FINAL REPORT

FLOWFIELD VISUALIZATION FOR SSME HOT GAS MANIFOLD

Contract NAS8-36090

15 July 1988

Prepared for

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION
MARSHALL SPACE FLIGHT CENTER, AL 35812

by

Robert P. Roger

Lockheed
Missiles & Space Company, Inc
Huntsville Engineering Center
4800 Bradford Blvd., Huntsville, AL 35807

FOREWORD

This final report was prepared by personnel of the Computational Mechanics Section of Lockheed's Huntsville Engineering Center. It constitutes final documentation of efforts performed under Contract NAS8-36090 for NASA-Marshall Space Flight Center.

The NASA-MSFC Contracting Officer's Representative for this research study was Dr. P.K. McConnaughey, ED32.

CONTENTS

LIST OF FIGURES

LIST OF FIGURES (Continued)

LIST OF FIGURES (Concluded)

1. INTRODUCTION

The objective of this research effort, as defined by NASA-Marshall Space Flight Center was two-fold: (1) to numerically simulate viscous subsonic flow in a proposed elliptical two-duct version of the fuel side Hot Gas Manifold (HGM) for the Space Shuttle Main Engine (SSME), and (2) to provide analytical support for SSME related numerical computational experiments, being performed by the Computational Fluid Dynamics staff in the Aerophysics Division of the Structures and Dynamics Laboratory at NASA-MSFC. Numerical results of HGM were calculations to complement both water flow visualization experiments and air flow visualization experiments and air experiments in two-duct geometries performed at NASA-MSFC and Rocketdyne. In addition, code modification and improvement efforts were to strengthen the CFD capabilities of NASA-MSFC for producing reliable predictions of flow environments within the SSME.

Full three-dimensional laminar and turbulent SSME/HGM computations were performed on a sparse grid model of the Phase II+ two-duct version for the fuel side manifold employing the Lockheed explicit PAGE Navier-Stokes code. The approach, methodology, and results of this effort are documented in an earlier interim technical report submitted in March 1986 (Ref. 1). These will not be reiterated here, and the reader is referred to that previous document for those details.

Emphasis in this report is placed on the second of the aforementioned objectives. The direction of these efforts was to obtain and modify, for application to SSME internal flow analyses, state-of-the-art incompressible full three-dimensional Navier-Stokes codes. The approach taken is described in Section 2. Sample results for both laminar and turbulent computations are presented in Section 3.

2. TECHNICAL APPROACH

2.1 GENERAL

Previous computational fluid dynamics results, reported in the interim report for this contract (Ref. 1), were performed with a finite difference code employing an explicit solution algorithm. Steady state was obtained from a trial initialization by performing successive iterations in time until all transients have evolved away. The size of the time step used in this procedure is a strong function of the density of points in the grid. The higher the density, the smaller the allowable time step. For this reason, relatively coarse grids were used, even for regions near the solid walls.

Experience has dictated that much larger nodal densities near the walls are desirable for more accurate computational predictions. This precludes using an explicit code because of the unnecessarily large number of time steps required to obtain a steady state solution. The implicit code INS3D, developed at NASA-Ames (Ref. 2), was one of two codes chosen for application to additional SSME related computations. The second implicit code considered was the FDNS3D incompressible code developed by Y.S. Chen (Refs. 3 and 4). The implicit solution algorithm incorporated into these codes allows for much larger time steps even for grids of larger nodal densities. A brief description of the approach and solution methodology for each of these codes is now presented for completeness.

2.1.1 INS3D Code

The INS3D code solves the three-dimensional incompressible Navier-Stokes equations in primitive variables. An implicit finite difference operator is used in a general curvilinear coordinate system. The solution

procedure uses the standard approximate factorization scheme. The pressure field solution is based on the concept of adding a time-like pressure term into the continuity equation via an artificial compressibility factor. This approach was first introduced by Chorin (Ref. 5) and later adopted by Steger and Kutler (Ref. 6) using an implicit approximate factorization scheme by Beam and Warming (Ref. 7). It is from these earlier developments that INS3D evolved (Ref. 2).

Values of the artificial compressibility factor are bounded in order not to influence the steady state mass conservation. In the INS3D methodology mass conservation is of crucial importance if a stable solution is to result. Since the continuity equation is modified to obtain a hyperbolic-type equation, pressure waves of finite speed will be introduced. The speed of propagation of these pressure waves depends on the magnitude of the compressibility parameter. When the pressure waves travel through a given location a pressure gradient is created there. Near boundaries, the viscous boundary layer must respond to this pressure fluctuation. To accelerate convergence and avoid slow fluctuations it is desirable that the time required for pressure waves to propagate through the region of interest be much less than the time needed for the boundary layer to fully adjust itself. This condition provides for a lower bound on the artificial compressibility factor. The upper bound on this factor comes not from the physics but from the effects of the approximate factorization of the governing equations. When the finite difference form of the equation is factored, higher order cross-differencing terms are added to the left-hand side of the equation. These added terms must be made smaller than the original terms everywhere in the computational domain. This condition results in an upper bound on the compressibility factor.

It is well known in the computational fluid dynamics community that the approximate factorization schemes which employ alternating direction type implicit methods have stability problems in three dimensions (Refs. 8 through 11). The INS3D code satisfactorily overcomes this difficulty by providing second and fourth order smoothing terms to the algorithm to ensure stability without adversely affecting mass conservation.

2.1.2 FDNS3D Code

The FDNS3D code solves the steady or unsteady Navier-Stokes equations in three-dimensional curvilinear coordinate space for both incompressible and compressible flow. In the discretization of the steady transport equations, the code employees a combined second and third order upwind differencing scheme to approximate the convective terms. Second order central differencing is used for the remaining terms in the transport equation. For time-dependent flow problems, a second order backward differencing scheme is used for the temporal term. A SIMPLE type velocity-pressure correction solution algorithm is used to couple the continuity and momentum equations. Compressibility effects are considered in the formulation of the pressure correction treatment. Velocity components, pressure, density, and temperature are updated using the solution of the correction equation. An alternate direction line relaxation matrix solver is applied to solve the system of linearized algebraic equations. A standard k-ε turbulence model and an extended version which employs an improved transport equation for the turbulent kinetic energy dissipation rate are included in the FDNS3D code.

Artificial fourth order smoothing techniques are used in the transport equations to obtain strong coupling between the velocity and pressure fields in many approaches. To ensure stability of the numerical solution, grid staggering between the velocity vectors and pressure nodes are used in a usual SIMPLE type algorithm. This method, however, has the drawback that the velocity components and the pressure are solved using different control volumes. This also requires a lot more bookkeeping in the coding. To ensure velocity-pressure coupling for the non-staggered grid system, FDNS3D employs an explicit dissipation (smoothing) term which is added to the pressure correction equation. This term is limited to a very small contribution to the correction equation so that mass conservation is preserved. Both approaches, staggered (Ref. 3) and non-staggered (Ref. 4), are coded as different versions of FDNS3D. The non-staggered version is used in the present analysis.

2.2 LAMINAR COMPUTATIONS

The viscous flows internal to the SSME/HGM involve complicated passages which tend to create regions of separation and swirling secondary flow patterns. In order to evaluate the incompressible codes on computational grids which would possess these characteristics two simple geometries were generated. The Lockheed algebraic grid code was employed to develop the two-dimensional grid shown in Fig. 2-1 and the three-dimensional grid shown in Fig. 2-2. The turnaround duct (TAD) geometry presented in Fig. 2-1 has the same dimensions as the Phase II+ version of the SSME/HGM TAD. Flow enters the lower part of the duct from the right, experiences a 180 deg turn and exits from the upper part of the duct, flowing toward the right. Adverse pressure gradients produced on the downstream side of the turn will tend to cause the boundary layer to separate. The three-dimensional flow around the turn in the square duct geometry presented in Fig. 2-2 will exhibit a swirling secondary flow similar to that exiting the right or left transfer ducts of the SSME/HGM. Because of symmetry considerations, no such swirl will be produced in the TAD configuration, even in three dimensions.

Both INS3D and FDNS3D were used to solve identical problems on these two geometries. Results of these computations are presented in Section 3.1.

2.3 TURBULENT COMPUTATIONS

Incorporated into the FDNS3D code for turbulent applications is a k-ϵ subroutine with stand and wall function treatment. This was used for the two-dimensional case using the TAD geometry. The INS3D code must be supplied with an external subroutine for computing the turbulence eddy viscosity array. For the two-dimensional TAD geometry, a standard Baldwin-Lomax calculation was coded and incorporated into the code.

For three-dimensional applications, especially to layer computations such as primary SSME/HGM geometries, INS3D was chosen to be the primary code.

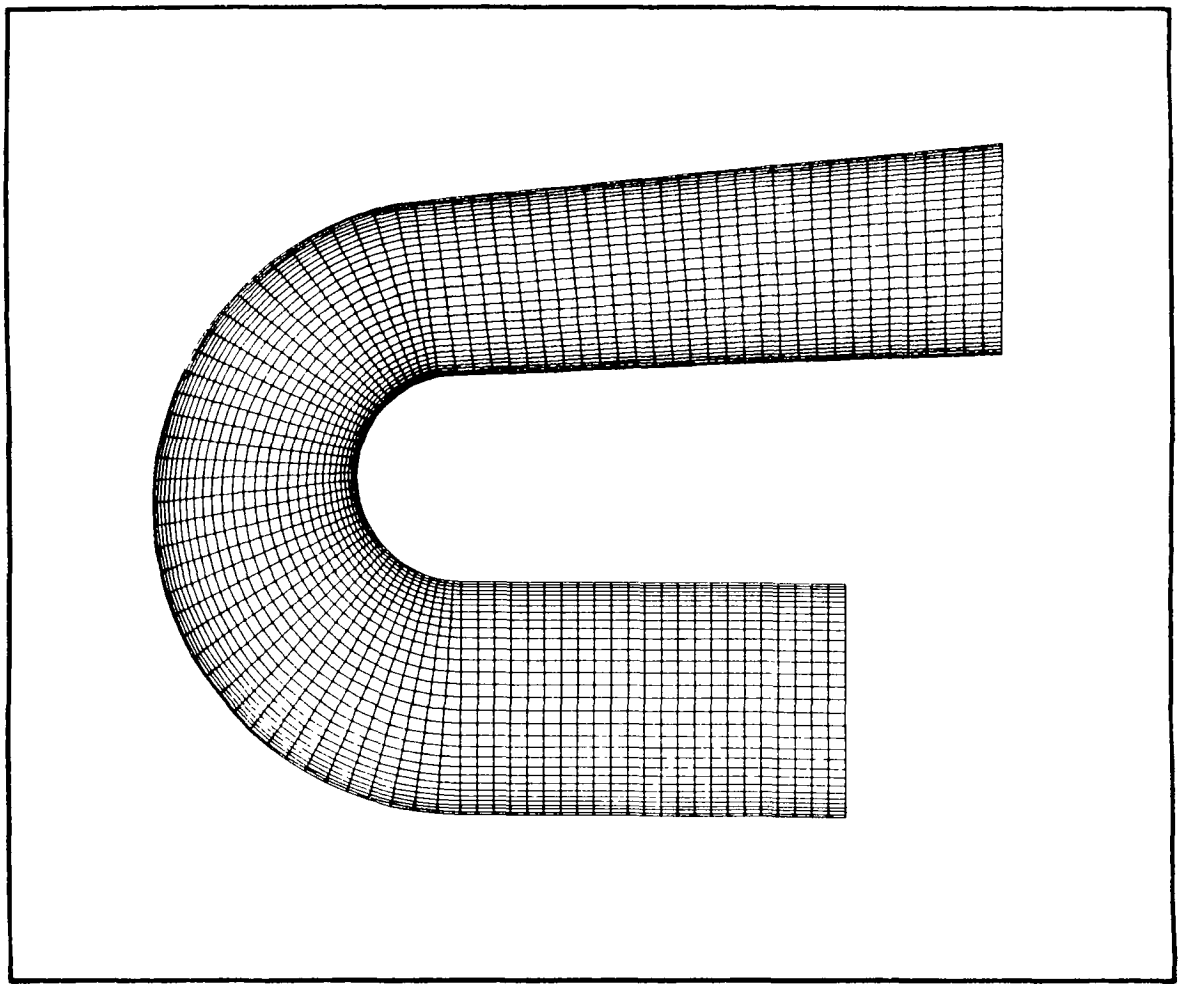

Fig. 2-1 TAD Computational Grid (33 x 117) for Two-Dimensional Test Solutions Using Both INS3D and FDNS3D

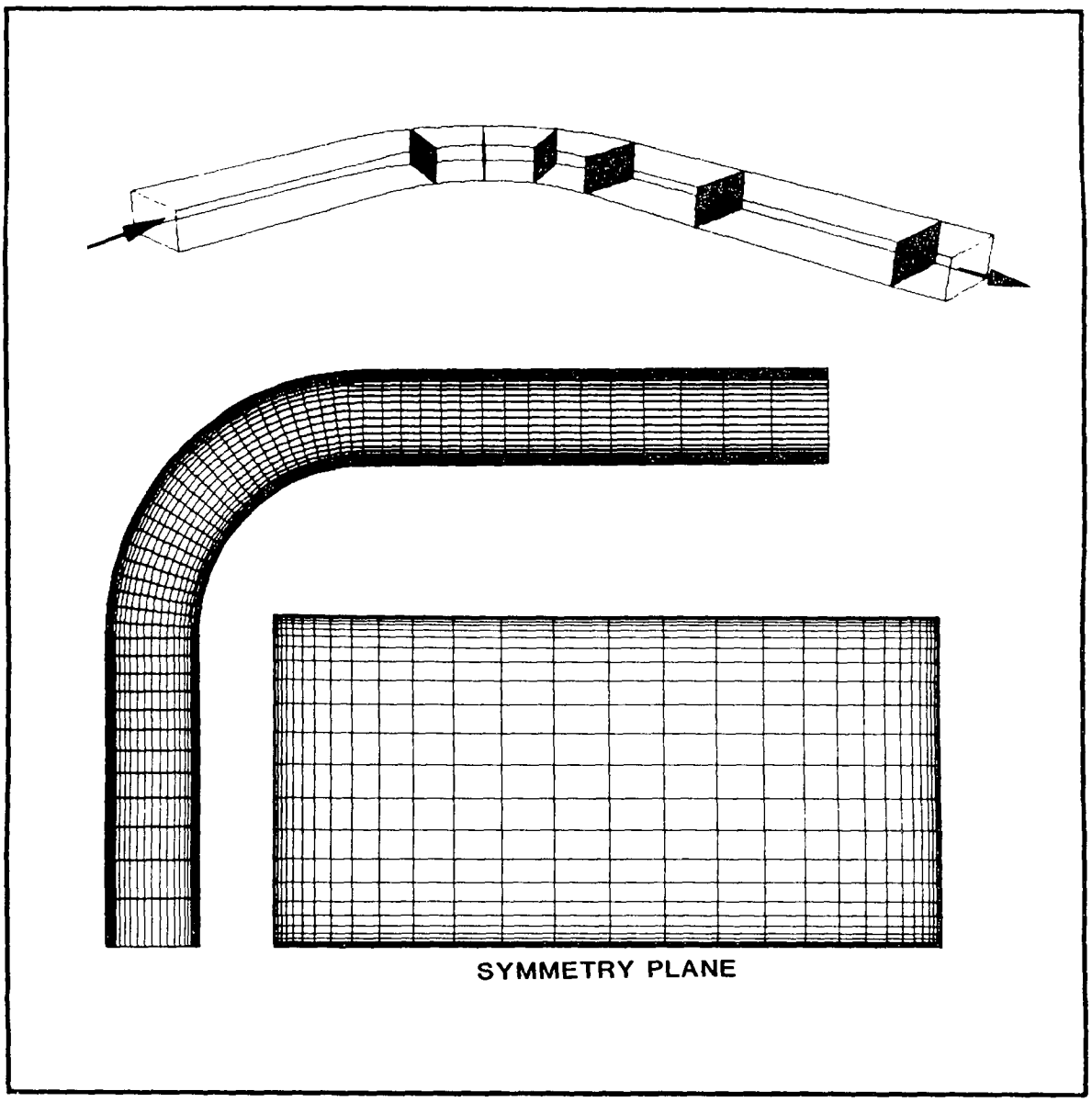

Fig. 2-2 Square Duct Computational Grid (54 x 31 x 27) for Three-Dimensional Laminar and Turbulent Test Solutions

The structure of the finite volume methodology of FDNS3D causes it to be about three times more storage intensive as INS3D. On the NASA-MSFC Cray-XMP, INS3D can be run as a single zone computation on a geometry consisting of no more than 165,000 nodes. FDNS3D can perform the same task but on a single zone of one-third as many grid points. Because of this limitation and because of the fact that the k-ε subroutine in FDNS3D seemed to produce consistent and reliable turbulent results (Ref. 12), it was decided that this treatment be incorporated into INS3D as a separate subroutine.

The approach for adding the k-ε computation to INS3D was to have INS3D compute the primative variables, pressure and velocity components, and have the turbulence eddy viscosity array updated using two modified subroutines from FDNS3D. The subroutines solve the transport equations for turbulent kinetic energy, k, and its dissipation rate, ε. Thus, the solution procedures are decoupled. However, since the k-ε solution strategy is iterative at each time stem in INS3D, the decoupling produces a minimal effect on the solution.

In this analysis, the turbulence k-ε model after Launder and Spalding (Ref. 13) is employed. The modeled kinetic energy and dissipation equations are given by

$$\frac{Dk}{Dt} = \frac{\partial}{\partial x_i} \frac{\nu_{eff}}{\alpha_k} \frac{\partial k}{\partial x_j} + P - \varepsilon$$

where

$$P = \nu_{eff} \frac{\partial u_i}{\partial x_j} + \frac{\partial \bar{u}_j}{\partial x_j} \frac{\partial \bar{u}_i}{\partial x_j},$$

and

$$\nu_t = C_\mu k^2 / \varepsilon, \; C_\mu = \text{proportionality constant},$$

and

$$\frac{D\varepsilon}{Dt} = \frac{\partial}{\partial x_i} \frac{\nu_{eff}}{\sigma_\varepsilon} \frac{\partial \varepsilon}{\partial x_i} + \frac{1}{\rho} \frac{\varepsilon}{k} (C_1 P - C_2 \rho \varepsilon)$$

where $\sigma\varepsilon$ is the turbulent kinetic energy dissipation Prandtl number, and C_1 and C_2 are model constants. Recommended values for the constants in the above equations are (Ref. 14):

$$C_1 = 1.44$$
$$C_2 = 1.92$$
$$C\mu = 0.09$$
$$\sigma_k = 1.0$$
$$\sigma_\varepsilon = 1.3$$

Much study has been made concerning the values of these constants (Ref. 15). The values above have produced the most consistent reliable results for flows involving streamline curvature and recirculation regions.

Boundary conditions treatment for the k-ε calculation is as follows:

- Inlet Plane - The strengths of k and ε were not known and so were determined using

  $$k_{in} = (\text{turbulence intensity}) \cdot (v_{in})^2$$

  and

  $$\varepsilon_{in} = (k_{in})^{1.5}/(\lambda L),$$

  where λ and L represent a constant and a characteristic length, respectively. Sometimes λL is called a turbulent length scale.

  In this analysis, the characteristic length L, length scale λ, and turbulent intensity are defined AS ΔY, 0.01 AND 0.0003, respectively. Parametric study on these variables in a single and multiple jet configuration was reported by Bai (Ref. 15). Different flow patterns and recirculation zone sizes are obtained near the inlet by varying these parameters. The above values were chosen because the effect is mainly limited to the inlet regions, as is the effect of varying the turbulent model constants C_1 and C_2.

- Exit Plane - For the exit plane, and the symmetric boundary at the center of the square duct, extrapolation enforcing normal gradients to zero was used.

- Solid Walls - Near the solid wall region the flux through the wall side interface is set to zero and the diffusive flux term on the left side of the equation is transferred into the source term on the right-hand side as a false source. This false source term is obtained from a "wall function" which is largely based on experimental data (Refs. 13 and 16). Recently more elaborate wall functions involving the division of the near wall region into two or three layers were proposed. The concept of the wall function is based on the assumption that the total stress τ_t is equal either to the Reynolds stress for the core or to the laminar stress for the viscous sublayer. In the current analysis, this wall boundary treatment is applied only to the turbulent kinetic energy and its dissipation rate equation.

3. RESULTS

3.1 LAMINAR COMPUTATIONS

Laminar results were obtained in the two-dimensional TAD geometry shown in Fig. 2-1 using both INS3D and FDNS3D for Reynolds number of 500 and 1000. In all cases an essentially converged solution was obtained in 1000 iterations. Results at a Reynolds number of 1000 are presented in Figs. 3-1 and 3-2. The two solutions are essentially the same. Both exhibit separation on the downstream side of the turn occurring at the same point on the wall and both shown the same pressure distribution on the inner and outer walls.

Since an axisymmetric option is available in FDNS3D, the TAD geometry was used to perform such a compilation at the same Reynolds number as the two-dimensional case. These results are given in Fig. 3-3. Separation of the boundary layer on the inner wall is again observed. It occurs, however, much earlier in the turn. This is of course due to the diffusive effect in the axisymmetric geometry as the flow traverses the turn.

A three-dimensional computation was performed with both codes on a square duct grid like the one shown in Fig. 2-2 but with the cross-plane nodal distribution shown in Fig. 3-4. The difference being that the grid is stretched to the solid walls but not to the symmetry plane. Like the two-dimensional results these solutions were essentially the same. Thus, only the INS3D results are presented, and these are shown in Figs. 3-5 through 3-7. Again, only 1000 iterations were needed to obtain a converged solution. This solution exhibits the characteristic secondary swirl which must be produced as the fluid negotiates the 90 deg turn. It also shows no separation along the inner wall at the computed Reynolds number of 790.

