MCAT Institute
Progress Report
92-010

P. 76

TURBULENCE MODELING FOR HYPERSONIC FLIGHT

Jorge E. Bardina

May 1992

NCC2-585

MCAT Institute
3933 Blue Gum Drive
San Jose, CA 95127

CASI

NCC 2-585

ANNUAL REPORT

	Page
Technical Proposal	1
Work Statement	2
Work Accomplished	3
Code Evaluation Criteria	4
Selection Criteria	5
Numerical methods	7
3-D Codes	10
RANS code	10
CNS code	13
FLO2 code	15
IZTUFF code	17
Codes Performance	19
Turbulence Simulation	23
Future Work	25
Acknowledgments	26
References	27
Figures	30
Appendix	
A. Methods	49
B. **RANS** Method	63

NCC 2-585

ANNUAL REPORT

Technical Proposal

The objective of the present work is to develop, verify and incorporate two-equation turbulence models which account for the effects of compressibility at high speeds into a three-dimensional Reynolds averaged Navier-Stokes (**RANS**) code and to provide documented model descriptions and numerical procedures so that they can be implemented into the NASP CFD codes.

Computational Fluid Dynamics (**CFD**) is recognized as a significant engineering design tool in modern hypersonic projects such as the National Aerospace Plane (**NASP**). The design of the NASP vehicle is a highly complex process. One of the critical tasks involved in such a process is the ability of the turbulence model to predict hypersonic viscous/inviscid interactions, mixing problems, transition, chemical nonequilibria, and a range of other turbulence phenomena.

Turbulence models are developed on the basis of insight gained from experimental and theoretical research. The complexity of turbulence requires that the mathematical models be guided by the flow physics in a rational and practical approach. Test and validation of new models with data of recognized quality is an essential step toward model acceptability. A database of high speed turbulent flows is currently being completed. This database provides a benchmark for testing and validation of new compressible models. Specific topics for the database are high-speed attached boundary layers with pressure gradients, supersonic shear layer mixing, and shock wave/boundary layer interactions.

The turbulence models will be implemented into the Ames 3-D Navier-Stokes codes selected on the basis of several criteria including speed, accuracy, user friendliness, and generality (grid, geometry, boundary conditions). The two-equation models will be validated first against the hypersonic database collection. A major consideration throughout the research effort is the development of improved compressibility corrections to the turbulence models and the identification of models which are superior in their predictive capabilities.

Work Statement

The following is the work statement proposed to be accomplished within the first research year:

1. Evaluate available NASA Ames 3-D RANS codes and make a selection based on speed, accuracy, user friendliness and generality.
2. Incorporate candidate compressible turbulent flow models into the 3-D code and validate against simple attached flows.
3. Initiate 3-D computations of flows contained in hypersonic database collection.

Work Accomplished

A summary of the work already accomplished in the first research year is outlined below:

1. Four NASA Ames 3-D RANS codes have been tested and evaluated against a flat plate boundary layer flow and an external supersonic flow. The codes used in this study are **RANS**, **CNS**, **FLO2**, and **IZTUFF**.
2. Based on speed, accuracy, and user friendliness for this research, the code **RANS** has been selected as the basic code to test the turbulence models. Work is in progress assisting in the implementation of the turbulence models into **CNS**, **FLO2** and **IZTUFF**.
3. The code **RANS** has been extended from thin boundary layer to full Navier-Stokes.
4. The $k - \omega$ two-equation turbulence model has been implemented into the base code. Tests base on supersonic flow at $M_\infty = 5$ on a cooled flat plate show good agreement with theory and with 2-D numerical simulations (Thomas Coakley's code).
5. A 24° laminar compression corner flow has been simulated and the extend of the separation zone compared against other numerical simulations (George Huang, Tom Coakley, Langley's codes).
6. A **NAS** account has been obtained to simulate 3-D turbulent flows.
7. Work is in progress in the writing of the numerical method of the base code including the turbulence model.

Code Evaluation Criteria

The main purpose of the first task was to evaluate and select the 3-D NASA-Ames Reynolds Averaged Navier-Stokes (RANS) code or codes in order to implement and test compressible turbulence models in the simulation of complex high-speed flows with turbulence interactions. A guideline of various selection criteria and a guideline of the different numerical methods were designed. The various selection criteria included speed, accuracy, user friendliness and generality. The numerical methods are described including equations, capability, discretizing method, convective terms treatment, boundary conditions, viscous terms treatment, conservative properties, iteration method, and time advance method.

These guidelines are shown next:

Selection Criteria

We are in the process of selecting the 3-D NASA-Ames Reynolds Averaged Navier-Stokes (RANS) code or codes in order to implement and test compressible turbulence models in the simulation of complex high-speed flows with turbulence interactions. The selection is based on various criteria including accuracy, efficiency, user friendliness, and generality.

- **Accuracy:**

  conservation of mass, momentum, and energy.
  shock capturing capability.
  present validations with experimental data.

- **Efficiency:**

  total convergence time to steady-state solutions.
  relative speed: cpu seconds/iteration grid-point.
  robustness.
  stability.
  implicit/explicit numerical scheme.
  implicit/explicit boundary conditions.
  data storage requirements.

- **User Friendliness:**

  user interface.

  instruction manual.

  input parameters definition.

  time-step definition.

  data transfer and management.

- **Generality:**

  Navier-Stokes method, PNS method.

  single/multiple grids.

  single/multiple patches in each grid.

  simple/complex geometries.

  simple/complex flows.

  single/general boundary conditions.

  pseudo-time relaxation, time accurate.

  turbulence models.

  chemistry capability (equilibrium, non-equilibrium).

Numerical Methods

A general description of the numerical method is required including the discretizing method, the treatment of the convective terms, boundary conditions, conservative properties, iteration or relaxation method, and time advance method.

- **Equations:**

  Mass, momentum, and energy equations.
  Identify Navier-Stokes terms not included.
  Present program status.
  Navier-Stokes, thin layer, Euler.
  elliptic, hyperbolic, parabolic.
  time relaxation, time accurate.
  compressible, incompressible.
  perfect gas, real gas.
  source terms.
  turbulence models.

- **Capability:**

  type of problems successfully solved.
  program verifications.
  documentation, user friendliness.
  program availability and support.

- **Discretizing method:**

  finite difference, finite volume, finite element.

  regular or staggered grid.

  simple or multiple grids.

  simple patch or multiple patches in each grid.

  structured, unstructured grids.

- **Convective terms treatment:**

  explicit or implicit method.

  central differencing, upwind differencing, or hybrid.

  artificial dissipation, natural dissipation.

  shock treatment, sonic treatment.

  flux-difference splitting, flux-vector splitting.

  first order, higher order.

  TVD, ENO, unconditionally stable.

- **Boundary conditions:**

  flow through (inflow, outflow, mixed, fixed).

  wall (adiabatic, fixed temperature, inviscid).

  symmetry (axis, planes, grid-axis).

  characteristic, primitive, or conservative variables.

  explicit or implicit scheme.

  general optional lists or code dependent.

- **Viscous terms treatment:**

  no viscous, laminar, and/or turbulent capability.

  explicit or implicit method.

  central differencing, upwind differencing, or hybrid.

  first order, higher order.

  turbulence models incorporated in the code.

  uncoupled or coupled relaxation of turbulence model equations.

  boundary conditions of turbulent quantities.

- **Conservative properties:**

  mass, momentum, energy.

  strongly conservative, weakly conservative, non-conservative or mass only conservative.

- **Iteration method:**

  approximate factorization, relaxation, or hybrid.

  matrix inversion, ADI, DDADI (diagonal-dominant ADI).

  non, over, or under relaxation.

  point, line ,plane substitution.

  Newton-Raphson acceleration.

- **Time advance method:**

  explicit/implicit.

  first order, higher order.

  pseudo-time relaxation, time accurate.

3-D Codes

The four NASA Ames Research Center codes included in this study are **RANS**, **CNS**, **FLO2**, and **IZTUFF**. All these codes have been previously used to simulate high speed compressible flows using the three-dimensional thin layer Reynolds Averaged Navier-Stokes equations. The schemes used in these codes are total variation diminishing (TVD) based on flux limiters in the higher-order fluxes. They are spatially second or higher-order accurate with first-order accuracy in the vicinity of shocks. None of the available codes have a manual or a standard code description available. These are all research codes under development and they have been previously used to simulate complex flows in complex geometries. A brief description of these codes is given below.

- **RANS** Code. This code is under development in the Experimental Fluid Dynamics Branch by the investigator of this research, Dr. Jorge Bardina. It is a 3-D implicit finite difference Reynolds-Averaged Navier-Stokes algorithm written in generalized curvilinear coordinates. It is based on a flux difference splitting method which is described in Appendix A under the denomination RANS method. It was developed based on the CSCM code[1-4] written by this researcher in the Aerothermodynamics Branch coupled with ideas obtained from Roe's flux difference method[5]. During this research work, this code has been extended to simulate the full Navier-Stokes equations with a general two-equation eddy viscosity turbulence model.

  Equations: It solves the 3-D compressible mass, momentum, and energy Navier-Stokes equations. Presently, it has the capability to solve the full Navier-Stokes, thin layer, and/or Euler equations.

Capability: It is a research code under development. Its base code has been extensively applied in the simulation complex 3-D hypersonic, supersonic, transonic, and low speed flows on complex configurations, internal and external flows. There is little code documentation available at present time. The investigator in this research has expertise with this code.

Discretizing method: It uses a finite difference on regular simple or multiple-patch grids.

Convective terms treatment: The convective terms are treated using higher order TVD implicit upwind flux difference splitting with flux limiters and no explicit artificial dissipation. The flux limiters are based on the "minmod" method of Osher and Chakravarthy[6]. The averaging procedure is similar to the Roe's scheme, but simpler; it is based on simple arithmetic averaging of primitive variables. However, the coefficients of the Jacobian matrices are more complex (see Appendix A). It captures shock within two to three grid points.

Boundary conditions: The boundary conditions are implicit and they are input through a general optional list of input coefficients. They include various flow conditions, such as, flow through (inflow, outflow, mixed, fixed), wall boundaries (inviscid, viscous, adiabatic, fixed temperature distribution), symmetry boundaries (axis, planes), grid axis singularity (in symmetry plane) along each coordinate direction.

Viscous terms treatment: The viscous terms are differenced using an implicit central difference algorithm. They include laminar and turbulent capability. The turbulence model incorporated in this code is the Baldwin and Lomax mixing length model[7]. Presently, a two equation $k - \omega$ turbulence model[8] has also been incorporated and tested.

Conservative properties: Conservation of mass, momentum and energy is obtained once convergence is achieved using second or higher order differences.

Iteration method: The iteration procedure is based on an alternating symmetric plane Gauss-Seidel relaxation procedure coupled with a DDADI diagonal-dominant approximate factorization within the relaxation plane. It provides the capability to perform full Navier-Stokes simulations and/or marching procedure similar to PNS methods. It includes a Newton-Raphson acceleration procedure for faster convergence. It has the capability to advance the solution in forward sweeps and/or backward sweeps. The solution procedure is generally from the inflow plane through the outflow plane firstly, and from the outflow plane through the inflow one secondly. In supersonic zones, the best procedure is to advance the solution in the streamwise direction only.

Time advance method: The time advancing is first order with capability for uniform time step or local time step based on local CFL number. The time step can be increased within the boundary layers following T.J. Coakley's approach to increase convergence speed[9]. It can be easily modified to perform time-accurate simulations (It has been done and tested previously).

- CNS Code. The Compressible Navier-Stokes (CNS) code is under development in the Applied Computational Fluids Branch. The researchers that provided this code for this research work included Dr. Terry Holts, Dr. Thomas Edwards, and Dr. Bradford Bennett. It is a 3-D thin layer Navier-Stokes solver in generalized curvilinear coordinates on regular zonal grids[10–13]. The flow solver is based on F3D flux vector splitting of Steger and Warming[14], and the zonal interface logic is derived from the TNS (transonic Navier-Stokes) code[15]. The general treatment of the flux vector splitting is briefly described in Appendix A as the Steger and Warming method.

  Equations: It solves the 3-D compressible mass, momentum, and energy Euler or thin layer Navier-Stokes equations. It includes an air model for equilibrium chemistry.

  Capability: It is a research code under development. It is based on the F3D solver that have been extensively applied into the simulation of 3-D transonic flows and several hypersonic external flows. There is no manual and little code documentation is available.

  Discretizing method: It uses a finite difference on regular zonal grids.

  Convective terms treatment: The flux vector splitting of Steger and Warming is applied in the streamwise direction, and central differencing is used in the crossflow directions. Explicit fourth-order (or second-order times pressure gradient) artificial dissipation is added to the right-hand-side in the crossflow directions to achieve stability with the central difference method.

  Boundary conditions: The boundary conditions are explicit and have to be defined in a special subroutine. Present boundary conditions include outflow, freestream, no slip, reflection, axis of symmetry, and singularity line at given grid indices.

Viscous terms treatment: The viscous terms are differenced using an implicit central difference algorithm. They include laminar and turbulent capability. The turbulence model incorporated in this code is the Baldwin and Lomax mixing length model[7].

Conservative properties: Conservation of mass, momentum and energy is obtained once convergence is achieved using second or higher order differences.

Iteration method: The flow solver is the F3D which is a two-factored flux split algorithm proposed by Ying and Steger[16]. The iteration procedure is a LU-ADI method based on the ADI approximate factorization of Beam and Warming[17]. The first left-hand side operator includes a block lower triangular matrix coupled with a block tridiagonal matrix in one crossed flow direction. The second left-hand side operator includes a block upper triangular matrix coupled with a block tridiagonal matrix in the other crossed flow direction.

Time advance method: The time advancing is first order with capability for uniform time step or local time step based on zonal CFL number.

- **FLO2** Code. This code is under development in the Aerothermodynamics Branch by Dr. Ethiraj Venkatapathy. It is a 3-D implicit, finite volume, thin layer, Reynolds Averaged Navier-Stokes method in generalized curvilinear coordinates[18-21]. This code is based on a LDU-ADI diagonal algorithm developed by Dr. S. Obayashi[18] with Roe's averaging[5] and finite differences with MUSCL extrapolation. Roe's flux difference splitting method used in this code is also briefly described in Appendix A.

  Equations: It solves the 3-D compressible mass, momentum, and energy Euler or thin layer Navier-Stokes equations. It solves ideal gas and/or chemically frozen flows.

  Capability: It is a research code under development and it has been extensively applied into the simulation of 3-D complex plume flows. There is no manual and little code documentation is available.

  Discretizing method: It uses finite differences on regular grids.

  Convective terms treatment: It is a TVD algorithm based on Roe's averaging and upwind flux-difference splitting approximate Riemann solver with entropy correction. The MUSCL differencing approach is used to obtain spatially second- or third-order accuracy with flux limiters. The flux limiters include the "minmod" limiter of Osher and Chakravarthy[6], second-order UNO, second-order differentiable limiter.

  Boundary conditions: The boundary conditions are treated explicitly. Present boundary conditions include inflow, outflow, freestream, no slip, reflection and axis of symmetry.

  Viscous terms treatment: The viscous terms are differenced using an implicit central difference algorithm. They include laminar and turbulent capability. The turbulence model incorporated in this code is the Baldwin and Lomax mixing length model[7].

Conservative properties: Conservation of mass, momentum and energy is obtained once convergence is achieved using second or higher order differences.

Iteration method: The iteration procedure is based on a scalar-matrix operator to obtain steady-state solutions. It is derived from an approximate diagonalization of the LU-ADI implicit operators in the implicit nonconservative form like Pulliam and Chaussee's diagonal algorithm[22]. It consists of three sweeps with a lower, a diagonal, and an upper operator in each coordinate direction.

Time advance method: The time advancing is first order with capability for uniform time step or local time step.

- **IZTUFF** Code. This code is under develop ment by Dr. Gregory Molvik in the Applied Computational Fluids Branch. It is a 3-D implicit finite volume thin-layer Reynolds-Averaged Navier-Stokes method in generalized curvilinear coordinates[23-24]. This code has mainly been used to compute the external forebody flow field of 3-D bodies, in conjunction with a parabolized Navier-Stokes (PNS) code to solve 3-D ideal, equilibrium-, or nonequilibrium-chemistry external flows. This code also uses Roe's method of flux difference splitting described in Appendix A.

  Equations: It solves the 3-D compressible mass, momentum, and energy thin layer Navier-Stokes equations.

  Capability: It is a research code under development. It has been extensively applied in the simulation of complex external configurations including nonequilibrium chemistry. There is little code documentation available at present times.

  Discretizing method: It uses finite volume on regular grids.

  Convective terms treatment: It is a TVD scheme using Roe's upwind-biased flux-difference splitting approximate Riemann solver[5] with entropy correction. The "minmod" flux limiter of Osher-Chakravarthy[6] is used to obtain spatially second- or third-order accuracy.

  Boundary conditions: The boundary conditions are explicit and include viscous wall, symmetry plane, and frozen supersonic inflow.

  Viscous terms treatment: The viscous terms are differenced using an implicit central difference algorithm. They include laminar and turbulent capability. The turbulence model incorporated in this code is the Baldwin and Lomax mixing length model[7].

Conservative properties: Conservation of mass, momentum and energy is maintained through the finite volume approach when convergence is achieved using second or higher order differences.

Iteration method: The iteration procedure is based on the 3-D ADI approximate factorization of Beam and Warming[17].

Time advance method: The time advancing is first order with capability for uniform time step or local time step based on local CFL number. It can be easily modified to perform time-accurate simulations.

Codes Performance

The predictive capabilities of the 3-D Reynolds averaged Navier-Stokes codes are well known through the different applications shown in available publications (see previous section). The main objective of the present work is to analyze and test this codes in order to gain a better understanding on their accuracy, efficiency, user friendliness, and generality.

A few tests of inviscid and laminar flows have been simulated. These simulations include a supersonic free flow, an hypersonic blunt body, and a supersonic flat plate. Some of the results of these simulations are shown below. In each case, the simulations used the same grid and started with the same initial flow field; the flow variables were read and rewritten following the particular non-dimensional system of each code.

Figure 1 shows the residual history of a supersonic free flow with $M_\infty = 4$. In this case, the mesh had 11x11x11 grid points equally spaced. The residuals were defined as the sum of the square residuals of the energy equation divided by the number of grid points and the CFL number. The energy residual is usually the largest one when it is compared with the residuals of the other conservative variables. The CFL factor is removed from the residuals in order to get independence from variations in the time step or CFL number. **IZTUFF** achieved a convergence of the order of 10^{-14} and in about 30 iterations with a $CFL = 50$; minor code modifications were needed in order to achieve convergence. **RANS** achieved a convergence of the order of 10^{-11} in about 30 iterations with a $CFL = 50$; faster convergence is obtained when the Newton-Raphson acceleration cycle is increased. **CNS** achieved a convergence of the order of 10^{-11} in about 290 iterations with a $TSTEP = 5$; simulations with higher values of $TSTEP$ produced negative pressure or density values, while lower values of this parameter increased the number of iterations needed to achieve convergence. **FLO2** did not achieved a fast convergence, the residuals were of the order of $7 \cdot 10^{-6}$ after 9000 iterations with a $CFL = 50$. Some of the differences can be explained by the differences in the treatment of the fluxes, however, the results are user dependent and code modifications can produce significant improvements.

