Scheme

The Euler implicit time integration of the preconditioned Navier-Stokes (Equation 49) is as follows [1]

$$\left[ \frac{P}{J\Delta t} + \delta_\xi E_Q + \delta_\eta F_Q + \delta_\zeta G_Q - S_Q \right] \Delta Q = R^n \quad (74)$$

In Equation 74,  $E_Q$  represents the Jacobian,  $\frac{\partial E}{\partial Q} - \frac{\partial E_v}{\partial Q}$  and  $S_Q$  is defined to be  $\frac{\partial S}{\partial Q}$ .

The operator  $\delta_\zeta E_Q \Delta Q$  is represented by

$$\frac{E_{Q_{i+\frac{1}{2}}} \Delta Q_{i+\frac{1}{2}} - E_{Q_{i-\frac{1}{2}}} \Delta Q_{i-\frac{1}{2}}}{\Delta \zeta} \quad (75)$$

where the interface states  $(i + \frac{1}{2})$  are defined by an upwind approach such as

$$E_{Q_{i+\frac{1}{2}}} \Delta Q_{i+\frac{1}{2}} \equiv E_{Q_{i+\frac{1}{2}}}^+ \Delta Q_i + E_{Q_{i+\frac{1}{2}}}^- \Delta Q_{i+1} \quad (76)$$

where

$$E_{Q_{i+\frac{1}{2}}}^\pm \equiv \frac{1}{2} \left( E_Q \pm P T_\xi \left| \lambda^p_{\xi,c} \right| T_\xi^{-1} \right)_{\frac{1}{2}} \quad (77)$$

In Equation 77, the modal matrices,  $T_\xi^p$  and  $(T_\xi^p)^{-1}$ , are constructed from diagonalization transformations of the form

$$T_\xi^p \lambda^p_{\xi,c} (T_\xi^p)^{-1} = P^{-1} \frac{\partial E}{\partial Q} \quad (78)$$

where  $\lambda^p_{\xi,c}$  is a diagonal matrix containing the eigenvalues of the preconditioned Euler system.  $R^n$  is the steady state residual from time level  $n$ . The Euler implicit formulation is obtained by forming a Taylor expansion around  $E$ ,  $F$  and  $G$ , defined as



$$E^{n+1} = E^n + \left( \frac{\partial E}{\partial t} \right)^n \Delta t + O(\Delta t^2) \quad (79)$$

where

$$\left( \frac{\partial E}{\partial t} \right)^{n+1} = \left( \frac{\partial E}{\partial Q} \right)^n \frac{\partial Q}{\partial t} = \left( \frac{\partial E}{\partial Q} \right)^n \frac{\Delta Q}{\Delta t} \quad (80)$$

and after combining equations

$$E^{n+1} = E^n + \left( \frac{\partial E}{\partial Q} \right)^n \Delta Q + O(\Delta t^2) \quad (81)$$

After substituting Equation 81 into Equation 74 the result is

$$P \frac{\Delta Q}{J \Delta t} + \delta_\xi (E^n + E_Q \Delta Q) + \delta_\eta (F^n + F_Q \Delta Q) + \delta_\zeta (G^n + G_Q \Delta Q) - (S^n + S_Q \Delta Q) = 0 \quad (82)$$

After rearranging and defining the steady state residual as

$$R^n = -(\partial_\xi E^n + \partial_\eta F^n + \partial_\zeta G^n - S^n) \quad (83)$$

the final equation becomes

$$P \frac{\Delta Q}{J \Delta t} + \delta_\xi (E_Q \Delta Q) + \delta_\eta (F_Q \Delta Q) + \delta_\zeta (G_Q \Delta Q) - (S_Q \Delta Q) = R^n \quad (84)$$

Due to storage requirements for solving Equation 84 directly, an approximate factorization scheme must be instituted as an alternative. The factorization scheme used to advance the solution in time is the preconditioned, diagonalized approximate factorization (DAF) scheme. The preconditioned version of the DAF scheme is written as follows:

$$\begin{aligned}
& [I - J\Delta t S_Q] T_\xi^p [I + J\Delta t \delta_\xi (\lambda_{\xi,c}^p - \lambda_{\xi,v})] (T_\xi^p)^{-1} T_\eta^p [I + J\Delta t \delta_\eta (\lambda_{\eta,c}^p - \lambda_{\eta,v})] (T_\eta^p)^{-1} T_\zeta^p \\
& [I + J\Delta t \delta_\zeta (\lambda_{\zeta,c}^p - \lambda_{\zeta,v})] (T_\zeta^p)^{-1} \Delta Q_{i,j,k} = J\Delta t P^{-1} R^n
\end{aligned} \tag{85}$$

and solved sequentially as

$$\begin{aligned}
& [I - J\Delta t S_Q] \Delta Q_{i,j,k}^* = J\Delta t P^{-1} R^n \\
& T_\xi^p [I + J\Delta t \delta_\xi (\lambda_{\xi,c}^p - \lambda_{\xi,v})] (T_\xi^p)^{-1} \Delta Q_{i,j,k}^{**} = \Delta Q_{i,j,k}^* \\
& T_\eta^p [I + J\Delta t \delta_\eta (\lambda_{\eta,c}^p - \lambda_{\eta,v})] (T_\eta^p)^{-1} \Delta Q_{i,j,k}^{***} = \Delta Q_{i,j,k}^{**} \\
& T_\zeta^p [I + J\Delta t \delta_\zeta (\lambda_{\zeta,c}^p - \lambda_{\zeta,v})] (T_\zeta^p)^{-1} \Delta Q_{i,j,k}^{****} = \Delta Q_{i,j,k}^{***}
\end{aligned} \tag{86}$$

$$Q_{i,j,k}^{n+1} = Q_{i,j,k}^n + \Delta Q_{i,j,k}$$

Again,  $T_\xi^p$  and  $(T_\xi^p)^{-1}$  are the modal matrices. An example of the modal matrices in one dimension for the preconditioned Euler system can be found in Appendix A. The diagonalizing procedure presented above relieves much of the expense involved with solving the system. The scheme reduces to solving a system of N scalar tridiagonals. Note that the first step of the DAF procedure is an approximation of the more exact expression

$$[P - J\Delta t S_Q] \Delta Q_{i,j,k}^* = J\Delta t R^n \tag{87}$$

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 Sherman-Morrison theorem deals with matrices that contain rank-one perturbations such as the preconditioning matrix given in Equation 50. The Sherman-Morrison theorem is given as

$$(A + uv^T)^{-1} = A^{-1} - \frac{A^{-1}uv^T A^{-1}}{1 + v^T A^{-1}u} \quad (88)$$

This theorem applied to the preconditioning matrix above is given by

$$P^{-1} = (I + \theta uv^T)^{-1} = I - \frac{uv^T}{1 + v^T u} \quad (89)$$

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.

## 5 Results

### 5.1 Planar Relaxation Results

The testing of the planar relaxation algorithm with subdomain coupling has been focused on the West-Korkegi intersecting-wedge geometry [14] 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. Figures 2 and 3 portray the grid and dimensions used for the West-Korkegi intersecting-wedge geometry and the channel-flow grid. The free-stream conditions for the intersecting-wedge simulations are:  $M_\infty = 3$ ,  $Re/m = 2.11e6$ ,  $T_\infty = 105$  K. The free-stream conditions for the channel-flow simulations are:  $M_\infty = 0.5$ ,  $Re/m = 3.52e5$ ,  $T_\infty = 105$  K. In both cases, the grid size is  $65 \times 125 \times 125$  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 4 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 than the baseline DAF scheme. This translates into 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 5 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 6 shows that there is little advantage to choosing a CFL higher than 150 for this case. Figure 6 shows the effect of the choice of sweep direction on the performance of the iteratively improved planar relaxation algorithm with  $l_{max} = 0$  and  $l_{max} = 1$ . As anticipated, 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 7 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 8 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  $l_{max} = 2$ . Although the number of iterations decreases by almost half, the time spent in the subiteration loop erases some of the benefits from the scheme.

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 9 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 second-order 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.

## 5.2 Preconditioning Results

The five test cases for the implementation of time-derivative preconditioning are the inviscid flow over a bump in a channel, flow over a flat-plate, flow through a two-dimensional United Technologies Research Center (UTRC) nozzle, flow between intersecting wedges, and finally, the flow through the channel analogue of the intersecting wedges. 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 the diagonalized approximate factorization scheme is used. Most cases involve both turbulent and laminar flow as well as multi-component gases. In all turbulent cases, the Wilcox (1998)  $k - \omega$  model is used. For all cases involving multiple gas species, a mixture of nitrogen and oxygen is used.