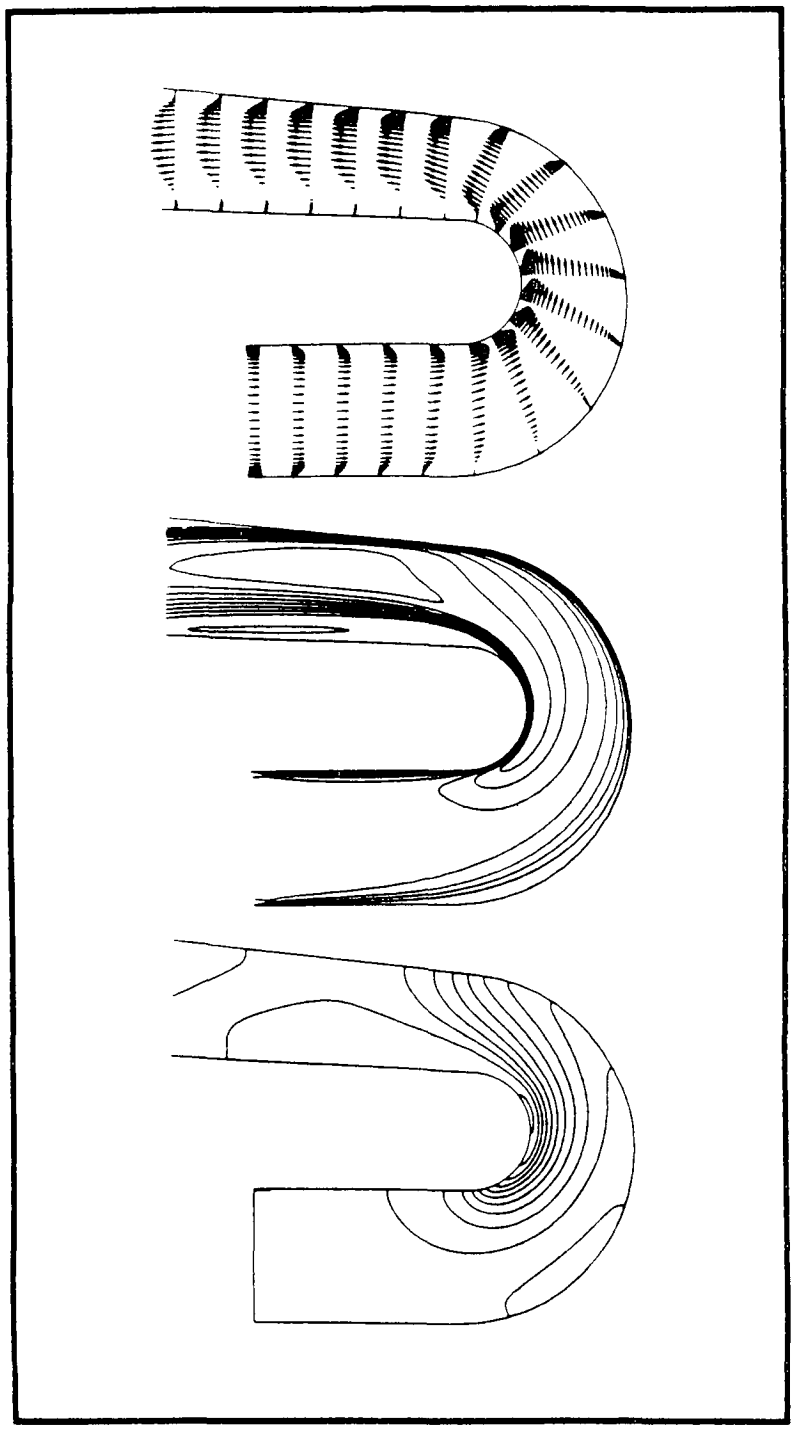

Fig. 3-1 Velocity Vectors, Velocity Magnitude Contours, (Middle) and Static Pressure Contours for Re = 1000 in Two-Dimensional TAD Using INS3D

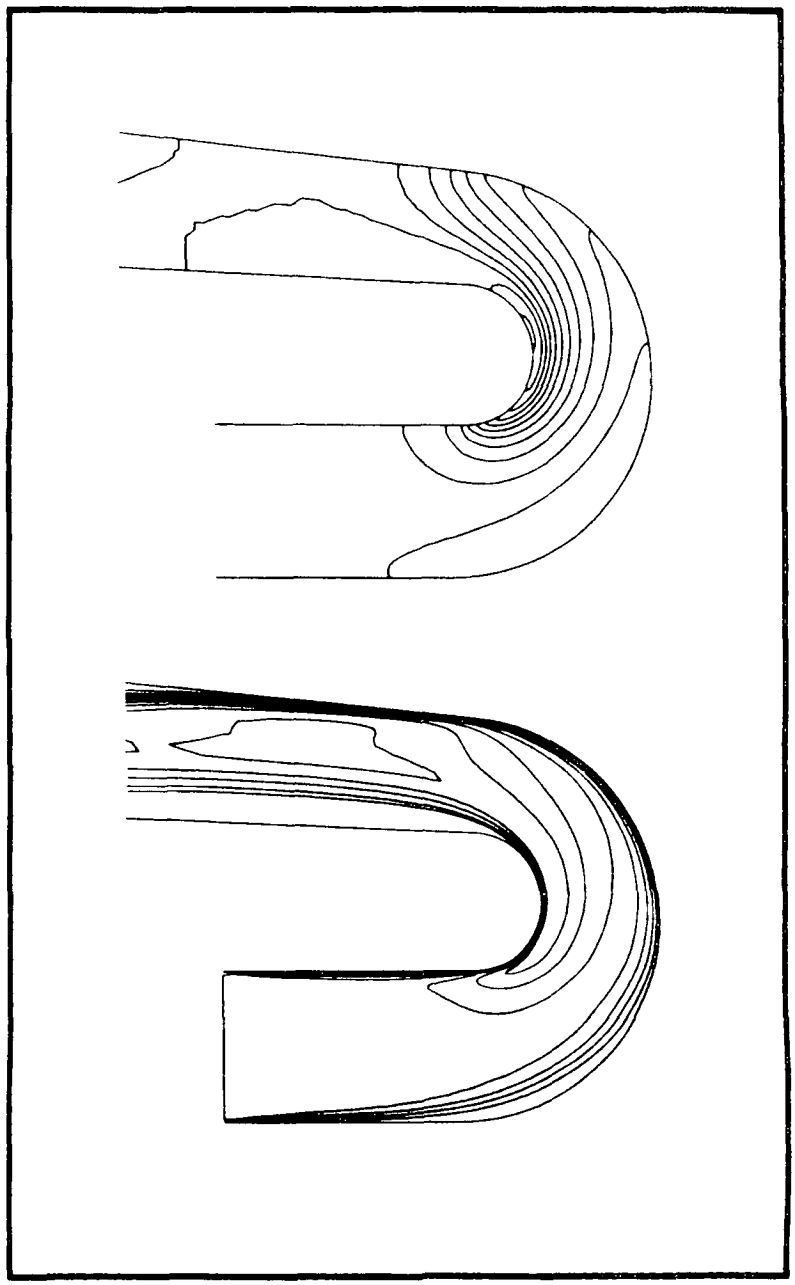

Fig. 3-2 Velocity Magnitude Contours, (Bottom) and Static Pressure Contours for Re = 1000 in Two-Dimensional TAD Using FDNS3D

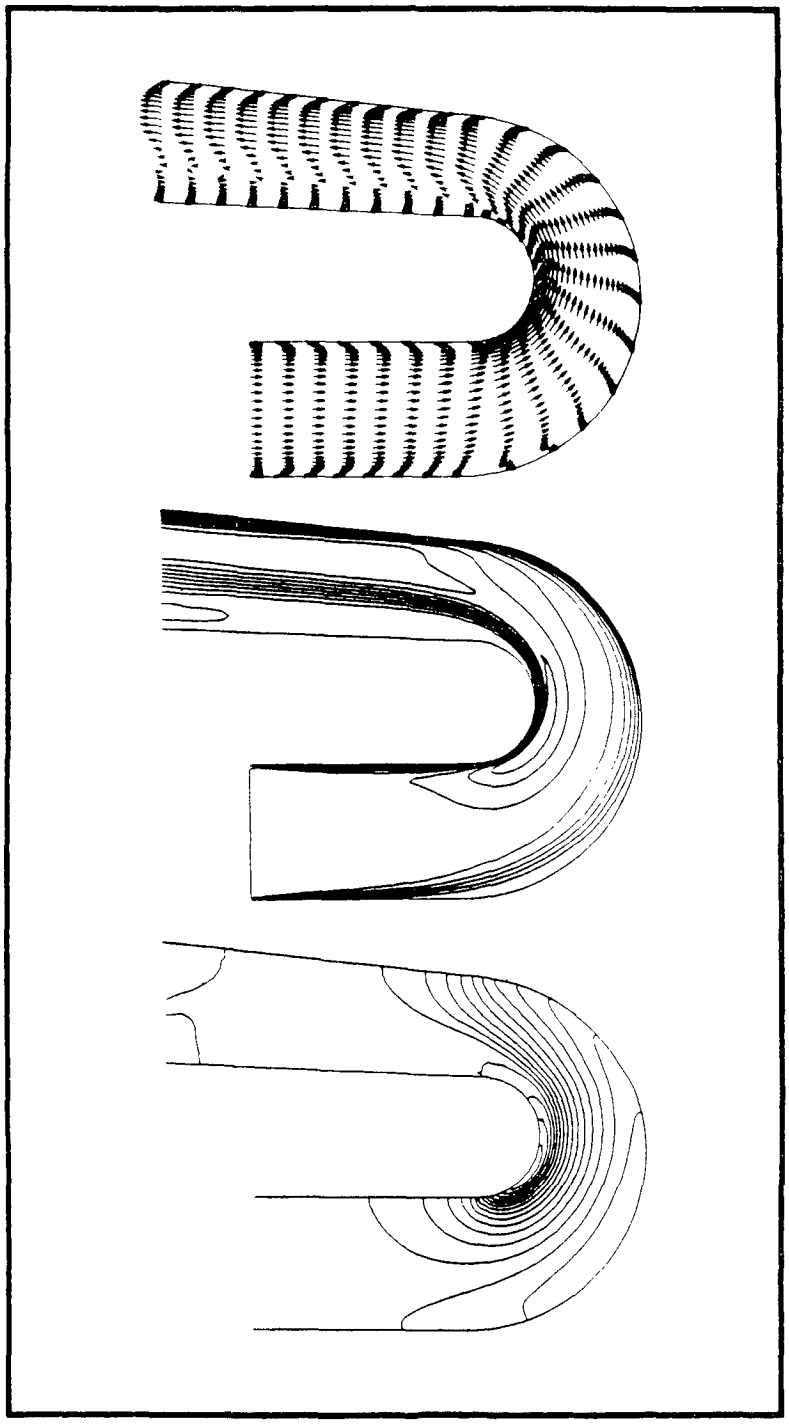

Fig. 3-3 Velocity Vectors, Velocity Magnitude Contours, (Middle), and Static Pressure Contours for Re = 1000 in Axisymmetric TAD Using FDNS3D

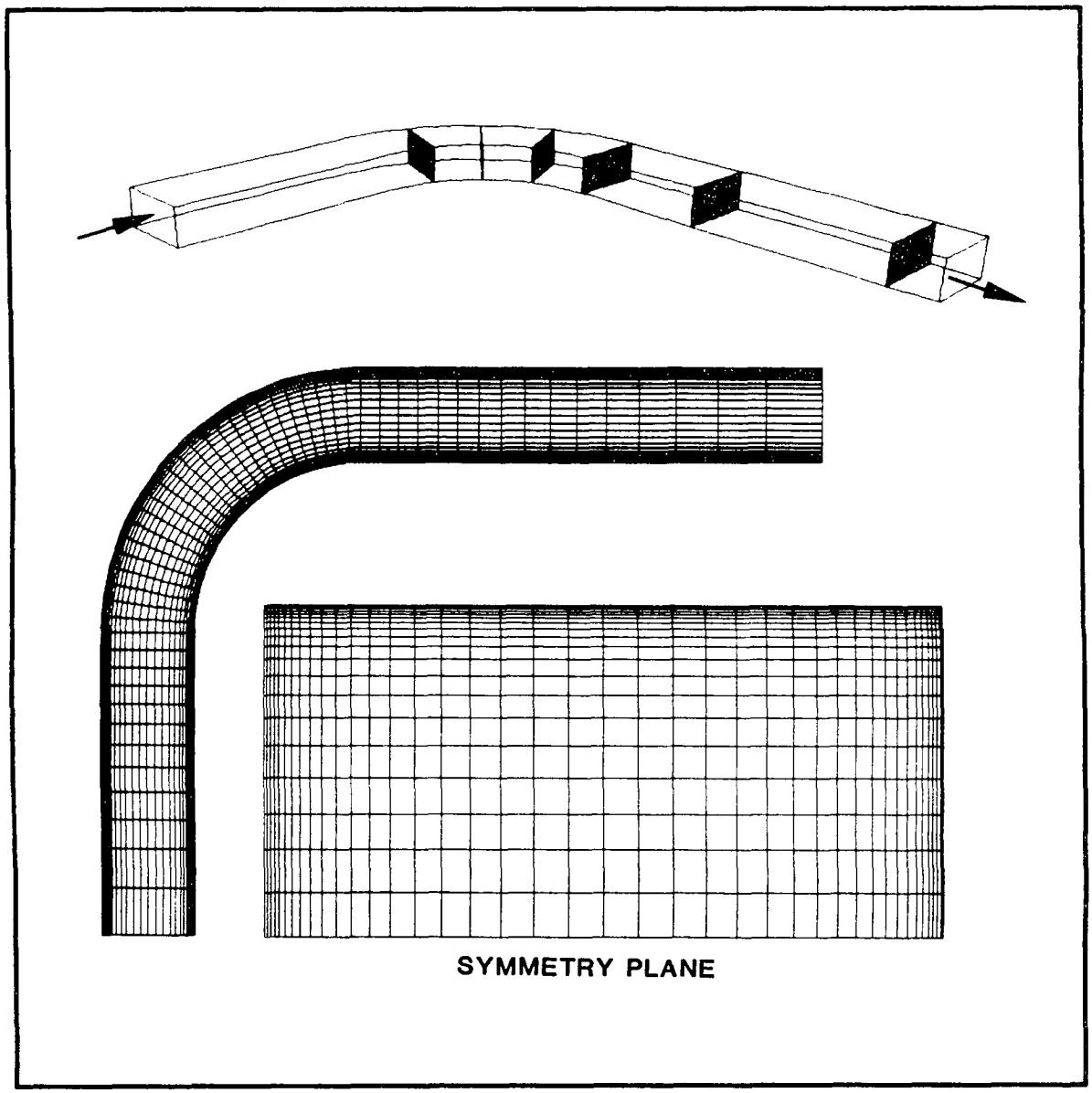

Fig. 3-4 Square Duct Computational Grid (54 x 41 x 31) for Initial Three-Dimensional Laminar Test Solution

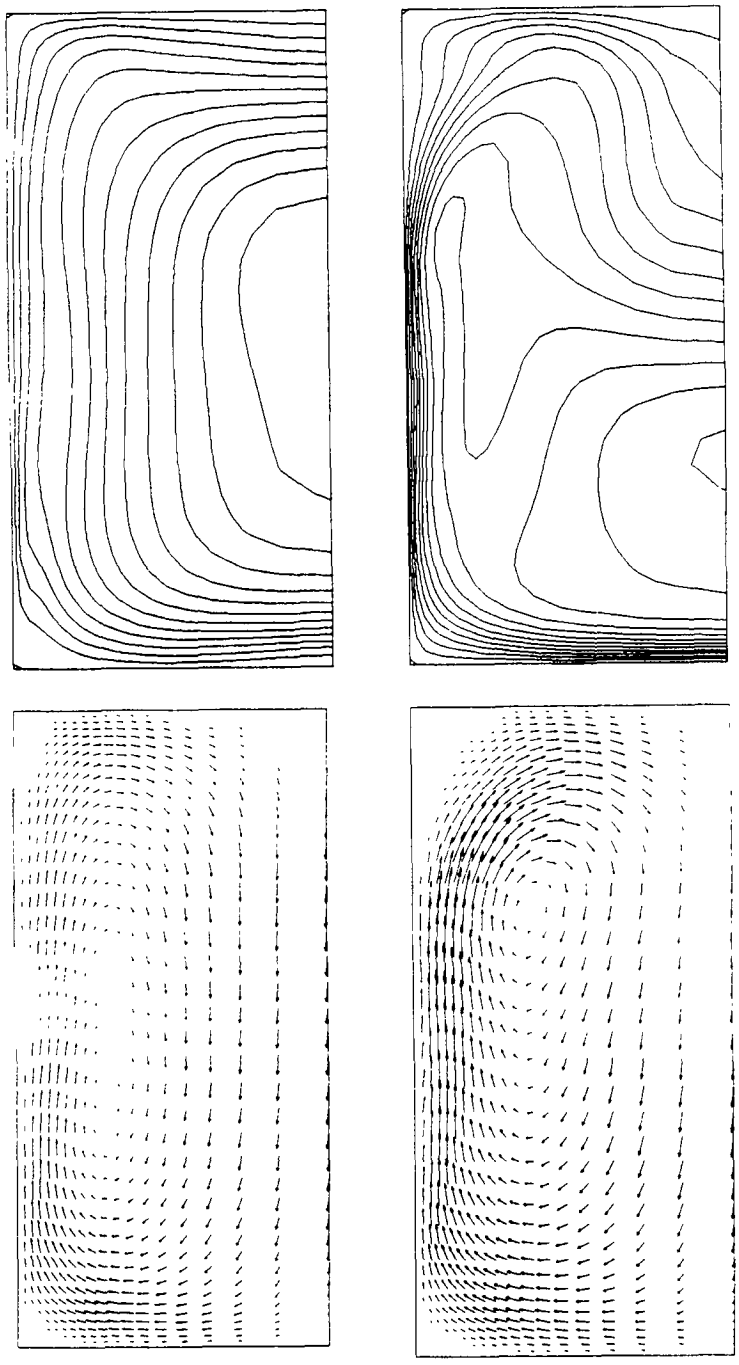

Fig. 3-5 Components of Velocity Vectors and Velocity Magnitude Contours in Square Duct Cross Planes Corresponding to 27 deg (Top) and 45 deg into the Turn for Laminar Flow at Re = 790 Using INS3D and Grid in Fig. 3-4

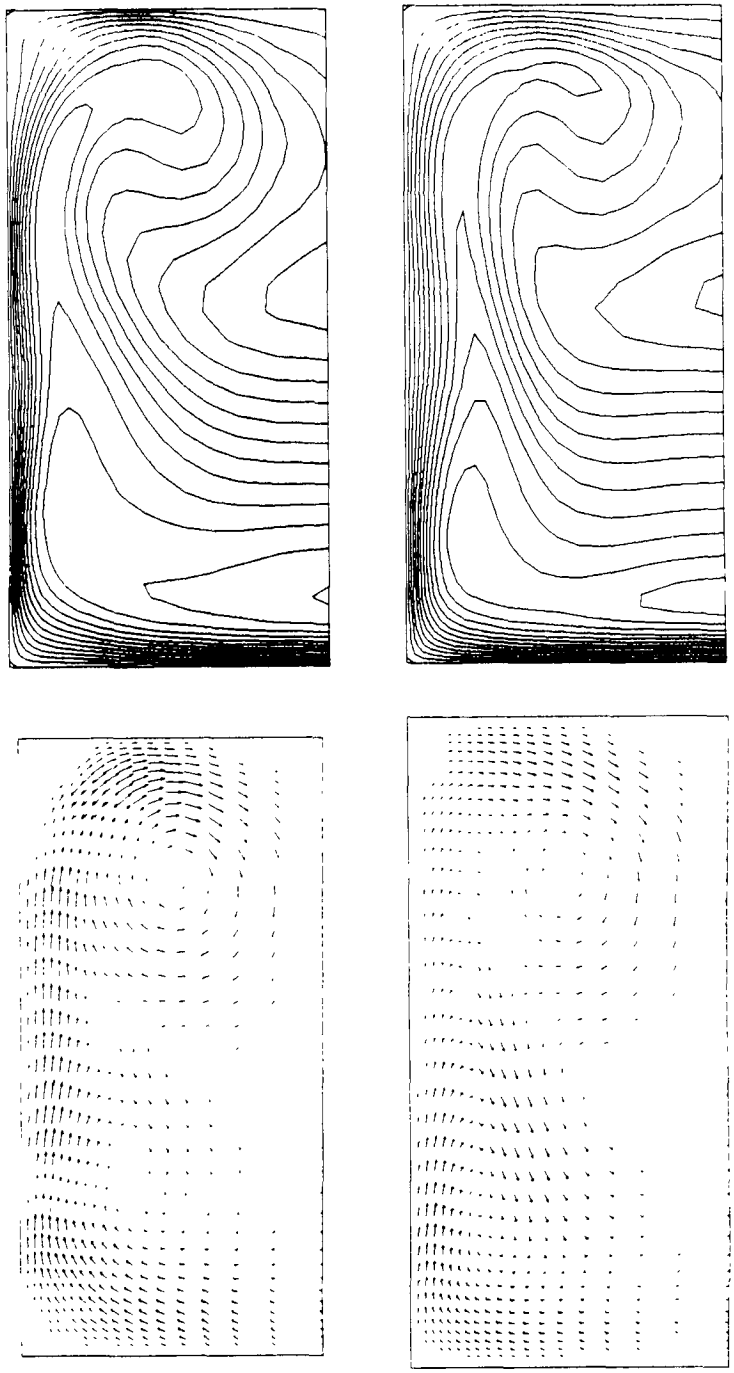

Fig. 3-6 Components of Velocity Vectors and Velocity Magnitude Contours in Square Duct Cross Planes Corresponding to 73 deg (Top) and 90 deg into the Turn for Laminar Flow at Re = 790 Using INS3D and Grid in Fig. 3-4

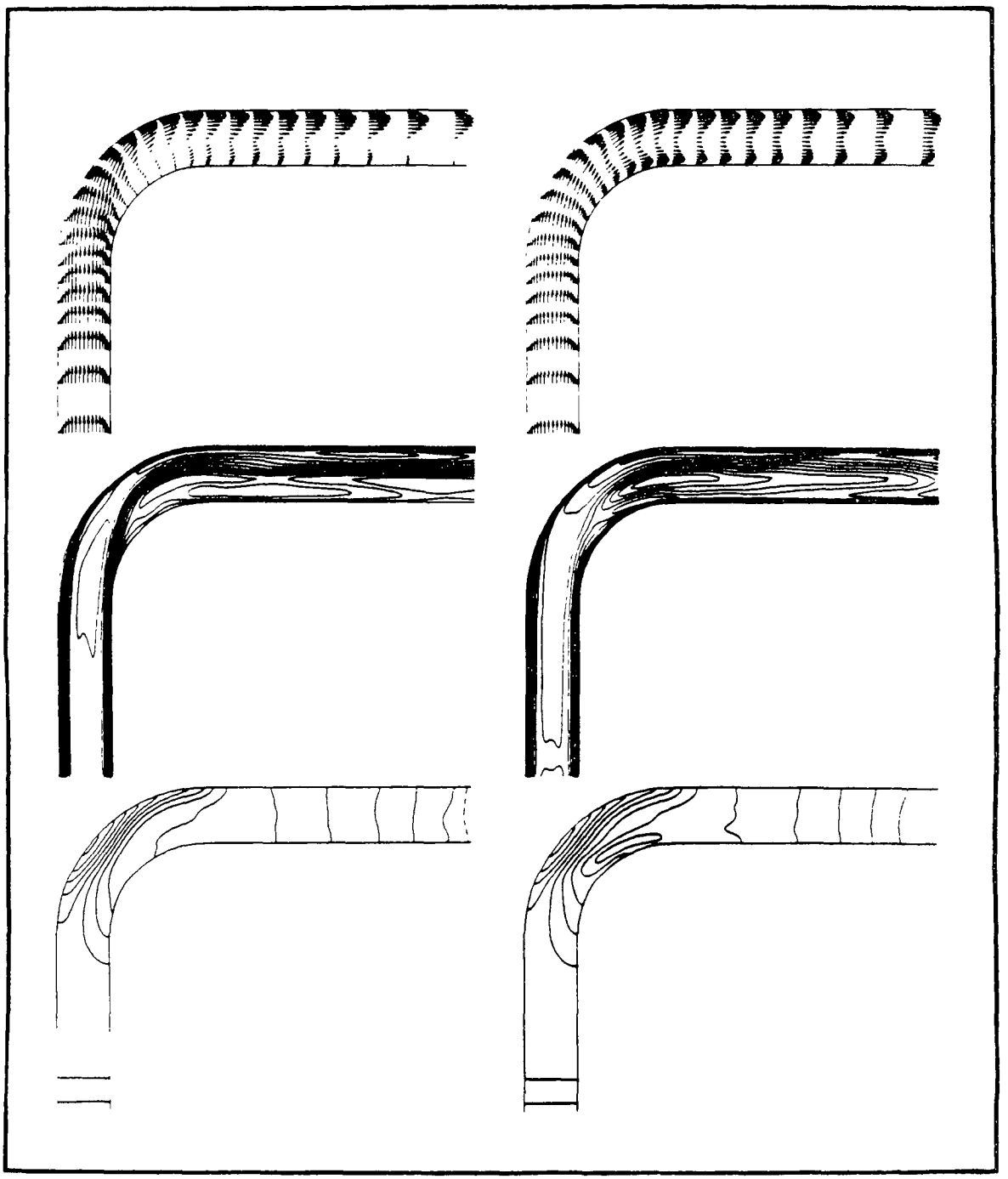

Fig. 3-7 Velocity Vectors, Velocity Magnitude Contours (Middle) and Static Pressure Contours at Square Duct Mid Plane (Left) and Three-Quarter Plane for Laminar Flow at Re = 790 Using INS3D and Grid in Fig. 3-4

Due to the sparseness of the grid at the symmetry plane boundary, it could be argued that the computed structure of the secondary flowfield shown in Figs. 3-5 and 3-6 might be grid dependent. To settle this question a second calculation at the same Reynolds number was made on the grid in Fig. 2-2. These results are presented for comparison in Figs. 3-8 through 3-10. Figure 3-11 contains results at cross-planes downstream of the turn showing the continued development of the swirl pattern.