Figure 2 shows the residual history of a supersonic laminar flow with $M_\infty = 4$ and $Re = 10^5$. on an adiabatic flat plate. The mesh had 51 points along the plate, 31 points from the wall through the free stream, and 31 points across the main flow. **RANS**, **CNS** and **FLO2** achieved a convergence of 10^{-9} in about 25, 400, and 3000 iterations, respectively. **RANS** was run with $CFL = 10000$ and four Newton-Raphson acceleration cycles. **CNS** was run with $TSTEP = 25$ and **FLO2** was run with $CFL = 25.0$. A slight improvement was observed with **FLO2** when $CFL = 2$, larger or smaller values of CFL decreased the convergence history of **FLO2**. Dr. E. Venkatapathy achieved convergence in about 1000 iterations instead of 3000 iterations when **FLO2** was run as a 2-D plane; this kind of improvement was also observed in the other codes. The convergence history of **IZTUFF** is not included because the residuals did not decrease after 1400 iterations with $CFL = 0.9$; since this code is in a developmental stage, it is considered that some code modifications are required in this version.

Although the residual history indicates the number of iterations needed to achieve convergence, different codes and methods require different amounts of time and memory to complete each iteration. The following table shows the MIPS (millions of instructions per second), MFLOPS (millions of floating operations per CPU second), and SPEED (CPU micro-seconds per iteration and per grid point) obtained with each code in the simulation of the flat plate in the Cray-YMP "eagle" at NASA Ames Research Center.

Table 1. Codes Performance in Cray-YMP

Code	MIPS	MFLOPS	SPEED
RANS	40	123	165.6
CNS	54	82	51.4
FLO2	34	124	50.7
IZTUFF	59	40	287.1

This table shows that the **CNS** and **FLO2** codes are the fastest codes per iteration, the **RANS** code is about 3.2 times slower and the **IZTUFF** code is about 5.6 times slower than the other two codes. All the codes have been run with a single procesor which has a maximum capability of about 300 MFLOPS. This Cray-YMP has 8 processors which can be used in parallel. Therefore, the speed of all these codes can be improved significantly.

Previous experience (see previous section) shows that all these codes have good shock capturing capabilities and conservation properties. A brief comparison of some of the results obtained in the flat plate simulations are shown below. Figure 3 shows one of the grid planes along the flow direction and the wall boundary. Figures 4a, 4b, 4c and 5a ,5b, 5c show the pressure contours and the velocity vectors in the vicinity of the inflow obtained with the different codes, respectively. The contours and velocity vectors of **RANS** and **FLO2** are quite similar, while the **CNS** pressure contours show some oscillations near the wall boundary and steeper velocity profiles. Some of these differences are due to the different higher-order methods implemented in the codes.

With regard to user friendliness and generality, all these codes are research codes and they are being currently used in the simulation of different flows of current scientific and engineering interest. The input parameters definitions are explained in each code, although there is no instructional manual or user friendly interface. The data transfer and management is relatively easy and no major problems were found in order to run these codes when no modifications were required. All these codes can simulate complex flows in complex geometries. There are strong research efforts undergoing with **CNS** in the Applied Computational Fluids Branch and with **FLO2** in the Aerothermodynamics Branch. The **IZTUFF** code also includes chemistry capability and a two-equation model, however, there is less accumulated experience with this code.

Turbulence Simulation

Following the work of Coakley and Huang[8] in the development and testing of two-equation turbulence models in 2-D flows, the $k - \omega$ model has been implemented into the 3-D **RANS** code. Work is in progress into the implementation of the $k - \epsilon$ baseline models. The model equations are well known and they are reported in several publications[8].

The simulation of a turbulent adiabatic flat plate flow was chosen for evaluation against 2-D results and empirical correlations. The free stream Mach number is $M_\infty = 5$, the Reynolds numbers are $Re_L = 1.4 \cdot 10^7$ and $Re_\theta = 10^4$, and the wall temperature is $T_w/T_{aw} = 1.0$. This is one of the cases previously selected to analyze turbulence models[8]. The numerical simulation was done using 3 parallel planes along the streamwise direction. The calculations were done in the middle plane, while the lateral planes were used to obtain the Jacobians needed in the 3-D code.

The numerical simulation was run 500 iterations with a $CFL = 500$ and required 67 CPU seconds in the Cray-YMP. The speed of the calculation was less than 49 CPU micro-seconds per iteration and per grid point. This speed is much faster than the one reported in Table 1 because this was a 2-D calculation, no fluxes were computed and solved in the crossed flow direction between the parallel planes.

Figure 6a shows the 61x61 regular grid used in each parallel plane. The grid points are clustered in the wall region and a blowup of the wall grid is shown in Figure 6b. Figure 7 shows the normalized temperature contours inside the boundary layer. The temperature rises inside the boundary layer and is cooled by the cold wall. Figure 8 shows a few selected profiles of velocity vectors. It shows the turbulent boundary layer profiles as well as the turning of the velocity vectors due to the lip shock. Figure 9 shows the first wall spacing in wall coordinates Y^+ along the wall, it varies between 0.2 and 0.35. Figure 10 shows the skin friction C_f distribution along the wall. Figures 11a and 11b show the turbulent kinetic energy K and mass weighted ρK profiles against wall distance Y^+, respectively; and Figures 12a and 12b show the specific dissipation rate ω and mass weighted $\rho\omega$ profiles against wall distance Y^+, respectively. Figures 13a and 13b show the calculated velocity profile compared with empirical correlations, the $1/7th$ power law and the compressible law of the wall, respectively. Similar results are reported in the 2-D calculations[8]. Figure 14 shows the turbulent eddy viscosity profile with a peak value at about 0.4 boundary layer distance, which is inside the range of values between 0.3 and 0.5 found in other cold-wall boundary layer[8]. Figure 15 shows a relatively good agreement between the computed skin friction and the van Driest II correlation. Although the agreement is within the observed experimental uncertainties in cooled-wall flows, the agreement is better in the 2-D calculations[8]. Finally, Figure 16 shows a relatively good agreement between the calculated 3-D and 2-D velocity profiles. This agreement is quite good for turbulent calculations done with different codes and different numerical methods.

The results of these simulations support the validation of the turbulence model incorporated into the 3-D **RANS** code.

Future Work

The main tasks to be accomplished next are:

1. Incorporate $k - \epsilon$ models into base code, and evaluate performance against 2-D flat plate flows and axisymmetric ogive-cylinder-flare , $M_\infty = 7$, of Kussoy, M.I., et al.
2. Test and evaluate improved numerics that take into account second law of thermodynamics and higher-order TVD flux limiters.
3. Study and development of compressibility corrections for 3-D turbulence models:
   - 3a. Simulate 3-D fin, $M_\infty = 8.2$, of Kussoy, M.I. et al.
   - 3b. Simulate 3-D intersecting SWBLI (2 fins), $M_\infty = 8.3$, of Kussoy, M.I. et al.
   - 3c. Simulate 3-D fin, $M_\infty = 6$, of Law, C.H.
   - 3d. Simulate 3-D swept compression corner, $M_\infty = 3$, of McKenzie, T.M., et al.
4. Supervise and deliver information needed to implement compressible two-equation models into the 3-D codes **CNS** with Brad Bennett and Tom Edwards, **FLO2** with Ethiraj Venkatapathy, and **IZTUFF** with Greg Molvik.

Acknowledgments

The author wishes to thank J.G. Marvin for his general guidance in this research and Dr. T.J. Coakley for his guidance and insightful discussions throughout this work. The author is also grateful to Drs. P.G. Huang, J.R. Viegas and F.R. Menter for their helpful discussions on turbulence modeling and graphic softwares. This research was performed in the Experimental Fluid Dynamics Branch at NASA Ames Research Center and supported by a grant from NASA to MCAT Institute (NCC 2-585).

References

1. Bardina, J., and Lombard, C.K.: " Three Dimensional CSCM Method for the Compressible Navier-Stokes Equations with Application to a Multi-Nozzle Exhaust Flowfield," AIAA-85-1193, 1985.

2. Bardina, J., and Lombard, C.K.: "Three Dimensional Hypersonic Flow Simulations with the CSCM Implicit Navier-Stokes Method," AIAA-87-1114-CP, 1987.

3. Bardina, J., Lombard, C.K., and Luh, R.C.C., "CSCM Three Dimensional Navier-Stokes Computational Aerodynamics for a Projectile Configuration at Transonic Velocities," AIAA-88-2585-CP, 1988.

4. Bardina, J., Lombard, C.K., and Yoon, S., "Efficient Design Analysis for a New Launch Vehicle using the 3-D CSCM Navier-Stokes Method," AIAA-89-0336, 1989.

5. Roe, P.L., "Approximate Riemann Solvers, Parameter vectors, and Difference Schemes," *J. Comp. Physics,* Vol. 43, 1983, pp. 357-372.

6. Chakravarthy, S.R., and Osher, S., "A New Class of High Accuracy TVD Schemes for Hyperbolic Conservation Laws," AIAA-85-0363, 1985.

7. Baldwin, B.S., and Lomax, H.: "Thin Layer Approximation and Algebraic Model for Separated Turbulent Flows," AIAA-73-0257, 1973.

8. Coakley, T.J., and Huang, P.G., "Turbulence Modeling for High Speed Flows," AIAA-92-0436, 1992.

9. Coakley, T.J., "Implicit Upwind methods for the Compressible Navier-Stokes Equations," *AIAA J.,* Vol. 23, No. 3, March 1985, p. 374.

10. Edwards, T.A., Chaussee, D.S., and Lawrence, S.L., "Comparisons of Four CFD codes as Applied to a Hypersonic All-body Vehicle," AIAA-87-2642, 1987.

11. Flores, J., Ryan, J.S., and Edwards, T.A., "CNS Code Development," Paper 12, *Fifth NASP Tech. Symp.*, Hampton, VA, October 18-21, 1988.

12. Ryan, J.S., Flores, J., and Chow, C.Y., "Development and Validation of CNS (Compressible Navier-Stokes) for Hypersonic Applications," AIAA-89-1839, 1989.

13. Venkatapathy, E., and Feiereisen, W.J., "3-D Plume Flow Computations with an Upwind Solver," AIAA-88-3158, 1988.

14. Steger, J.L., and Warming, R.F., "Flux Vector Splitting of the Inviscid Gasdynamic Equations with Application to Finite Difference Methods," *J. of Comp. Physics,* Vol. 40, 1981, pp. 263-293.

15. Holst, T.L., Thomas, S.D., Kaynak, U., Gundy, K.L, Flores, J., and Chaderjian, N.M., "Computational Aspects of Zonal Algorithms for Solving the Compressible Navier-Stokes Equations in Three Dimensions," *Int. Symp. on Comp. Fluid Dyn.*, Tokyo, Sept. 9-12, 1985.

16. Ying, S.X., Steger, J.L., Schiff, L.B., and Baganoff, D., "Numerical Simulation of Unsteady, Viscous, High Angle of Attack Flows using a Partially Flux Split Algorithm," AIAA-86-2179, 1986.

17. Beam, R.M., and Warming, R.F., "An Implicit Factored Scheme for the Compressible Navier-Stokes equations," *AIAA J.,* Vol. 16, No. 4, April 1978, pp.393-402.

18. Obayashi, S., Matsushima, K., Fujii, K., and Kuwahara, K., "Improvements in Efficiency and Reliability for Navier-Stokes Computations using the LU-ADI Factorization Algorithm," AIAA-86-0338, 1986.

19. Obayashi, S., "Numerical Simulation of Underexpanded Plumes using Upwind Algorithms," AIAA-88-4360-CP, 1988.

20. Ruffin, S.M., Venkatapathy, E., Keener, E.R., and Nagaraj, N., "Computational Design Aspects of a NASP Nozzle Afterbody Experiment," AIAA-89-0446, 1989.

21. Venkatapathy, E., and Feiereisen, W.J., "Computational Analysis of Plume Induced Separation," AIAA-91-0711, 1991.

22. Pulliam, T.H., and Chaussee, D.S., "A Diagonal Form of an Implicit Approximate Factorization Algorithm," *J. of Comp. Physics,* Vol. 39, pp. 347-363, 1981.

23. Molvik, G.A., *A Computational Model for the prediction of Hypersonic, Reacting Flows,* Thesis in Mech. Engrg., The Pennsylvania State Univ., December 1989.

24. Molvik, G.A., and Merkle, C.L., "A Set of Strongly Coupled, Upwind Algorithms for Computing Flows in Chemical Nonequilibrium," AIAA-89-0199, 1989.