The best measure of the stiffness of an equation system is the condition number,  $C$ , defined as the ratio of the largest eigenvalue of the system to the smallest eigenvalue:

$$C = \frac{\lambda_{\max}}{\lambda_{\min}} \quad (90)$$

Figure 10 shows a direct correlation of the effect of the preconditioning of [11] on the condition number of the Euler equations. The disparity of the eigenvalues of the non-preconditioned system is readily apparent, while the preconditioned system demonstrates the desired correction to the Euler equations and eigenvalues. It is interesting to note that at approximately Mach 0.4, the eigenvalues of the non-preconditioned system appears to be slightly better conditioned than the preconditioned system.

### 5.2.1 Inviscid flow over a bump in a channel

The first example of the validation of the preconditioning strategy is the inviscid flow over a bump in a channel. For this particular case, the Euler equations are solved, thus Reynolds-number effects are not present. Figure 11 portrays the grid used for the calculations, while Figure 12 shows the convergence histories for Mach numbers of 0.5, 0.05, and 0.005. The reference conditions used were the same ones as in Table 1, along with the ramping done to obtain the inputs for the Mach 0.05 and Mach 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.



### 5.2.2 Two-dimensional flow over flat plate

Figure 13 shows the 65x129 grid for the flat-plate simulation. The first run was used 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 given in Table 1. 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 14 presents convergence histories for preconditioned and non-preconditioned laminar flow 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 verifies that the solution is in fact incorrect. The Blasius solution is calculated 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:

$$\delta(x) = \frac{5.0x}{\sqrt{\text{Re}_x}} \quad (91)$$

Table 2 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 obtained without preconditioning.

Figures 15 and 16 show the results for a turbulent, calorically perfect and a turbulent, two-species air flow over a flat plate, respectively. The convergence histories for each case are very similar to that of the laminar flow in Figure 14. In all three examples, the preconditioned Mach 0.5 case shows a marked improvement in convergence over the non-preconditioned result. 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 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.

As proof of this conjecture, a few cases were run holding the Reynolds number constant. The cases were all run at a Reynolds number of  $1.11 \times 10^7$ , the same as the Mach 0.5 case from the previous results. The Mach number was again ramped down from Mach 0.5 down to 0.005 while increasing the pressure by a factor of 10 each time in order to maintain the correct Reynolds number. Figure 17, for laminar flow, shows that the convergence rates of all the non-preconditioned cases are much slower than their preconditioned counterparts. Unlike the previous laminar cases, the Reynolds number effects have been removed, and the fact that the preconditioner scales all inviscid eigenvalues to the order of the local velocity shows its true effect on the

convergence behavior. For the lower Mach number cases, the residual has almost converged to four orders of magnitude within 5000 iterations. Over 5000 iterations, the non-preconditioned residual only decreases a maximum of 2.5 orders of magnitude.

As can be seen by Figure 10, the condition number asymptotically goes down to 2.6 as the Mach number is lowered. Thus at lower Mach numbers it is expected that the convergence histories should be independent of Mach number. This is shown in Figure 18 for the turbulent flow of a calorically perfect gas. The convergence history for the Mach 0.05 and Mach 0.005 are nearly identical. Also, like the laminar case for the same Reynolds number, all three preconditioned cases converge much faster than their non-preconditioned counterparts. The convergence rate for the non-preconditioned system slows down as the Mach number is lowered.

Finally, the strength of preconditioning can be seen most readily in Figure 19. This figure represents the flow for a turbulent, two-species air flow over a flat-plate at a constant Reynolds number. Here, there is an even larger disparity in the convergence rates of the preconditioned system versus those of the non-preconditioned system for the lower Mach numbers ( $\text{Mach} < 0.1$ ). The difference is from two to three orders of magnitude in residual drop for the lower Mach numbers. Also as can be seen in Figure 19, the non-preconditioned residuals tend to flat line and no longer show a tendency to converge further. In contrast, the preconditioned residual maintains its downward trend.

### 5.2.3 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 20 shows the grid used in the calculation, while Figures 21 and 22 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 Equation 51 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 23 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  $K \sim 0.1$ . 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 flow fields 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.

#### 5.2.4 Three-dimensional flow through intersecting wedges

Figures 24-27 present results from simulations of subsonic viscous flow through the West-Korkegi intersecting wedge geometry (shown in Figure 2). Free-stream conditions are chosen to be the same as in the two-dimensional flat-plate case. Figure 24 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 convergence 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 25 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 26, 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).

Due to the slow convergence histories, it is necessary to prove that this degradation in convergence rate is due to Reynolds number effects. A laminar case was run for the intersecting wedge geometry, holding the Reynolds number constant at

$1.11 \times 10^7$ . As expected, the preconditioned residual histories, shown in Figure 27, reveal a drastic improvement over the ones in Figure 24. The largest improvement can be seen when comparing the Mach 0.005 cases. In Figure 27, the residual has almost reached convergence at four orders of magnitude reduction; whereas in Figure 24 the residual has started leveling off around a 2.5 order of reduction in magnitude. The most striking feature in Figure 27 is the fact that the convergence history has become nearly independent of Mach number. As is readily apparent, the convergence histories for Mach 0.05 and Mach 0.005 are almost identical. This is expected due the condition number being virtually the same for both Mach numbers as seen in Figure 10.

#### **5.2.5 Three-dimensional flow through a channel**

The final case tested was the channel analogue of the West-Korkegi intersecting wedge geometry. Again, Figure 3 displays the computational grid used for the ramped Mach number simulations. Figure 28 shows the convergence histories for the ramped Mach number flows of a laminar flow through the channel. As previously observed, the preconditioned Mach 0.5 case converges faster than the non-preconditioned case. A notable difference is that there are more oscillations visible in the non-preconditioned residual histories that are not evidenced in the flat-plate solution.

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 Equation 91. Table 3 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.

Figures 29 and 30 reveal how the convergence histories are affected by turbulent, calorically perfect and turbulent, thermally perfect two species air flow through the channel, respectively. Once more, the oscillations that were not present in the flat plate solution are very noticeable. Although these oscillations are present, both residual histories maintain their downward trend towards convergence.

Finally, an important property of the preconditioned DAF scheme is verified in Figure 31. Mach 3, turbulent flow was simulated through the channel in order to prove that the preconditioned system will revert back to the non-preconditioned system as the Mach number approaches and becomes greater than one. It is apparent from this figure that the two schemes are in fact identical for the turbulent, supersonic flow through the channel, thus demonstrating the validity of the preconditioned diagonalized approximate factorization scheme for all flow speeds.

## 6 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 gases 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.



## 7 Future Work

Future work will include the validation of the preconditioning model for low-speed reacting flows as well as implementing a change in the definition of the total enthalpy. Originally, the total enthalpy was defined as Equation 29, repeated below for convenience

$$H = h + \frac{1}{2}(u^2 + v^2 + w^2) \quad (29)$$

This definition of the total enthalpy does not take into effect the turbulent kinetic energy parameter,  $k$ . Once this parameter has been added, the new total enthalpy becomes:

$$H = h + \frac{1}{2}(u^2 + v^2 + w^2) + k \quad (92)$$

With this addition, a few terms are added to the modal matrices in Appendix A. The  $\sigma$  in Appendix A is representative of the modal matrix with and without the addition of the turbulent kinetic energy. Using  $\sigma$  equal to one represents the influence of the turbulent kinetic energy on the modal matrices. By using  $\sigma$  equal to zero, the matrix reverts back to its previous state.

The issue with low Reynolds number effects 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.

## 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] A. J. Chorin, "A Numerical Method for Solving Incompressible Viscous Flow Problems," *J. of Computational Physics*, Vol. 2, 1967, pp. 12-26.
- [3] Choi, Y.H., and Merkle, C.L., "The Application of Preconditioning in Viscous Flows," *J. Computational Physics*, Vol. 105, 1993, pp. 207-233.
- [4] Turkel, E., "Preconditioned Methods for Solving the Incompressible and Low Speed Compressible Equations," *J. Computational Physics*, Vol. 72, 1987, pp. 277-298.
- [5] Weiss, J.M. and Smith, W.A., "Preconditioning Applied to Variable and Constant Density Flows," *AIAA Journal*, Vol. 33, 1995, p. 2050.
- [6] Wilcox, D. C., "Turbulence Modeling for CFD," DCW Industries, (2000).
- [7] Anderson, D. A., J.C. Tannehill and R.H. Pletcher, "Computational fluid mechanics and heat transfer," Taylor and Francis, (1997).
- [8] McBride, B.J., HeimeL, S., Ehlers, J.G. and Gordon, S., "Thermodynamic Properties to 6000 K for 210 Substances Involving the first 18 Elements," NASA SP-3001, 1963.
- [9] McBride, B.J., Gordon, S. and Reno, M.A., "Thermodynamic Data for Fifty Reference Elements," NASA TP-3287, 1993.
- [10] Edwards, J.R. "A Low-Diffusion Flux-Splitting Scheme for Navier-Stokes Calculations," *Computers & Fluids*, Vol. 26, No. 6, 1997, pp. 635-659.
- [11] 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.
- [12] van Leer, B., "Towards the Ultimate Conservative Difference Scheme V. A Second Order Sequel to Gudunov's Method," *J. of Computational Physics*, Vol. 32, pp. 101-136, 1979.
- [13] Venkatakrishnan, V., "On the Accuracy of Limiters and Convergence to Steady State Solutions," AIAA 93-0880, 1993.