The secondary flow pattern for the second case is clearly different from that using the first grid. The larger density of grid points near the outer wall and symmetry plane intersection has resolved a second counter-rotating swirl pattern that appears after mid-turn and grows to be the same size as the primary swirl a position near the exit plane of the geometry. Because of the resolution of finer details with flow using the grid in Fig. 2-2, this grid was employed in the turbulent computation for the same geometry.

3.2 TURBULENT COMPUTATIONS

Two-dimensional and axisymmetric turbulent results in the TAD geometry using FDNS3D with the standard k-ε model are displayed in Figs. 3-12 and 3-13. No boundary layer separation was observed in either case for a Reynolds number of one million. Effects of the increase in cross sectional area for the axisymmetric case is evident when the two results are contrasted. Each case was run for 1000 iterations on the NASA-MSFC Cray-XMP.

The geometry in Fig. 2-2 was used to run a turbulent computation at Reynolds number of 40,000 using INS3D with the k-ε subroutine extracted from FDNS3D. This calculation was again run for 1000 iterations. For comparison the laminar computation and the turbulent computation Cray-XMP CPU run times were 0.7×10^{-4} seconds per iteration per node and 2.0×10^{-4} seconds per iteration per node. Figures 3-14 through 3-20 show various aspects of the turbulent square duct computation. The secondary

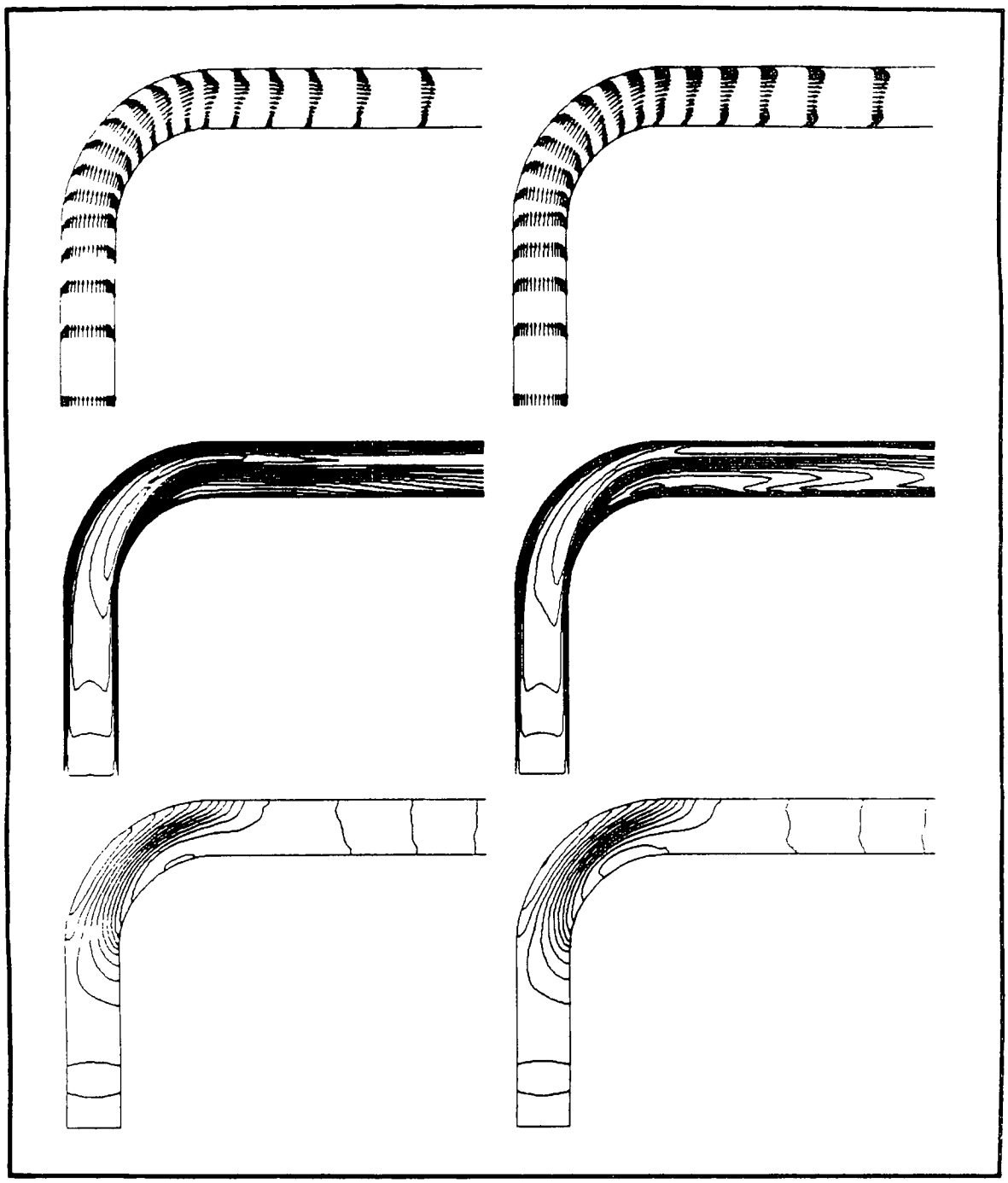

Fig. 3-8 Velocity Vectors, Velocity Magnitude Contours (Middle) and Static Pressure Contours at Square Duct Mid Plane (Left) and Three-Quarter Plane for Laminar Flow at Re = 790 Using INS3D and Grid in Fig. 2-2

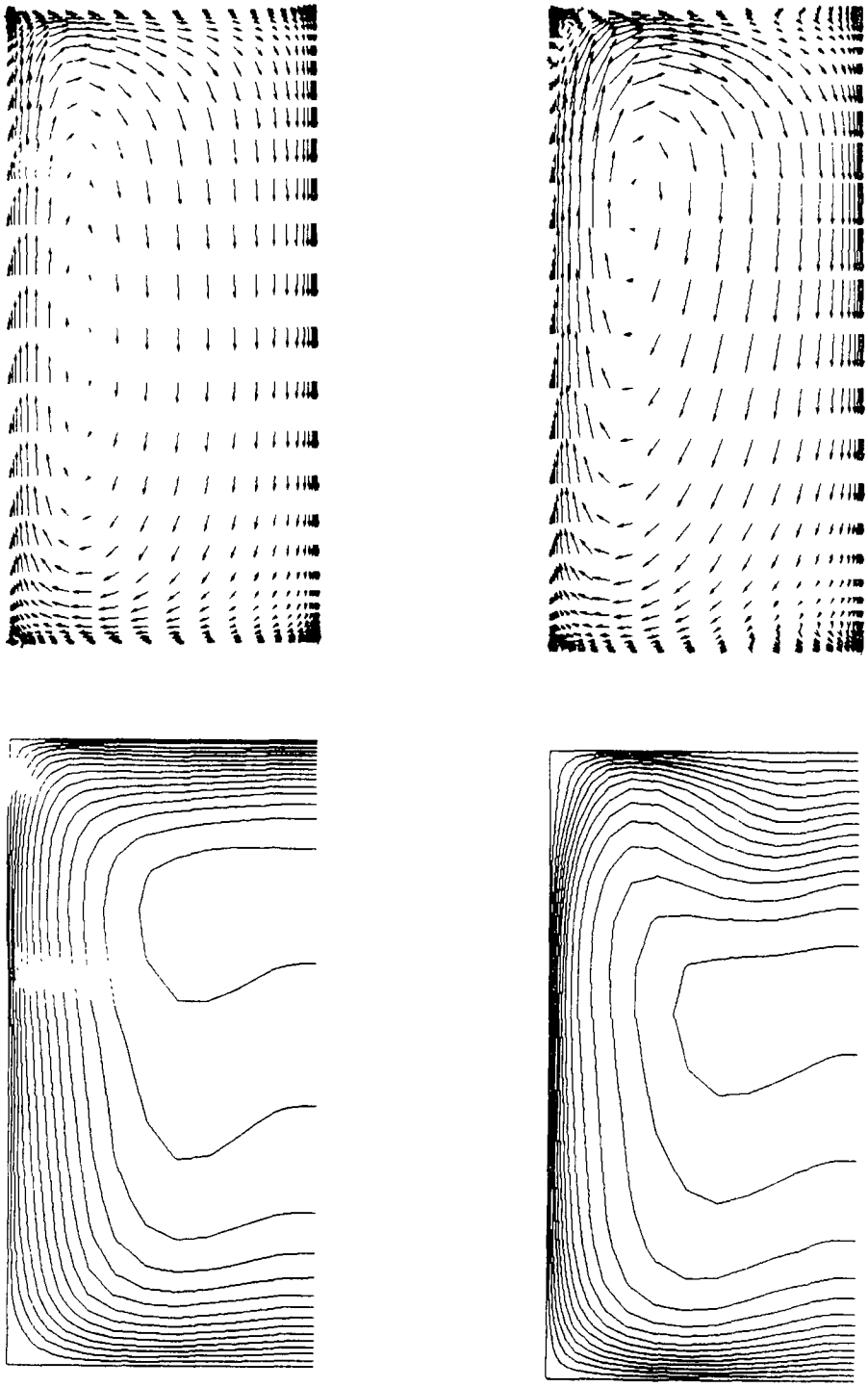

Fig. 3-9 Components of Velocity Vectors and Velocity Magnitude Contours in Square Duct Cross Planes Corresponding to 27 deg (Top) and 45 deg into the Turn for Laminar Flow at Re = 790 Using INS3D and Grid in Fig. 2-2

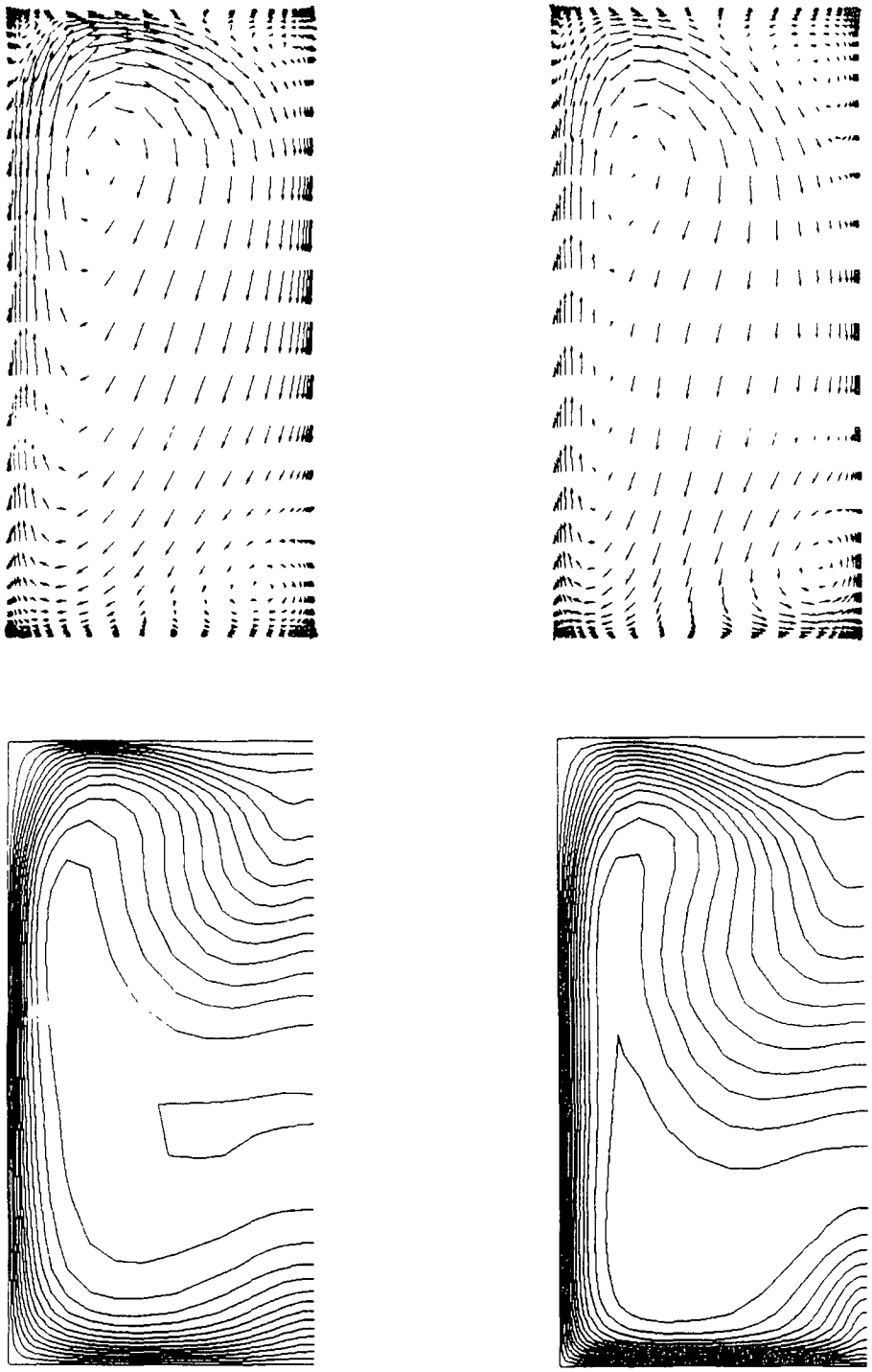

Fig. 3-10 Components of Velocity Vectors and Velocity Magnitude Contours in Square Duct Cross Planes Corresponding to 73 deg (Top) and 90 deg into the Turn for Laminar Flow at Re = 790 Using INS3D and Grid in Fig. 2-2

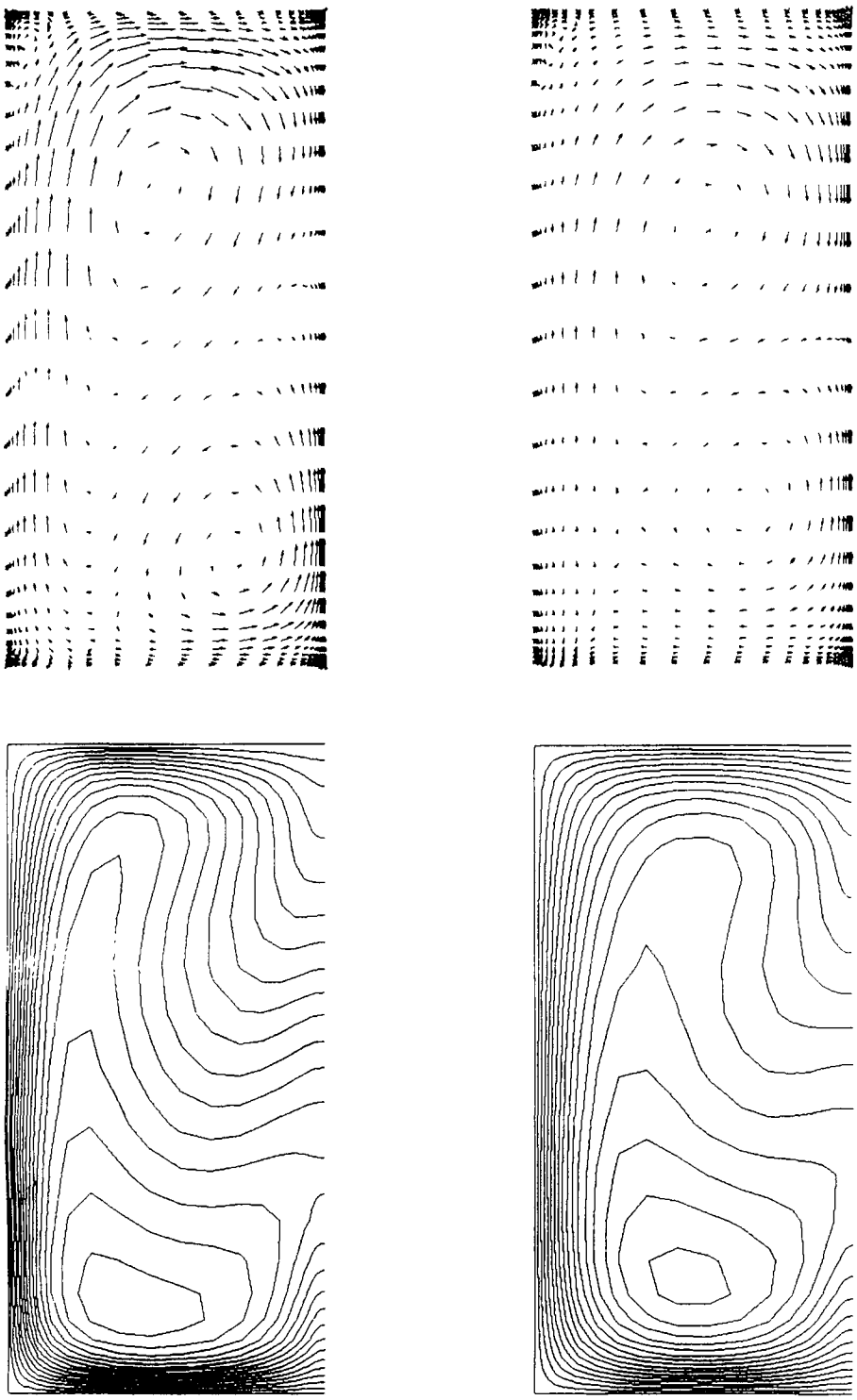

Fig. 3-11 Components of Velocity Vectors and Velocity Magnitude Contours in Square Duct Cross Planes Corresponding to Two Duct Widths Downstream of Turn (Top) and from Duct Widths Downstream of Turn for Laminar Flow at Re = 790 Using INS3D and Grid in Fig. 2-2

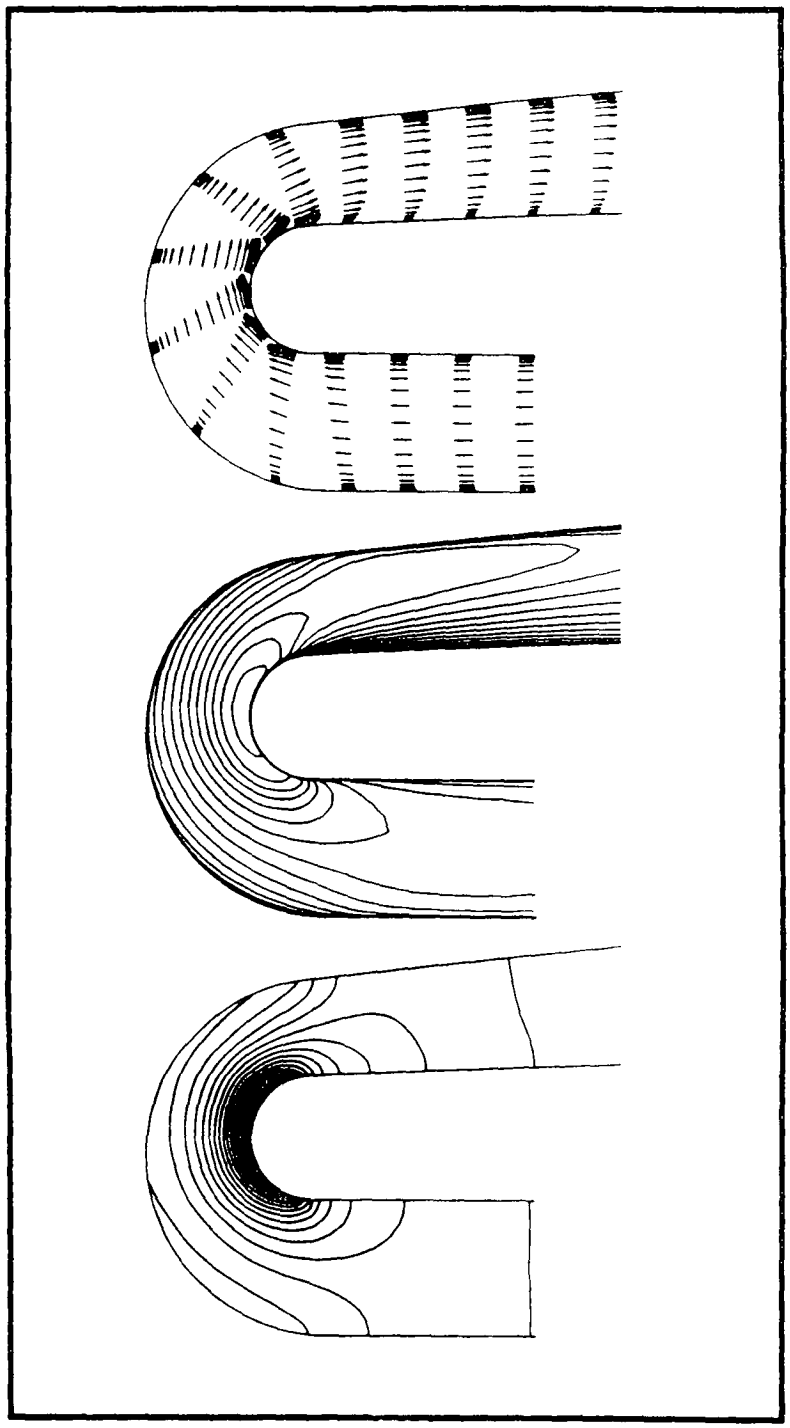

Fig. 3-12 Velocity Vectors (Top), Velocity Magnitude Contours and Static Pressure Contours (Bottom) for Two-Dimensional Turbulent Computations at Re = 1,000,000 Using FDNS3D with Standard k-ε Model