Figure 1a. RESIDUAL HISTORY

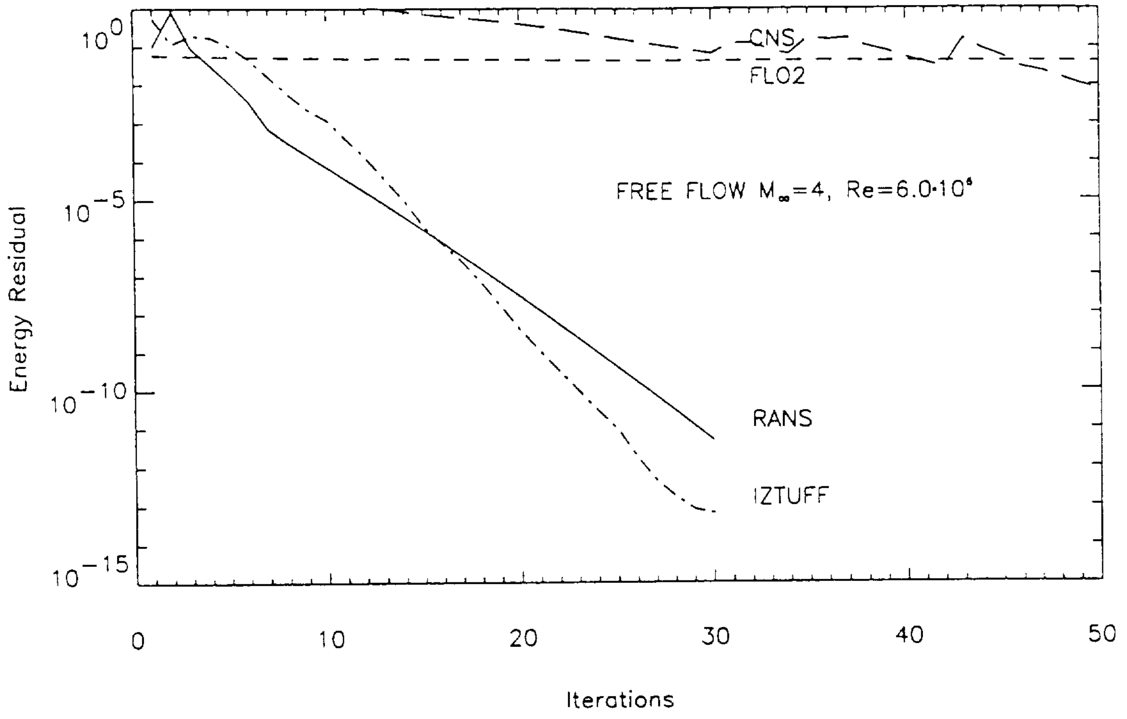


Figure 1b. RESIDUAL HISTORY

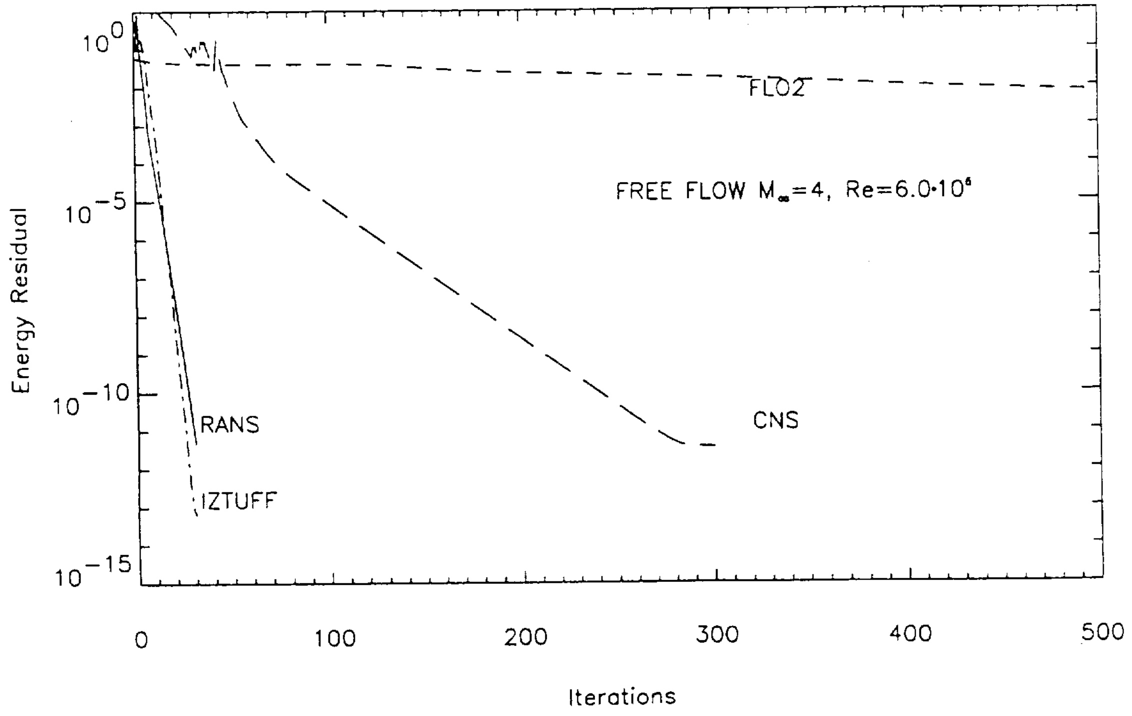


Figure 2a. RESIDUAL HISTORY

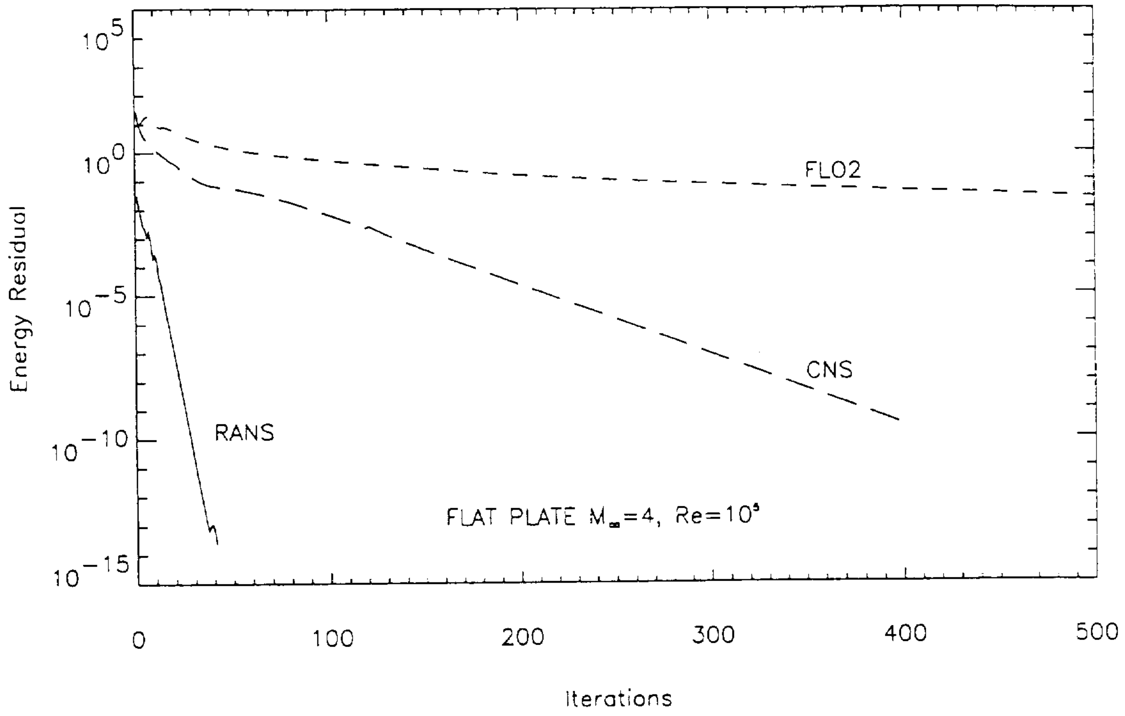
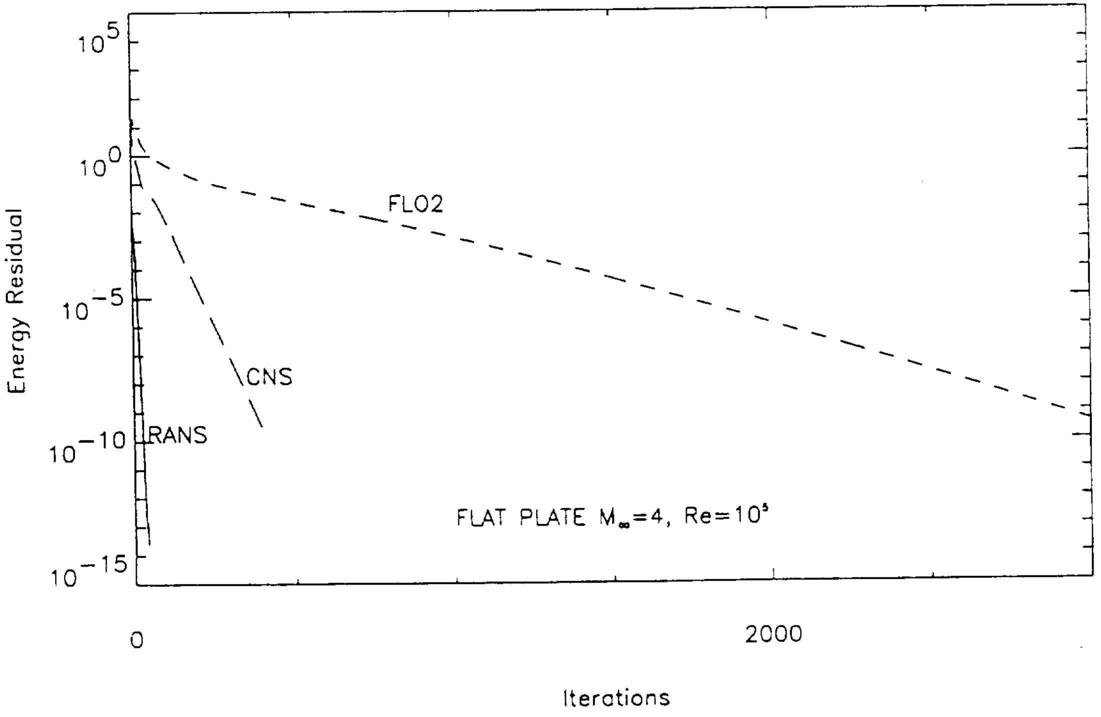


Figure 2b. RESIDUAL HISTORY



51×31 GRID

Figure 3.

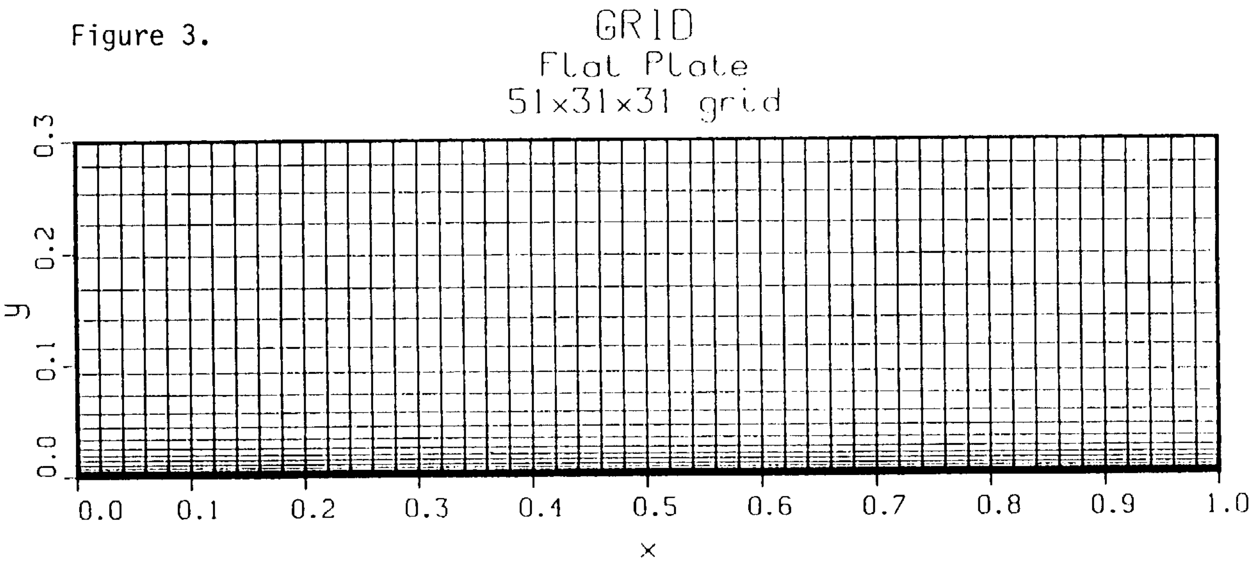


Figure 4a.

NORMALIZED PRESSURE
Flat Plate M=4
RANS

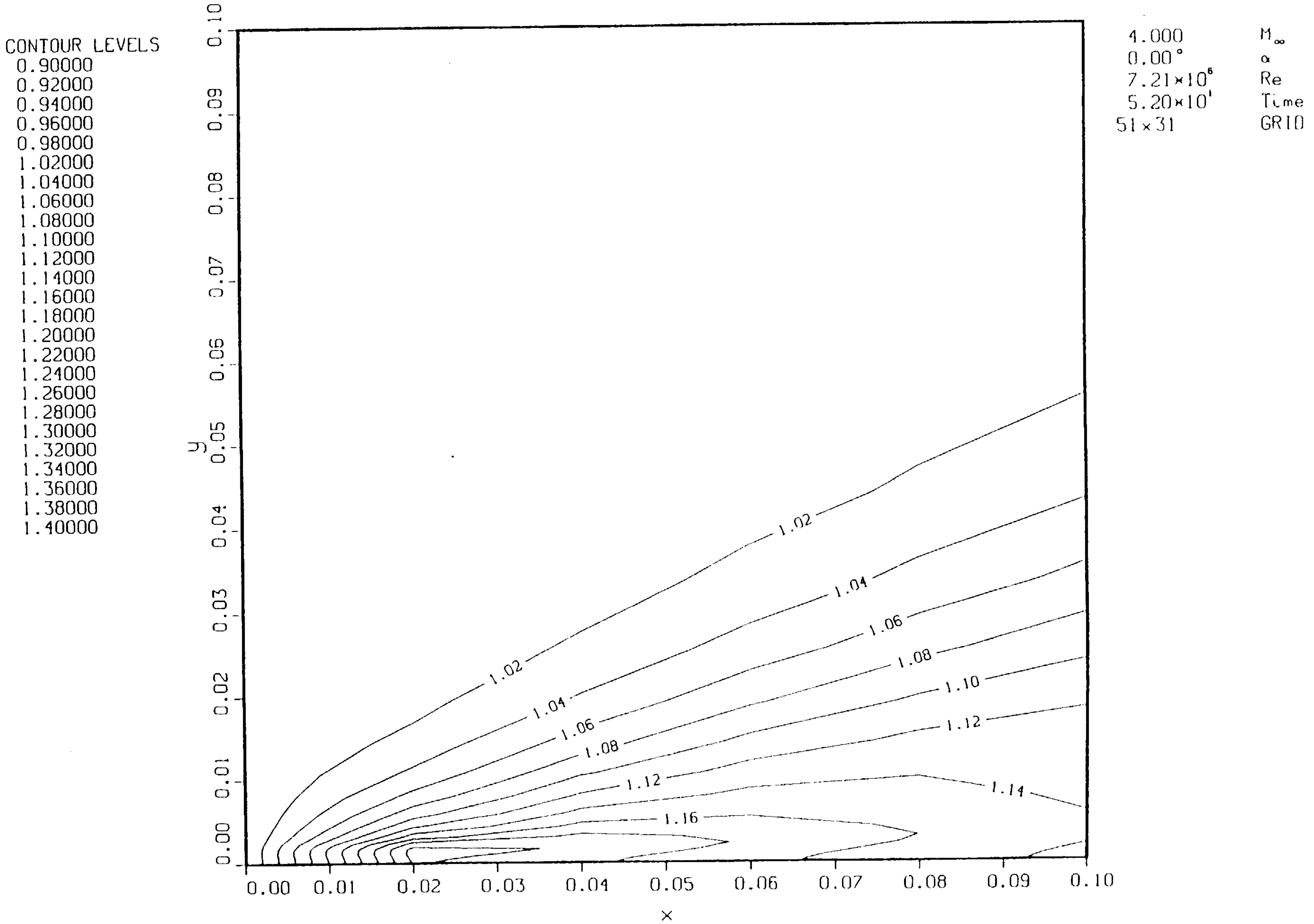
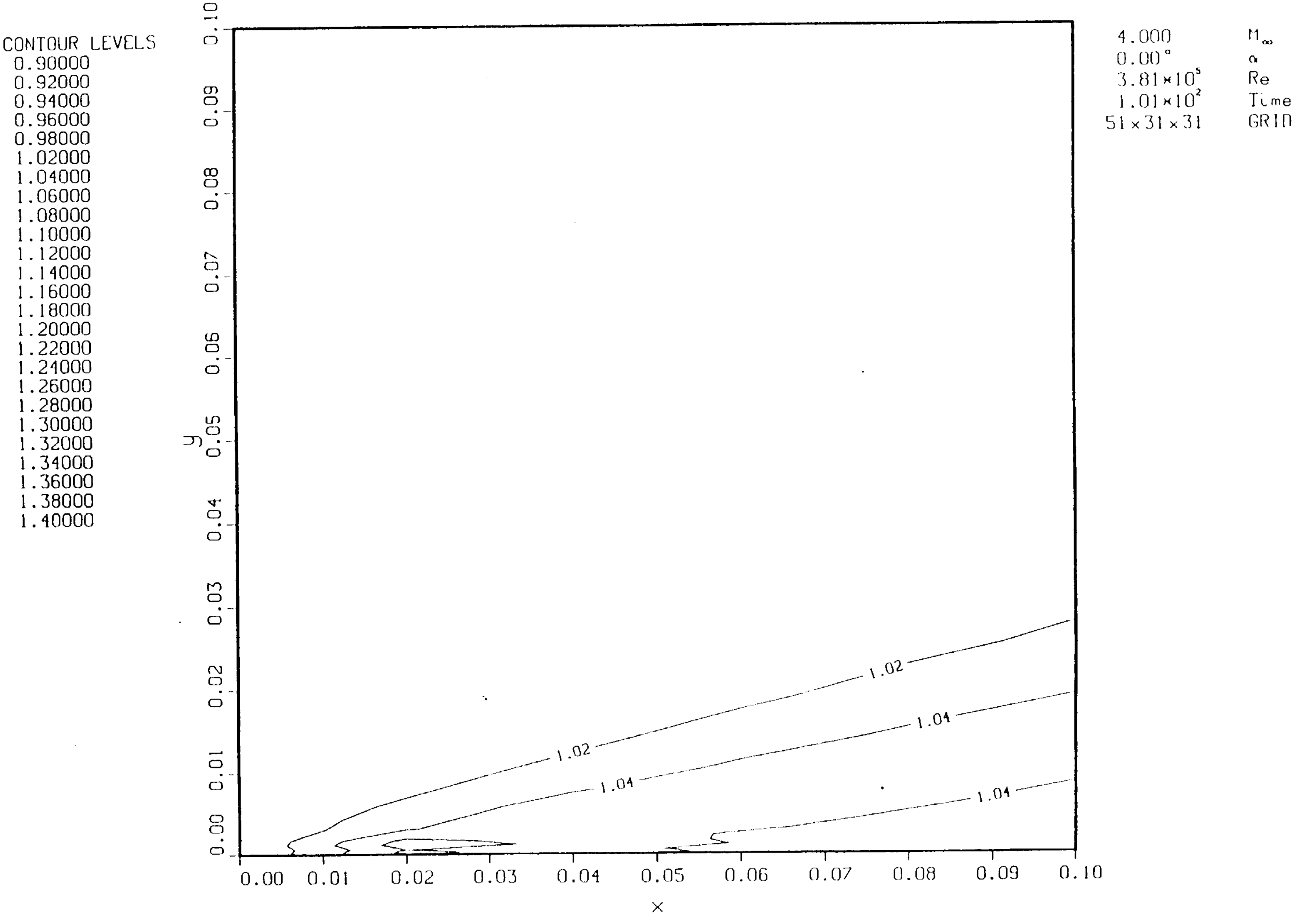


Figure 4b.

NORMALIZED PRESSURE
'Flat Plate'
'CNS'



PLOT 10 16.03.22 FRI 9 AUG, 1991 JOB=bardina , NASA AMES RESEARCH CENTER DISSPLA 11.0

Figure 4c.

NORMALIZED PRESSURE
'Flat Plate'
'FLO3D'

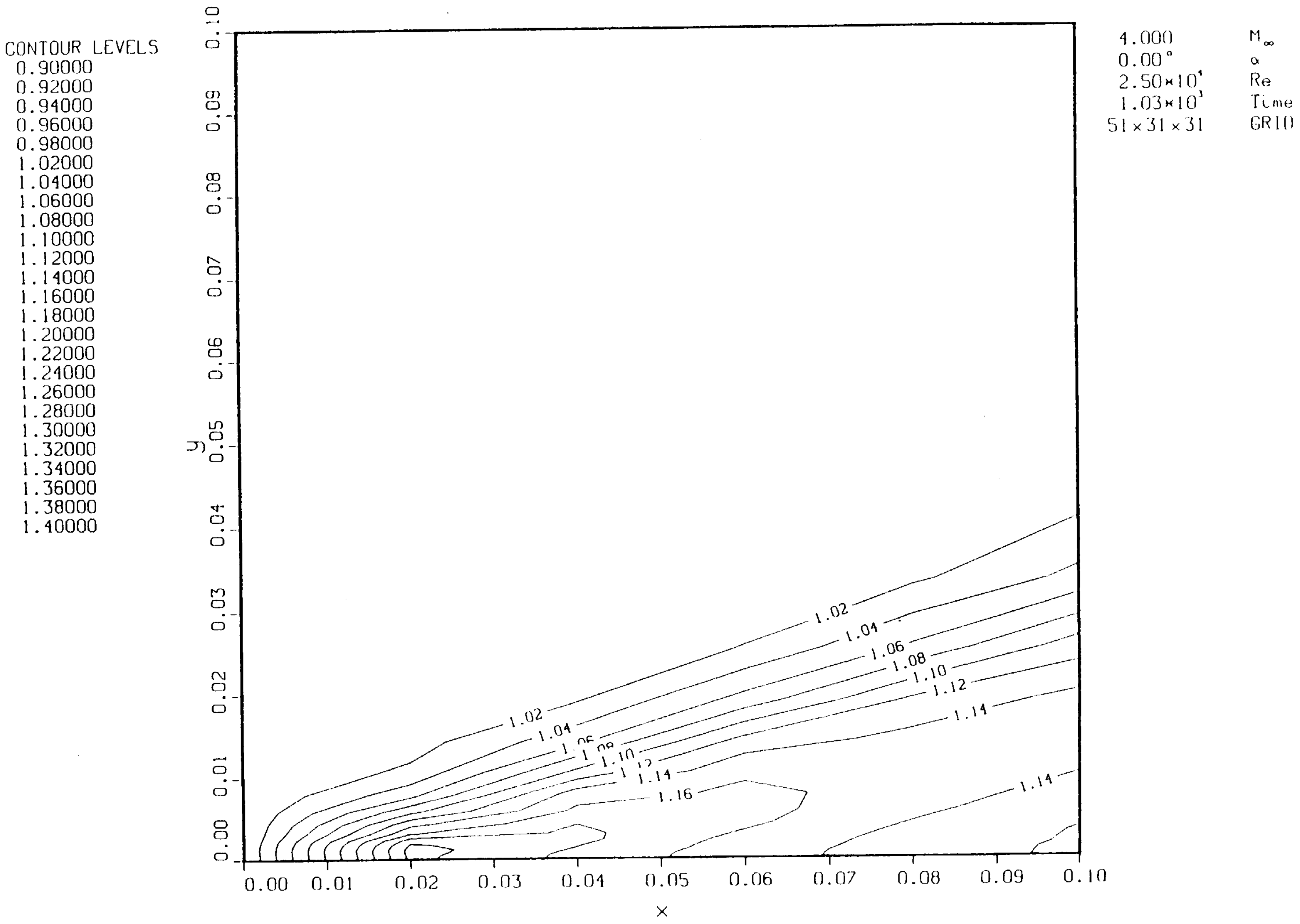
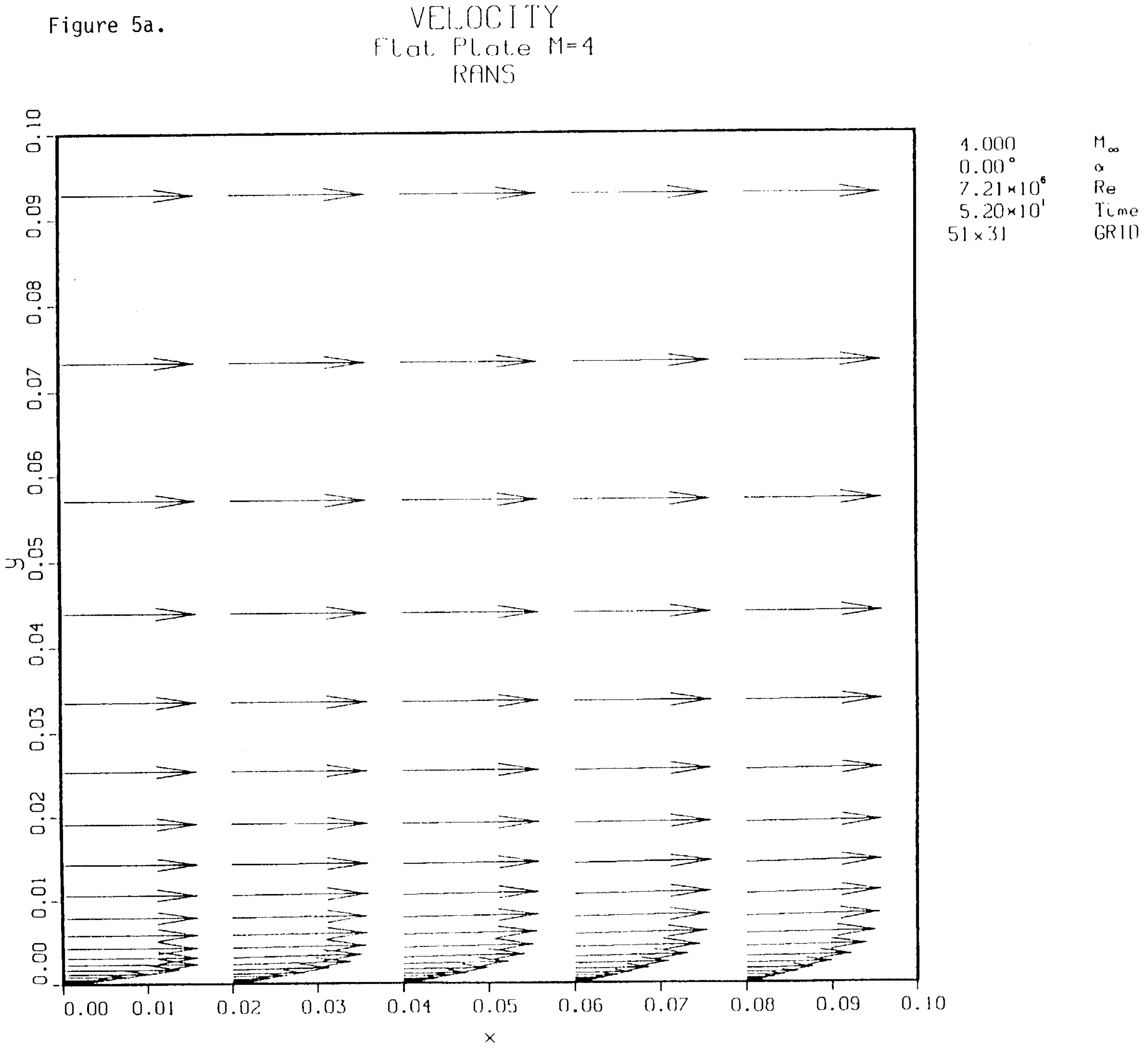


Figure 5a.