[14] 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.

## Appendix

### A Modal Matrices

The modal matrices of the preconditioned Euler system are constructed from diagonalization transformations of the form

$$T^p \lambda^p c (T^p)^{-1} = P^{-1} \frac{\partial E}{\partial Q}$$

Based on the one dimensional equation set for a turbulent, two component gas, T and T<sup>-1</sup> are given in Cartesian coordinates as

$$T = \begin{bmatrix} \rho & 0 & Y_1 & Y_1 A & Y_1 B & 0 & 0 \\ 0 & \rho & Y_2 & Y_2 A & Y_2 B & 0 & 0 \\ 0 & 0 & 1 & A & B & 0 & 0 \\ 0 & 0 & u & uA + C\rho & uB + \rho D & 0 & 0 \\ 0 & 0 & \frac{u^2}{2} + k\sigma & GA + \rho uC + \frac{E}{\gamma - 1} & GB + \rho uD + \frac{F}{\gamma - 1} & \rho\sigma & 0 \\ 0 & 0 & k & kA & kB & \rho & 0 \\ 0 & 0 & \omega & \omega A & \omega B & 0 & \rho \end{bmatrix}$$

where

$$A = \frac{(u' + a') - u}{2a'a} \quad B = \frac{u - (u' - a')}{2a'a^2} \quad C = \frac{(u' + a') - u - \frac{1}{2a'}[(u' + a') - u]^2}{\rho a'^2 M_R^2}$$

$$D = \frac{[u - (u' + a')][u - (u' - a')]}{\rho a' a^2 M_R^2} \quad E = \frac{(u' + a') - u}{2a'} \quad F = \frac{u - (u' - a')}{2a'} \quad G = \left( \frac{u^2}{2} + k\sigma \right)$$

$$T^{-1} = \begin{bmatrix} \frac{1}{\rho} & 0 & \frac{-Y_1}{\rho} & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{\rho} & \frac{-Y_2}{\rho} & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 - \frac{M^2(\gamma-1)}{2} & \frac{(\gamma-1)u}{a^2} & -\frac{\gamma-1}{a^2} & \frac{(\gamma-1)\sigma}{a^2} & 0 \\ 0 & 0 & Hu^2 - Lu & L - (\gamma-1)u & \gamma-1 & -(\gamma-1)\sigma & 0 \\ 0 & 0 & Hu^2 - Mu & M - (\gamma-1)u & \gamma-1 & -(\gamma-1)\sigma & 0 \\ 0 & 0 & -\frac{k}{\rho} & 0 & 0 & \frac{1}{\rho} & 0 \\ 0 & 0 & -\frac{\omega}{\rho} & 0 & 0 & 0 & \frac{1}{\rho} \end{bmatrix}$$

where

$$u' \pm a' = \frac{1}{2} \left[ \left( 1 + M_{ref}^2 \right) u \pm \sqrt{u^2 \left( 1 - M_{ref}^2 \right)^2 + 4V_{ref}^2} \right] \quad \text{and} \quad M_{ref}^2 = \frac{V_{ref}^2}{a^2}$$

$$H = \frac{\gamma-1}{2} \quad L = \frac{a^2 M_{ref}^2}{(u' + a') - u} \quad N = \frac{a^2 M_{ref}^2}{(u' - a') - u}$$

The constant,  $\sigma$ , represents the effect of  $k$ , the turbulent kinetic energy parameter on the modal matrices. If  $\sigma$  equals 1, the matrix includes the effect of  $k$ , whereas if  $\sigma$  equals 0, the effect is taken out of the matrix.