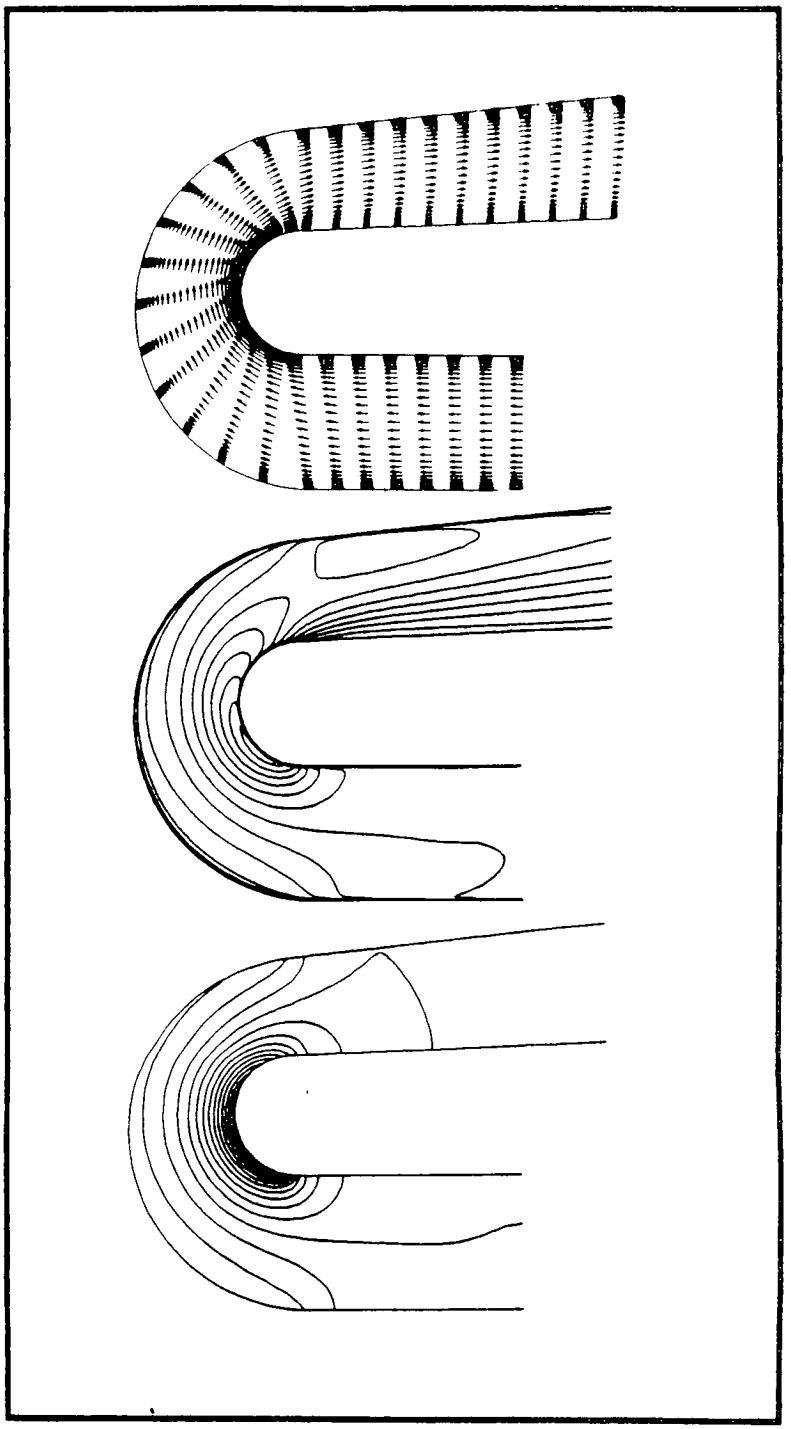

Fig. 3-13 Velocity Vectors (Top), Velocity Magnitude Contours and Static Pressure Contours (Bottom) for Axisymmetric Turbulent Computations at Re = 1,000,000 Using FDNS3D with Standard k-ε Model

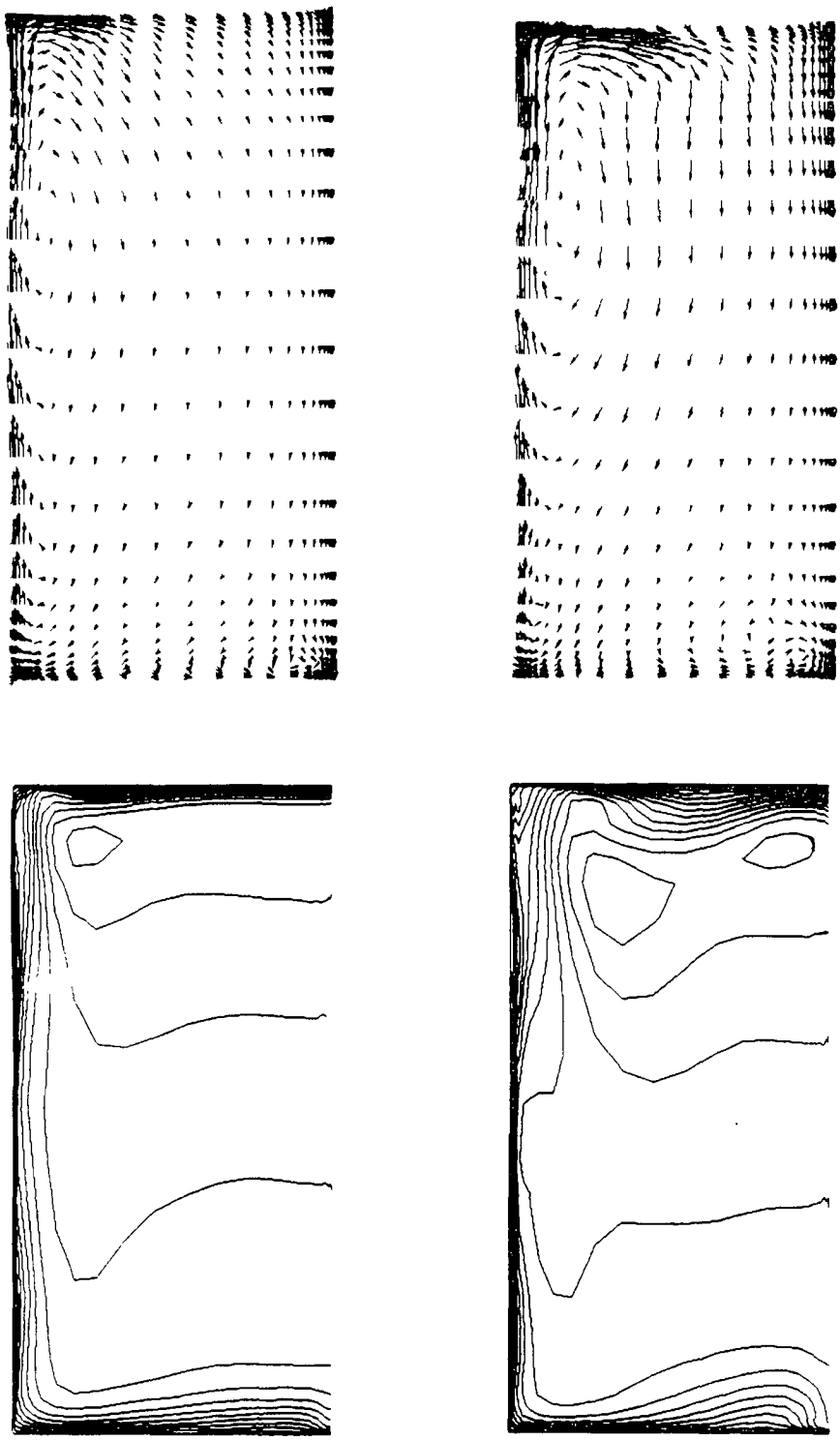

Fig. 3-14 Components of Velocity Vectors and Velocity Magnitude Contours in Square Duct Cross Planes Corresponding to 27 deg (Top) and 45 deg into the Turn for Turbulent Flow at Re = 40,000 Using INS3D and Standard k-ε Model

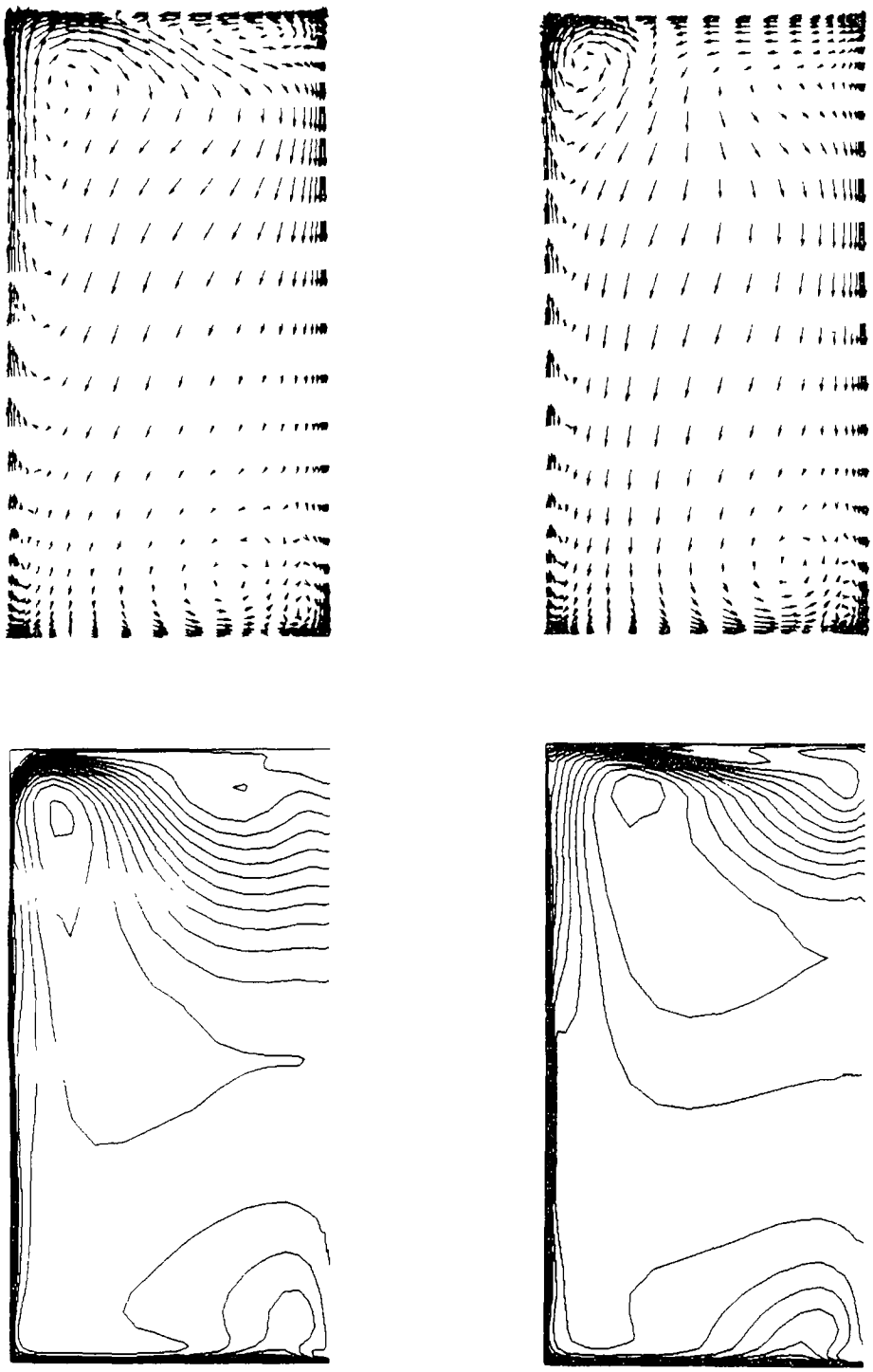

Fig. 3-15 Components of Velocity Vectors and Velocity Magnitude Contours in Square Duct Cross Planes Corresponding to 73 deg (Top) and 90 deg into the Turn for Turbulent Flow at Re = 40,000 Using INS3D and Standard k-ε Model

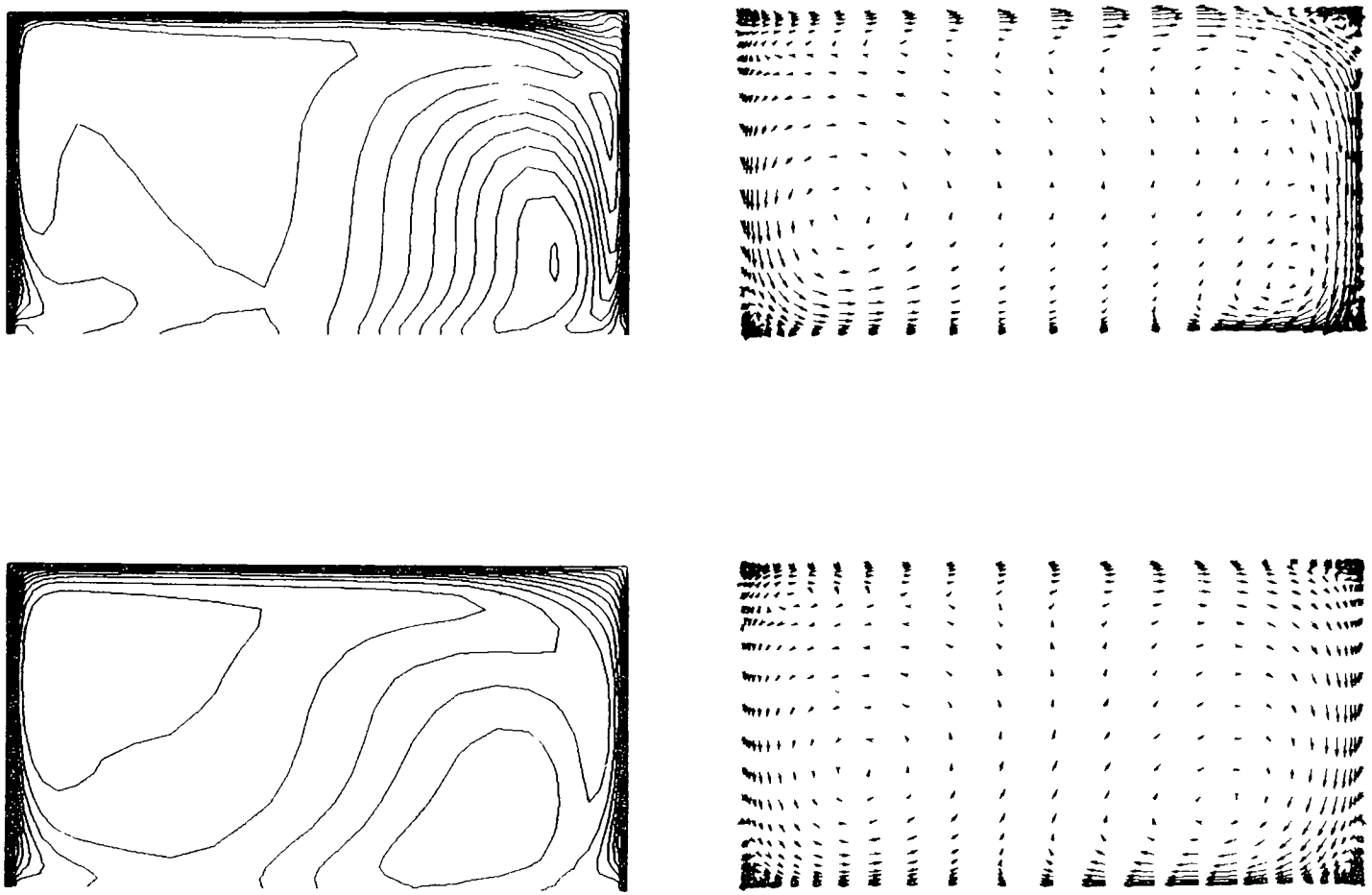

Fig. 3-16 Components of Velocity Vectors and Velocity Magnitude Contours in Square Duct Cross Planes Corresponding to Two Duct Widths Downstream of Turn (Top) and from Duct Widths Downstream of Turn for Turbulent Flow at Re = 40,000 Using INS3D and Standard k-ε Model

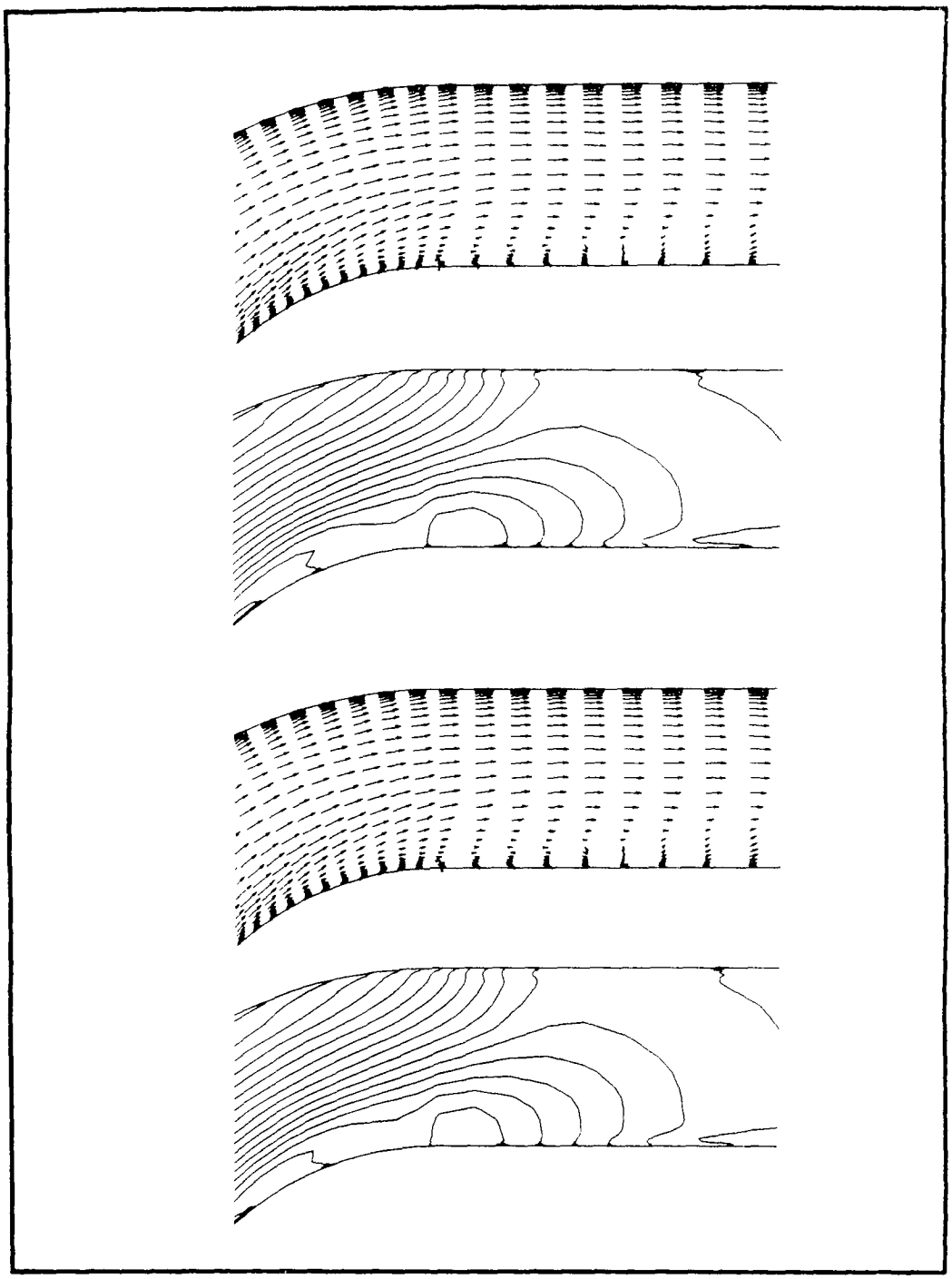

Fig. 3-17 Velocity Vectors and Static Pressure Contours at Symmetry Plane (Upper Plots) and Two Planes Above, in a Region Just Downstream of 90 deg Turn in Square Duct for Turbulent Computation at Re = 40,000 Using INS3D with Standard k-ε Model

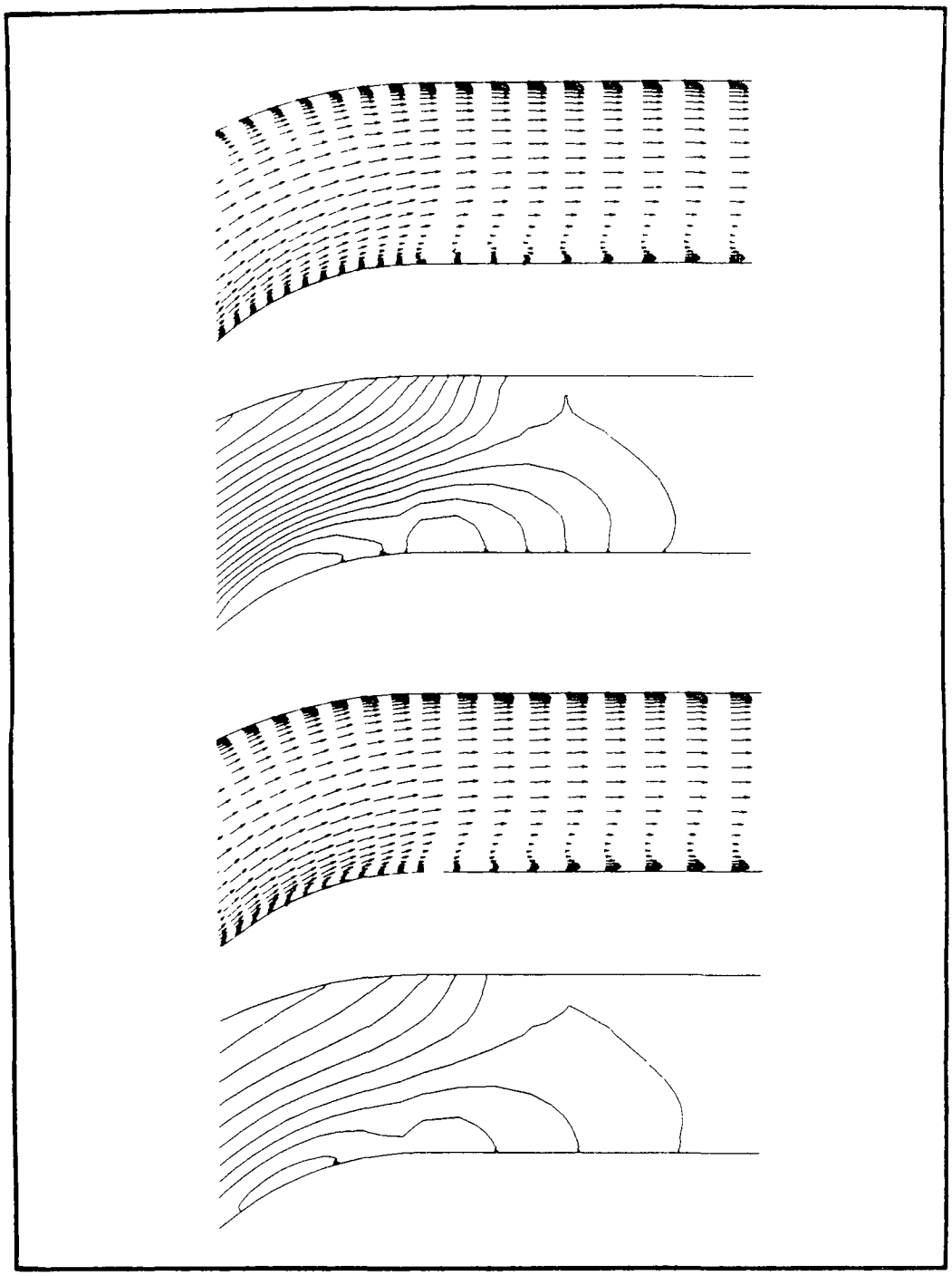

Fig. 3-18 Velocity Vectors and Static Pressure Contours at the Three-Quarter Plane (Upper Plots) and Two Planes Below, in a Region Just Downstream of 90 deg Turn in Square Duct for Turbulent Computations at Re = 40,000 Using INS3D with Standard k-ε Model