LOT 13 09.54.43 WED 1 APR, 1992 JOB-bordino , NASA AMES RESEARCH CENTER DISSPLA 11.0

Figure 5b.

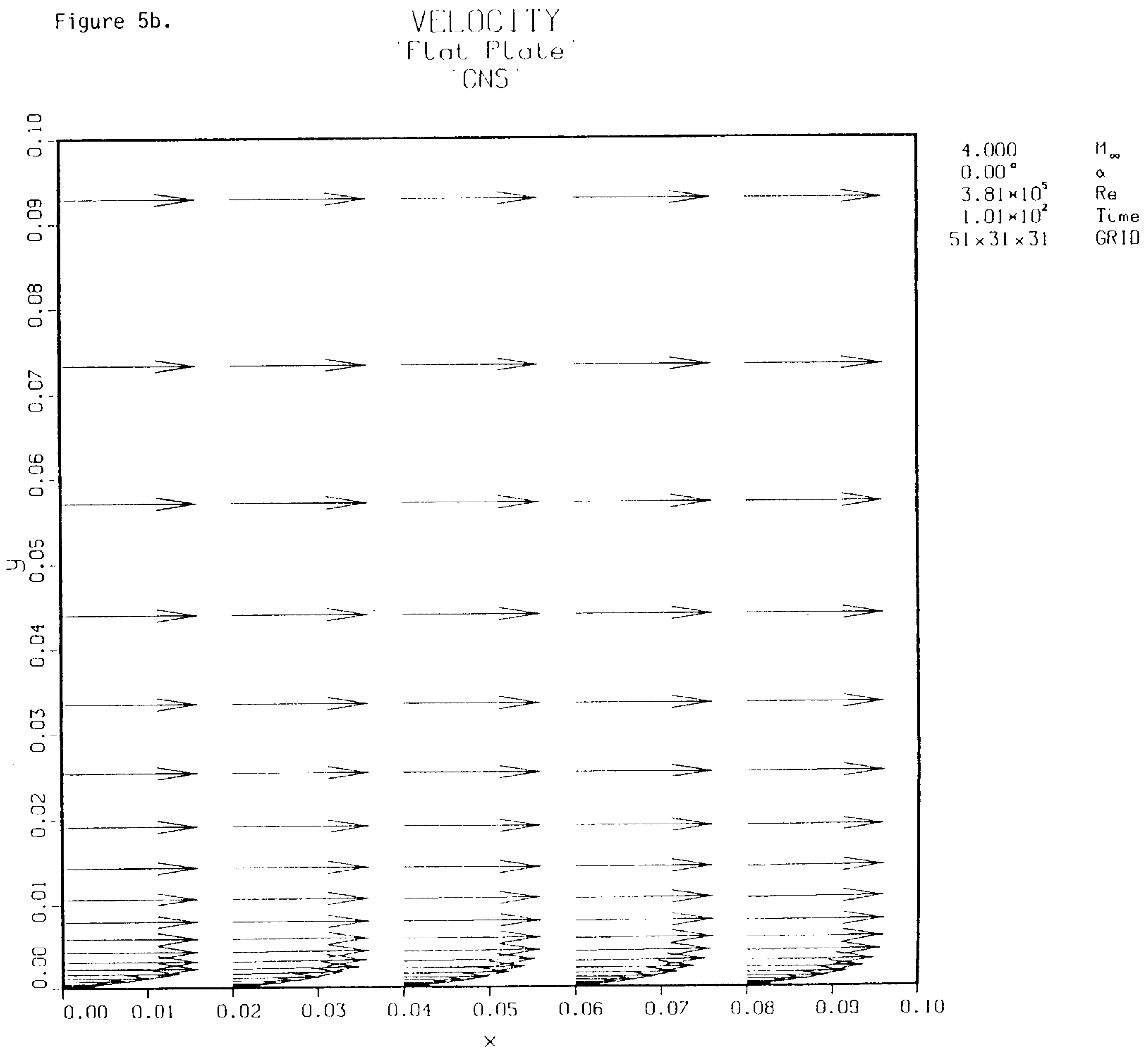

VELOCITY
'Flat Plate'
'CNS'
4.000 M∞
0.00° α
3.81×10⁵ Re
1.01×10² Time
51×31×31 GRID
y
0.00
0.01
0.02
0.03
0.04
0.05
0.06
0.07
0.08
0.09
0.10
x
0.00
0.01
0.02
0.03
0.04
0.05
0.06
0.07
0.08
0.09
0.10

Figure 5c.

VELOCITY
'Flat Plate'
'FL03D'

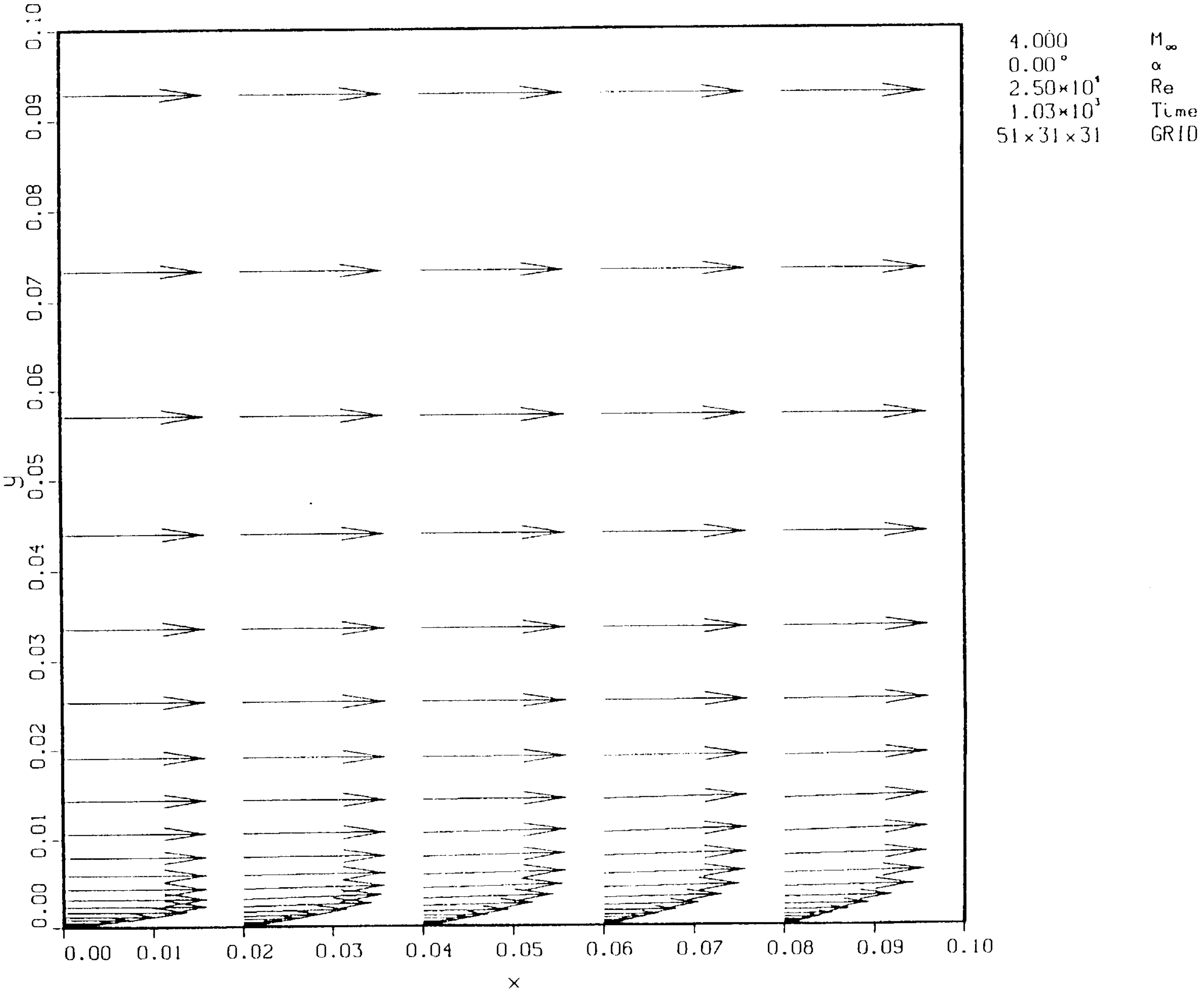
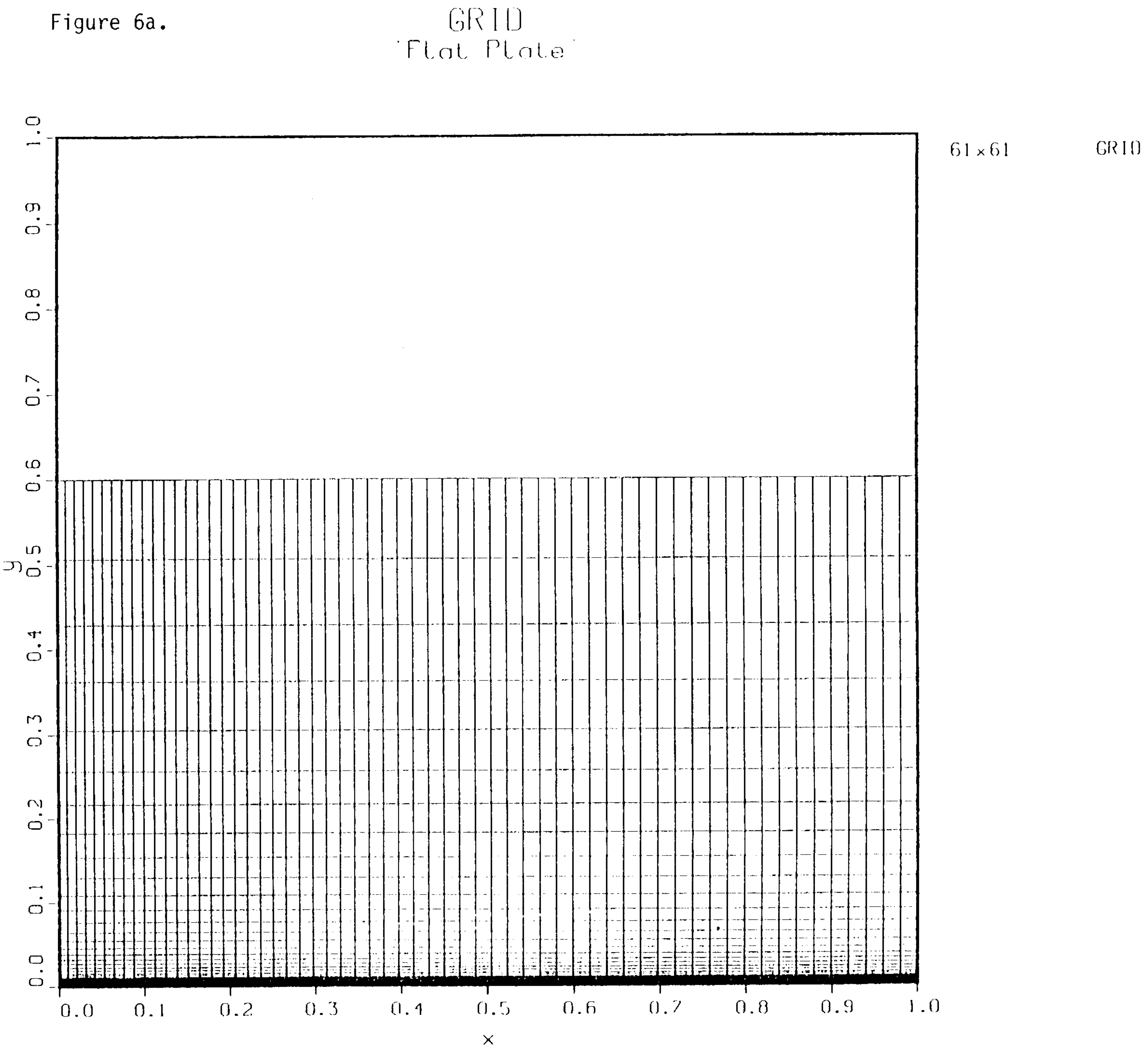


Figure 6a.



PLOT 1 12.48.18 TUES 11 FEB, 1992 JOB-pordino , NASA AMES RESEARCH CENTER DISSPLA 11.0

Figure 6b.