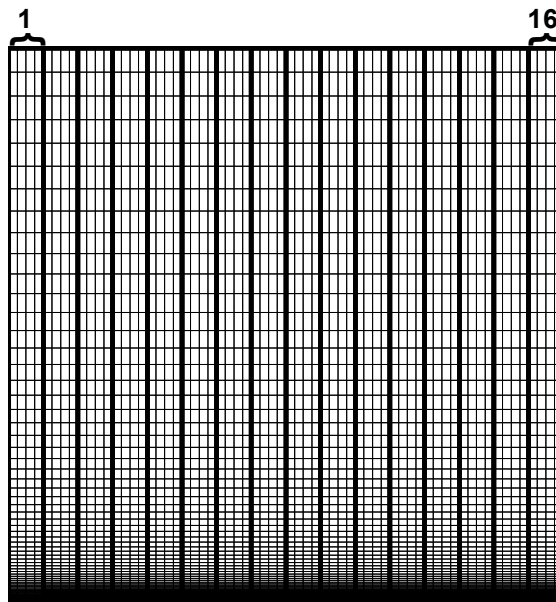


Figure 1: Example of decomposition of flat-plate

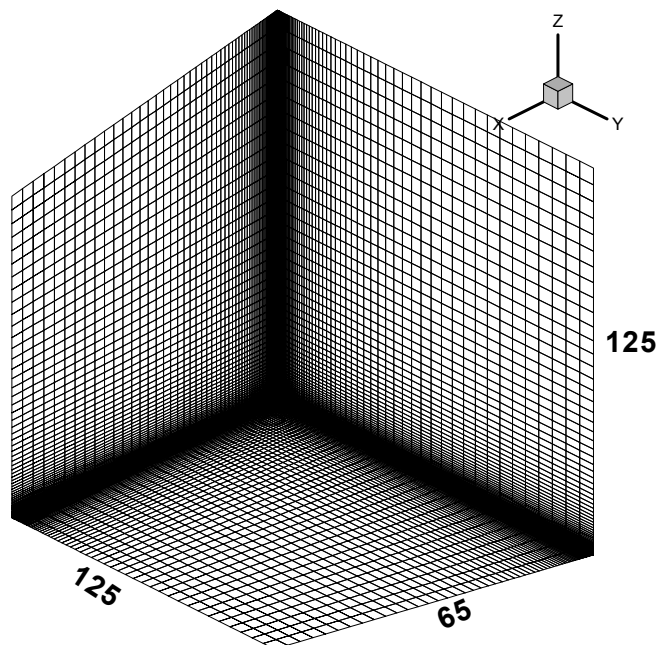


Figure 2: Computational grid for West-Korkegi intersecting-wedges

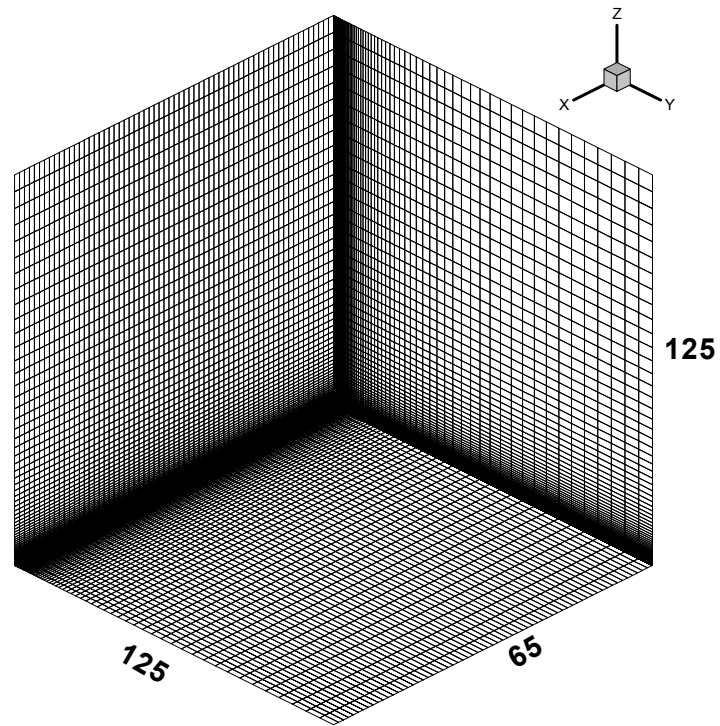


Figure 3: Computational grid for channel-flow

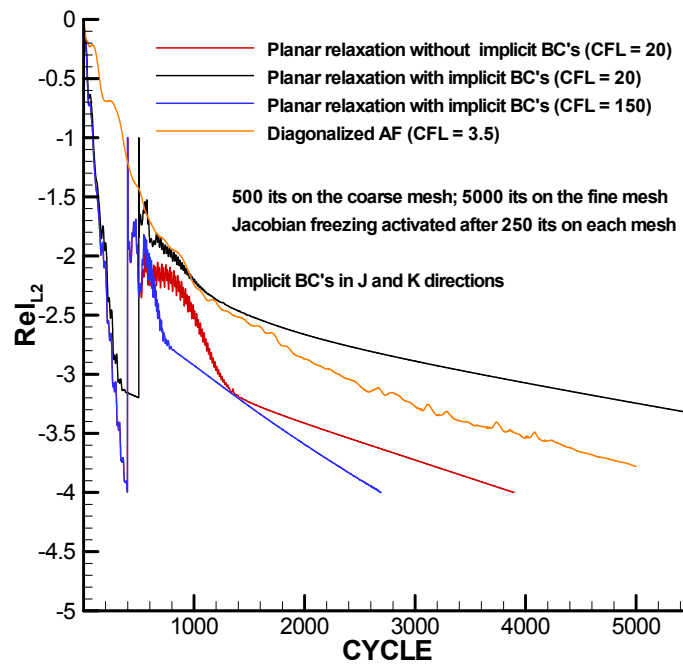


Figure 4: Convergence of planar relaxation scheme

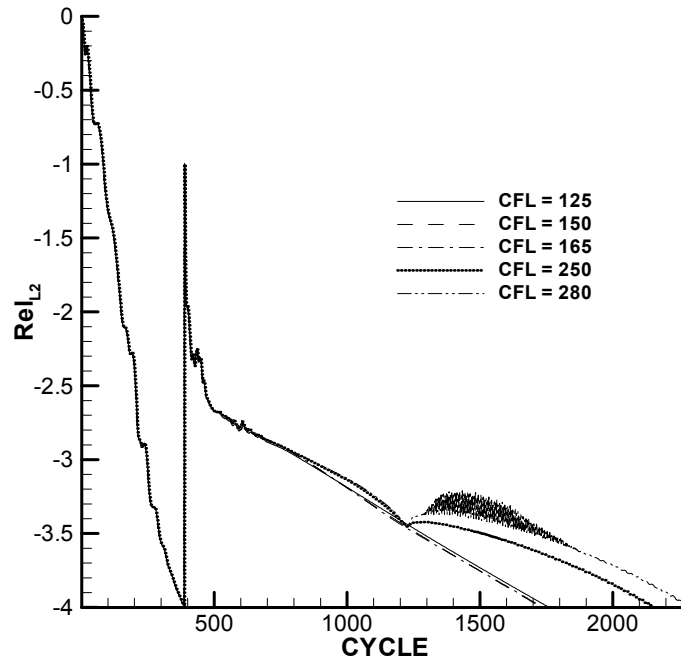


Figure 5: Effect of CFL number on convergence

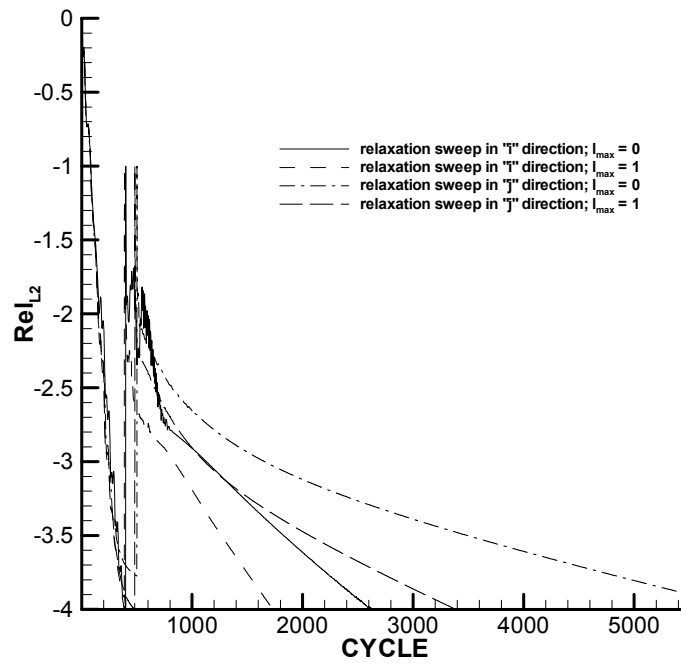


Figure 6: Effect of sweep direction on convergence



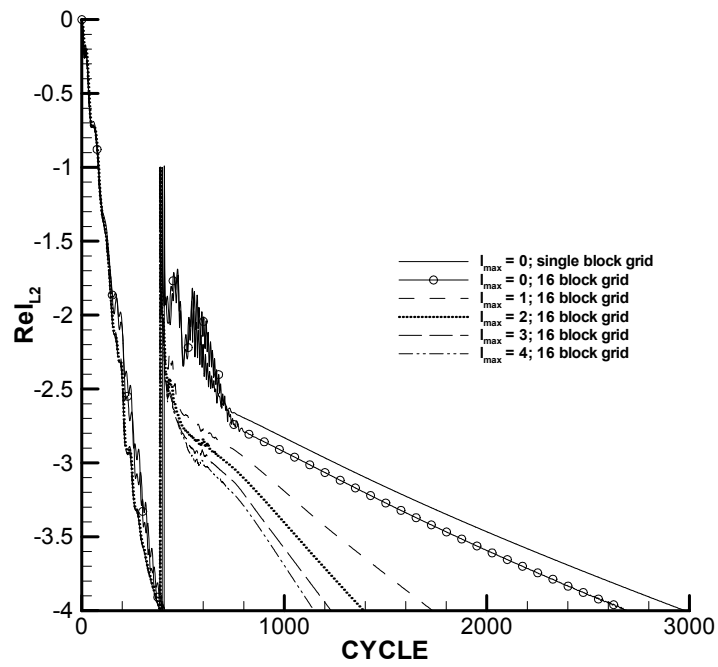


Figure 7: Effect of subiteration number on convergence

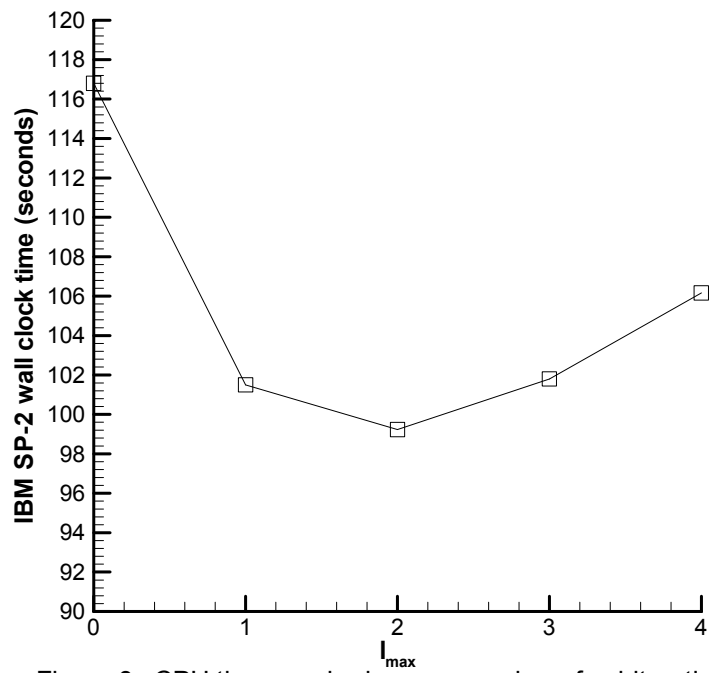


Figure 8: CPU time required versus number of subiterations

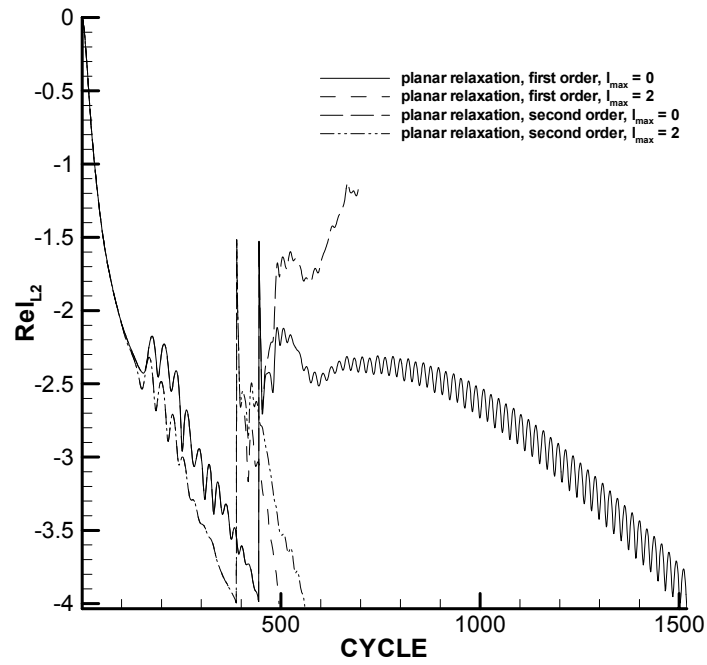