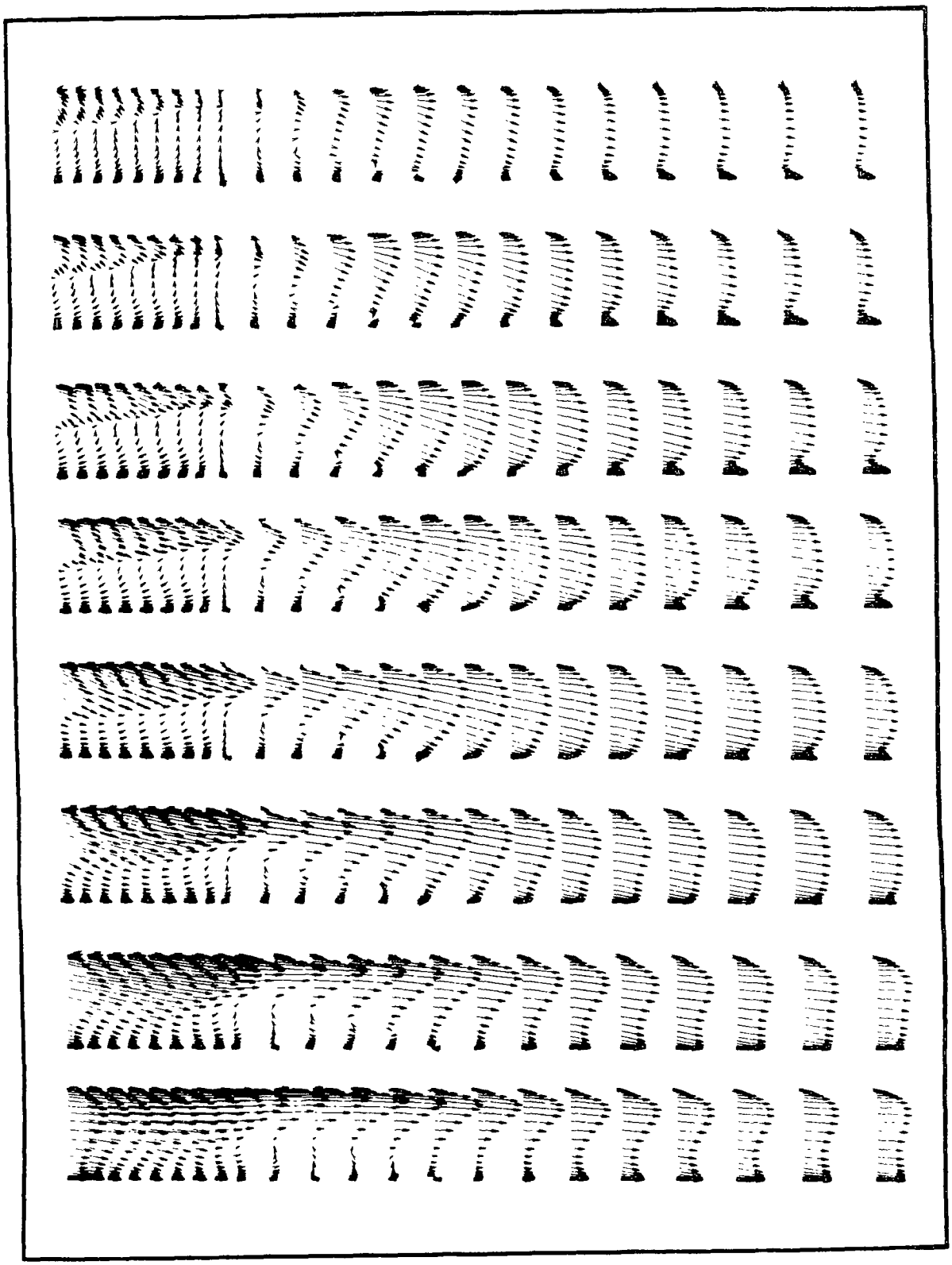

Fig. 3-19 Components of Velocity Vectors (for Turbulent Computation) in Vertical Planes Starting at First Plane Off Inner Wall (Top) and Proceeding Eight Planes Toward Mid Channel in Region Just Past Mid turn in Square Duct to Three Duct Widths Downstream

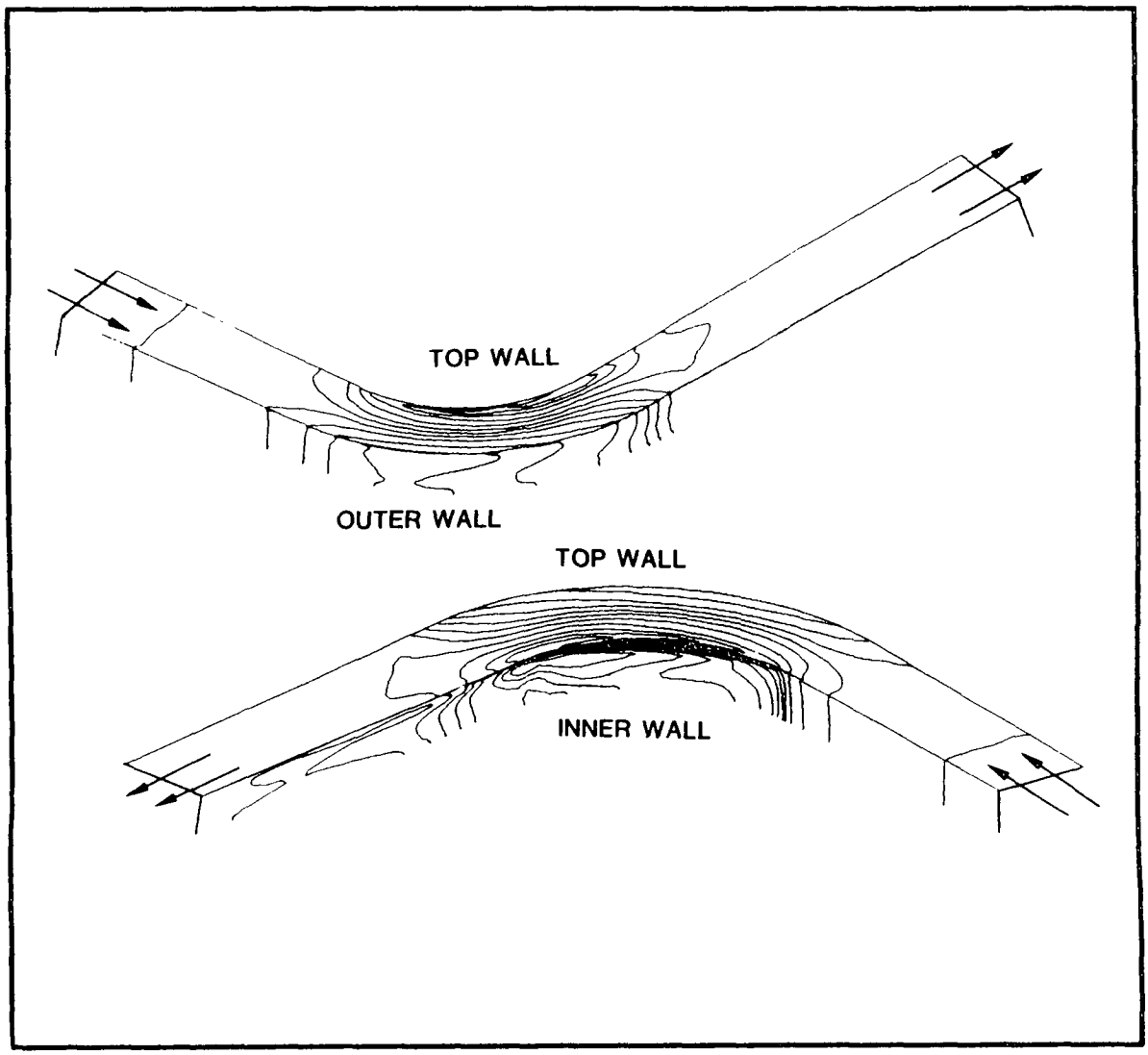

Fig. 3-20 Surface Pressure Contours for Turbulent Computation in Square Duct Using INS3D with Standard k-ε Model

flow swirl structure evolves in a manner similar to the laminar case with the exception that the position of the primary clockwise rotating pattern is more tightly fitted into the upper inner wall corner of the duct. This is also accompanied by a streamwise adverse pressure gradient on the inner wall which is stronger near the symmetry plane than at the upper wall. The pressure hill causes the boundary layer to separate for a small region downstream of the turn causing a separation bubble on this wall.

Evidence of this is clearly visible in the plots shown in Figs. 3-17 through 3-19. In Fig. 3-20, we display the pressure distribution on the inner, outer, and upper wall of the duct.

4. CONCLUDING REMARKS

The Computational Mechanics staff at the Lockheed-Huntsville Engineering Center have worked closely with NASA-MSFC Computational Fluid Dynamics personnel to provide NASA with up-to-date analytical computational results and computational tools for SSME engine redesign. Full three-dimensional laminar and turbulent SSME/HGM computations have been performed and results compared to water flow and air flow measurements (Ref. 1). Experience was acquired with two state-of-the-art Incompressible Navier-Stokes codes and both have been implemented on the NASA-MSFC supercomputer facility. Results of the latter work is reported in this document. Each of the codes, INS3D and FDNS3D, yield essentially the same results using identical computational grids. The INS3D code is, however, much less storage intensive and is recommended for large, multi-zone, three-dimensional computations.

5. REFERENCES

1. Roger, R.P., "Viscous Flow Computations for Elliptical Two-Duct Version of the SSME Hot Gas Manifold," LMSC-HEC TR D065161, Lockheed Missiles & Space Company, Huntsville, Ala., March 1986.

2. Kwak, D., J.L.C. Chang, S.P. Shanks, and S.R. Chakravarthy, "A Three-Dimensional Imcompressible Navier-Stokes Flow Solver Using Primitive Variable," AIAA J., Vol. 24, No. 3, March 1986.

3. Chen, Y.S., "A Computer Code for Three-Dimensional Incompressible Flows Using Nonorthogonal Body-Fitted Coordinate System," NASA-CR-178818, March 1986.

4. Chen, Y.S., " A Curvilinear CFD Code Using a Second Order Upwinding Scheme for High Reynolds Number Flow," to be published (private communication).

5. Chorin, A.J., "A Numerical Method for Solving Incompressible Viscous Flow Problems," J. Comp. Physics, Vol. 2, 1967, pp. 12-26.

6. Steger, J.L., and P. Kutler, "Implicit Finite-Difference Procedures for the Computation of Vortex Wakes," AIAA J., Vol. 15, No. 4, 1977, pp. 581-590.

7. Beam, R.M., and R.F. Warming, "An Implicit Finite-Difference Algorithm for Hyperbolic Systems in Conservation Law Form," J. Comp. Physics, Vol. 22, September 1976, pp. 87-110.

8. Warming, R.F., and R.M. Warming, "An Extension of A-Stability to Alternating Direction Implicit Methods," BIT, Vol. 19, 1979, p. 395.

9. Dwoyer, D.L., and F.C. Thames, "Accuracy and Stability of Time-Split Finite-Difference Schemes," AIAA Paper 81-1005, 1984.

10. South, J.C., "Recent Advances in Computational Transonic Aerodynamics," AIAA Paper 85-0366, 1985.

11. Briley, W.R., R.C. Buggeln, and H. McDonald, "Solution of the Three-Dimensional Navier-Stokes Equations for a Steady Laminar Horseshoe Vortex Flow," AIAA Paper 85-1520, 1985.

12. Chen, Y.S., "Development of a High Accuracy Finite Difference Upwinding Differencing Scheme for Viscous Flow Computations," Fifth SSME CFD Working Group Meeting, NSAA-Marshall Space Flight Center, Ala., April 1987.

13. Launder, B.E., and D.B. Spalding, Mathematical Models of Turbulence, Academic Press, New York, 1972.

14. Lilly, D.G., and D.L. Rhode, "A Computer Code for Swirling Turbulent Axisymmetric Recirculating Flow in Practical Isothermal Combustion Geometries," NASA CR-3442, 1982.

15. Bai, S.D., "Numerical Simulation of Multiple Jet Iteration," Ph.D. Thesis, University of Alabama in Huntsville, 1986.

16. Launder, B.E., and D.B. Spalding, "The Numerical Computation of Turbulent Flows," Comp. Meth. in Appl. Mech. and Eng., North-Holland, Vol. 3, p. 269, 1974.