GRID
Flat Plate

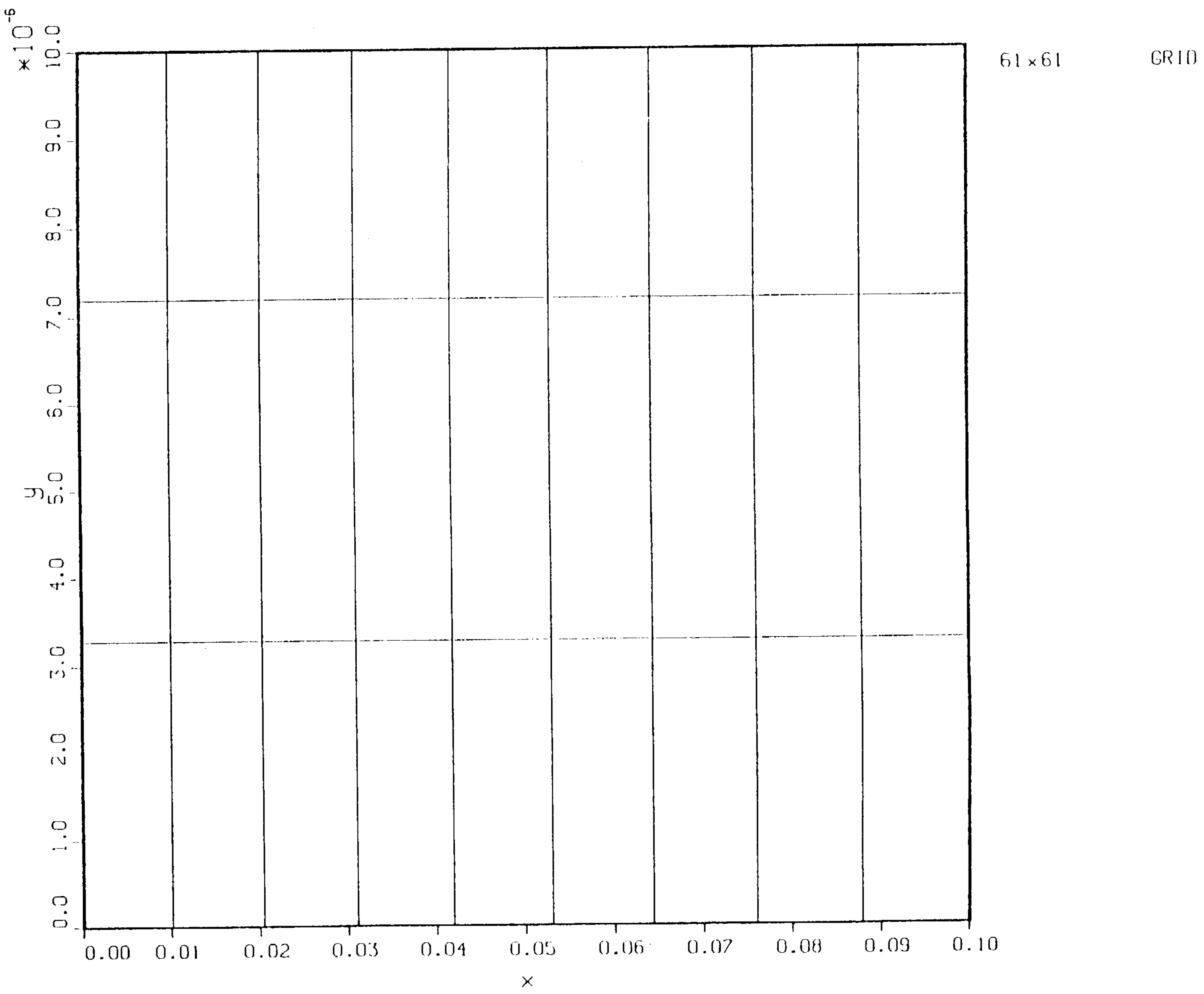


Figure 7. NORMALIZED TEMPERATURE
Adiabatic Flat Plate
M=5 k-w model

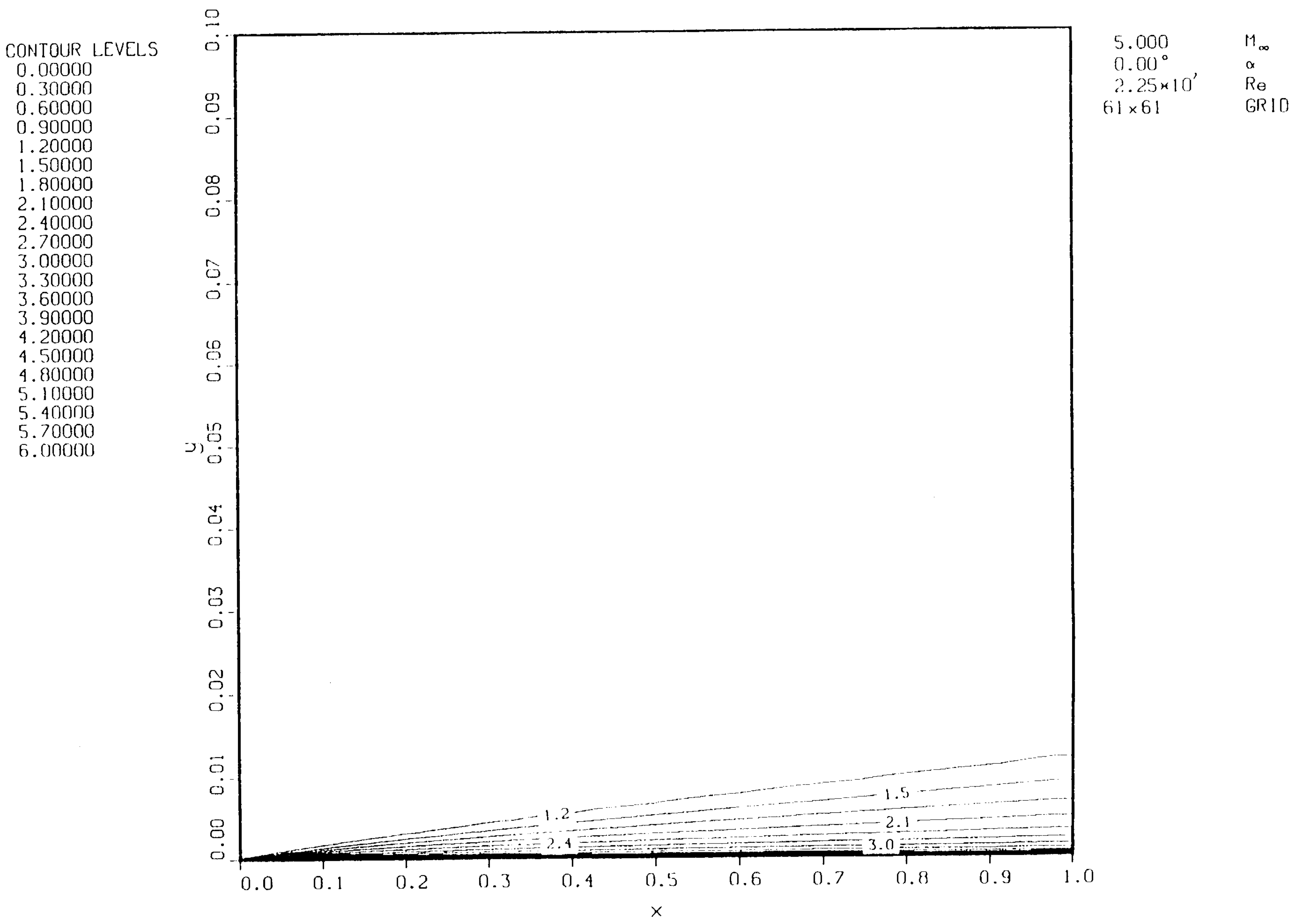
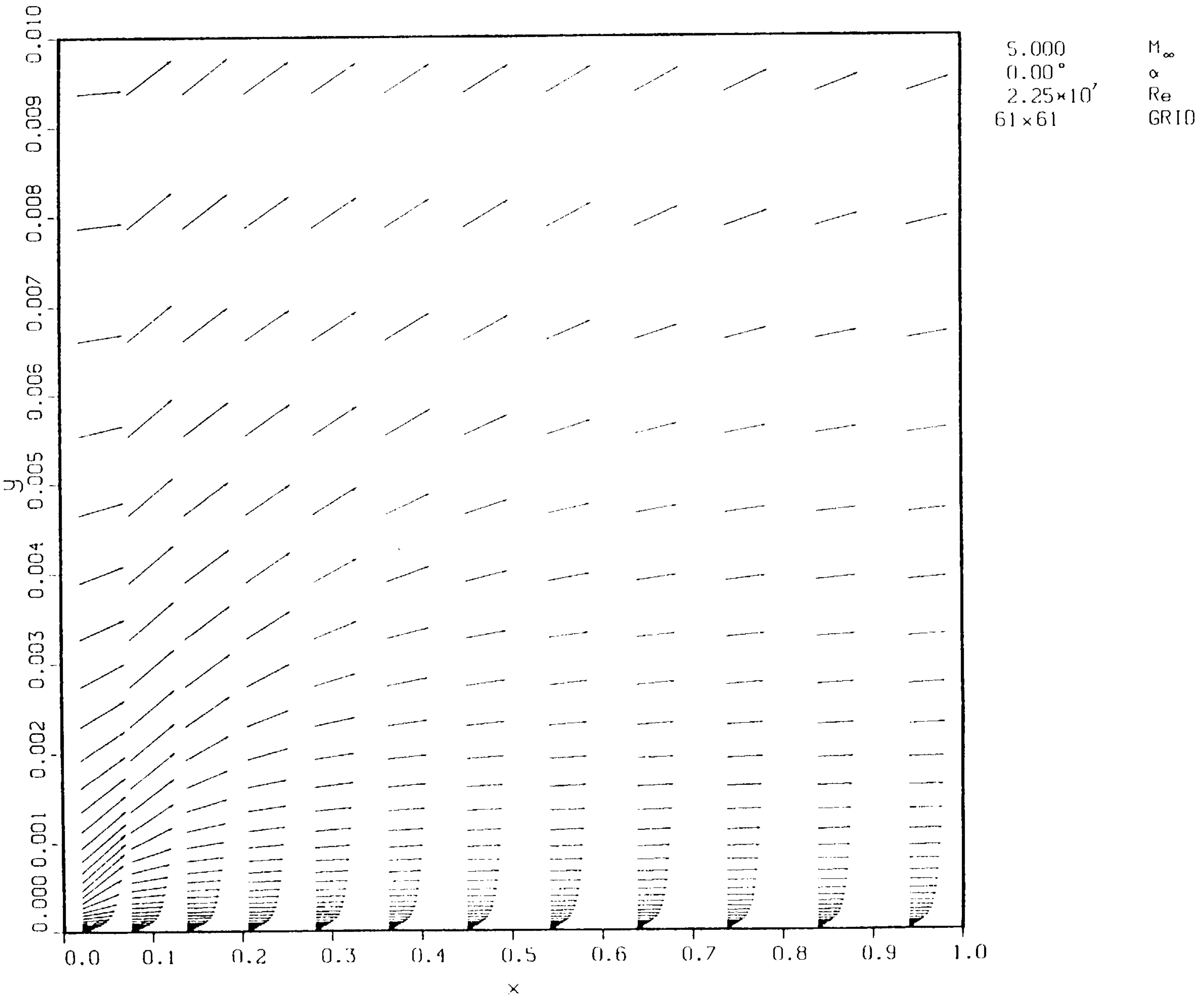


Figure 8.

VELOCITY
'Adiabatic Flat Plate'
'M=5 k-w model'



PLOT 8 12.48.55 TUES 11 FEB, 1992 JOE-bardino , NASA AMES RESEARCH CENTER DISSPLA 11.0

Figure 9. First Wall Spacing

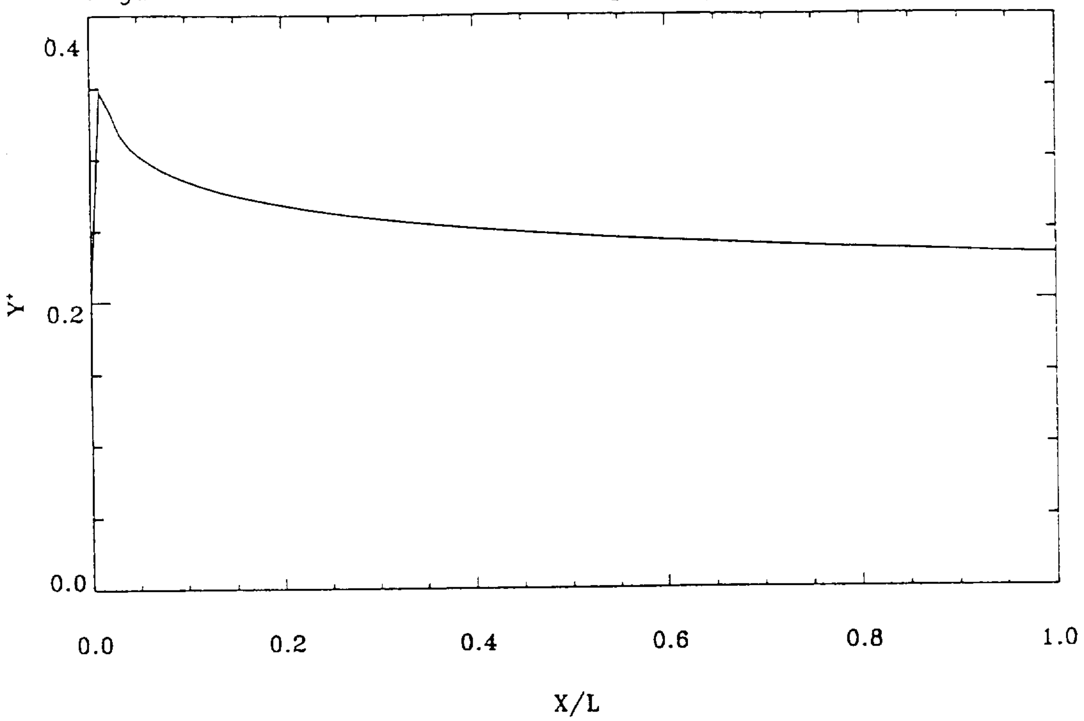


Figure 10. SKIN FRICTION

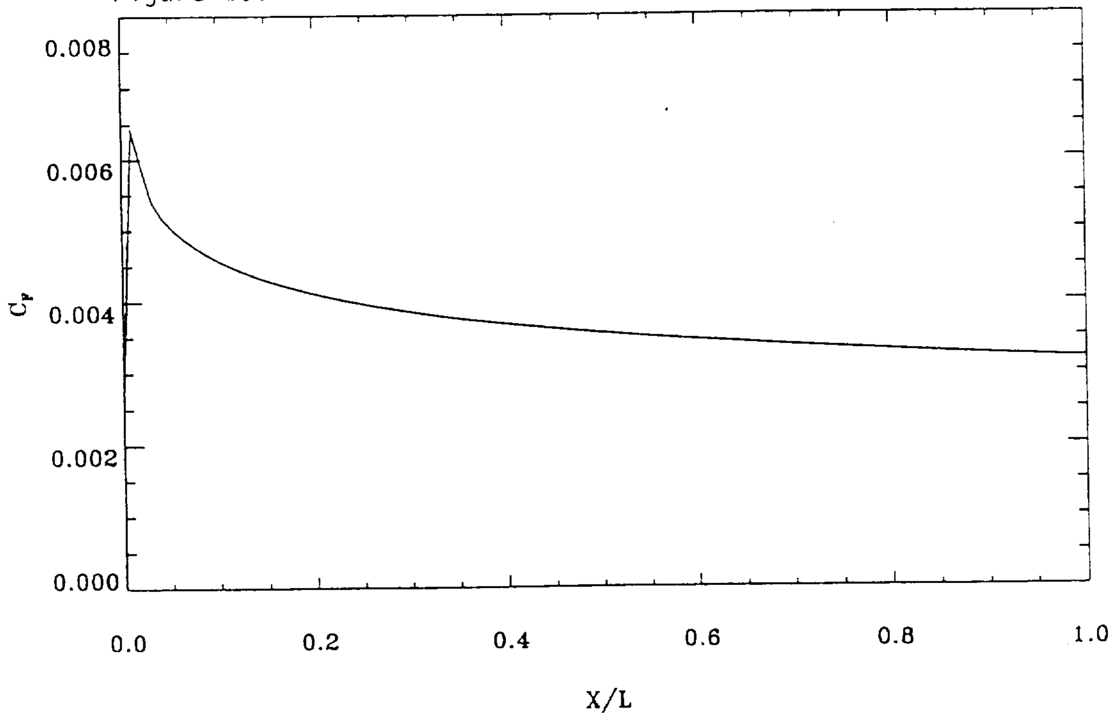


Figure 11a. TURBULENT KINETIC ENERGY

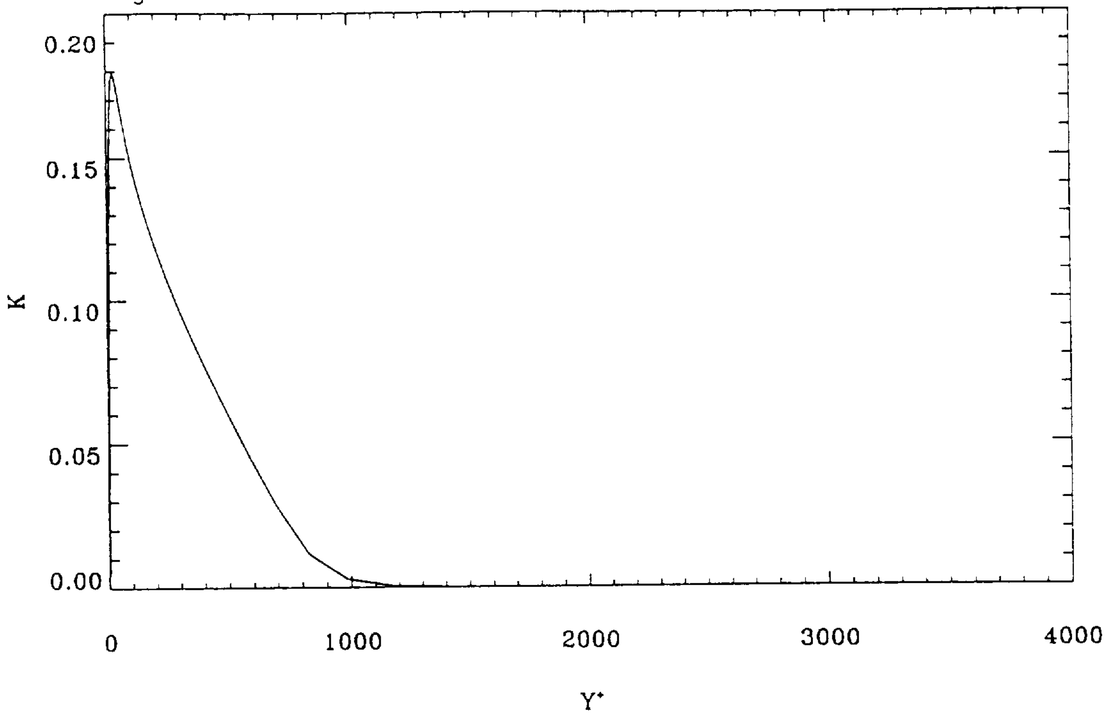


Figure 11b. TURBULENT KINETIC ENERGY

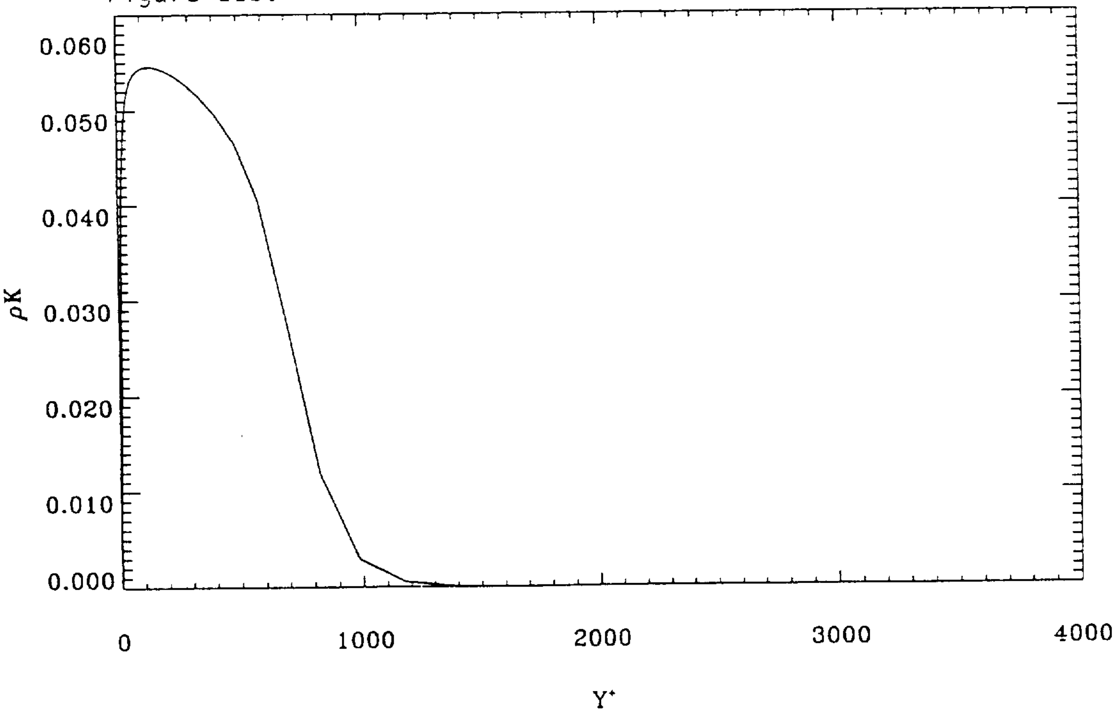


Figure 12a. ω SPECIFIC DISSIPATION RATE

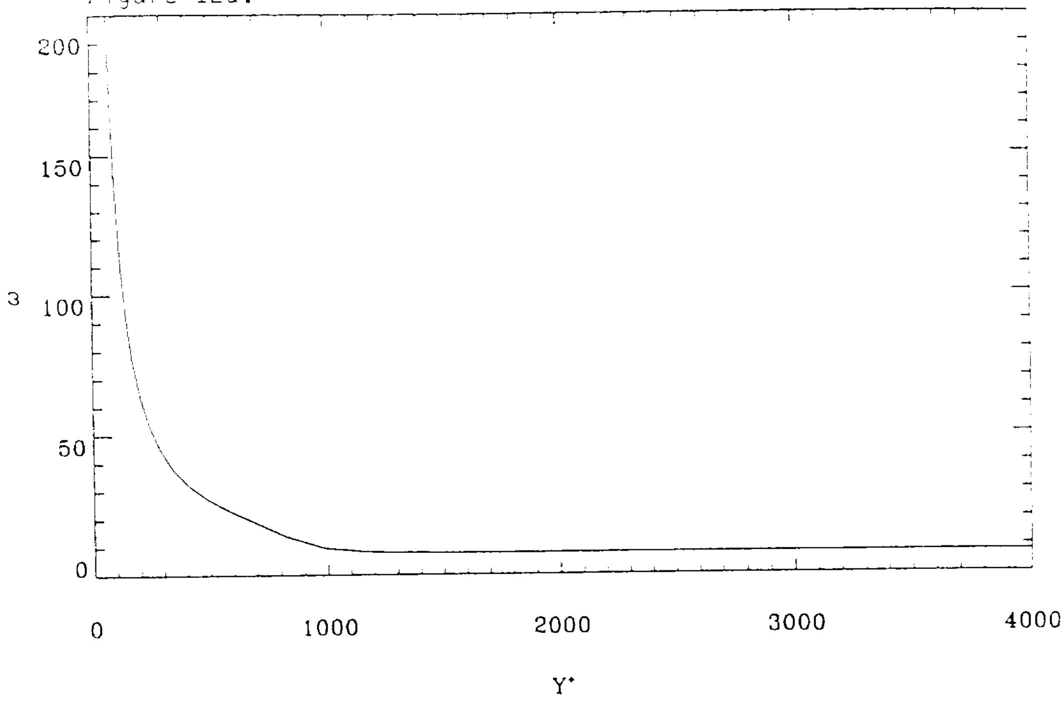


Figure 12b. $\rho\omega$ SPECIFIC DISSIPATION RATE

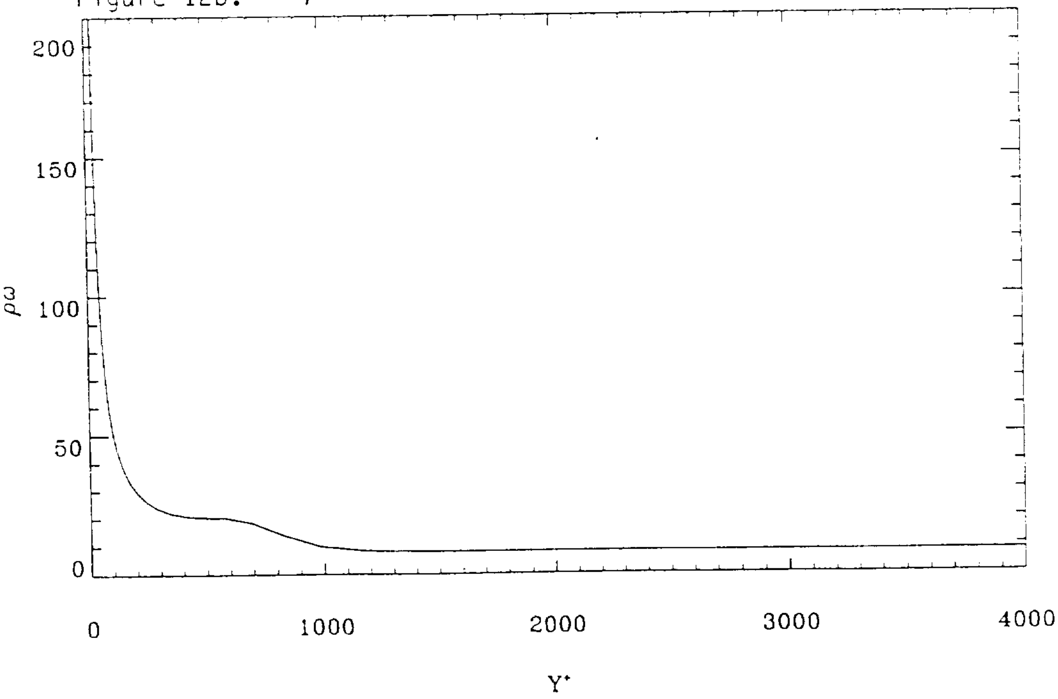


Figure 13a.