Figure 9: Effect of subiterations on subsonic channel-flow convergence

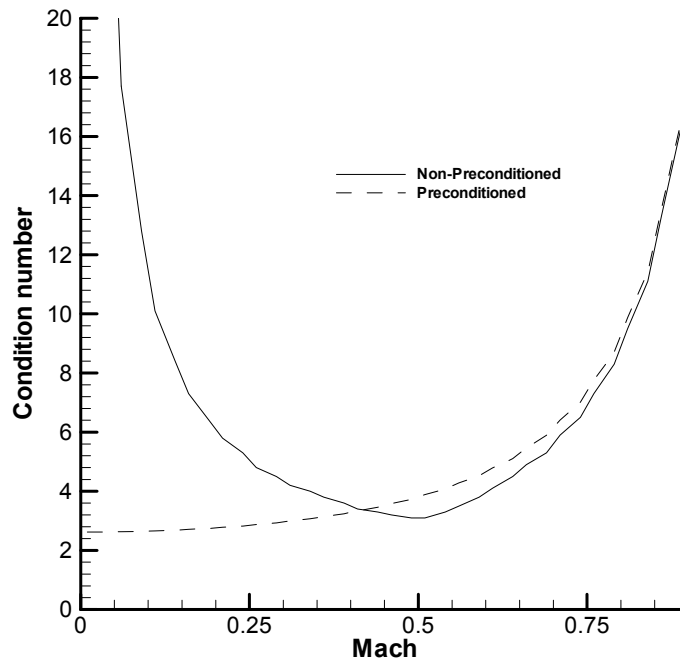
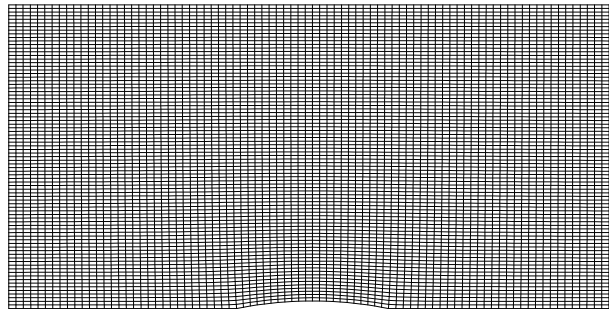


Figure 10: Condition number versus Mach number



85

85

Figure 11: Computational grid for flow over a bump

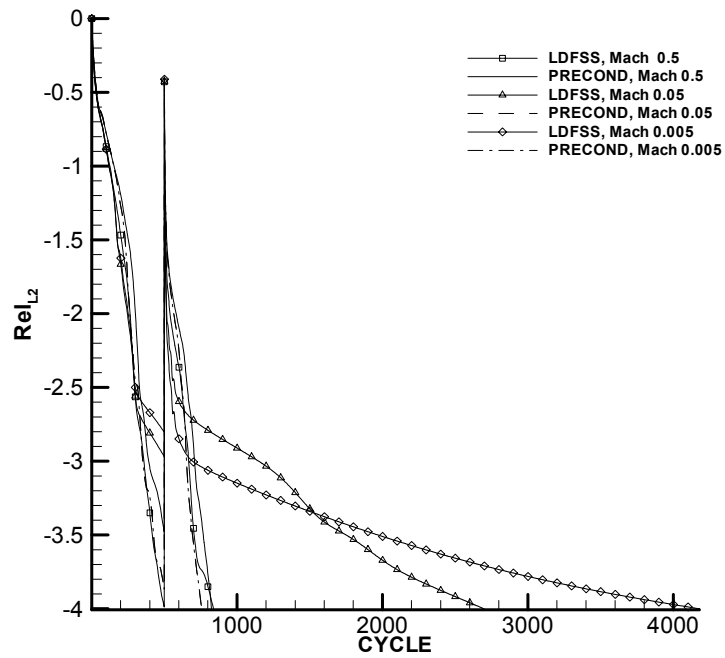
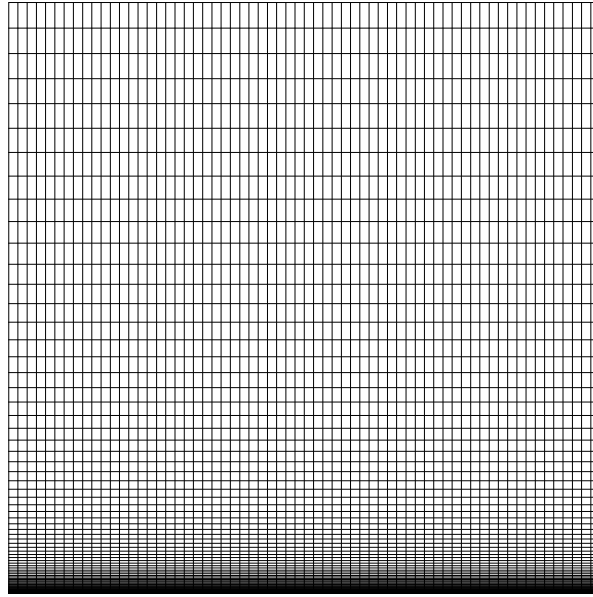


Figure 12: Convergence history for various Mach numbers for flow over a bump



125

65

Figure 13: Computational grid for flow over a flat-plate

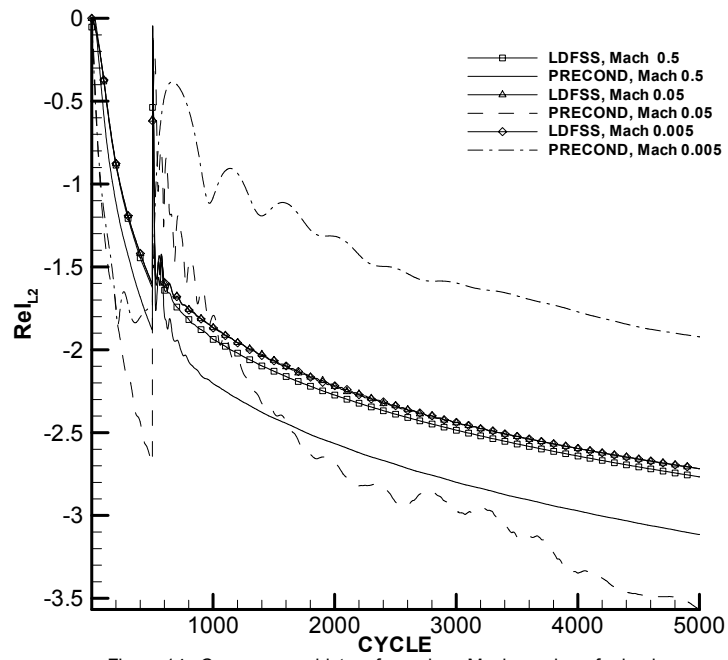


Figure 14: Convergence history for various Mach numbers for laminar flow over a flat-plate

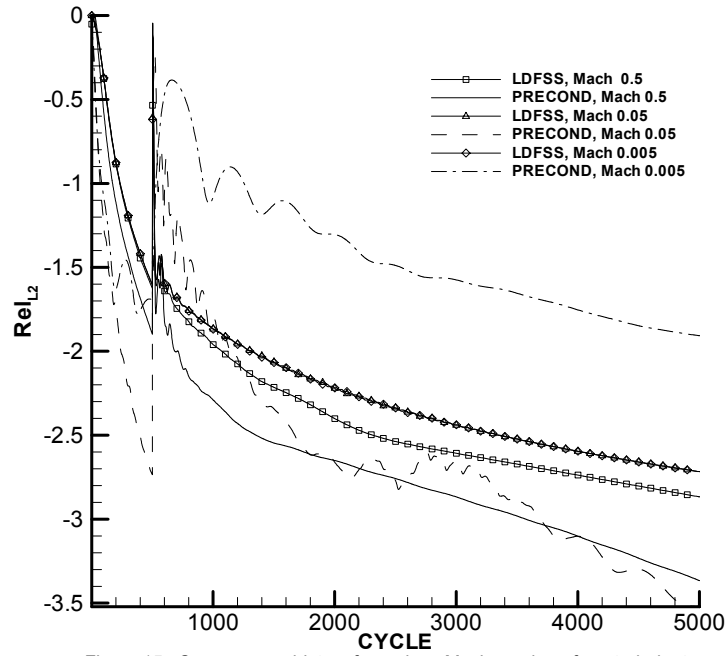


Figure 15: Convergence history for various Mach numbers for a turbulent, calorically perfect flow over a flat-plate