Appendix A

INSED k-ε TURBULENCE MODEL SUBROUTINE LISTING

```
C*********************************************************************
      SUBROUTINE KETURB
     1 (RENL,VNUT,ITT,ITMAX)
C***** SET DIMENSIONS ***********************************************
      DIMENSION SS(54,31,27,6),VNUT(46656)
      COMMON/XYZQ/Q(4,46656),X(46656),Y(46656),Z(46656)
      COMMON/FDNS1/ DK(46656),DE(46656),TT(1),VISE(46656)
      COMMON
     1/VAR/UCO(46656),VCO(46656),WCO(46656),F1(46656)
     1/TRAN/ TJO(46656),DSX(46656),DSY(46656),DSZ(46656)
      COMMON
     1/PROP/ DEN(46656),VISC,DENIN,FLOWIN,DENN1
     1/TUR/ SIGK,SIGE,CMU,C1,C2,CMU1,CMU2,E,CK,HINUM,SMNUM,ANV1(7000),
     2 YN(7000),YN1(7000),SINX(7000),SINY(7000),SINZ(7000),ANW1(7000),
     3 IBC(7000),JBC(7000),KBC(7000),IITY(7000),
     4 GEN(46656),MC(46656),IJLO(46656),IITO
      COMMON
     1/COEF/ AP(46656),SU(46656),SP(46656),SUK(46656),
     2   AE(46656),AW(46656),AN(46656),
     3   AS(46656),AT(46656),AB(46656),AP0(46656)
      COMMON
     1/LIMT/ L,M,LT,MT,L1,L2,M1,M2,ISWK,ALK,ALVIS,N,N1,N2,IG,NT,
     3   ISU,ITU,JSU,JTU,KSU,KTU,CBE,L3,L4,M3,M4,N3,N4,ERRK,ERRDE
C
C---------------CONVERT INS3D GEOMETRY TO FDNS3D--------------------
C
      LLMAX=31
      KKMAX=27
      JJMAX=54

      NNTOT=LLMAX*KKMAX*JJMAX
C
      IF (ITT.EQ.ITMAX) THEN
      IF(ITMAX.LT.10) GO TO 999
C
      PRINT *, 'WRITING RESTART QQQ AT ITT = ',ITT
      DO 692 II=1,JJMAX
      DO 692 JJ2=1,LLMAX
      DO 692 KK2=1,KKMAX
      INOD= II+(KK2-1)*JJMAX+(JJ2-1)*JJMAX*KKMAX
      SS(II,JJ2,KK2,1)= 1.
      SS(II,JJ2,KK2,6)= 1.
      SS(II,JJ2,KK2,2)= Q(2,INOD)
      SS(II,JJ2,KK2,3)= Q(3,INOD)
      SS(II,JJ2,KK2,4)= Q(4,INOD)
      SS(II,JJ2,KK2,5)= Q(1,INOD)/0.4+0.5* (Q(2,INOD)**2
     1     + Q(3,INOD)**2+Q(4,INOD)**2)
  692 CONTINUE
      FSMACH=0.0
      ALPHA=0.0
      TIME=1.
      WRITE(23,190) JJMAX,LLMAX,KKMAX
      WRITE(23,195) FSMACH,ALPHA,RENL,TIME
      WRITE(23,195) ((((SS(II,JJ2,KK2,N6),II=1,JJMAX),JJ2=1,LLMAX),
     1             KK2=1,KKMAX),N6=1,6)
C
C------------------ TAGGING K-E & TUR. VISC C---------------------
C
      WRITE(23,195) (DK(II),II=1,NNTOT), (DE(II),II=1,NNTOT)
     .          ,(VNUT(II),II=1,NNTOT)
```

```
      CLOSE(23)
      IDB=1
      IF(IDB.EQ.1) GO TO 999
C
C-----------------------------XYZ FILE C-------------------------
C
      PRINT *, 'WRITING RESTART XYZ AT ITT = ',ITT
      DO 694 II=1,JJMAX
      DO 694 JJ2=1,LLMAX
      DO 694 KK2=1,KKMAX
      INOD= II+(KK2-1)*JJMAX+(JJ2-1)*JJMAX*KKMAX
      SS(II,JJ2,KK2,1)= X(INOD)
      SS(II,JJ2,KK2,2)= Y(INOD)
      SS(II,JJ2,KK2,3)= Z(INOD)
  694 CONTINUE
C     OPEN(7,FILE='SQD52X.BIN',STATUS='NEW',FORM='UNFORMATTED')
      WRITE(2,190) JJMAX,LLMAX,KKMAX
      WRITE(2,195)(((SS(II,JJ2,KK2,1),II=1,JJMAX),JJ2=1,LLMAX),
     1             KK2=1,KKMAX),
     2         (((SS(II,JJ2,KK2,2),II=1,JJMAX),JJ2=1,LLMAX),
     3             KK2=1,KKMAX),
     4         (((SS(II,JJ2,KK2,3),II=1,JJMAX),JJ2=1,LLMAX),
     5             KK2=1,KKMAX)
  999 CONTINUE
      PRINT *,' STOP AT KETURB NT=NTMAX & ITT=',ITT
      STOP
      END IF
  190 FORMAT(3I6)
  195 FORMAT(8(E12.5,1X))
C
      DO 197 II=1,JJMAX
      DO 197 JJ2=1,LLMAX
      DO 197 KK2=1,KKMAX
      DO 197 JVAR=1,4
      INOD= II+(KK2-1)*JJMAX+(JJ2-1)*JJMAX*KKMAX
      SS(II,JJ2,KK2,JVAR)= Q(JVAR,INOD)
  197 CONTINUE
C
      CALL IVA4(IDATA,IG,NLIMT,IDUM,2,2,1,0)
      CALL IVA4(IMN,JMN,KMN,ISWK,10,21,30,6)
      CALL RVA4(RDUM,ALK,ALVIS,RDUM,1000.,0.2,0.3,0.)
C
C-----CONSTANTS
C
      DENN1=0.1160*22.876/63.7/0.4500
      SIGF = 0.85
      CALL RVA4(DENIN,VISC,SIGU,SIGK, 1.0, 1.0/RENL, 1.0,  1.0)
      CALL RVA4(SIGE,CMU,C1,C2, 1.30, 0.09, 1.43, 1.92)
      CALL RVA4(E,CK,PI,P1,     9.01069, 0.40000, 3.141592654, 1.000)
      CALL RVA4(SIGK,SIGE,C1,C2, 1.0, 1.15, 1.15, 1.9)
      HINUM=1.E30
      SMNUM=1.E-30
      CMU1=CMU**0.25
      CMU2=CMU**0.75
C
C------------------EXAMPLE DATA (90-DEG BEND)-----------------------
C
        CALL IVA4(L,M,N,LL,JJMAX,LLMAX,KKMAX,0)
        CALL IVA4(L0,M0,N0,LT, L+1, M+1, N+1, L-1)
        CALL IVA4(MT,NT,L1,L2, M-1, N-1,   0,   0)
```

```
        CALL IVA4(M1,M2,N1,N2,   0,   0,   0,   0)
        CALL IVA4(ISU,JSU,KSU,ITU, 2, 2, 2, LT)
        CALL IVA4(JTU,KTU,L3,L4, MT, NT,  0,  0)
        CALL IVA4(M3,M4,N3,N4, 0, 0, 0, 0)
C
C------------------SQUARE DUCT 90 DEGREE BEND --------------------
C
C     WRITE(6,1112)
 1112 FORMAT(2X,' INTO KETURB ##')
C
C---------------GET BOUNDARY CONTROL PARAMETERS-----------------
C
      CALL DIRCOS
C
C-------------------INITIALIZE DENSITY/VISCOSITY----------------
C
      TURINT=0.0003
      ALAMDA=0.01
      INODX= 1
      INODY= 1+(M-1)*L*N
      RLENG=ABS(Y(INODX)-Y(INODY))
      DO 121 INOD=1,NNTOT
      DEN(INOD) =DENIN
      VNUT(INOD)=0.0
      VISE(INOD)=VISC
      UMEAN2=(Q(2,INOD)**2+Q(3,INOD)**2+Q(4,INOD)**2)
      IF(ITT.GT.1) GO TO 121
      DK(INOD)=0.5*TURINT*UMEAN2
      DE(INOD)=DK(INOD)**1.5/(ALAMDA*RLENG)
 121  CONTINUE
C
C----------------SET TURBULENCE QUANTITIES-----------------------
C
      IF(ITT.GT.1) GO TO 123
      DO 122 INOD=1,NNTOT
      IF(MC(INOD).EQ.0) THEN
         DK(INOD)=AMAX1(1.E-05,DK(INOD))
         DE(INOD)=AMAX1(1.E-05,DE(INOD))
      ENDIF
 122  CONTINUE
 123  CONTINUE
C
C--------------CALCULATE GRID TRANSFORMATION COEFFICIENTS---------
C
      CALL TRANF(ITT)
C
      DO 800 I=1,L
      DO 800 J=1,M
      DO 800 K=1,N
C
      INOD= I+(K-1)*L+(J-1)*L*N
      IP1=MIN0(I+1,L)
      IM1=MAX0(I-1,1)
      JP1=MIN0(J+1,M)
      JM1=MAX0(J-1,1)
      KP1=MIN0(K+1,N)
      KM1=MAX0(K-1,1)
      PEW=0.5
      PNS=0.5
      PTB=0.5
```

```
      IF(I.EQ.1.OR.I.EQ.L) PEW=1.0
      IF(J.EQ.1.OR.J.EQ.M) PNS=1.0
      IF(K.EQ.1.OR.K.EQ.N) PTB=1.0
C
      INOD= I  +(K-1)  *L+(J-1)  *L*N
      IECC= IP1+(K-1)  *L+(J-1)  *L*N
      IWCC= IM1+(K-1)  *L+(J-1)  *L*N
      ICNC= I  +(K-1)  *L+(JP1-1)*L*N
      ICSC= I  +(K-1)  *L+(JM1-1)*L*N
      ICCT= I  +(KP1-1)*L+(J-1)  *L*N
      ICCB= I  +(KM1-1)*L+(J-1)  *L*N
C
      P1=PEW*(X(IECC)-X(IWCC))
      P2=PNS*(X(ICNC)-X(ICSC))
      P3=PTB*(X(ICCT)-X(ICCB))
      Q1=PEW*(Y(IECC)-Y(IWCC))
      Q2=PNS*(Y(ICNC)-Y(ICSC))
      Q3=PTB*(Y(ICCT)-Y(ICCB))
      R1=PEW*(Z(IECC)-Z(IWCC))
      R2=PNS*(Z(ICNC)-Z(ICSC))
      R3=PTB*(Z(ICCT)-Z(ICCB))
      PTR=1.0/(P1*(Q2*R3-Q3*R2)-P2*(Q1*R3-Q3*R1)+P3*(Q1*R2-Q2*R1))
      CXC=PTR*(Q2*R3-Q3*R2)
      CYC=-PTR*(P2*R3-P3*R2)
      CZC=PTR*(P2*Q3-P3*Q2)
      EXC=-PTR*(Q1*R3-Q3*R1)
      EYC=PTR*(P1*R3-P3*R1)
      EZC=-PTR*(P1*Q3-P3*Q1)
      SXC=PTR*(Q1*R2-Q2*R1)
      SYC=-PTR*(P1*R2-P2*R1)
      SZC=PTR*(P1*Q2-P2*Q1)
      TJOD=TJO(INOD)*DEN(INOD)
      UCO(INOD)=TJOD*(Q(2,INOD)*CXC+Q(3,INOD)*CYC+Q(4,INOD)*CZC)
      VCO(INOD)=TJOD*(Q(2,INOD)*EXC+Q(3,INOD)*EYC+Q(4,INOD)*EZC)
      WCO(INOD)=TJOD*(Q(2,INOD)*SXC+Q(3,INOD)*SYC+Q(4,INOD)*SZC)
 800  CONTINUE
C
C----------------EVALUATE TURBULENT VISCOSITY----------------
C
      IF(ITT.GT.1) GO TO 316
      DO 310 INOD=1,NNTOT
      IF(DK(INOD) .LE. SMNUM) DK(INOD)=SMNUM
      IF(DE(INOD) .LE. SMNUM) DE(INOD)=SMNUM
      IF(DE(INOD) .LE. SMNUM) GO TO 312
      TURVIS=DEN(INOD)*CMU*DK(INOD)**2/DE(INOD)+VISC
      GO TO 314
 312  TURVIS=VISC
 314  CONTINUE
      VISE(INOD)=VISE(INOD)+ALVIS*(TURVIS-VISE(INOD))
 310  CONTINUE
 316  CONTINUE
C
C-----------------CALCULATE INLET MASS FLOW RATE-----------------
C
      FLOWIN=0.0
      AREA=0.0
      I=1
      DO 45 J=2,MT
      DO 45 K=1,N
      INOD= I  +(K-1)  *L+(J-1)  *L*N
```

```
      AREA=AREA+1.0
      FLOWIN=FLOWIN+UCO(INOD)
   45 CONTINUE
C      WRITE(6,300) IDATA,AREA,FLOWIN
      ITER=C
C--------------------SOLUTION PROCEDURES START-------------------------
C
    1 CONTINUE
C############################
C############################
      CALL CALCK(ITT)
C############################
C############################
      CALL CALCE(ITT)
C############################
C############################
C
      IF(ITER.GT.1) GO TO 844
C
C-------------------BOUNDARY CONDITIONS--------------------
C
      DO 15 I=2,L
      DO 15 J=2,MT
      INOD1= I      +(J-1)  *L*N
      INOD2= I  + L+(J-1)  *L*N
      Q(2,INOD1)=Q(2,INOD2)
      Q(3,INOD1)=Q(3,INOD2)
   15 CONTINUE
C
C-------------EAST OUT (BASED ON INFLOW MASS FLOW RATE)-----------------
C
      UINC=0.0
      FLOW=0.0
      DAREA=0.0
      DO 50 J=2,MT
      DO 50 K=1,N
      INOD1= LT  +(K-1)  *L+(J-1)  *L*N
      INOD2= L   +(K-1)  *L+(J-1)  *L*N
      FLOW=FLOW+UCO(INOD1)
      DAREA=DAREA+TJO(INOD2)**0.67
   50 CONTINUE
      UINC=(FLOW-FLOWIN)
      DO 51 J=2,MT
      DO 51 K=1,N
      INOD1= LT  +(K-1)  *L+(J-1)  *L*N
      INOD2= L   +(K-1)  *L+(J-1)  *L*N
      VCO(INOD2)=VCO(INOD1)
      WCO(INOD2)=WCO(INOD1)
   51 UCO(INOD2)=UCO(INOD1)-UINC*TJO(INOD2)**0.67/DAREA
C
      DO 52 J=2,MT
      DO 52 K=2,NT
C
      IECC= L  +(K-1) *L+ (J-1) *L*N
      IWCC= LT +(K-1) *L+ (J-1) *L*N
      ICNC= L  +(K-1) *L+ (J)   *L*N
      ICSC= L  +(K-1) *L+ (J-2) *L*N
      ICCT= L  +(K)   *L+ (J-1) *L*N
      ICCB= L  +(K-2) *L+ (J-1) *L*N
C
```

```
      IF(MC(IECC).EQ.0) THEN
       P1=1.0*(X(IECC)-X(IWCC))
       P2=0.5*(X(ICNC)-X(ICSC))
       P3=0.5*(X(ICCT)-X(ICCB))
       Q1=1.0*(Y(IECC)-Y(IWCC))
       Q2=0.5*(Y(ICNC)-Y(ICSC))
       Q3=0.5*(Y(ICCT)-Y(ICCB))
       R1=1.0*(Z(IECC)-Z(IWCC))
       R2=0.5*(Z(ICNC)-Z(ICSC))
       R3=0.5*(Z(ICCT)-Z(ICCB))
       PTR=1.0/(P1*(Q2*R3-Q3*R2)-P2*(Q1*R3-Q3*R1)+P3*(Q1*R2-Q2*R1))
       CXC=PTR*(Q2*R3-Q3*R2)
       CYC=-PTR*(P2*R3-P3*R2)
       CZC=PTR*(P2*Q3-P3*Q2)
       EXC=-PTR*(Q1*R3-Q3*R1)
       EYC=PTR*(P1*R3-P3*R1)
       EZC=-PTR*(P1*Q3-P3*Q1)
       SXC=PTR*(Q1*R2-Q2*R1)
       SYC=-PTR*(P1*R2-P2*R1)
       SZC=PTR*(P1*Q2-P2*Q1)
       Q(2,IECC)= UCO(IECC)*(EYC*SZC-SYC*EZC)+
     1           VCO(IECC)*(SYC*CZC-CYC*SZC)+
     2           WCO(IECC)*(CYC*EZC-EYC*CZC)
       Q(3,IECC)= UCO(IECC)*(EZC*SXC-SZC*EXC)+
     1           VCO(IECC)*(SZC*CXC-CZC*SXC)+
     2           WCO(IECC)*(CZC*EXC-EZC*CXC)
       Q(4,IECC)= UCO(IECC)*(EXC*SYC-SXC*EYC)+
     1           VCO(IECC)*(SXC*CYC-CXC*SYC)+
     2           WCO(IECC)*(CXC*EYC-EXC*CYC)
C      TT(IECC)=TT(IWCC)
       DK(IECC)=DK(IWCC)
       DE(IECC)=DE(IWCC)
       ENDIF
   52 CONTINUE
  844 CONTINUE
       IF(ITER.GT.1) THEN
       DO 521 J=2,MT
       DO 521 K=2,NT
       IECC= L  +(K-1) *L+ (J-1) *L*N
       IWCC= LT +(K-1) *L+ (J-1) *L*N
       DK(IECC)=DK(IWCC)
       DE(IECC)=DE(IWCC)
  521 CONTINUE
       END IF
C
C----------------EVALUATE TURBULENT VISCOSITY-------------------------
C
       DO 318 INOD=1,NNTOT
       IF(DK(INOD) .LE. SMNUM) DK(INOD)=SMNUM
       IF(DE(INOD) .LE. SMNUM) DE(INOD)=SMNUM
       IF(DE(INOD) .LE. SMNUM) GO TO 322
       TURVIS=DEN(INOD)*CMU*DK(INOD)**2/DE(INOD)+VISC
       GO TO 324
  322  TURVIS=VISC
  324  CONTINUE
       VISE(INOD)=VISE(INOD)+ALVIS*(TURVIS-VISE(INOD))
       VNUT(INOD)=TURVIS-VISC
       IF(VNUT(INOD).LE.0.0) VNUT(INOD)=0.0
  318  CONTINUE
C
```

```
C---------------------CONVERGENCE CHECK---------------------------------
C
      EREXT=0.01
      NODMN=IMN+(KMN-1)*L+(JMN-1)*L*N
      DO 911 I=1,L
      FLOW=0.0
      UMON=0.0
      DO 912 J=2,MT
      DO 912 K=1,NT
      INOD= I  +(K-1)  *L+(J-1)  *L*N
      FLOW=FLOW+UCO(INOD)
      UMON=UMON+UCO(INOD)*UCO(INOD)
 912  CONTINUE
 911  CONTINUE
      ERRK=ERRK/(0.5*TURINT*FLOW*FLOW)
      IF(ITER.EQ.1) WRITE(6,826)
  826 FORMAT(10X,'ITER  ','NODMN ', 4X,'U',10X,'DK',9X,
     . 'DE',9X,'ERRK',7X,'VISE',7X,'VNUT',7X,'P'/)
      WRITE(6,824) ITER,NODMN,Q(2,NODMN),DK(NODMN),DE(NODMN)
     .             ,ERRK,VISE(NODMN),VNUT(NODMN),Q(1,NODMN)
      IF(ITER .GE. 20 .AND. ERRK .GT. 1.E05) GO TO 99
      IF(ITER .GE. NLIMT .OR. ERRK .LE. EREXT) GO TO 99
      ITER=ITER+1
      ERRK=0.0
      GO TO 1
  824 FORMAT(2X,'MONITOR ',2I6,7(1X,E10.4))
C
C-------------------PRINT OUT SOLUTIONS---------------------------
C
   99 CONTINUE
  100 FORMAT(15I5)
  300 FORMAT(1X,I5,6E11.3)
  400 FORMAT(1X)
C 500  FORMAT(4I5,3E12.4)
C
C#####################################################################
C
      DO 690 II=1,JJMAX
      DO 690 JJ2=1,LLMAX
      DO 690 KK2=1,KKMAX
      DO 690 JVAR=1,4
      INOD= II+(KK2-1)*JJMAX+(JJ2-1)*JJMAX*KKMAX
      Q(JVAR,INOD) = SS(II,JJ2,KK2,JVAR)
  690 CONTINUE
      RETURN
      END
C
C############################
      SUBROUTINE DIRCOS
C############################
C
      COMMON/FDNS1/ DK(46656),DE(46656),TT(1),VISE(46656)
      COMMON/XYZQ/Q(4,46656),X(46656),Y(46656),Z(46656)
      COMMON
     1/TRAN/ TJO(46656),DSX(46656),DSY(46656),DSZ(46656)
      COMMON
     1/TUR/ SIGK,SIGE,CMU,C1,C2,CMU1,CMU2,E,CK,HINUM,SMNUM,ANV1(7000),
     2 YN(7000),YN1(7000),SINX(7000),SINY(7000),SINZ(7000),ANW1(7000),
     3 IBC(7000),JBC(7000),KBC(7000),IITY(7000),
     4 GEN(46656),MC(46656),IJLO(46656),IITO
```

```
     1/LIMT/ L,M,LT,MT,L1,L2,M1,M2,ISWK,
     2   ALK,ALVIS,N,N1,N2,IG,NT,
     3   ISU,ITU,JSU,JTU,KSU,KTU,CBE,L3,L4,M3,M4,N3,N4,
     4   ERRK,ERRDE
C-----------------------------------------------------------------------
C
C-------------SET DOMAIN BLOCKAGE CONTROL PARAMETER-------------------
C
C-----SCALAR BLOCKAGE :   MC(INOD)=1
C
      DO 777 I=1,L
      DO 777 J=1,M
      DO 777 K=1,N
      INOD= I + (K-1)*L + (J-1)*L*N
      MC(INOD)=0
      IF(I.GE.L1.AND.I.LE.L2.AND.J.GE.M1.AND.J.LE.M2.AND.
     1                           K.GE.N1.AND.K.LE.N2) MC(INOD)=1
      IF(I.GE.L3.AND.I.LE.L4.AND.J.GE.M3.AND.J.LE.M4.AND.
     1                           K.GE.N3.AND.K.LE.N4) MC(INOD)=1
C
C-----FOR SQUARE DUCT
C
      IF(J.EQ.1.OR.J.EQ.M.OR.K.EQ.N)                    MC(INOD)=1
 777  CONTINUE
C
C-----CALCULATE BOUNDARY GRID SIZES AND ORIENTATIONS
C
      III=1
      DO 30 I=2,LT
      DO 30 J=2,MT
      DO 30 K=2,NT
C
      INOD= I + (K-1)*L + (J-1)*L*N
      IF(MC(INOD) .NE. 0) GO TO 30
C
C---  27 NODE IDENDIFICATIONS
C  CENTER PLANE
C
      IWCC= I-1 +(K-1)*L +(J-1)*L*N
      IWNC= I-1 +(K-1)*L +(J)  *L*N
      IWSC= I-1 +(K-1)*L +(J-2)*L*N
C      ICCC= I   +(K-1)*L +(J-1)*L*N
      ICNC= I   +(K-1)*L +(J)  *L*N
      ICSC= I   +(K-1)*L +(J-2)*L*N
      IECC= I+1 +(K-1)*L +(J-1)*L*N
      IENC= I+1 +(K-1)*L +(J)  *L*N
      IESC= I+1 +(K-1)*L +(J-2)*L*N
C
C  TOP PLANE
C
      IWCT= I-1 +(K)  *L +(J-1)*L*N
C      IWNT= I-1 +(K)  *L +(J)  *L*N
C      IWST= I-1 +(K)  *L +(J-2)*L*N
      ICCT= I   +(K)  *L +(J-1)*L*N
      ICNT= I   +(K)  *L +(J)  *L*N
      ICST= I   +(K)  *L +(J-2)*L*N
      IECT= I+1 +(K)  *L +(J-1)*L*N
C      IENT= I+1 +(K)  *L +(J)  *L*N
C      IEST= I+1 +(K)  *L +(J-2)*L*N
C
```

```
C  BOTTON PLANE
C
      IWCB= I-1 +(K-2)*L +(J-1)*L*N
C      IWNB= I-1 +(K-2)*L +(J)  *L*N
C      IWSB= I-1 +(K-2)*L +(J-2)*L*N
      ICCB= I   +(K-2)*L +(J-1)*L*N
      ICNB= I   +(K-2)*L +(J)  *L*N
      ICSB= I   +(K-2)*L +(J-2)*L*N
      IECB= I+1 +(K-2)*L +(J-1)*L*N
C      IENB= I+1 +(K-2)*L +(J)  *L*N
C      IESB= I+1 +(K-2)*L +(J-2)*L*N
C
      MCT=MC(IECC)+MC(IWCC)+MC(ICNC)+MC(ICSC)+
     1    MC(ICCT)+MC(ICCB)
      IF(MCT .EQ. 0) GO TO 30
      IF(MC(ICNC) .EQ. 0) GO TO 2
C
C-----NORTH
C
      IBC(III)=I
      JBC(III)=J
      KBC(III)=K
      IITY(III)=1
      I1=I+1
      I2=I-1