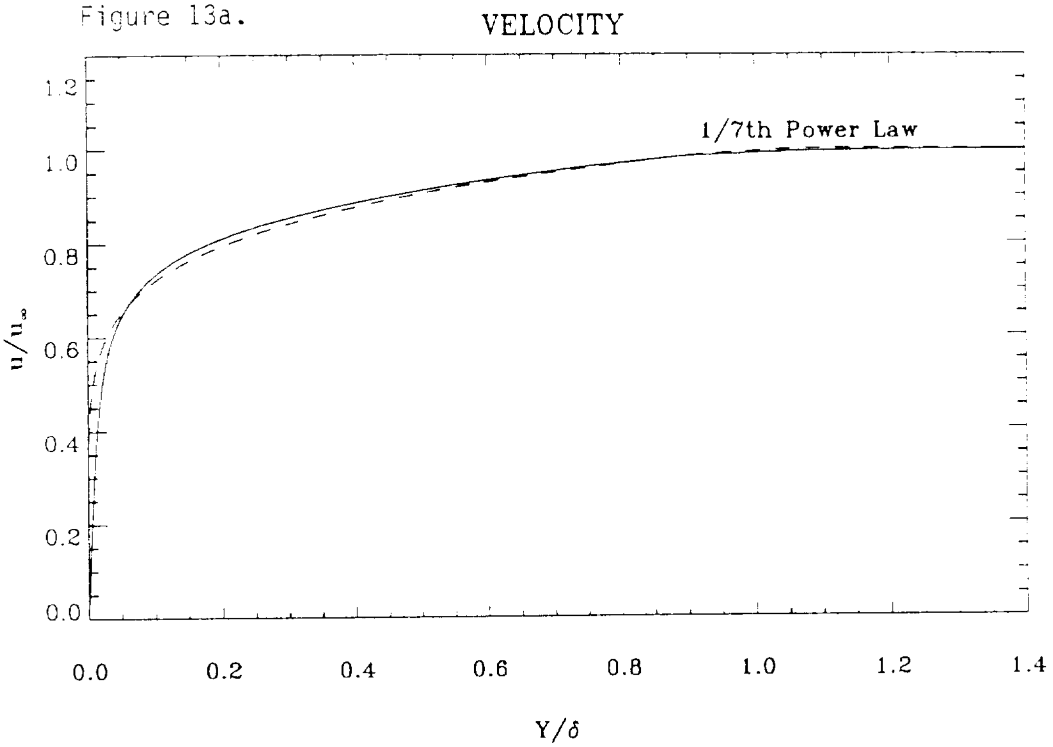


Figure 13b.

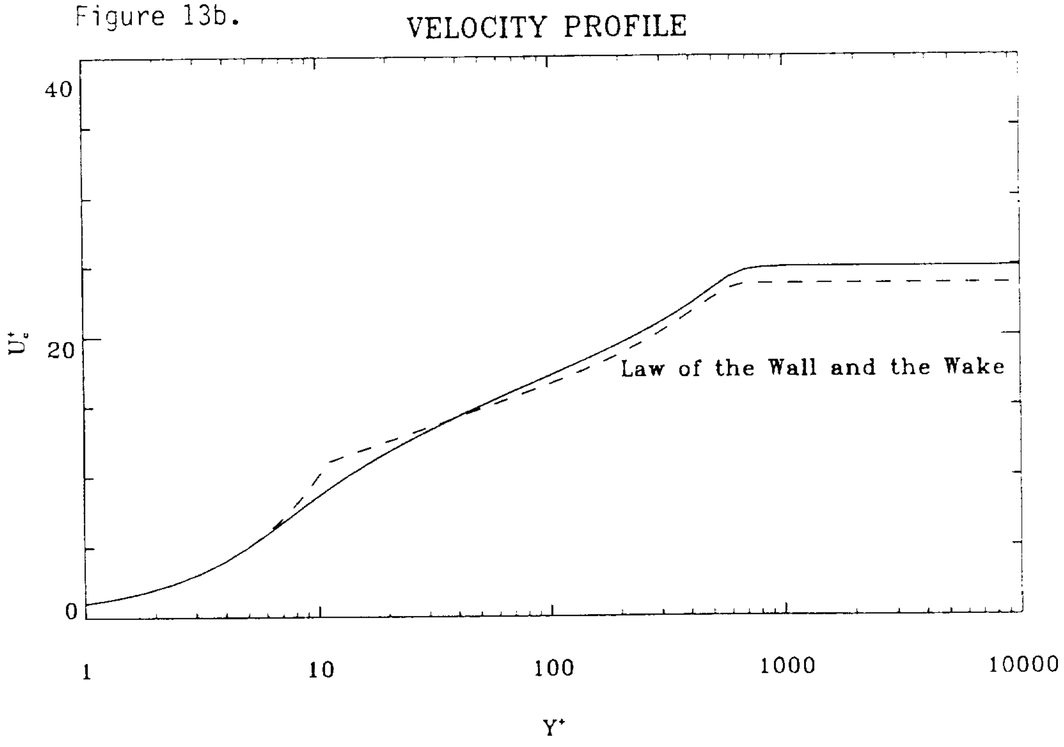


Figure 14. EDDY VISCOSITY

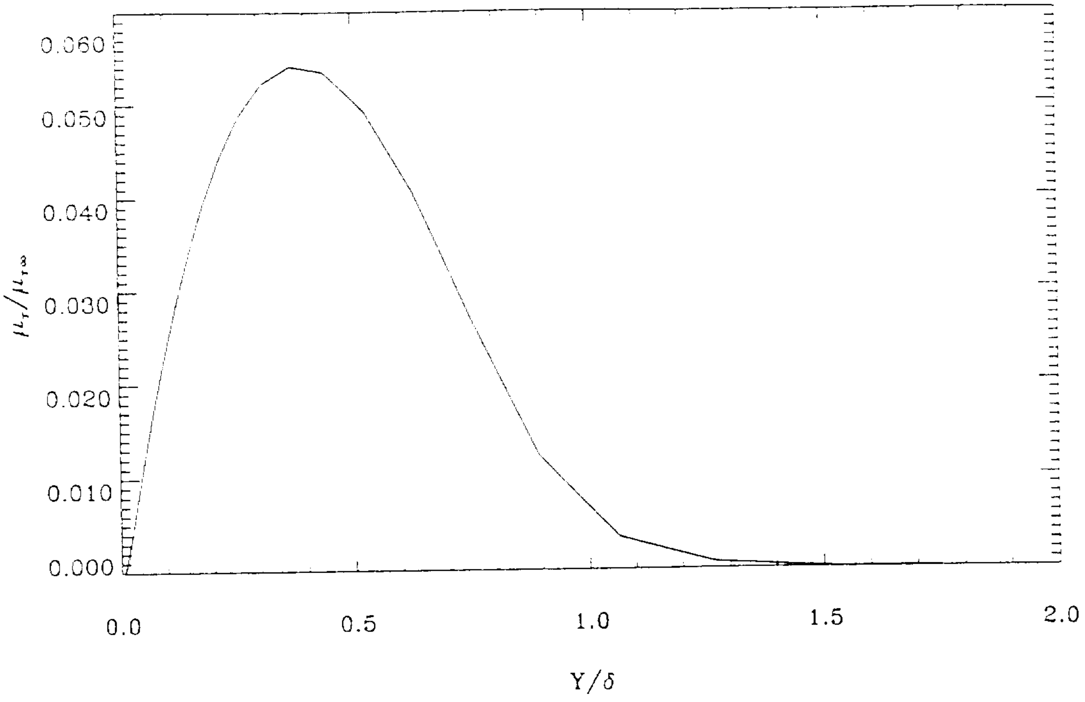


Figure 15. SKIN FRICTION

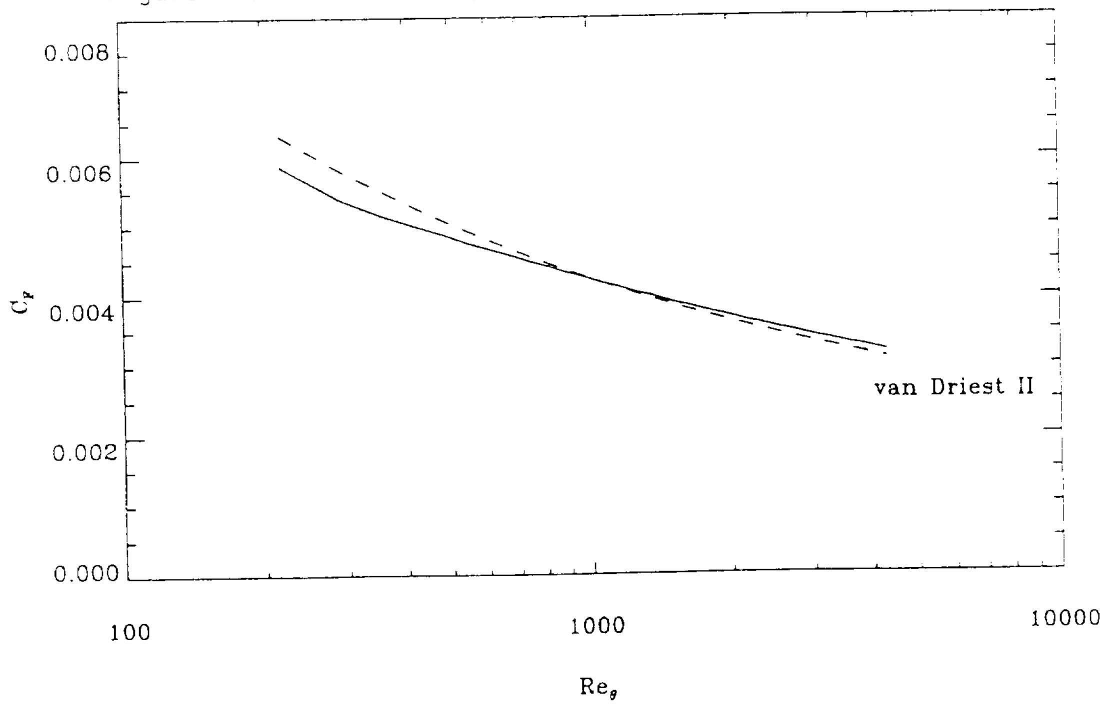
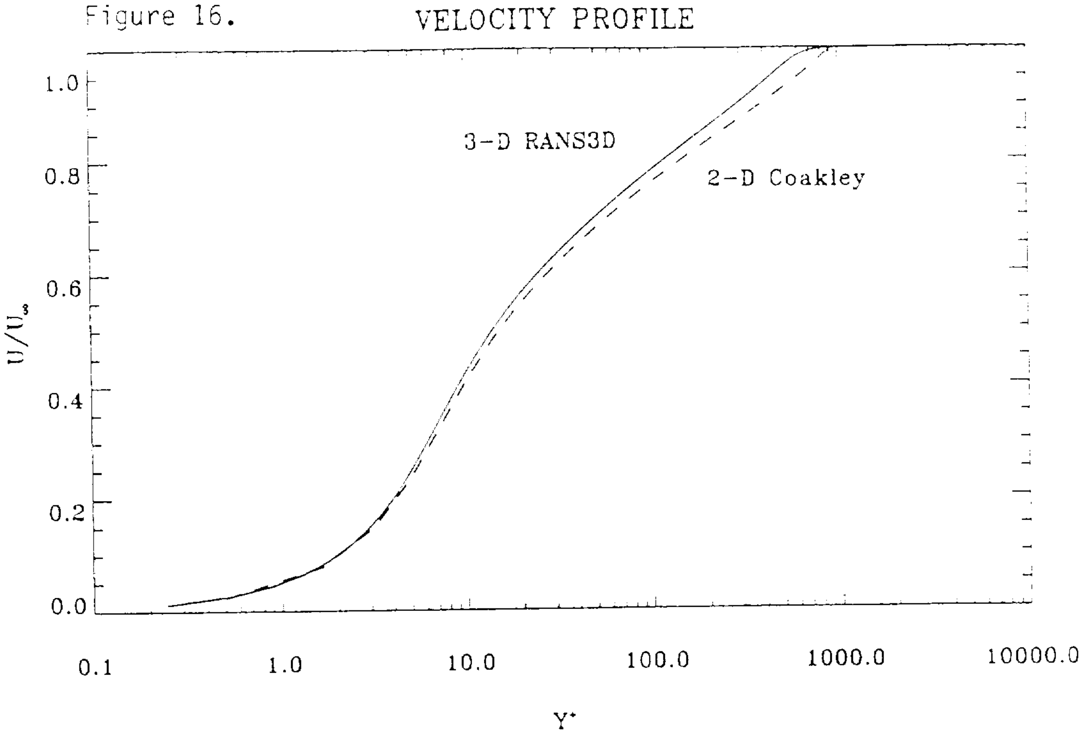


Figure 16.


VELOCITY PROFILE
3-D RANS3D
2-D Coakley
U/U_∞
1.0
0.8
0.6
0.4
0.2
0.0
0.1
1.0
10.0
100.0
1000.0
10000.0
Y^+

APPENDIX A

APPENDIX A. METHODS

		Page
I.	CONSERVATION LAWS	50
Ia.	CARTESIAN COORDINATES	50
Ib.	GENERAL CURVILINEAR COORDINATES	50
Ic.	GALILEAN TRANSFORMATION PROPERTY	51
II.	FLUX-VECTOR SPLITTING	52
IIa.	STEGER AND WARMING METHOD	52
IIb.	VAN LEER METHOD	55
III.	FLUX-DIFFERENCE SPLITTING	56
IIIa.	ROE METHOD	59
IIIb.	CSCM METHOD	61
IIIc.	RANS METHOD	62

I. CONSERVATION LAWS

Ia. CARTESIAN COORDINATES

The three-dimensional conservation laws of mass, momentum, and energy for the inviscid equations of gasdynamics can be written in the strong conservation form as

$$\frac{\partial U}{\partial t} + \frac{\partial F_x(U)}{\partial x} + \frac{\partial F_y(U)}{\partial y} + \frac{\partial F_z(U)}{\partial z} = 0 \tag{1.1}$$

where the conservative variables U are

$$U = \begin{pmatrix} U_1 \\ U_2 \\ U_3 \\ U_4 \\ U_5 \end{pmatrix} = \begin{pmatrix} \rho \\ \rho u \\ \rho v \\ \rho w \\ e \end{pmatrix} \tag{1.2}$$

and the flux vectors F_x, F_y, F_z are defined as

$$F_x = \begin{pmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ \rho uw \\ u(e+p) \end{pmatrix} \quad F_y = \begin{pmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ \rho vw \\ v(e+p) \end{pmatrix} \quad F_z = \begin{pmatrix} \rho w \\ \rho uw \\ \rho vw \\ \rho w^2 + p \\ w(e+p) \end{pmatrix} \tag{1.3}$$

The primitive variables are the density ρ, the three velocity components (u, v, w) and the static pressure p. For a perfect gas, the total energy per unit volume, e, is related to the primitive variables according to the equation of state which for a perfect gas leads to

$$e = p/(\gamma - 1) + \rho(u^2 + v^2 + w^2)/2 \tag{1.4}$$

where γ is the ratio of specific heat.

Ib. GENERAL CURVILINEAR COORDINATES

The transformation of the conservation laws (eq. 1.1) from (x, y, z) Cartesian coordinates to (ξ, η, ψ) general curvilinear coordinates is written in strong conservation form as

$$\frac{\partial JU}{\partial t} + \frac{\partial F_1(U)}{\partial \xi} + \frac{\partial F_2(U)}{\partial \eta} + \frac{\partial F_3(U)}{\partial \psi} = 0 \tag{1.5}$$

with the flux vectors F_1, F_2, F_3 defined as

$$\begin{pmatrix} F_1 \\ F_2 \\ F_3 \end{pmatrix} = J \begin{pmatrix} \xi_x & \xi_y & \xi_z \\ \eta_x & \eta_y & \eta_z \\ \psi_x & \psi_y & \psi_z \end{pmatrix} \begin{pmatrix} F_x \\ F_y \\ F_z \end{pmatrix} \tag{1.6}$$

and the transformation matrix has the following property

$$\begin{pmatrix} \xi_x & \xi_y & \xi_z \\ \eta_x & \eta_y & \eta_z \\ \psi_x & \psi_y & \psi_z \end{pmatrix} \begin{pmatrix} x_\xi & y_\xi & z_\xi \\ x_\eta & y_\eta & z_\eta \\ x_\psi & y_\psi & z_\psi \end{pmatrix} = I \tag{1.7}$$

where I is the identity matrix and J is the Jacobian of the coordinate transformation matrix.

$$J = x_\xi(y_\eta z_\psi - y_\psi z_\eta) + x_\eta(y_\psi z_\xi - y_\xi z_\psi) + x_\psi(y_\xi z_\eta - y_\eta z_\xi) \tag{1.8}$$

where the subindex notation of the matrix coefficients means partial differentiation with respect to the subindex variable.