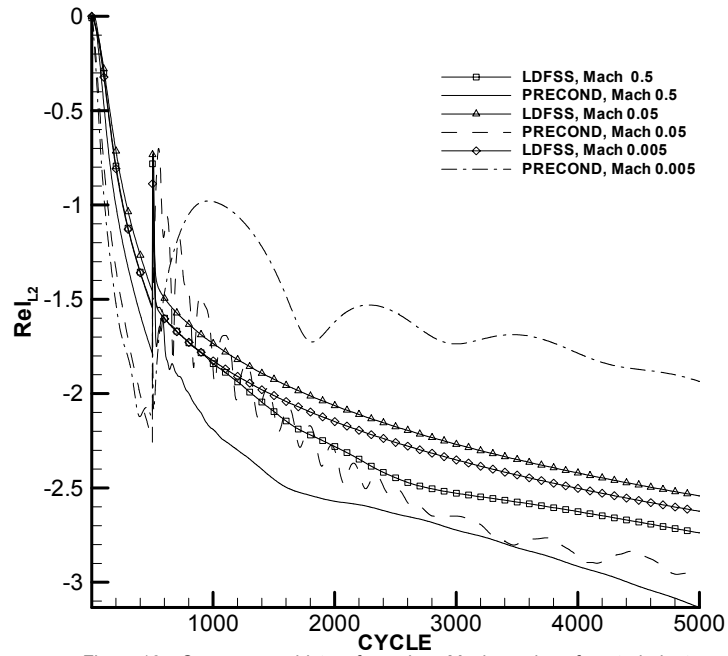


Figure 16 : Convergence history for various Mach numbers for a turbulent, two-species air flow over a flat-plate

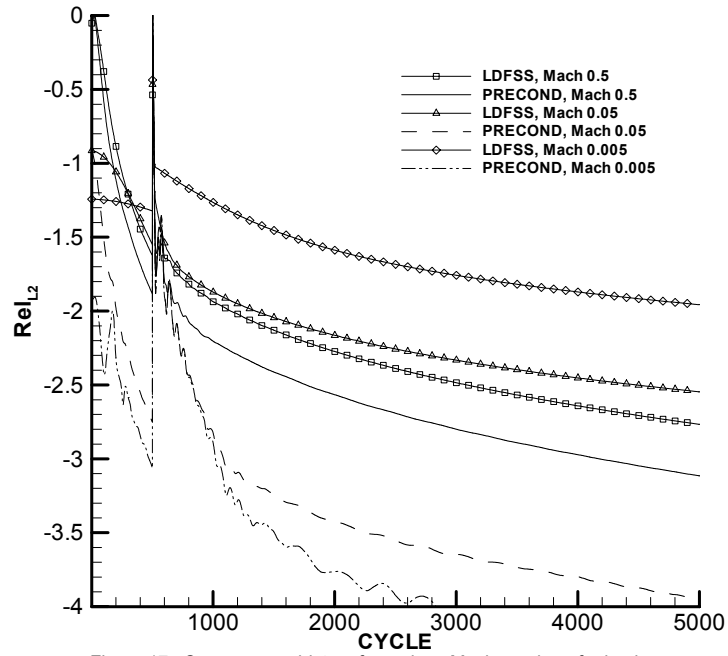


Figure 17: Convergence history for various Mach numbers for laminar flow over a flat-plate at a Reynolds number of  $1.11 \times 10^7$ .

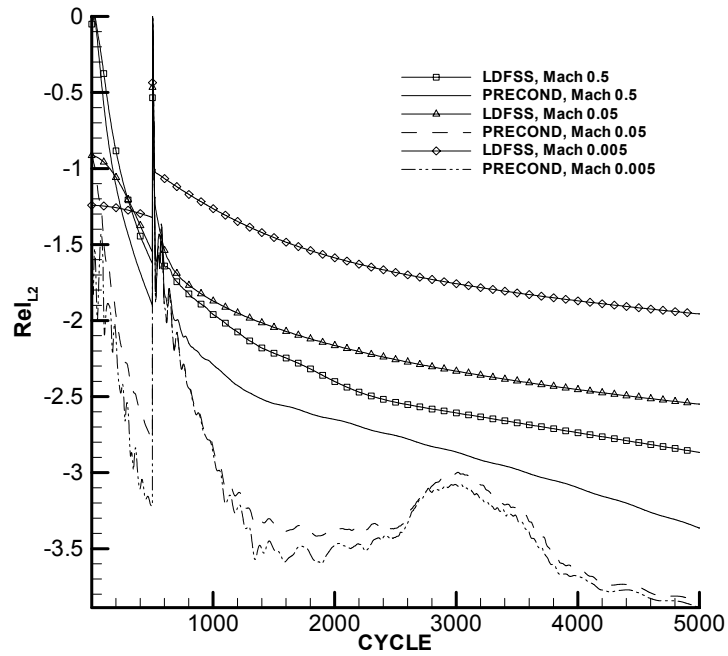


Figure 18: Convergence history for various Mach numbers for a turbulent, calorically perfect flow over a flat-plate at a Reynolds number of  $1.11 \times 10^7$ .

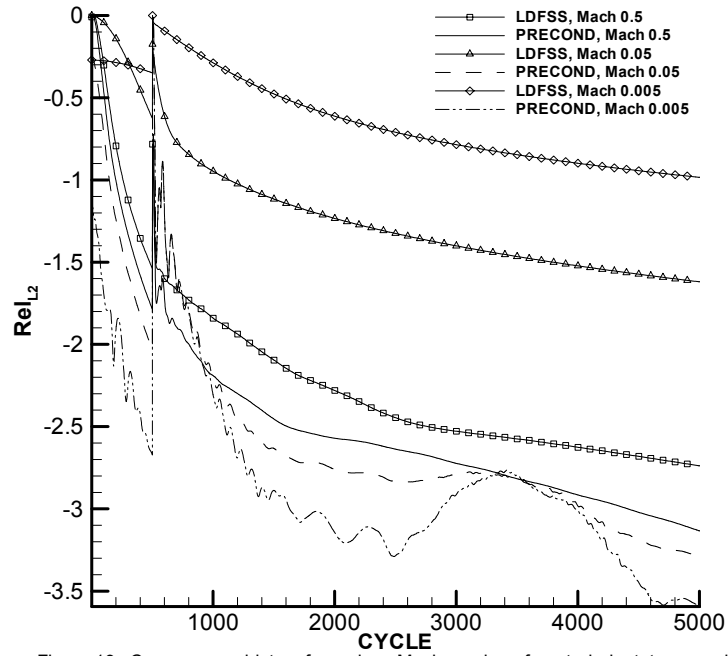


Figure 19: Convergence history for various Mach numbers for a turbulent, two-species air flow over a flat-plate at a Reynolds number of  $1.11 \times 10^7$