      K1=K+1
      K2=K-1
      IF(MC(IWNC).EQ.0) I2=I
      IF(MC(IENC).EQ.0) I1=I
      IF(MC(ICNB).EQ.0) K2=K
      IF(MC(ICNT).EQ.0) K1=K
      J1=J+1
      J2=J-1
C
      INOD3= I +(K -1)  *L+(J1-1) *L*N
      INOD4= I +(K -1)  *L+(J2-1) *L*N
      INOD5= I +(K1-1)  *L+(J1-1) *L*N
      INOD6= I +(K2-1)  *L+(J1-1) *L*N
      INODC= I1 +(K -1) *L+(J1-1) *L*N
      INODH= I2 +(K -1) *L+(J1-1) *L*N
C
      P1=(Y(INOD5)-Y(INOD6))*(Z(INODC)-Z(INODH))-
     1   (Z(INOD5)-Z(INOD6))*(Y(INODC)-Y(INODH))
      P2=(Z(INOD5)-Z(INOD6))*(X(INODC)-X(INODH))-
     1   (X(INOD5)-X(INOD6))*(Z(INODC)-Z(INODH))
      P3=(X(INOD5)-X(INOD6))*(Y(INODC)-Y(INODH))-
     1   (Y(INOD5)-Y(INOD6))*(X(INODC)-X(INODH))
      PQ=SQRT(P1*P1+P2*P2+P3*P3)
      P4=P1/PQ
      P5=P2/PQ
      P6=P3/PQ
      R1=ABS(1.-P4**2)
      R2=ABS(1.-P5**2)
      R3=ABS(1.-P6**2)
      SINX(III)=SQRT(R1)
      SINY(III)=SQRT(R2)
      SINZ(III)=SQRT(R3)
      Q1=X(INOD)-X(INOD3)
      Q2=Y(INOD)-Y(INOD3)
      Q3=Z(INOD)-Z(INOD3)
```

```
      AA=SQRT((Q1-P4)**2+(Q2-P5)**2+(Q3-P6)**2)
      CC=1.0
      BB=SQRT(Q1*Q1+Q2*Q2+Q3*Q3)
      COTH=(BB*BB+CC*CC-AA*AA)/(2*BB*CC)
      YN(III)=BB*ABS(COTH)
      Q1=X(INOD3)-X(INOD4)
      Q2=Y(INOD3)-Y(INOD4)
      Q3=Z(INOD3)-Z(INOD4)
      BB=SQRT(Q1*Q1+Q2*Q2+Q3*Q3)
      AA=SQRT((Q1-P4)**2+(Q2-P5)**2+(Q3-P6)**2)
      COTH=(BB*BB+CC*CC-AA*AA)/(2*BB*CC)
      YN1(III)=(BB*ABS(COTH)+YN(III))*0.5
      IJLO(INOD)=III
      III=III+1
    2 CONTINUE
      IF(MC(ICSC) .EQ. 0) GO TO 3
C
C-----SOUTH
C
      IBC(III)=I
      JBC(III)=J
      KBC(III)=K
      IITY(III)=2
      I1=I+1
      I2=I-1
      K1=K+1
      K2=K-1
      IF(MC(IWSC).EQ.0) I2=I
      IF(MC(IESC).EQ.0) I1=I
      IF(MC(ICSB).EQ.0) K2=K
      IF(MC(ICST).EQ.0) K1=K
      J1=J-1
      J2=J+1
C
      INOD3= I +(K -1)  *L+(J1-1) *L*N
      INOD4= I +(K -1)  *L+(J2-1) *L*N
      INOD5= I +(K1-1)  *L+(J1-1) *L*N
      INOD6= I +(K2-1)  *L+(J1-1) *L*N
      INODC= I1 +(K -1) *L+(J1-1) *L*N
      INODH= I2 +(K -1) *L+(J1-1) *L*N
C
      P1=(Y(INOD5)-Y(INOD6))*(Z(INODC)-Z(INODH))-
     1   (Z(INOD5)-Z(INOD6))*(Y(INODC)-Y(INODH))
      P2=(Z(INOD5)-Z(INOD6))*(X(INODC)-X(INODH))-
     1   (X(INOD5)-X(INOD6))*(Z(INODC)-Z(INODH))
      P3=(X(INOD5)-X(INOD6))*(Y(INODC)-Y(INODH))-
     1   (Y(INOD5)-Y(INOD6))*(X(INODC)-X(INODH))
      PQ=SQRT(P1*P1+P2*P2+P3*P3)
      P4=P1/PQ
      P5=P2/PQ
      P6=P3/PQ
      R1=ABS(1.-P4**2)
      R2=ABS(1.-P5**2)
      R3=ABS(1.-P6**2)
      SINX(III)=SQRT(R1)
      SINY(III)=SQRT(R2)
      SINZ(III)=SQRT(R3)
      Q1=X(INOD)-X(INOD3)
      Q2=Y(INOD)-Y(INOD3)
      Q3=Z(INOD)-Z(INOD3)
```

```
      AA=SQRT((Q1-P4)**2+(Q2-P5)**2+(Q3-P6)**2)
      CC=1.0
      BB=SQRT(Q1*Q1+Q2*Q2+Q3*Q3)
      COTH=(BB*BB+CC*CC-AA*AA)/(2*BB*CC)
      YN(III)=BB*ABS(COTH)
      Q1=X(INOD3)-X(INOD4)
      Q2=Y(INOD3)-Y(INOD4)
      Q3=Z(INOD3)-Z(INOD4)
      BB=SQRT(Q1*Q1+Q2*Q2+Q3*Q3)
      AA=SQRT((Q1-P4)**2+(Q2-P5)**2+(Q3-P6)**2)
      COTH=(BB*BB+CC*CC-AA*AA)/(2*BB*CC)
      YN1(III)=(BB*ABS(COTH)+YN(III))*0.5
      IJLO(INOD)=III
      III=III+1
    3 CONTINUE
      IF(MC(IECC) .EQ. 0) GO TO 4
C
C-----EAST
C
      IBC(III)=I
      JBC(III)=J
      KBC(III)=K
      IITY(III)=3
      J1=J+1
      J2=J-1
      K1=K+1
      K2=K-1
      IF(MC(IESC).EQ.0) J2=J
      IF(MC(IENC).EQ.0) J1=J
      IF(MC(IECB).EQ.0) K2=K
      IF(MC(IECT).EQ.0) K1=K
      I1=I+1
      I2=I-1
C
      INODA= I1 +(K1 -1)*L+(J-1)  *L*N
      INODB= I1 +(K2 -1)*L+(J-1)  *L*N
      INODC= I1 +(K -1) *L+(J1-1) *L*N
      INODD= I1 +(K -1) *L+(J2-1) *L*N
      INODE= I1 +(K -1) *L+(J-1)  *L*N
      INODF= I2 +(K -1) *L+(J-1)  *L*N
C
      P1=(Y(INODA)-Y(INODB))*(Z(INODC)-Z(INODD))-
     1   (Z(INODA)-Z(INODB))*(Y(INODC)-Y(INODD))
      P2=(Z(INODA)-Z(INODB))*(X(INODC)-X(INODD))-
     1   (X(INODA)-X(INODB))*(Z(INODC)-Z(INODD))
      P3=(X(INODA)-X(INODB))*(Y(INODC)-Y(INODD))-
     1   (Y(INODA)-Y(INODB))*(X(INODC)-X(INODD))
      PQ=SQRT(P1*P1+P2*P2+P3*P3)
      P4=P1/PQ
      P5=P2/PQ
      P6=P3/PQ
      R1=ABS(1.-P4**2)
      R2=ABS(1.-P5**2)
      R3=ABS(1.-P6**2)
      SINX(III)=SQRT(R1)
      SINY(III)=SQRT(R2)
      SINZ(III)=SQRT(R3)
      Q1=X(INOD)-X(INODE)
      Q2=Y(INOD)-Y(INODE)
      Q3=Z(INOD)-Z(INODE)
```

```
      AA=SQRT((Q1-P4)**2+(Q2-P5)**2+(Q3-P6)**2)
      CC=1.0
      BB=SQRT(Q1*Q1+Q2*Q2+Q3*Q3)
      COTH=(BB*BB+CC*CC-AA*AA)/(2*BB*CC)
      YN(III)=BB*ABS(COTH)
      Q1=X(INODE)-X(INODF)
      Q2=Y(INODE)-Y(INODF)
      Q3=Z(INODE)-Z(INODF)
      BB=SQRT(Q1*Q1+Q2*Q2+Q3*Q3)
      AA=SQRT((Q1-P4)**2+(Q2-P5)**2+(Q3-P6)**2)
      COTH=(BB*BB+CC*CC-AA*AA)/(2*BB*CC)
      YN1(III)=(BB*ABS(COTH)+YN(III))*0.5
      IJLO(INOD)=III
      III=III+1
    4 CONTINUE
      IF(MC(IWCC) .EQ. 0) GO TO 5
C
C-----WEST
C
      IBC(III)=I
      JBC(III)=J
      KBC(III)=K
      IITY(III)=4
      J1=J+1
      J2=J-1
      K1=K+1
      K2=K-1
      IF(MC(IWSC).EQ.0) J2=J
      IF(MC(IWNC).EQ.0) J1=J
      IF(MC(IWCB).EQ.0) K2=K
      IF(MC(IWCT).EQ.0) K1=K
      I1=I-1
      I2=I+1
C
      INODA= I1 +(K1 -1)*L+(J-1)  *L*N
      INODB= I1 +(K2 -1)*L+(J-1)  *L*N
      INODC= I1 +(K -1) *L+(J1-1) *L*N
      INODD= I1 +(K -1) *L+(J2-1) *L*N
      INODE= I1 +(K -1) *L+(J-1)  *L*N
      INODF= I2 +(K -1) *L+(J-1)  *L*N
C
      P1=(Y(INODA)-Y(INODB))*(Z(INODC)-Z(INODD))-
     1   (Z(INODA)-Z(INODB))*(Y(INODC)-Y(INODD))
      P2=(Z(INODA)-Z(INODB))*(X(INODC)-X(INODD))-
     1   (X(INODA)-X(INODB))*(Z(INODC)-Z(INODD))
      P3=(X(INODA)-X(INODB))*(Y(INODC)-Y(INODD))-
     1   (Y(INODA)-Y(INODB))*(X(INODC)-X(INODD))
      PQ=SQRT(P1*P1+P2*P2+P3*P3)
      P4=P1/PQ
      P5=P2/PQ
      P6=P3/PQ
      R1=ABS(1.-P4**2)
      R2=ABS(1.-P5**2)
      R3=ABS(1.-P6**2)
      SINX(III)=SQRT(R1)
      SINY(III)=SQRT(R2)
      SINZ(III)=SQRT(R3)
      Q1=X(INOD)-X(INODE)
      Q2=Y(INOD)-Y(INODE)
      Q3=Z(INOD)-Z(INODE)
```

```
      AA=SQRT((Q1-P4)**2+(Q2-P5)**2+(Q3-P6)**2)
      CC=1.0
      BB=SQRT(Q1*Q1+Q2*Q2+Q3*Q3)
      COTH=(BB*BB+CC*CC-AA*AA)/(2*BB*CC)
      YN(III)=BB*ABS(COTH)
      Q1=X(INODE)-X(INODF)
      Q2=Y(INODE)-Y(INODF)
      Q3=Z(INODE)-Z(INODF)
      BB=SQRT(Q1*Q1+Q2*Q2+Q3*Q3)
      AA=SQRT((Q1-P4)**2+(Q2-P5)**2+(Q3-P6)**2)
      COTH=(BB*BB+CC*CC-AA*AA)/(2*BB*CC)
      YN1(III)=(BB*ABS(COTH)+YN(III))*0.5
      IJLO(INOD)=III
      III=III+1
    5 CONTINUE
      IF(MC(ICCT) .EQ. 0) GO TO 6
C
C-----TOP
C
      IBC(III)=I
      JBC(III)=J
      KBC(III)=K
      IITY(III)=5
      I1=I+1
      I2=I-1
      J1=J+1
      J2=J-1
      IF(MC(IWCT).EQ.0) I2=I
      IF(MC(IECT).EQ.0) I1=I
      IF(MC(ICST).EQ.0) J2=J
      IF(MC(ICNT).EQ.0) J1=J
      K1=K+1
      K2=K-1
C
      INOD1= I +(K1 -1) *L+(J-1)  *L*N
      INOD2= I +(K2 -1) *L+(J-1)  *L*N
      INOD5= I +(K1-1)  *L+(J1-1) *L*N
      INOD7= I +(K1-1)  *L+(J2-1) *L*N
      INODA= I1 +(K1 -1)*L+(J-1)  *L*N
      INODG= I2 +(K1-1) *L+(J-1)  *L*N
C
      P1=(Y(INOD5)-Y(INOD7))*(Z(INODA)-Z(INODG))-
     1   (Z(INOD5)-Z(INOD7))*(Y(INODA)-Y(INODG))
      P2=(Z(INOD5)-Z(INOD7))*(X(INODA)-X(INODG))-
     1   (X(INOD5)-X(INOD7))*(Z(INODA)-Z(INODG))
      P3=(X(INOD5)-X(INOD7))*(Y(INODA)-Y(INODG))-
     1   (Y(INOD5)-Y(INOD7))*(X(INODA)-X(INODG))
      PQ=SQRT(P1*P1+P2*P2+P3*P3)
      P4=P1/PQ
      P5=P2/PQ
      P6=P3/PQ
      R1=ABS(1.-P4**2)
      R2=ABS(1.-P5**2)
      R3=ABS(1.-P6**2)
      SINX(III)=SQRT(R1)
      SINY(III)=SQRT(R2)
      SINZ(III)=SQRT(R3)
      Q1=X(INOD)-X(INOD1)
      Q2=Y(INOD)-Y(INOD1)
      Q3=Z(INOD)-Z(INOD1)
```

```
      AA=SQRT((Q1-P4)**2+(Q2-P5)**2+(Q3-P6)**2)
      CC=1.0
      BB=SQRT(Q1*Q1+Q2*Q2+Q3*Q3)
      COTH=(BB*BB+CC*CC-AA*AA)/(2*BB*CC)
      YN(III)=BB*ABS(COTH)
      Q1=X(INOD1)-X(INOD2)
      Q2=Y(INOD1)-Y(INOD2)
      Q3=Z(INOD1)-Z(INOD2)
      BB=SQRT(Q1*Q1+Q2*Q2+Q3*Q3)
      AA=SQRT((Q1-P4)**2+(Q2-P5)**2+(Q3-P6)**2)
      COTH=(BB*BB+CC*CC-AA*AA)/(2*BB*CC)
      YN1(III)=(BB*ABS(COTH)+YN(III))*0.5
      IJLO(INOD)=III
      III=III+1
    6 CONTINUE
      IF(MC(ICCB) .EQ. 0) GO TO 30
C
C ----BOTTON
C
      IBC(III)=I
      JBC(III)=J
      KBC(III)=K
      IITY(III)=6
      I1=I+1
      I2=I-1
      J1=J+1
      J2=J-1
      IF(MC(IWCB).EQ.0) I2=I
      IF(MC(IECB).EQ.0) I1=I
      IF(MC(ICSB).EQ.0) J2=J
      IF(MC(ICNB).EQ.0) J1=J
      K1=K-1
      K2=K+1
C
      INOD1= I +(K1 -1) *L+(J-1)  *L*N
      INOD2= I +(K2 -1) *L+(J-1)  *L*N
      INOD5= I +(K1-1)  *L+(J1-1) *L*N
      INOD7= I +(K1-1)  *L+(J2-1) *L*N
      INODA= I1 +(K1 -1)*L+(J-1)  *L*N
      INODG= I2 +(K1-1) *L+(J-1)  *L*N
C
      P1=(Y(INOD5)-Y(INOD7))*(Z(INODA)-Z(INODG))-
     1   (Z(INOD5)-Z(INOD7))*(Y(INODA)-Y(INODG))
      P2=(Z(INOD5)-Z(INOD7))*(X(INODA)-X(INODG))-
     1   (X(INOD5)-X(INOD7))*(Z(INODA)-Z(INODG))
      P3=(X(INOD5)-X(INOD7))*(Y(INODA)-Y(INODG))-
     1   (Y(INOD5)-Y(INOD7))*(X(INODA)-X(INODG))
      PQ=SQRT(P1*P1+P2*P2+P3*P3)
      P4=P1/PQ
      P5=P2/PQ
      P6=P3/PQ
      R1=ABS(1.-P4**2)
      R2=ABS(1.-P5**2)
      R3=ABS(1.-P6**2)
      SINX(III)=SQRT(R1)
      SINY(III)=SQRT(R2)
      SINZ(III)=SQRT(R3)
      Q1=X(INOD)-X(INOD1)
      Q2=Y(INOD)-Y(INOD1)
      Q3=Z(INOD)-Z(INOD1)
```

```
      AA=SQRT((Q1-P4)**2+(Q2-P5)**2+(Q3-P6)**2)
      CC=1.0
      BB=SQRT(Q1*Q1+Q2*Q2+Q3*Q3)
      COTH=(BB*BB+CC*CC-AA*AA)/(2*BB*CC)
      YN(III)=BB*ABS(COTH)
      Q1=X(INOD1)-X(INOD2)
      Q2=Y(INOD1)-Y(INOD2)
      Q3=Z(INOD1)-Z(INOD2)
      BB=SQRT(Q1*Q1+Q2*Q2+Q3*Q3)
      AA=SQRT((Q1-P4)**2+(Q2-P5)**2+(Q3-P6)**2)
      COTH=(BB*BB+CC*CC-AA*AA)/(2*BB*CC)
      YN1(III)=(BB*ABS(COTH)+YN(III))*0.5
      IJLO(INOD)=III
      III=III+1
   30 CONTINUE
      IITO=III-1
C      WRITE(6,100) L,M,N,IITO
  100 FORMAT(4I5)
      RETURN
      END
C
C*******************************
      SUBROUTINE TRANF(ITT)
C*******************************
C
      COMMON/FDNS1/ DK(46656),DE(46656),TT(1),VISE(46656)
      COMMON/XYZQ/Q(4,46656),X(46656),Y(46656),Z(46656)
      COMMON
     1/VAR/UCO(46656),VCO(46656),WCO(46656),F1(46656)
     1/TRAN/ TJO(46656),DSX(46656),DSY(46656),DSZ(46656)
      COMMON
     1/PROP/ DEN(46656),VISC,DENIN,FLOWIN,DENN1
     1/TUR/ SIGK,SIGE,CMU,C1,C2,CMU1,CMU2,E,CK,HINUM,SMNUM,ANV1(7000),
     2 YN(7000),YN1(7000),SINX(7000),SINY(7000),SINZ(7000),ANW1(7000),
     3 IBC(7000),JBC(7000),KBC(7000),IITY(7000),
     4 GEN(46656),MC(46656),IJLO(46656),IITO
      COMMON
     1/COEF/ AP(46656),SU(46656),SP(46656),SUK(46656),
     2   AE(46656),AW(46656),AN(46656),
     3   AS(46656),AT(46656),AB(46656),AP0(46656)
      COMMON
     1/LIMT/ L,M,LT,MT,L1,L2,M1,M2,ISWK,ALK,ALVIS,N,N1,N2,IG,NT,
     3   ISU,ITU,JSU,JTU,KSU,KTU,CBE,L3,L4,M3,M4,N3,N4,ERRK,ERRDE
C
C-----CALCULATE GRID TRANSFORMATION COEFFICIENTS
C
      DO 40 I=1,L
      DO 40 J=1,M
      DO 40 K=1,N
C
      INOD= I  +(K-1)*L +(J-1)*L*N
      IP1=MIN0(I+1,L)
      IM1=MAX0(I-1,1)
      JP1=MIN0(J+1,M)
      JM1=MAX0(J-1,1)
      KP1=MIN0(K+1,N)
      KM1=MAX0(K-1,1)
      PEW=0.5
      PNS=0.5
      PTB=0.5
```

```
      IF(I.EQ.1.OR.I.EQ.L) PEW=1.0
      IF(J.EQ.1.OR.J.EQ.M) PNS=1.0
      IF(K.EQ.1.OR.K.EQ.N) PTB=1.0
C
      INOD= I  +(K-1)  *L+(J-1)  *L*N
      IECC= IP1+(K-1)  *L+(J-1)  *L*N
      IWCC= IM1+(K-1)  *L+(J-1)  *L*N
      ICNC= I  +(K-1)  *L+(JP1-1)*L*N
      ICSC= I  +(K-1)  *L+(JM1-1)*L*N
      ICCT= I  +(KP1-1)*L+(J-1)  *L*N
      ICCB= I  +(KM1-1)*L+(J-1)  *L*N
C
      P1=PEW*(X(IECC)-X(IWCC))
      P2=PNS*(X(ICNC)-X(ICSC))
      P3=PTB*(X(ICCT)-X(ICCB))
      Q1=PEW*(Y(IECC)-Y(IWCC))
      Q2=PNS*(Y(ICNC)-Y(ICSC))
      Q3=PTB*(Y(ICCT)-Y(ICCB))
      R1=PEW*(Z(IECC)-Z(IWCC))
      R2=PNS*(Z(ICNC)-Z(ICSC))
      R3=PTB*(Z(ICCT)-Z(ICCB))
C
C      PTR=1.0/(P1*(Q2*R3-Q3*R2)-P2*(Q1*R3-Q3*R1)+P3*(Q1*R2-Q2*R1))
      AD1=P1*(Q2*R3-Q3*R2)
      AD2=-P2*(Q1*R3-Q3*R1)
      AD3=P3*(Q1*R2-Q2*R1)
      AD4=(AD1+AD2+AD3)
      IF(AD4.EQ.0.)
     1 WRITE(6,1111)INOD,P1,P2,P3,Q1,Q2,Q3,R1,R2,R3,AD1,AD2,AD3,PTR
      PTR=1.0/AD4
      TJO(INOD)=ABS(1.0/PTR)
  40  CONTINUE
 1111 FORMAT(1X,3I3,1X,9E10.5/,10X,4E10.5)
C
      DO 50 I=1,L
      DO 50 J=1,M
      DO 50 K=1,N
      INOD= I  +(K-1)  *L+(J-1)  *L*N
      IF(I.GT.1) THEN
      IWCC= I-1+(K-1)  *L+(J-1)  *L*N
      DSX(INOD)=SQRT((X(INOD)-X(IWCC))**2+(Y(INOD)-Y(IWCC))**2
     1           +(Z(INOD)-Z(IWCC))**2)
      ENDIF
      IF(J.GT.1) THEN
      ICSC= I  +(K-1)  *L+(J-1-1)*L*N
C
      DSY(INOD)=SQRT((X(INOD)-X(ICSC))**2+(Y(INOD)-Y(ICSC))**2
     1           +(Z(INOD)-Z(ICSC))**2)
      ENDIF
      IF(K.GT.1) THEN
      ICCB= I  +(K-1-1)*L+(J-1)  *L*N
C
      DSZ(INOD)=SQRT((X(INOD)-X(ICCB))**2+(Y(INOD)-Y(ICCB))**2
     1           +(Z(INOD)-Z(ICCB))**2)
      ENDIF
  50  CONTINUE
      RETURN
      END
C
C#######################################################
```

```
      SUBROUTINE CALCK(ITT)
C#################################################
C
      COMMON/FDNS1/ DK(46656),DE(46656),TT(1),VISE(46656)
      COMMON/XYZQ/Q(4,46656),X(46656),Y(46656),Z(46656)
      COMMON
     1/VAR/UCO(46656),VCO(46656),WCO(46656),F1(46656)
     1/TRAN/ TJO(46656),DSX(46656),DSY(46656),DSZ(46656)
      COMMON
     1/PROP/ DEN(46656),VISC,DENIN,FLOWIN,DENN1
     1/TUR/ SIGK,SIGE,CMU,C1,C2,CMU1,CMU2,E,CK,HINUM,SMNUM,ANV1(7000),
     2 YN(7000),YN1(7000),SINX(7000),SINY(7000),SINZ(7000),ANW1(7000),
     3 IBC(7000),JBC(7000),KBC(7000),IITY(7000),
     4 GEN(46656),MC(46656),IJLO(46656),IITO
      COMMON
     1/COEF/ AP(46656),SU(46656),SP(46656),SUK(46656),
     2   AE(46656),AW(46656),AN(46656),
     3   AS(46656),AT(46656),AB(46656),AP0(46656)
      COMMON
     1/LIMT/ L,M,LT,MT,L1,L2,M1,M2,ISWK,ALK,ALVIS,N,N1,N2,IG,NT,
     3   ISU,ITU,JSU,JTU,KSU,KTU,CBE,L3,L4,M3,M4,N3,N4,ERRK,ERRDE
C
C------TRANSPORT EQUATIONS LINERAIZATION AND SOLVER
C
      ERRK=0.0
      PI=3.141592654
C
C-----PRESSURE CORRECTION SOLVER STARTS FROM 10
C
      CALL IVA4(IS,IT,JS,JT, ISU, ITU, JSU, JTU)
      CALL IVA4(KS,KT,II,JX, KSU, KTU,   0,   0)
      CALL RVA4(CE,CW,CN,CS, 0.0, 0.0, 0.0, 0.0)
      CALL RVA4(CT,CB,P1,P2, 0.0, 0.0, 0.0, 0.0)
C
C-----U, V, W, TM, K & E EQUATIONS
C
      DO 22 I=IS-1,IT+1
      DO 22 J=JS-1,JT+1
      DO 22 K=KS-1,KT+1
      INOD = I +(K-1)*L +(J-1)*L*N
      F1(INOD)=VISE(INOD)/SIGK
  22  CONTINUE
C
C-----CALCULATE THE SOURCE TERMS OF TRANSPORT EQNS.
C
      PI=3.141592654
      PCC=0.0
C
C-----EVALUATE LINK COEFF. AND SOURCE TERMS
C
      DO 20 I=IS,IT
      DO 20 J=JS,JT
      DO 20 K=KS,KT
C
C---  27 NODE IDENDIFICATIONS
C
C  CENTER PLANE
C
      IWCC= I-1 +(K-1)*L +(J-1)*L*N
      IWNC= I-1 +(K-1)*L +(J)  *L*N
```

```
      IWSC= I-1 +(K-1)*L +(J-2)*L*N
      INOD= I   +(K-1)*L +(J-1)*L*N
      ICNC= I   +(K-1)*L +(J)  *L*N
      ICSC= I   +(K-1)*L +(J-2)*L*N
      IECC= I+1 +(K-1)*L +(J-1)*L*N
      IENC= I+1 +(K-1)*L +(J)  *L*N
      IESC= I+1 +(K-1)*L +(J-2)*L*N
C
C  TOP PLANE
C
      IWCT= I-1 +(K)  *L +(J-1)*L*N
C      IWNT= I-1 +(K)  *L +(J)  *L*N
C      IWST= I-1 +(K)  *L +(J-2)*L*N
      ICCT= I   +(K)  *L +(J-1)*L*N
      ICNT= I   +(K)  *L +(J)  *L*N
      ICST= I   +(K)  *L +(J-2)*L*N
      IECT= I+1 +(K)  *L +(J-1)*L*N
C      IENT= I+1 +(K)  *L +(J)  *L*N
C      IEST= I+1 +(K)  *L +(J-2)*L*N
C
C  BOTTON PLANE
C
      IWCB= I-1 +(K-2)*L +(J-1)*L*N