Ic. GALILEAN TRANSFORMATION PROPERTY

Since the Euler equations are invariant under a Galilean transformation, it is sufficient to consider

$$\frac{\partial U}{\partial t} + \frac{\partial F(U)}{\partial x} = 0 \tag{1.9}$$

as the one-dimensional problem associated with the flux-vector splitting and the flux-difference vector splitting methods. The one-dimensional conservative variables U and flux vector F are defined as

$$U = \begin{pmatrix} U_1 \\ U_2 \\ U_3 \end{pmatrix} = \begin{pmatrix} \rho \\ \rho u \\ e \end{pmatrix} \quad F = \begin{pmatrix} \rho u \\ \rho u^2 + p \\ u(e+p) \end{pmatrix} \tag{1.10}$$

In the 1-D case, the primitive variables are the density ρ, the velocity u and the static pressure p. For a perfect gas, the total energy per unit volume, e, is related to the primitive variables according to the equation of state as

$$e = p/(\gamma - 1) + \rho(u^2)/2 \tag{1.11}$$

II. FLUX-VECTOR SPLITTING

IIa. STEGER AND WARMING METHOD

Steger and Warming[1] have proposed the following flux-vector splitting method. Since the flux vector F is an homogeneous function of degree one in U, it can be expressed in terms of its Jacobian matrix A as

$$F = \frac{\partial F}{\partial U} U = AU \tag{2.1}$$

where F can be rewritten as

$$F = \begin{pmatrix} \rho u \\ \rho u^2 + p \\ u(e + p) \end{pmatrix} = \begin{pmatrix} \rho u \\ (3-\gamma)(\rho u)^2/2\rho + e/(\gamma - 1) \\ \gamma e(\rho u)/\rho - (\gamma - 1)(\rho u)^3/2\rho^2 \end{pmatrix} \tag{2.2}$$

or as a function of the conservative variables as

$$F = \begin{pmatrix} U_2 \\ (3-\gamma){U_2}^2/2U_1 + U_3/(\gamma - 1) \\ \gamma U_3 U_2/U_1 - (\gamma - 1){U_2}^3/2{U_1}^2 \end{pmatrix} \tag{2.3}$$

and the Jacobian matrix A becomes

$$A = \frac{\partial F}{\partial U} = \begin{pmatrix} 0 & 1 & 0 \\ -(3-\gamma)u^2/2 & (3-\gamma)u & (\gamma - 1) \\ -uH + (\gamma-1)u^3/2 & H - (\gamma - 1)u^2 & \gamma u \end{pmatrix} \tag{2.4}$$

and the total enthalpy per unit mass, H, is

$$H = (\epsilon + p)/\rho = \gamma e/\rho - (\gamma - 1)u^2/2 \tag{2.5a}$$

$$H = \gamma p/\rho(\gamma - 1) + u^2/2 = c^2/(\gamma - 1) + u^2/2 \tag{2.5b}$$

The flux vector F can be rewritten using a similarity transformation as

$$F = AU = S\Lambda S^{-1}U \tag{2.6}$$

where Λ is a diagonal matrix whose coefficients are the eigenvalues of the Jacobian matrix A,

$$diag\Lambda = (u, u + c, u - c) \tag{2.7}$$

$$c = \sqrt{\gamma p/\rho} \tag{2.8}$$

and S is the column-eigenvector matrix

$$S = \begin{pmatrix} 1 & 0 & 0 \\ u & 1 & 0 \\ u^2/2 & u & 1 \end{pmatrix} \begin{pmatrix} 1 & 1 & 1 \\ 0 & c & -c \\ 0 & c^2/(\gamma - 1) & c^2/(\gamma - 1) \end{pmatrix} \tag{2.9a}$$

$$S^{-1} = \begin{pmatrix} 1 & 0 & -(\gamma - 1)/c^2 \\ 0 & 1/2c & (\gamma - 1)/2c^2 \\ 0 & -1/2c & (\gamma - 1)/2c^2 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ -u & 1 & 0 \\ u^2/2 & -u & 1 \end{pmatrix} \tag{2.9b}$$

The flux vector splitting is defined by the decomposition of the eigenvalue matrix Λ into a non-negative and a non-positive diagonal matrix,

$$\Lambda = \Lambda^+ + \Lambda^- \tag{2.10a}$$

$$\Lambda^+ = (\Lambda + \mid \Lambda \mid)/2 \tag{2.10b}$$

$$\Lambda^- = (\Lambda - \mid \Lambda \mid)/2 \tag{2.10c}$$

where $\mid \Lambda \mid$ is defined as a matrix whose coefficients are equal to the absolute value of the corresponding coefficients of the matrix Λ. Thus, Λ^+ includes only the non-negative eigenvalues of Λ and Λ^- includes only the non-positive eigenvalues of Λ, respectively. Following this splitting method, the flux vector F is rewritten as

$$F = S(\Lambda^+ + \Lambda^-)S^{-1}U = (A^+ + A^-)U = F^+ + F^- \tag{2.11}$$

The spatial derivatives of F^+ and F^- are usually approximated with the standard implicit first- and second-order backward and forward difference operators, respectively. Flux limiters are introduced in the higher-order difference operators to eliminate oscillations in the presence of discontinuities, such as shock waves. The Euler equations (1.8) become

$$\frac{\partial U}{\partial t} + \frac{\partial F^+}{\partial x} + \frac{\partial F^-}{\partial x} = 0 \tag{2.12}$$

and the difference operators are

$$\delta_x^- F_i^+ = (F_i^+ - F_{i-1}^+)/\delta x + \phi_i^-(F_i^+ - 2F_{i-1}^+ + F_{i-2}^+)/2\delta x \tag{2.13}$$

$$\delta_x^+ F_i^- = (F_{i+1}^- - F_i^-)/\delta x - \phi_i^+(F_i^- - 2F_{i+1}^- + F_{i+2}^-)/2\delta x \tag{2.14}$$

with first-order when $\phi = 0$ and second-order when $\phi = 1$. Other higher-order difference operators are expressed in a similar form, such as Fromm scheme and third-order biased scheme. Equations (2.13) and (2.14) can also be written as a extrapolation at cell interfaces

$$\delta_x^- F_i^+ = (F_{i+1/2}^+ - F_{i-1/2}^+)/\delta x \tag{2.15}$$

$$\delta_x^+ F_i^- = (F_{i+1/2}^- - F_{i-1/2}^-)/\delta x \tag{2.16}$$

where

$$F_{i+1/2}^+ = F_i^+ + \phi_{i+1/2}^-(F_i^+ - F_{i-1}^+)/2 \tag{2.17}$$

$$F_{i-1/2}^- = F_i^- + \phi_{i-1/2}^+(F_i^- - F_{i+1}^-)/2 \tag{2.18}$$

with the appropriate definition of ϕ at cell interfaces.

An alternative approach proposed by van Leer[2] is the MUSCL method. In this method, the variables U are extrapolated toward the interfaces and then the flux vectors F^+ and F^- are evaluated. Therefore

$$\delta_x F_i = (\quad F^+(U_{i+1/2}^-) - F^+(U_{i-1/2}^-) + F^-(U_{i+1/2}^+) - F^-(U_{i-1/2}^+) \quad)/\delta x \tag{2.19}$$

and

$$U_{i-1/2}^+ = U_i + \phi_i^+(U_{i+1} - U_i)/2 \tag{2.20}$$

$$U_{i+1/2}^- = U_i + \phi_i^-(U_i - U_{i-1})/2 \tag{2.21}$$

IIb. VAN LEER METHOD

Although the flux vector F is a continuous differentiable function, the fluxes split F^+ and F^- of Steger and Warming described in the above previous section are not differentiable at zeros of the eigenvalues. A method proposed by van Leer defines the split fluxes as continuous differentiable functions including sonic and stagnation points. The flux vector F expressed as a function of density ρ, sound speed c, and Mach number $M = u/c$ becomes

$$F = \begin{pmatrix} \rho c M \\ \rho c^2 \left(M^2 + 1/\gamma\right) \\ \rho c^3 M \left(M^2/2 + 1/(\gamma - 1)\right) \end{pmatrix} \tag{2.22}$$

The formulas for the flux split given in terms of the Mach number are for supersonic flows, $\mid M \mid \geq 1$,

$$F^+ = F \qquad F^- = 0 \qquad \text{for } M \geq 1 \tag{2.23}$$

$$F^+ = 0 \qquad F^- = F \qquad \text{for } M \leq -1 \tag{2.24}$$

and for subsonic flow, $\mid M \mid \leq 1$, the flux vectors are defined as

$$F^{\pm} = \begin{pmatrix} \pm\rho c((M \pm 1)/2)^2 \\ \pm\rho c^2((M \pm 1)/2)^2((\gamma - 1)M \pm 2)/\gamma \\ \pm\rho c^3((M \pm 1)/2)^2((\gamma - 1)M \pm 2)^2/2(\gamma^2 - 1) \end{pmatrix} \tag{2.25}$$

III. FLUX-DIFFERENCE SPLITTING

The one-dimensional conservation laws of mass, momentum, and energy for the inviscid gasdynamics equations can be written in a discrete space as

$$\delta_t U = -\Delta t \; \delta_x F(U) / \Delta x \tag{3.1}$$

There are significant differences in the numerical methods between most finite volume and finite difference approaches. One of these differences is the location of the cell areas.

- Finite Volume method.

In a finite volume approach, the flux difference splitting are defined as follows

$$\delta_x F_i = \delta_x F_i^+ + \delta_x F_i^- \tag{3.2}$$

where $F_i \equiv F(U_i)$ and the flux differences are

$$\delta_x F_i^+ = F_{i+\frac{1}{2}}^+ - F_{i-\frac{1}{2}}^+ \qquad \text{and} \qquad \delta_x F_i^- = F_{i+\frac{1}{2}}^- - F_{i-\frac{1}{2}}^- \tag{3.3}$$

and

$$\Delta x = (x_{i+1} - x_{i-1})/2 \tag{3.4}$$

The flux difference may be rewritten also as

$$\delta F_i = h_{i+\frac{1}{2}} - h_{i-\frac{1}{2}} \tag{3.5}$$

- Finite Difference method.

In a finite difference approach, the flux difference splitting are defined as

$$\frac{\delta_x F}{\Delta x} = \frac{\delta_x F^+}{\Delta x^+} + \frac{\delta_x F^-}{\Delta x^-} \tag{3.6}$$

and a first-order upwind method gives the following equations

$$\delta_x F_i^+ = F_i^+ - F_{i-1}^+ \tag{3.7}$$

$$\delta_x F_i^- = F_{i+1}^- - F_i^- \tag{3.8}$$

$$\Delta_x^+ = x_i - x_{i-1} \tag{3.9}$$

$$\Delta_x^- = x_{i+1} - x_i \tag{3.10}$$

In both methods, the fluxes are split in delta form based on a Jacobian matrix A which follows 'Roe's property U,'

$$\delta F \equiv A\,\delta U \tag{3.11}$$

Under a similarity transformation, the matrix A is expressed as

$$A = S\,\Lambda\,S^{-1} \tag{3.12}$$

where Λ is a diagonal matrix with the eigenvalues of the matrix A as diagonal coefficients, the matrix S is composed by the eigenvectors of the matrix A. The matrix $S^{-1}\,\delta U$ is also the expression of the characteristic variable representation of the conservative variables U. The positive and negative flux differences are defined according to the sign of the eigenvalues as

$$\delta F_i^+ = A^+\,(U_i - U_{i-1}) \qquad \text{and} \qquad \delta F_i^- = A^-\,(U_{i+1} - U_i) \tag{3.13}$$

where

$$A^+ = S\,\Lambda^+\,S^{-1} \qquad \text{and} \qquad A^- = S\,\Lambda^-\,S^{-1} \tag{3.14}$$

$$\Lambda^+ = (\Lambda + \mid \Lambda \mid)/2 \qquad \text{and} \qquad \Lambda^- = (\Lambda - \mid \Lambda \mid)/2 \tag{3.15}$$

In both methods, finite volume and finite difference, the flux differences may be combined into a equivalent expression as (H.C. Yee, NASA TM 101088)

$$\delta F_i = h_{i+\frac{1}{2}} - h_{i-\frac{1}{2}} \tag{3.19}$$

where

$$h_{i+\frac{1}{2}} = (F_{i+1} + F_i - \mid A \mid (U_{i+1} - U_i))/2 \tag{3.20}$$

where $\mid A \mid$ is the Jacobian matrix with the absolute value of its eigenvalue coefficients,

$$\mid A \mid = S \mid \Lambda \mid S^{-1} \tag{3.21}$$

The main differences between the numerical methods are due to the choice of finite volume or finite difference approach, the definition of the Jacobian matrix A, and the implicit operator procedures..

IIIa. ROE METHOD

The Roe's method described here is a finite volume, flux difference splitting method using Roe's averaged variables. This method uses the above defined equations (3.19, 3.20, and 3.21) and it is based on the property that the the flux F is an homogeneous function of degree one of the conservative variables U. Therefore

$$F = AU \tag{3a.1}$$

where A is the Jacobian matrix of the flux vector F. Thus, at each nodal point i

$$F_i = F(U_i) = A_i U_i \tag{3a.2}$$

On the other hand, at the cell interfaces between the cells x_i and x_{i+1}, Roe defined a set of averaged variables to define the coefficients of the Jacobian matrix A subject to the following condition

$$\delta F = F_{i+1} - F_i = A(U_{i+1} - U_i) = A\,\delta U = S\Lambda S^{-1}\delta U \tag{3a.3}$$

The average density is defined as the geometric-averaged value of the nodal points, the averaged velocity components and total enthalpy are defined as weighted-averaged nodal values

$$\rho^\star_{i+\frac{1}{2}} = \sqrt{\rho_{i+1}\rho_i} \tag{3a.4}$$

$$u^\star_{i+\frac{1}{2}} = (Wu_{i+1} + u_i)/(W+1) \tag{3a.5}$$

$$H^\star_{i+\frac{1}{2}} = (WH_{i+1} + H_i)/(W+1) \tag{3a.6}$$

$$c^\star = \sqrt{H^\star - u^{\star 2}/2} \tag{3a.7}$$

$$W = \sqrt{\rho_{i+1}/\rho_i} \tag{3a.8}$$

Thus, the Jacobian matrix becomes

$$A = \frac{\partial F}{\partial U} = \begin{pmatrix} 0 & 1 & 0 \\ -(3-\gamma)u^{\star 2}/2 & (3-\gamma)u^\star & (\gamma - 1) \\ -u^\star H^\star + (\gamma-1)u^{\star 3}/2 & H^\star - (\gamma-1)u^{\star 2} & \gamma u^\star \end{pmatrix} \tag{3a.9}$$

and the column-eigenvector matrix and its inverse are

$$S = \begin{pmatrix} 1 & 0 & 0 \\ u^\star & 1 & 0 \\ u^{\star 2}/2 & u^\star & 1 \end{pmatrix} \begin{pmatrix} 1 & 1 & 1 \\ 0 & c & -c \\ 0 & c^{\star 2}/(\gamma - 1) & c^{\star 2}/(\gamma - 1) \end{pmatrix} \tag{3a.10}$$

$$S^{-1} = \begin{pmatrix} 1 & 0 & -(\gamma - 1)/c^{\star 2} \\ 0 & 1/2c^\star & (\gamma - 1)/2c^{\star 2} \\ 0 & -1/2c^\star & (\gamma - 1)/2c^{\star 2} \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ -u^\star & 1 & 0 \\ u^{\star 2}/2 & -u^\star & 1 \end{pmatrix} \tag{3a.11}$$

and the diagonal matrix Λ is

$$diag \Lambda = (u^\star, u^\star + c^\star, u^\star - c^\star) \tag{3a.12}$$

IIIb. CSCM METHOD

This method is a finite difference, flux difference splitting method using 'Roe's property U'. This method uses the above defined equations (3.6 through 3.18) and it is based on the following property

$$\delta F = A\,\delta U \tag{3b.1}$$

The flux difference splitting is defined based on

$$\delta F^{\pm} = S I^{\pm} S^{-1} A\,\delta U \tag{3b.2}$$

where I is the identity matrix and it is splitted into a non-negative matrix and a non-positive matrix according to the sign of the coefficients

$$diag I^{\pm} = \pm 1 \cdot sign(\bar{u}, \bar{u} + \bar{c}, \bar{u} - \bar{c}) \tag{3b.3}$$

where the overbar denotes arithmetic averaging.

The matrix A is defined as

$$A = \begin{pmatrix} 0 & 1 & 0 \\ (\gamma - 1)(\bar{u}^2 - \frac{\overline{u^2}}{2}) - \bar{u}\frac{\overline{\rho u}}{\bar{\rho}} & (2-\gamma)\bar{u} + \frac{\overline{\rho u}}{\bar{\rho}} & \gamma - 1 \\ \bar{u}(\gamma(\bar{u}^2 - \frac{\overline{u^2}}{2}) - \bar{u}\frac{\overline{\rho u}}{\bar{\rho}} - \frac{\overline{\gamma p}}{\bar{\rho}(\gamma-1)}) & \frac{\overline{u^2}}{2} - \bar{u}(\gamma\bar{u} - \frac{\overline{\rho u}}{\bar{\rho}}) + \frac{\overline{\gamma p}}{\bar{\rho}(\gamma-1)} & \gamma\bar{u} \end{pmatrix} \tag{3b.4}$$

and the matrix S represents a transformation matrix from the conservative variables to characteristic variables,

$$S = \begin{pmatrix} 1 & 0 & 0 \\ \bar{u} & 1 & 0 \\ \overline{u^2}/2 & \bar{u} & 1 \end{pmatrix} \begin{pmatrix} -\bar{\rho} & \bar{\rho}/2 & \bar{\rho}/2 \\ 0 & \bar{\rho}\bar{c}/2 & -\bar{\rho}\bar{c}/2 \\ 0 & \overline{\gamma p}/2(\gamma-1) & \overline{\gamma p}/2(\gamma-1) \end{pmatrix} \tag{3b.5}$$

$$S^{-1} = \begin{pmatrix} -1/\bar{\rho} & 0 & (\gamma-1)/\overline{\gamma p} \\ 0 & 1/\bar{\rho}\bar{c} & (\gamma-1)/\overline{\gamma p} \\ 0 & -1/\bar{\rho}\bar{c} & (\gamma-1)/\overline{\gamma p} \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ -\bar{u} & 1 & 0 \\ \bar{u}^2 - \overline{u^2}/2 & -\bar{u} & 1 \end{pmatrix} \tag{3b.6}$$

IIIc. RANS METHOD

The RANS's method is similar to Roe's flux difference splitting method, however it is based on simple arithmetic averaging of primitive variables. The flux differences are also defined as

$$\delta F = F_{i+1} - F_i = A(U_{i+1} - U_i) = A\,\delta U = S\Lambda S^{-1}\delta U \tag{3c.1}$$

in order to preserve the conservative property of the fluxes. The column-eigenvector matrix S and its inverse are

$$S = \begin{pmatrix} 1 & 0 & 0 \\ \bar{u} & 1 & 0 \\ \overline{u^2}/2 & \bar{u} & 1 \end{pmatrix} \begin{pmatrix} -\bar{\rho} & \bar{\rho}/2 & \bar{\rho}/2 \\ 0 & \bar{\rho}(\tilde{c} - \tilde{d})/2 & -\bar{\rho}(\tilde{c} + \tilde{d})/2 \\ 0 & \overline{\gamma p}/2(\gamma - 1) & \overline{\gamma p}/2(\gamma - 1) \end{pmatrix} \tag{3c.2}$$

$$S^{-1} = \begin{pmatrix} -1/\bar{\rho} & 0 & (\gamma - 1)/\overline{\gamma p} \\ 0 & 1/\bar{\rho}\tilde{c} & (1 + \tilde{d}/\tilde{c})(\gamma - 1)/\overline{\gamma p} \\ 0 & -1/\bar{\rho}\tilde{c} & (1 - \tilde{d}/\tilde{c})(\gamma - 1)/\overline{\gamma p} \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ -\bar{u} & 1 & 0 \\ \bar{u}^2 - \overline{u^2}/2 & -\bar{u} & 1 \end{pmatrix} \tag{3c.3}$$

and the diagonal matrix Λ is defined with the eigenvalues of the matrix A as

$$diag\Lambda = (\bar{u}, \tilde{u} + \tilde{c}, \tilde{u} - \tilde{c}) \tag{3c.4}$$

and the coefficients

$$\tilde{u} = (\bar{u} + \overline{pu}/\bar{p})/2 \tag{3c.5}$$

$$\tilde{d} = (\bar{u} - \overline{pu}/\bar{p})/2 \tag{3c.6}$$

$$\tilde{c} = \sqrt{(\overline{\gamma p}/\bar{\rho}) + \tilde{d}^2} \tag{3c.7}$$

APPENDIX B

APPENDIX B. RANS METHOD

Page

I. CONSERVATION LAWS 64

II. FLUX-DIFFERENCES 66

III. HIGHER-ORDER TVD FLUXES 69

IV. IMPLICIT NUMERICAL METHOD 70

RANS METHOD

I. CONSERVATION LAWS

The conservation laws of mass, momentum, and energy for the 3-D Navier-Stokes equations can be expressed as

$$\frac{\partial U}{\partial t} + \frac{\partial F_1}{\partial x_1} + \frac{\partial F_2}{\partial x_2} + \frac{\partial F_3}{\partial x_3} = 0 \qquad (1a)$$

or in compressed vector notation (where repeated sub-indices in any term imply summation over the index range, sub-index t following a comma imply partial differentiation with respect to time, and sub-index numbers following a comma imply partial differentiation with respect to the respective spatial coordinate directions) as

$$U_{,t} + F_{j,j} = 0 \qquad (1b)$$

The vector U represents the conservative dependent variables,

$$U = (\rho, \rho u_1, \rho u_2, \rho u_3, e) \qquad (2)$$

and F_j are the flux vectors in the respective Cartesian coordinates direction x_j

$$\begin{aligned} F_j =& (\rho u_j, \rho u_1 u_j + p\delta_{1j} - \tau_{1j}, \rho u_2 u_j + p\delta_{2j} - \tau_{2j}, \\ & \rho u_3 u_j + p\delta_{3j} - \tau_{3j}, (e+p)u_j - u_i\tau_{ij} - \kappa T_{,j}) \end{aligned} \qquad (3)$$

ρ is the fluid density, u_j are the velocity components in each coordinate direction, p is the static pressure, κ is the thermal conductivity, T is the fluid temperature, and e is the total energy.