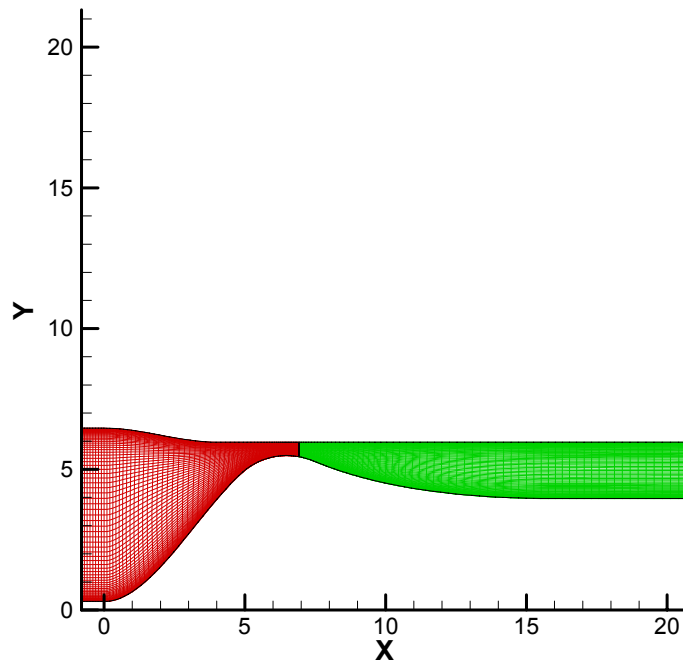


Figure 20: Computational grid for UTRC nozzle flow

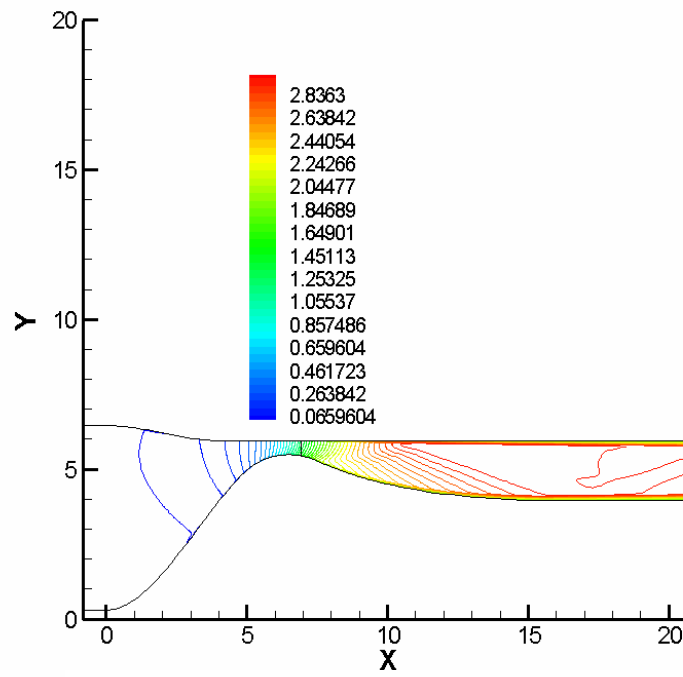


Figure 21: Mach number contours for UTRC nozzle flow

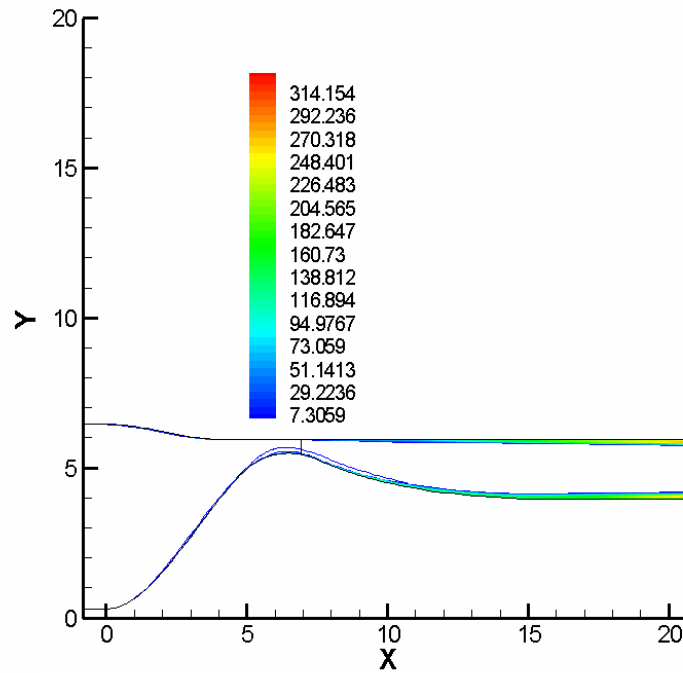


Figure 22: Normalized eddy viscosity for UTRC nozzle flow



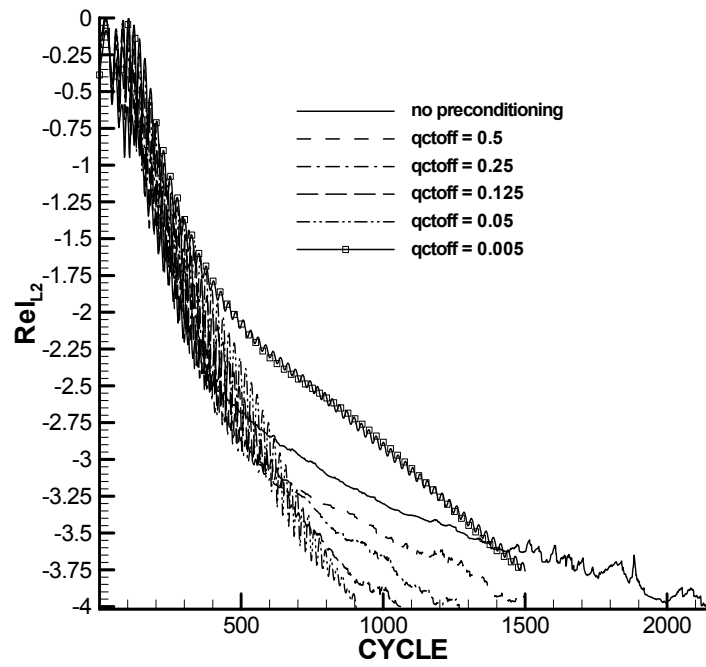


Figure 23: Convergence histories: elliptic part of UTRC nozzle calculation

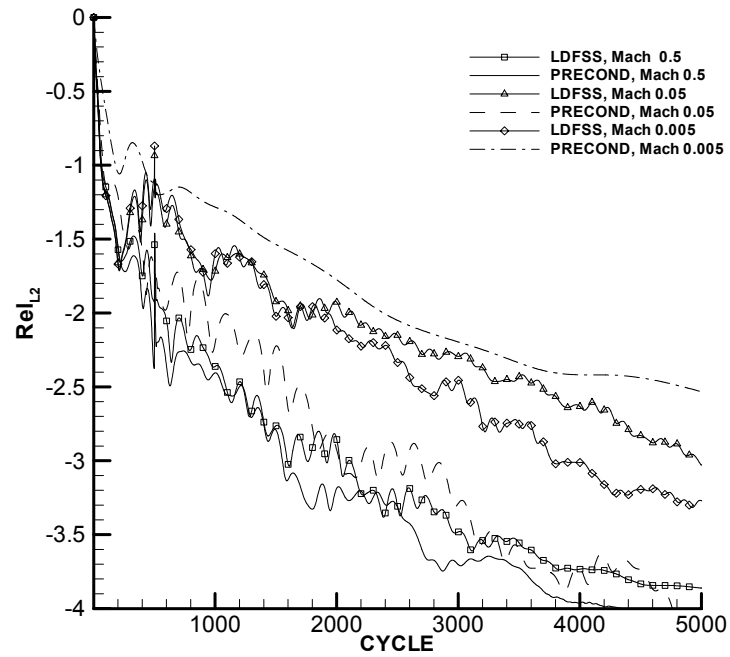


Figure 24: Convergence history for various Mach numbers for laminar flow through West-Korkegi intersecting-wedges

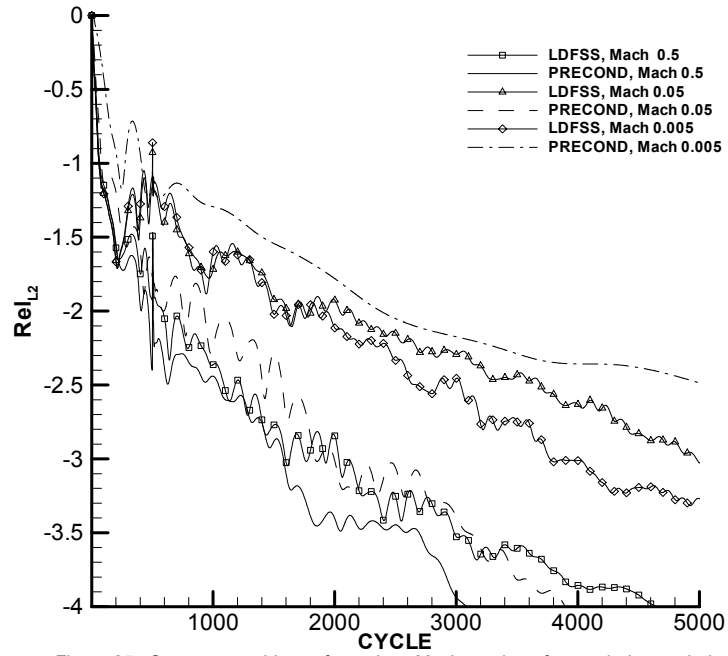


Figure 25: Convergence history for various Mach numbers for a turbulent, calorically perfect flow through the West-Korkegi intersecting-wedges

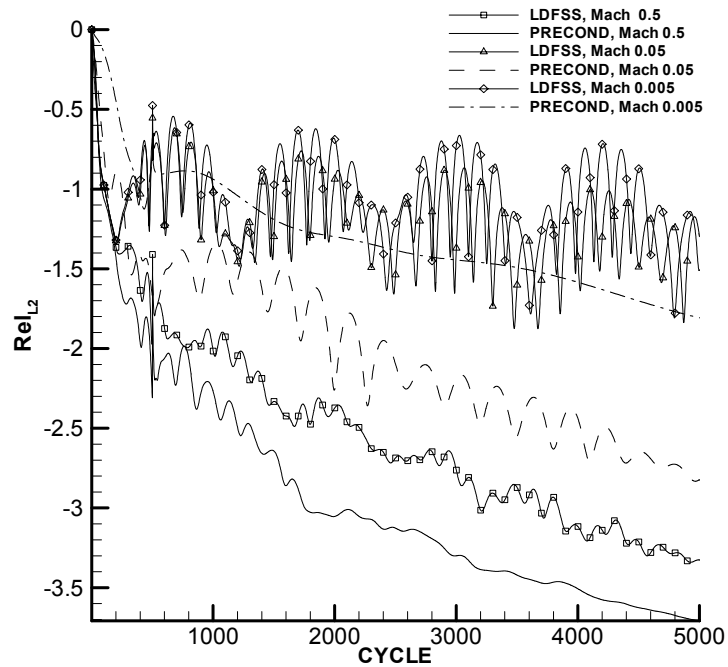


Figure 26: Convergence history for various Mach numbers for a turbulent, two-species air flow through the West-Korkegi intersecting-wedges

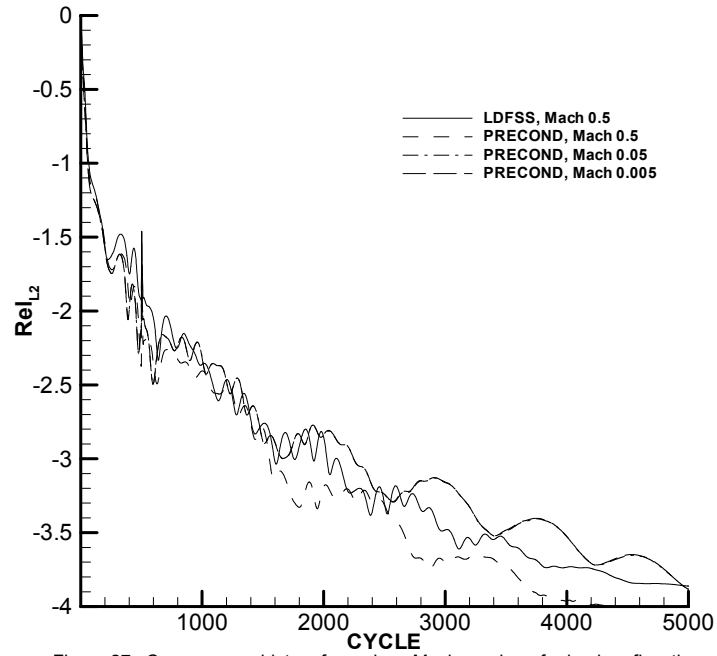


Figure 27: Convergence history for various Mach numbers for laminar flow through West-Korkegi intersecting wedges at a Reynolds number of  $1.11 \times 10^7$

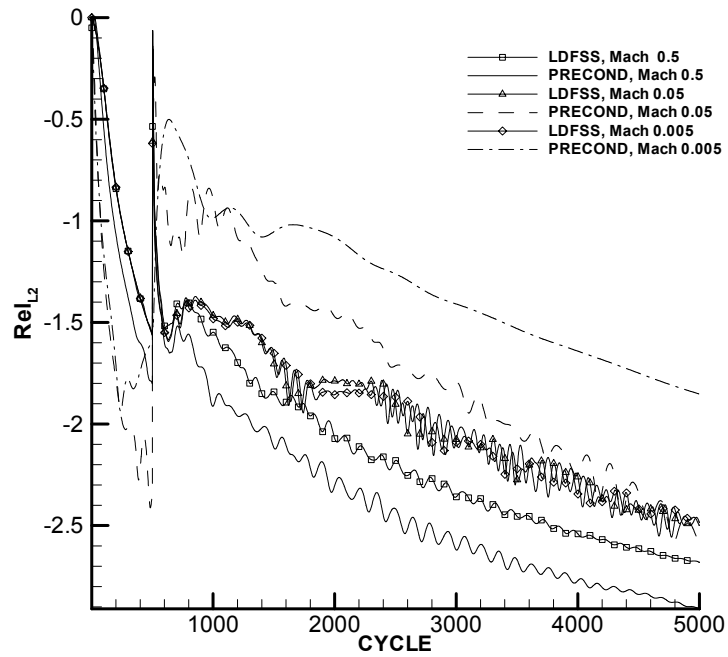


Figure 28: Convergence history for various Mach numbers for laminar flow through a channel

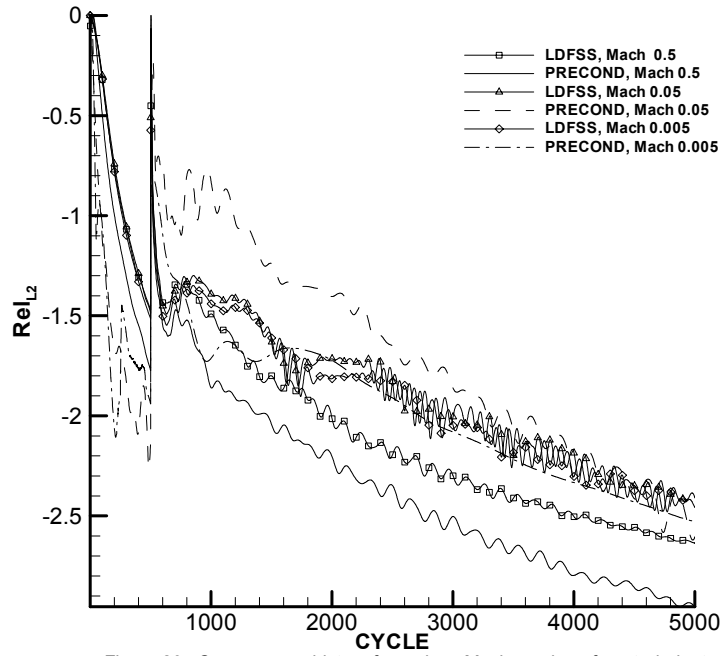


Figure 29: Convergence history for various Mach numbers for a turbulent, calorically perfect gas flow through a channel

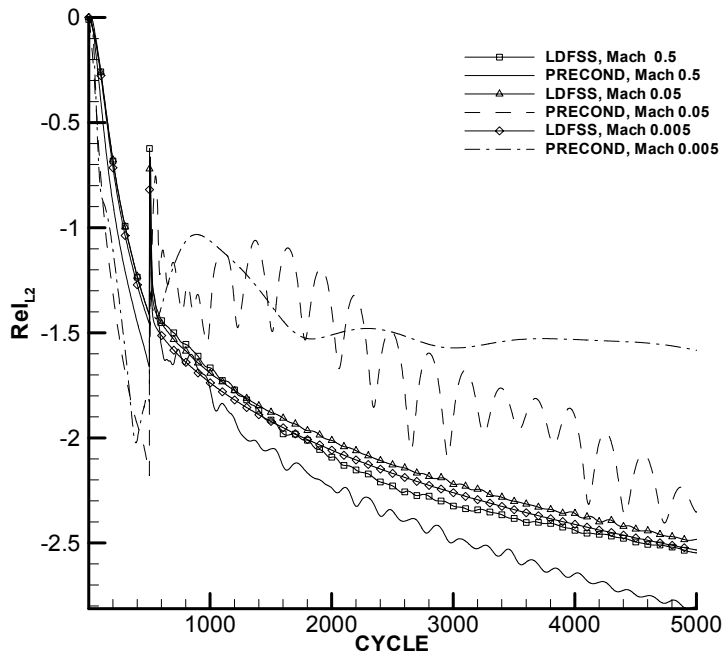


Figure 30: Convergence history for various Mach numbers for a turbulent, two-species air flow through a channel

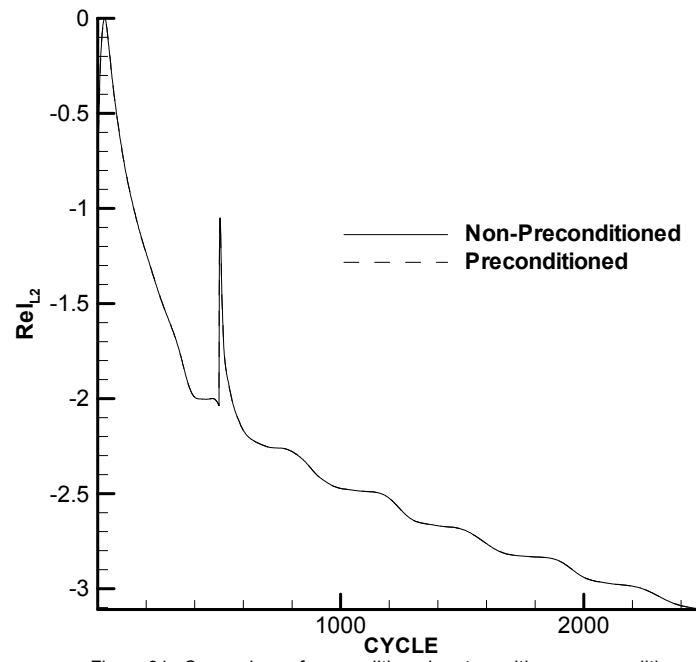


Figure 31: Comparison of preconditioned system with non-preconditioned system for supersonic flow

**Table 1: Reference conditions at Mach 0.5**

Static Pressure	Pa	101751.9
Static Density	kg/m <sup>3</sup>	1.177
Static Temperature	K	300
Reynolds Number	per meter	1.11E+07

**Table 2: Boundary layer thickness for laminar flow over flat-plate**

<b>X-loc.</b> (meters)	<b><math>\delta</math> (meters) at Mach 0.05</b>			<b><math>\delta</math> (meters) at Mach 0.005</b>		
	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 3: Boundary layer thickness for laminar flow through a Channel**

<b>X-loc.</b> (meters)	<b><math>\delta</math> (meters) at Mach 0.05</b>			<b><math>\delta</math> (meters) at Mach 0.005</b>		
	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	