The viscous stress tensor τ_{ij} including the turbulent eddy viscosity is defined as

$$\tau_{ij} = 2(\mu + \mu_T)(S_{ij} - \delta_{ij}S_{nn}/3) \qquad (4)$$

where δ_{ij} is the second-order isotropic Kronecker delta.

For a perfect gas, the pressure is related to the total energy as

$$e = p/(\gamma - 1) + \rho u_j u_j/2 \tag{5}$$

where γ is the ratio of the specific heats at constant pressure and volume, and the temperature is related to the density and pressure through the equation of state

$$p = R\rho T \tag{6}$$

The conservation laws written in generalized curvilinear coordinates $\xi_i = \xi_i(x_j)$ are

$$U_{,t} + \xi_{i,j} F_{j,\tilde{i}} = 0 \tag{7}$$

where $\tilde{i}$ represents the curvilinear coordinate direction ξ_i. The conservation laws can be rewritten in weakly-conservative form as

$$(JU)_{,t} + \overline{J\xi_{i,j}}\, F_{j,\tilde{i}} = 0 \tag{8}$$

or in strongly-conservation form as

$$(JU)_{,t} + (J\xi_{i,j}F_j)_{,\tilde{i}} = 0 \tag{9}$$

where J is the Jacobian of the coordinate transformation

$$J = \epsilon_{ijk}\, x_{1,\tilde{i}}\, x_{2,\tilde{j}}\, x_{3,\tilde{k}} \tag{10}$$

and ϵ_{ijk} is the third-order isotropic alternating tensor. The metric coefficients are evaluated at the interface between the grid points of the spatial difference, according to their mathematical definition as

$$J\xi_{i,s} = 0.5\, \epsilon_{ijk}\, \epsilon_{snm}\, x_{n,\tilde{j}}\, x_{m,\tilde{k}} \tag{11}$$

The present version of the **RANS** code uses the weakly-conservative form shown in the above equation 8 . Different tests and experiments in subsonic, transonic, and supersonic flows have shown that the conservation properties are maintained once convergence is achieved with a second- or higher-order scheme. An strongly-conservative version of the code is in progress.

II. FLUX-DIFFERENCES

All flux differences are treated implicitly in order to increase stability and to be able to use large increments of time or CFL numbers. The numerical scheme for the viscous fluxes is second-order central difference, while the numerical scheme for the inviscid fluxes is higher-order TVD upwind flux-difference splitting. The higher-order TVD scheme is based on the Osher-Chakravarthy "*minmod*" method applied to the flux differences; it has the capability of to represent various higher-order differences: first-order upwind, second-order upwind, third-order upwind biased, second-order Fromm scheme, and other combinations of upwind and central differences.

The differentiation of the inviscid fluxes is similar to Roe's flux difference splitting method, however it is based on simple arithmetic averaging of primitive variables. Each flux difference term $F_{j,\tilde{i}}$ of the conservation law equations is defined as

$$\delta F = F_{\tilde{i}+1} - F_{\tilde{i}} = A\,(U_{\tilde{i}+1} - U_{\tilde{i}}) = A\,\delta U \qquad (12)$$

in order to preserve the conservative property of the fluxes (the subindex j is omitted in equation 12 for simplicity). The Jacobian matrix A is decomposed using a similarity transformation as

$$A = S\Lambda S^{-1} \qquad (13)$$

where Λ is a diagonal matrix whose coefficients are the eigenvalues of the matrix A, and S is the column-eigenvector matrix of A. The matrix S is further decomposed in two matrices

$$S = MT \qquad (14)$$

where M represents the transformation matrix of conservative and primitive variable differences, and T represents the transformation matrix between characteristic and primitive variable differences. For the flux differences in the ξ_1 coordinate direction, these matrices are defined as

$$M = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ \bar{u}_1 & 1 & 0 & 0 & 0 \\ \bar{u}_2 & 0 & 1 & 0 & 0 \\ \bar{u}_3 & 0 & 0 & 1 & 0 \\ \overline{u_i u_i}/2 & \bar{u}_1 & \bar{u}_2 & \bar{u}_3 & 1 \end{pmatrix} \tag{15a}$$

$$M^{-1} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ -\bar{u}_1 & 1 & 0 & 0 & 0 \\ -\bar{u}_2 & 0 & 1 & 0 & 0 \\ -\bar{u}_3 & 0 & 0 & 1 & 0 \\ \bar{u}_i \bar{u}_i - \overline{u_i u_i}/2 & -\bar{u}_1 & -\bar{u}_2 & -\bar{u}_3 & 1 \end{pmatrix} \tag{15b}$$

$$T = \begin{pmatrix} -\bar{\rho} & 0 & 0 & \bar{\rho}/2 & \bar{\rho}/2 \\ 0 & \overline{x}'_{1,\tilde{2}} & \overline{x}'_{1,\tilde{3}} & \overline{\xi}'_{1,1}\bar{\rho}(\tilde{c} - \tilde{d})/2 & -\overline{\xi}'_{1,1}\bar{\rho}(\tilde{c} + \tilde{d})/2 \\ 0 & \overline{x}'_{2,\tilde{2}} & \overline{x}'_{2,\tilde{3}} & \overline{\xi}'_{1,2}\bar{\rho}(\tilde{c} - \tilde{d})/2 & -\overline{\xi}'_{1,2}\bar{\rho}(\tilde{c} + \tilde{d})/2 \\ 0 & \overline{x}'_{3,\tilde{2}} & \overline{x}'_{3,\tilde{3}} & \overline{\xi}'_{1,3}\bar{\rho}(\tilde{c} - \tilde{d})/2 & -\overline{\xi}'_{1,3}\bar{\rho}(\tilde{c} + \tilde{d})/2 \\ 0 & 0 & 0 & \overline{\gamma p}/2(\gamma - 1) & \overline{\gamma p}/2(\gamma - 1) \end{pmatrix} \tag{16b}$$

$$T^{-1} = \begin{pmatrix} -1/\bar{\rho} & 0 & 0 & 0 & (\gamma - 1)/\overline{\gamma p} \\ 0 & \hat{x}'_{1,\tilde{2}} & \hat{x}'_{2,\tilde{2}} & \hat{x}'_{3,\tilde{2}} & 0 \\ 0 & \hat{x}'_{1,\tilde{3}} & \hat{x}'_{2,\tilde{3}} & \hat{x}'_{3,\tilde{3}} & 0 \\ 0 & \overline{\xi}'_{1,1}/\bar{\rho}\tilde{c} & \overline{\xi}'_{1,2}/\bar{\rho}\tilde{c} & \overline{\xi}'_{1,3}/\bar{\rho}\tilde{c} & (1 + \tilde{d}/\tilde{c})(\gamma - 1)/\overline{\gamma p} \\ 0 & -\overline{\xi}'_{1,1}/\bar{\rho}\tilde{c} & -\overline{\xi}'_{1,2}/\bar{\rho}\tilde{c} & -\overline{\xi}'_{1,3}/\bar{\rho}\tilde{c} & (1 - \tilde{d}/\tilde{c})(\gamma - 1)/\overline{\gamma p} \end{pmatrix} \tag{16b}$$

where the arithmetic averaging is denoted by the overbar symbol. Although the density $\bar{\rho}$ could be factored in the T and T^{-1} transformation matrices, as it has been pointed out by T.J. Coakley, the density factor has been introduced in order to write the characteristic variables in nondimensional form. The prime symbol ($\overline{x}'$) is used to denote unity magnitude of the vector, thus

$$\overline{x}'_{i,\tilde{j}} = \overline{x}_{i,\tilde{j}}/|\xi_j| \tag{17a}$$

$$\overline{\xi}'_{i,j} = \overline{\xi}_{i,j}/|\xi_j| \tag{17b}$$

$$|\xi_j| = \sqrt{\xi^2_{j,1} + \xi^2_{j,2} + \xi^2_{j,3}} \tag{17c}$$

with no summation of repeated indices in the above equation (17). The hat symbol ($\hat{x}$) is used to denote the inverse matrix coefficients of the non-orthogonal matrices.

$$\hat{x}'_{i,\tilde{2}} = (\overline{x}'_{1,\tilde{2}} - \overline{x}'_{1,\tilde{3}}(\overline{x}_{i,\tilde{2}} \cdot \overline{x}_{i,\tilde{3}}))(|\xi_2||\xi_3|/J|\xi_1|)^2 \tag{18a}$$

$$= (\overline{x}'_{1,\tilde{2}} - \overline{x}'_{1,\tilde{3}}(\overline{x}_{i,\tilde{2}} \cdot \overline{x}_{i,\tilde{3}}))(1 - (\overline{x}_{i,\tilde{2}} \cdot \overline{x}_{i,\tilde{3}})^2) \tag{18b}$$

$$= (\overline{x}'_{2,\tilde{3}}\overline{\xi}'_{1,3} - \overline{x}'_{3,\tilde{3}}\overline{\xi}'_{1,2})(|\xi_2||\xi_3|/J|\xi_1|) \tag{18c}$$

which are all equivalent expressions.

The diagonal coefficients of the eigenvalue matrix Λ for the flux differences in the ξ_1 coordinate direction are

$$\text{diag}\Lambda = (\overline{\xi}_{1,j}\bar{u}_j, \overline{\xi}_{1,j}\tilde{u}_j, \overline{\xi}_{1,j}\tilde{u}_j, 0.5\overline{\xi}_{1,j}(\bar{u}_j + \tilde{u}_j) + \tilde{c}, 0.5\overline{\xi}_{1,j}(\bar{u}_j + \tilde{u}_j) - \tilde{c}) \tag{19}$$

where

$$\tilde{c} = \sqrt{\overline{\xi}^2_{1,j}\,\overline{\gamma p}/\bar{\rho} + \tilde{d}^2} \tag{20}$$

$$\tilde{d} = \overline{\xi}_{1,j}\widetilde{\delta u}_j \tag{21}$$

$$\widetilde{\delta u}_j = (\bar{u}_j - \overline{\rho u}_j/\bar{\rho})/2 \tag{22}$$

$$\tilde{u}_j = (\bar{u}_j + \overline{\rho u}_j/\bar{\rho})/2 \tag{23}$$

The inviscid first-order flux differences terms (see equation 12) are split according to the sign of the eigenvalues as

$$\delta F = \nabla F^+ + \Delta F^- \tag{24a}$$

$$\nabla F^+ = A^+\,\nabla U = S\Lambda^+S^{-1}\,\nabla U \tag{24b}$$

$$\Delta F^- = A^-\,\Delta U = S\Lambda^-S^{-1}\,\Delta U \tag{24c}$$

where ∇ is the backward-difference operator and Δ is the forward-difference operator. The coefficients of the diagonal matrices $\Lambda^{\pm}$ are defined as

$$\lambda^+ = (\lambda + |\lambda|)/2 \tag{25a}$$

$$\lambda^- = (\lambda - |\lambda|)/2 \tag{25b}$$

Unconditionally stable Euler implicit methods are constructed by forward upwind differencing flux differences with positive eigenvalues and backward differencing flux differences with negative eigenvalues.

III. HIGHER-ORDER TVD FLUXES

Higher-order spatial TVD flux differences in the right-hand-side of the inviscid terms of the conservation law equations are defined by using the "*minmod*" limiter of Osher and Chakravarthy applied on the complete split flux difference instead of the characteristic variable difference. For a given ξ_i direction and at the j grid point,

$$\begin{aligned}
\delta_j F &= \Delta_{j-\frac{1}{2}} F^+ + \Delta_{j+\frac{1}{2}} F^- \\
&+ c \Big[+ \frac{1-\phi}{4} \Big((\Delta_{j+\frac{1}{2}} F^- - \widetilde{\Delta_{j+\frac{3}{2}}} F^-) + (\Delta_{j-\frac{1}{2}} F^+ - \widetilde{\widetilde{\Delta_{j-\frac{3}{2}}}} F^+) \Big) \\
&\quad + \frac{1+\phi}{4} \Big((\widetilde{\Delta_{j+\frac{1}{2}}} F^+ - \Delta_{j-\frac{1}{2}} F^+) + (\widetilde{\widetilde{\Delta_{j-\frac{1}{2}}}} F^- - \Delta_{j+\frac{1}{2}} F^-) \Big) \Big] \qquad (26)
\end{aligned}$$

where the first-order terms, the second-order upwind-difference increment terms, and the second-order central difference increment terms are shown in the first, second, and third line of the above equation 26, respectively. $0 \leq c \leq 1$ is an input factor that allows to change from first-order to higher-order differences monotonically. $\widetilde{\Delta F}$ and $\widetilde{\widetilde{\Delta F}}$ are the limited split flux differences defined by

$$\begin{aligned}
\widetilde{\Delta_{j+\frac{1}{2}}} F^+ &= minmod(\Delta_{j+\frac{1}{2}} F^+, \beta \Delta_{j-\frac{1}{2}} F^+) \\
\widetilde{\Delta_{j+\frac{3}{2}}} F^- &= minmod(\Delta_{j+\frac{3}{2}} F^-, \beta \Delta_{j+\frac{1}{2}} F^-) \\
\widetilde{\widetilde{\Delta_{j-\frac{3}{2}}}} F^+) &= minmod(\Delta_{j-\frac{3}{2}} F^+, \beta \Delta_{j-\frac{1}{2}} F^+) \\
\widetilde{\widetilde{\Delta_{j-\frac{1}{2}}}} F^-) &= minmod(\Delta_{j-\frac{1}{2}} F^-, \beta \Delta_{j+\frac{1}{2}} F^-) \qquad (27)
\end{aligned}$$

The *minmod* function is defined as

$$minmod(a, b) = sign(a)\, max(0, min(|a|, \, b sign(a))) \qquad (28)$$

and returns a value of zero if the product $ab \le 0$, or a value equal to the argument with the minimum magnitude if both arguments a and b have the same sign. β is a compression parameter determined in the range given by

$$1 \le \beta \le \frac{3-\phi}{1-\phi} \tag{29}$$

The maximum value of β is used because it introduces less dissipation. The numerical scheme is second-order upwind when $\phi = -1$ ($\beta = 2$), Fromm's scheme when $\phi = 0$ ($\beta = 3$), third-order upwind biased when $\phi = 1/3$ ($\beta = 4$), and second-order central-difference when $\phi = 1$ ($\beta \to \infty$).

For example, the second-order flux-difference split upwind scheme with $c = 1$ is

$$\begin{aligned} \delta_j F =& \frac{3}{2}\Delta_{j-\frac{1}{2}}F^+ - \frac{1}{2}\widetilde{\widetilde{\Delta_{j-\frac{3}{2}}F^+}} \\ &+ \frac{3}{2}\Delta_{j+\frac{1}{2}}F^- - \frac{1}{2}\widetilde{\Delta_{j+\frac{3}{2}}F^-} \end{aligned} \tag{30}$$

where the *minmod* function

$$0 \le \widetilde{\widetilde{\Delta_{j-\frac{3}{2}}F^+}} = minmod(\Delta_{j-\frac{3}{2}}F^+, 2\Delta_{j-\frac{1}{2}}F^+) \le 2\Delta_{j-\frac{1}{2}}F^+ \tag{31a}$$

$$0 \ge \widetilde{\Delta_{j+\frac{3}{2}}F^-} = minmod(\Delta_{j+\frac{3}{2}}F^-, 2\Delta_{j+\frac{1}{2}}F^-) \ge 2\Delta_{j+\frac{1}{2}}F^- \tag{31b}$$

Thus, the TVD scheme limits the split fluxes with a factor between 0.5 and 1.5 times the first-order flux-difference,

$$\frac{1}{2}\Delta_{j-\frac{1}{2}}F^+ \le \frac{3}{2}\Delta_{j-\frac{1}{2}}F^+ - \frac{1}{2}\widetilde{\widetilde{\Delta_{j-\frac{3}{2}}F^+}} \le \frac{3}{2}\Delta_{j-\frac{1}{2}}F^+ \tag{32a}$$

$$\frac{1}{2}\Delta_{j+\frac{1}{2}}F^- \ge \frac{3}{2}\Delta_{j+\frac{1}{2}}F^- - \frac{1}{2}\widetilde{\Delta_{j+\frac{3}{2}}F^-} \ge \frac{3}{2}\Delta_{j+\frac{1}{2}}F^- \tag{32b}$$

IV. IMPLICIT NUMERICAL METHOD

The numerical method is based on an implicit "method of planes" symmetric Gauss-Seidell relaxation scheme. The data is conveniently stored on successive planes along the streamwise coordinate, and the system of equations is solved in each successive plane along the forward direction, first, and along the backward direction, afterwards. In each plane, the solution is obtained by using a two level pseudo time dependent relaxation procedure based on a diagonally dominant approximate factorization DDADI. The space marching alternating directional sweeps in the streamwise direction are von Neumann unconditionally stable for zones of subsonic and streamwise separated and reversed flows as well as supersonic flow. The space marching method results in improved propagation of nonlinear effects to accelerate convergence to steady state, much as do the more restrictive PNS techniques.

The diagonal dominant approximate factorization of the left-hand-side of the conservation law equations including the implicit viscous terms leads to the following block tridiagonal equation sequence for the ξ_1 plane relaxation method

$$(-A^+_{\xi_2},\, D,\, A^-_{\xi_2})\,\delta U^* = -RHS^{n,n+1} \tag{33a}$$

$$(-A^+_{\xi_3},\, D,\, A^-_{\xi_3})\,\delta U \;\; = -D\,\delta U^* \tag{33b}$$

The diagonally dominant matrix D involves the first-order split Jacobian matrices and the Jacobian matrices of the viscous terms of each coordinate direction

$$D = I + A^+_{\xi_1} - A^-_{\xi_1} + A^+_{\xi_2} - A^-_{\xi_2} + A^+_{\xi_3} - A^-_{\xi_3} \tag{34}$$

and the solution is updated from time step n to time step $n+1$

$$U^{n+1} = U^n + \delta U^n \tag{35}$$

Observe that the RHS of equation 33a has an exponent $n, n+1$ because some terms in the streamwise direction are already updated at time step $n+1$ due to the plane relaxation procedure.

A Newton-Raphson acceleration procedure is obtained by solving each plane twice or more times in each relaxation sequence. This procedure produces significant improvement in the propagation of nonlinear effects due to the nonlinearity of the Jacobian matrix A.