C      IWNB= I-1 +(K-2)*L +(J)  *L*N
C      IWSB= I-1 +(K-2)*L +(J-2)*L*N
      ICCB= I   +(K-2)*L +(J-1)*L*N
      ICNB= I   +(K-2)*L +(J)  *L*N
      ICSB= I   +(K-2)*L +(J-2)*L*N
      IECB= I+1 +(K-2)*L +(J-1)*L*N
C      IENB= I+1 +(K-2)*L +(J)  *L*N
C      IESB= I+1 +(K-2)*L +(J-2)*L*N
C
      DENC=1.0
      IF(MC(INOD).EQ.1) DENC=DENN1
      P1=0.5*(X(IECC)-X(IWCC))
      P2=0.5*(X(ICNC)-X(ICSC))
      P3=0.5*(X(ICCT)-X(ICCB))
      Q1=0.5*(Y(IECC)-Y(IWCC))
      Q2=0.5*(Y(ICNC)-Y(ICSC))
      Q3=0.5*(Y(ICCT)-Y(ICCB))
      R1=0.5*(Z(IECC)-Z(IWCC))
      R2=0.5*(Z(ICNC)-Z(ICSC))
      R3=0.5*(Z(ICCT)-Z(ICCB))
      PTR=1.0/(P1*(Q2*R3-Q3*R2)-P2*(Q1*R3-Q3*R1)+P3*(Q1*R2-Q2*R1))
      CXQ=PTR*(Q2*R3-Q3*R2)
      CYQ=-PTR*(P2*R3-P3*R2)
      CZQ=PTR*(P2*Q3-P3*Q2)
      EXQ=-PTR*(Q1*R3-Q3*R1)
      EYQ=PTR*(P1*R3-P3*R1)
      EZQ=-PTR*(P1*Q3-P3*Q1)
      SXQ=PTR*(Q1*R2-Q2*R1)
      SYQ=-PTR*(P1*R2-P2*R1)
      SZQ=PTR*(P1*Q2-P2*Q1)
      GAE=0.5*(F1(IECC)+F1(INOD))
      GAW=0.5*(F1(IWCC)+F1(INOD))
      GAN=0.5*(F1(ICNC)+F1(INOD))
      GAS=0.5*(F1(ICSC)+F1(INOD))
      GAT=0.5*(F1(ICCT)+F1(INOD))
      GAB=0.5*(F1(ICCB)+F1(INOD))
      CE=0.5*(UCO(INOD)+UCO(IECC))*DENC
```

```
      CW=0.5*(UCO(INOD)+UCO(IWCC))*DENC
      CN=0.5*(VCO(INOD)+VCO(ICNC))*DENC
      CS=0.5*(VCO(INOD)+VCO(ICSC))*DENC
      CT=0.5*(WCO(INOD)+WCO(ICCT))*DENC
      CB=0.5*(WCO(INOD)+WCO(ICCB))*DENC
C
      FACTEW=0.5*(DSX(INOD)+DSX(IECC))
      FACTNS=0.5*(DSY(INOD)+DSY(ICNC))
      FACTTB=0.5*(DSZ(INOD)+DSZ(ICCT))
      TXEW=(CXQ**2+CYQ**2+CZQ**2)*TJO(INOD)
      TYNS=(EXQ**2+EYQ**2+EZQ**2)*TJO(INOD)
      TZTB=(SXQ**2+SYQ**2+SZQ**2)*TJO(INOD)
      TXYN=(CXQ*EXQ+CYQ*EYQ+
     1      CZQ*EZQ)*0.25*TJO(INOD)
      TYZN=(EXQ*SXQ+EYQ*SYQ+
     1      EZQ*SZQ)*0.25*TJO(INOD)
      TZXE=(SXQ*CXQ+SYQ*CYQ+
     1      SZQ*CZQ)*0.25*TJO(INOD)
      DDE=GAE*TXEW*FACTEW/DSX(IECC)+(GAN-GAS)*TXYN+(GAT-GAB)*TZXE
      DDW=GAW*TXEW*FACTEW/DSX(INOD)  +(GAS-GAN)*TXYN+(GAB-GAT)*TZXE
      DDN=GAN*TYNS*FACTNS/DSY(ICNC)+(GAE-GAW)*TXYN+(GAT-GAB)*TYZN
      DDS=GAS*TYNS*FACTNS/DSY(INOD)  +(GAW-GAE)*TXYN+(GAB-GAT)*TYZN
      DDT=GAT*TZTB*FACTTB/DSZ(ICCT)+(GAE-GAW)*TZXE+(GAN-GAS)*TYZN
      DDB=GAB*TZTB*FACTTB/DSZ(INOD)  +(GAW-GAE)*TZXE+(GAS-GAN)*TYZN
C
C-----HYBRID SCHEME--------
C
      AE(INOD)=(AMAX1(ABS(0.5*CE),DDE)-0.5*CE)
      AW(INOD)=(AMAX1(ABS(0.5*CW),DDW)+0.5*CW)
      AN(INOD)=(AMAX1(ABS(0.5*CN),DDN)-0.5*CN)
      AS(INOD)=(AMAX1(ABS(0.5*CS),DDS)+0.5*CS)
      AT(INOD)=(AMAX1(ABS(0.5*CT),DDT)-0.5*CT)
      AB(INOD)=(AMAX1(ABS(0.5*CB),DDB)+0.5*CB)
C-------------------------
      CPG=(CE-CW+CN-CS+CT-CB)
      CP0=ABS(CPG)
      ANEQ= (GAE+GAN)*TXYN
      ASEQ=-(GAE+GAS)*TXYN
      ANWQ=-(GAW+GAN)*TXYN
      ASWQ= (GAW+GAS)*TXYN
      AETQ= (GAE+GAT)*TZXE
      AEBQ=-(GAE+GAB)*TZXE
      AWTQ=-(GAW+GAT)*TZXE
      AWBQ= (GAW+GAB)*TZXE
      ANTQ= (GAN+GAT)*TYZN
      ANBQ=-(GAN+GAB)*TYZN
      ASTQ=-(GAS+GAT)*TYZN
      ASBQ= (GAS+GAB)*TYZN
      SU(INOD)=ANEQ*DK(IENC)+
     1  ANWQ*DK(IWNC) +ASEQ*DK(IESC)+
     2  ASWQ*DK(IWSC) +AETQ*DK(IECT)+
     3  AEBQ*DK(IECB) +AWTQ*DK(IWCT)+
     4  AWBQ*DK(IWCB) +ANTQ*DK(ICNT)+
     5  ANBQ*DK(ICNB) +ASTQ*DK(ICST)+
     6  ASBQ*DK(ICSB)
      SUK(INOD)=CP0
C
      UCXI=0.5*(Q(2,IECC)-Q(2,IWCC))
      UEDA=0.5*(Q(2,ICNC)-Q(2,ICSC))
      USCI=0.5*(Q(2,ICCT)-Q(2,ICCB))
```

```
      VCXI=0.5*(Q(3,IECC)-Q(3,IWCC))
      VEDA=0.5*(Q(3,ICNC)-Q(3,ICSC))
      VSCI=0.5*(Q(3,ICCT)-Q(3,ICCB))
      WCXI=0.5*(Q(4,IECC)-Q(4,IWCC))
      WEDA=0.5*(Q(4,ICNC)-Q(4,ICSC))
      WSCI=0.5*(Q(4,ICCT)-Q(4,ICCB))
      UX=UCXI*CXQ+UEDA*EXQ+USCI*SXQ
      UY=UCXI*CYQ+UEDA*EYQ+USCI*SYQ
      UZ=UCXI*CZQ+UEDA*EZQ+USCI*SZQ
      VX=VCXI*CXQ+VEDA*EXQ+VSCI*SXQ
      VY=VCXI*CYQ+VEDA*EYQ+VSCI*SYQ
      VZ=VCXI*CZQ+VEDA*EZQ+VSCI*SZQ
      WX=WCXI*CXQ+WEDA*EXQ+WSCI*SXQ
      WY=WCXI*CYQ+WEDA*EYQ+WSCI*SYQ
      WZ=WCXI*CZQ+WEDA*EZQ+WSCI*SZQ
      GEN(INOD)=VISE(INOD)*((UY+VX)**2+(VZ+WY)**2+(WX+UZ)**2+
     1          2*(UX*UX+VY*VY+WZ*WZ)-PCC*2*(UX+VY+WZ)**2/3.)
      IF(MC(INOD).GT.0) GEN(INOD)=0.0
   20 CONTINUE
C-------------------------------------------------------------------------
C-----CALCULATE SOURCE TERMS FOR CALCK
C      GEN IS CREATED ABOVE DO-LOOP
C-----K-SOURCE
C
      DO 55 I=IS,IT
      DO 55 J=JS,JT
      DO 55 K=KS,KT
      INOD= I+(K-1)*L+(J-1)*L*N
      SUTM=GEN(INOD)+0.0
      P1=DEN(INOD)**2
      SP(INOD)=-SUK(INOD)-TJO(INOD)*CMU*P1*DK(INOD)/VISE(INOD)
      SU(INOD)=SUTM*TJO(INOD)+SUK(INOD)*DK(INOD)+SU(INOD)
   55 CONTINUE
C
C-----MODIFY WALL BOUNDARY CONDITIONS THRU WALL FUNCTIONS
C
      IF(IG .NE. 2) GO TO 410
C
C-----EVALUATE WALL BOUNDARY CONDITIONS USING WALL FUNCTIONS
C
      DO 150 III=1,IITO
      I=IBC(III)
      J=JBC(III)
      K=KBC(III)
C
C-----K
C
      INOD= I+(K-1)*L+(J-1)*L*N
      YPLN1=DEN(INOD)*SQRT(DK(INOD))*CMU1*YN(III)/VISC
      IF(YPLN1.LE.11.65) GO TO 111
      TMULT=DEN(INOD)*SQRT(DK(INOD))*CMU1*CK/ALOG(E*YPLN1)
      GO TO 112
  111 TMULT=VISC/YN(III)
  112 TAUN1=-TMULT
      GO TO (1,2,3,4,5,6), IITY(III)
    1 CONTINUE
C
C-----NORTH
C
      ICNC= I    +(K-1)*L +(J)   *L*N
```

```
      DDU=Q(2,INOD)-Q(2,ICNC)
      DDV=Q(3,INOD)-Q(3,ICNC)
      DDW=Q(4,INOD)-Q(4,ICNC)
      P1=DDU**2+DDV**2+DDW**2
      GEN(INOD)=P1*TAUN1**2/VISE(INOD)
      SU(INOD)=HINUM*SQRT(P1)*ABS(TAUN1)/DEN(INOD)/SQRT(0.09)
      SP(INOD)=-HINUM
      AN(INOD)=0.0
      GO TO 150
    2 CONTINUE
C
C-----SOUTH
C
      ICSC= I   +(K-1)*L +(J-2)*L*N
      DDU=Q(2,INOD)-Q(2,ICSC)
      DDV=Q(3,INOD)-Q(3,ICSC)
      DDW=Q(4,INOD)-Q(4,ICSC)
      P1=DDU**2+DDV**2+DDW**2
      GEN(INOD)=P1*TAUN1**2/VISE(INOD)
      SU(INOD)=HINUM*SQRT(P1)*ABS(TAUN1)/DEN(INOD)/SQRT(0.09)
      SP(INOD)=-HINUM
      AS(INOD)=0.0
      GO TO 150
    3 CONTINUE
C
C-----EAST
C
      IECC= I+1 +(K-1)*L +(J-1)*L*N
      DDU=Q(2,INOD)-Q(2,IECC)
      DDV=Q(3,INOD)-Q(3,IECC)
      DDW=Q(4,INOD)-Q(4,IECC)
      P1=DDU**2+DDV**2+DDW**2
      GEN(INOD)=P1*TAUN1**2/VISE(INOD)
      SU(INOD)=HINUM*SQRT(P1)*ABS(TAUN1)/DEN(INOD)/SQRT(0.09)
      SP(INOD)=-HINUM
      AE(INOD)=0.0
      GO TO 150
    4 CONTINUE
C
C-----WEST
C
      IWCC= I-1 +(K-1)*L +(J-1)*L*N
      DDU=Q(2,INOD)-Q(2,IWCC)
      DDV=Q(3,INOD)-Q(3,IWCC)
      DDW=Q(4,INOD)-Q(4,IWCC)
      P1=DDU**2+DDV**2+DDW**2
      GEN(INOD)=P1*TAUN1**2/VISE(INOD)
      SU(INOD)=HINUM*SQRT(P1)*ABS(TAUN1)/DEN(INOD)/SQRT(0.09)
      SP(INOD)=-HINUM
      AW(INOD)=0.0
      GO TO 150
    5 CONTINUE
C
C-----TOP
C
      ICCT= I   +(K)  *L +(J-1)*L*N
      DDU=Q(2,INOD)-Q(2,ICCT)
      DDV=Q(3,INOD)-Q(3,ICCT)
      DDW=Q(4,INOD)-Q(4,ICCT)
      P1=DDU**2+DDV**2+DDW**2
```

```
      GEN(INOD)=P1*TAUN1**2/VISE(INOD)
      SU(INOD)=HINUM*SQRT(P1)*ABS(TAUN1)/DEN(INOD)/SQRT(0.09)
      SP(INOD)=-HINUM
      AT(INOD)=0.0
      GO TO 150
    6 CONTINUE
C
C-----BOTTOM
C
      ICCB= I   +(K-2)*L +(J-1)*L*N
      DDU=Q(2,INOD)-Q(2,ICCB)
      DDV=Q(3,INOD)-Q(3,ICCB)
      DDW=Q(4,INOD)-Q(4,ICCB)
      P1=DDU**2+DDV**2+DDW**2
      GEN(INOD)=P1*TAUN1**2/VISE(INOD)
      SU(INOD)=HINUM*SQRT(P1)*ABS(TAUN1)/DEN(INOD)/SQRT(0.09)
      SP(INOD)=-HINUM
      AB(INOD)=0.0
      GO TO 150
  150 CONTINUE
C
  410 CONTINUE
C
C-----EAST OUT
C
      I=IT
      DO 200 J=2,MT
      DO 200 K=2,NT
      INOD= I +(K-1)*L +(J-1)*L*N
      AE(INOD)=0.0
  200 CONTINUE
C
      DO 511 I=IS-1,IT+1
      DO 511 J=JS-1,JT+1
      DO 511 K=KS-1,KT+1
      INOD= I +(K-1)*L +(J-1)*L*N
  511 F1(INOD)=0.0
C
C-----LINK COEFF. ASSEMBLY AND BLOCKAGES
C
      DO 500 I=IS,IT
      DO 500 J=JS,JT
      DO 500 K=KS,KT
      INOD= I +(K-1)*L +(J-1)*L*N
      ANAB=AE(INOD)+AW(INOD)+AN(INOD)+AS(INOD)+AT(INOD)+
     1     AB(INOD)+AP0(INOD)
      AP(INOD)=ANAB-SP(INOD)
      PDUV=1.0
      SCAL=1.0
      IF(IJLO(INOD).NE.0) SCAL=SMNUM
      IF(MC(INOD).GE.1) THEN
        AP(INOD)=ALK
        AN(INOD)=0.0
        AS(INOD)=0.0
        AE(INOD)=0.0
        AW(INOD)=0.0
        AT(INOD)=0.0
        AB(INOD)=0.0
        SU(INOD)=DK(INOD)
        PDUV=0.0
```

```
      ENDIF
C
C-----UNDER-RELAXATION
C
      P1=AMAX1(SMNUM,(AP(INOD)/0.8-ANAB))
      AP(INOD)=AP(INOD)/ALK
C
      IECC= I+1+(K-1)  *L+(J-1)  *L*N
      IWCC= I-1+(K-1)  *L+(J-1)  *L*N
      ICNC= I  +(K-1)  *L+(J)    *L*N
      ICSC= I  +(K-1)  *L+(J-2)  *L*N
      ICCT= I  +(K)    *L+(J-1)  *L*N
      ICCB= I  +(K-2)  *L+(J-1)  *L*N
C
      RD=-AP(INOD)*DK(INOD)+AE(INOD)*DK(IECC)+
     1 AW(INOD)*DK(IWCC)+ AN(INOD)*DK(ICNC)+
     2 AS(INOD)*DK(ICSC)+ AT(INOD)*DK(ICCT)+ AB(INOD)*
     3 DK(ICCB)+SU(INOD)+ PDUV*(1.0-ALF)*AP(INOD)*DK(INOD)
      SU(INOD)=RD
      ERRK=ERRK+ABS(RD*SCAL)
 500  CONTINUE
C
C-----LINEAR EQUATIONS SLOVER
C
      CALL LINERX(1,ISWK,IS,JS,KS,IT,JT,KT,L,M,N,F1)
C
C-----CALCULATE MAXIMUM CORRECTION OF CURRENT ITERATION
C
      DO 555 I=IS,IT
      DO 555 J=JS,JT
      DO 555 K=KS,KT
      INOD= I +(K-1)*L +(J-1)*L*N
      DK(INOD)=DK(INOD)+F1(INOD)
 555  CONTINUE
      RETURN
      END
C
C#########################################################
      SUBROUTINE CALCE(ITT)
C#########################################################
C
      COMMON/FDNS1/ DK(46656),DE(46656),TT(1),VISE(46656)
      COMMON/XYZQ/Q(4,46656),X(46656),Y(46656),Z(46656)
      COMMON
     1/VAR/UCO(46656),VCO(46656),WCO(46656),F1(46656)
     1/TRAN/ TJO(46656),DSX(46656),DSY(46656),DSZ(46656)
      COMMON
     1/PROP/ DEN(46656),VISC,DENIN,FLOWIN,DENN1
     1/TUR/ SIGK,SIGE,CMU,C1,C2,CMU1,CMU2,E,CK,HINUM,SMNUM,ANV1(7000),
     2 YN(7000),YN1(7000),SINX(7000),SINY(7000),SINZ(7000),ANW1(7000),
     3 IBC(7000),JBC(7000),KBC(7000),IITY(7000),
     4 GEN(46656),MC(46656),IJLO(46656),IITO
      COMMON
     1/COEF/ AP(46656),SU(46656),SP(46656),SUK(46656),
     2   AE(46656),AW(46656),AN(46656),
     3   AS(46656),AT(46656),AB(46656),AP0(46656)
      COMMON
     1/LIMT/ L,M,LT,MT,L1,L2,M1,M2,ISWK,ALK,ALVIS,N,N1,N2,IG,NT,
     3   ISU,ITU,JSU,JTU,KSU,KTU,CBE,L3,L4,M3,M4,N3,N4,ERRK,ERRDE
C
```

```
C------TRANSPORT EQUATIONS LINERAIZATION AND SOLVER
C
      ERRDE=0.0
      PI=3.141592654
      CALL IVA4(IS,IT,JS,JT, ISU, ITU, JSU, JTU)
      CALL IVA4(KS,KT,II,JX, KSU, KTU,   0,   0)
      CALL RVA4(CE,CW,CN,CS, 0.0, 0.0, 0.0, 0.0)
      CALL RVA4(CT,CB,P1,P2, 0.0, 0.0, 0.0, 0.0)
C
C----- E EQUATIONS
C
      DO 22 I=IS-1,IT+1
      DO 22 J=JS-1,JT+1
      DO 22 K=KS-1,KT+1
      INOD = I +(K-1)*L +(J-1)*L*N
      F1(INOD)=VISE(INOD)/SIGE
  22  CONTINUE
C
C-----EVALUATE LINK COEFF. AND SOURCE TERMS
C
      DO 20 I=IS,IT
      DO 20 J=JS,JT
      DO 20 K=KS,KT
C
C---  27 NODE IDENDIFICATIONS
C
C  CENTER PLANE
C
      IWCC= I-1 +(K-1)*L +(J-1)*L*N
      IWNC= I-1 +(K-1)*L +(J)  *L*N
      IWSC= I-1 +(K-1)*L +(J-2)*L*N
      INOD= I   +(K-1)*L +(J-1)*L*N
      ICNC= I   +(K-1)*L +(J)  *L*N
      ICSC= I   +(K-1)*L +(J-2)*L*N
      IECC= I+1 +(K-1)*L +(J-1)*L*N
      IENC= I+1 +(K-1)*L +(J)  *L*N
      IESC= I+1 +(K-1)*L +(J-2)*L*N
C
C  TOP PLANE
C
      IWCT= I-1 +(K)  *L +(J-1)*L*N
C      IWNT= I-1 +(K)  *L +(J)  *L*N
C      IWST= I-1 +(K)  *L +(J-2)*L*N
      ICCT= I   +(K)  *L +(J-1)*L*N
      ICNT= I   +(K)  *L +(J)  *L*N
      ICST= I   +(K)  *L +(J-2)*L*N
      IECT= I+1 +(K)  *L +(J-1)*L*N
C      IENT= I+1 +(K)  *L +(J)  *L*N
C      IEST= I+1 +(K)  *L +(J-2)*L*N
C
C  BOTTON PLANE
C
      IWCB= I-1 +(K-2)*L +(J-1)*L*N
C      IWNB= I-1 +(K-2)*L +(J)  *L*N
C      IWSB= I-1 +(K-2)*L +(J-2)*L*N
      ICCB= I   +(K-2)*L +(J-1)*L*N
      ICNB= I   +(K-2)*L +(J)  *L*N
      ICSB= I   +(K-2)*L +(J-2)*L*N
      IECB= I+1 +(K-2)*L +(J-1)*L*N
C      IENB= I+1 +(K-2)*L +(J)  *L*N
```

```
C       IESB= I+1 +(K-2)*L +(J-2)*L*N
C
        DENC=1.0
        IF(MC(INOD).EQ.1) DENC=DENN1
        P1=0.5*(X(IECC)-X(IWCC))
        P2=0.5*(X(ICNC)-X(ICSC))
        P3=0.5*(X(ICCT)-X(ICCB))
        Q1=0.5*(Y(IECC)-Y(IWCC))
        Q2=0.5*(Y(ICNC)-Y(ICSC))
        Q3=0.5*(Y(ICCT)-Y(ICCB))
        R1=0.5*(Z(IECC)-Z(IWCC))
        R2=0.5*(Z(ICNC)-Z(ICSC))
        R3=0.5*(Z(ICCT)-Z(ICCB))
        PTR=1.0/(P1*(Q2*R3-Q3*R2)-P2*(Q1*R3-Q3*R1)+P3*(Q1*R2-Q2*R1))
        CXQ=PTR*(Q2*R3-Q3*R2)
        CYQ=-PTR*(P2*R3-P3*R2)
        CZQ=PTR*(P2*Q3-P3*Q2)
        EXQ=-PTR*(Q1*R3-Q3*R1)
        EYQ=PTR*(P1*R3-P3*R1)
        EZQ=-PTR*(P1*Q3-P3*Q1)
        SXQ=PTR*(Q1*R2-Q2*R1)
        SYQ=-PTR*(P1*R2-P2*R1)
        SZQ=PTR*(P1*Q2-P2*Q1)
        GAE=0.5*(F1(IECC)+F1(INOD))
        GAW=0.5*(F1(IWCC)+F1(INOD))
        GAN=0.5*(F1(ICNC)+F1(INOD))
        GAS=0.5*(F1(ICSC)+F1(INOD))
        GAT=0.5*(F1(ICCT)+F1(INOD))
        GAB=0.5*(F1(ICCB)+F1(INOD))
        CE=0.5*(UCO(INOD)+UCO(IECC))*DENC
        CW=0.5*(UCO(INOD)+UCO(IWCC))*DENC
        CN=0.5*(VCO(INOD)+VCO(ICNC))*DENC
        CS=0.5*(VCO(INOD)+VCO(ICSC))*DENC
        CT=0.5*(WCO(INOD)+WCO(ICCT))*DENC
        CB=0.5*(WCO(INOD)+WCO(ICCB))*DENC
C
        FACTEW=0.5*(DSX(INOD)+DSX(IECC))
        FACTNS=0.5*(DSY(INOD)+DSY(ICNC))
        FACTTB=0.5*(DSZ(INOD)+DSZ(ICCT))
        TXEW=(CXQ**2+CYQ**2+CZQ**2)*TJO(INOD)
        TYNS=(EXQ**2+EYQ**2+EZQ**2)*TJO(INOD)
        TZTB=(SXQ**2+SYQ**2+SZQ**2)*TJO(INOD)
        TXYN=(CXQ*EXQ+CYQ*EYQ+
       1      CZQ*EZQ)*0.25*TJO(INOD)
        TYZN=(EXQ*SXQ+EYQ*SYQ+
       1      EZQ*SZQ)*0.25*TJO(INOD)
        TZXE=(SXQ*CXQ+SYQ*CYQ+
       1      SZQ*CZQ)*0.25*TJO(INOD)
        DDE=GAE*TXEW*FACTEW/DSX(IECC)+(GAN-GAS)*TXYN+(GAT-GAB)*TZXE
        DDW=GAW*TXEW*FACTEW/DSX(INOD)  +(GAS-GAN)*TXYN+(GAB-GAT)*TZXE
        DDN=GAN*TYNS*FACTNS/DSY(ICNC)+(GAE-GAW)*TXYN+(GAT-GAB)*TYZN
        DDS=GAS*TYNS*FACTNS/DSY(INOD)  +(GAW-GAE)*TXYN+(GAB-GAT)*TYZN
        DDT=GAT*TZTB*FACTTB/DSZ(ICCT)  +(GAE-GAW)*TZXE+(GAN-GAS)*TYZN
        DDB=GAB*TZTB*FACTTB/DSZ(INOD)  +(GAW-GAE)*TZXE+(GAS-GAN)*TYZN
C
C-------------------HYBRID SCHEME-----------------------------------
C
        AE(INOD)=(AMAX1(ABS(0.5*CE),DDE)-0.5*CE)
        AW(INOD)=(AMAX1(ABS(0.5*CW),DDW)+0.5*CW)
        AN(INOD)=(AMAX1(ABS(0.5*CN),DDN)-0.5*CN)
```

```
      AS(INOD)=(AMAX1(ABS(0.5*CS),DDS)+0.5*CS)
      AT(INOD)=(AMAX1(ABS(0.5*CT),DDT)-0.5*CT)
      AB(INOD)=(AMAX1(ABS(0.5*CB),DDB)+0.5*CB)
C
C-----------------------------------------------------------------------
C
      CPG=(CE-CW+CN-CS+CT-CB)
      CP0=ABS(CPG)
      ANEQ= (GAE+GAN)*TXYN
      ASEQ=-(GAE+GAS)*TXYN
      ANWQ=-(GAW+GAN)*TXYN
      ASWQ= (GAW+GAS)*TXYN
      AETQ= (GAE+GAT)*TZXE
      AEBQ=-(GAE+GAB)*TZXE
      AWTQ=-(GAW+GAT)*TZXE
      AWBQ= (GAW+GAB)*TZXE
      ANTQ= (GAN+GAT)*TYZN
      ANBQ=-(GAN+GAB)*TYZN
      ASTQ=-(GAS+GAT)*TYZN
      ASBQ= (GAS+GAB)*TYZN
      SU(INOD)=ANEQ*DE(IENC)+
     1  ANWQ*DE(IWNC)+ASEQ*DE(IESC)+
     2  ASWQ*DE(IWSC)+AETQ*DE(IECT)+
     3  AEBQ*DE(IECB)+AWTQ*DE(IWCT)+
     4  AWBQ*DE(IWCB)+ANTQ*DE(ICNT)+
     5  ANBQ*DE(ICNB)+ASTQ*DE(ICST)+
     6  ASBQ*DE(ICSB)
      SUK(INOD)=CP0
  20  CONTINUE
C
C--------------------CALCULATE SOURCE TERMS FOR CALCE--------------------
C
      PI=3.141592654
      PCC=0.0
C
C-----E-SOURCE
C
      DO 65 I=IS,IT
      DO 65 J=JS,JT
      DO 65 K=KS,KT
      INOD= I + (K-1)*L + (J-1)*L*N
      TMDE=DE(INOD)
      IF(TMDE.LE.1.E-30) TMDE=1.E-30
      SUTM=C1*CMU*GEN(INOD)*P1*DK(INOD)/
     1           VISE(INOD)+0.0
      TMDK=DK(INOD)+SMNUM
      SP(INOD)=-SUK(INOD)-TJO(INOD)*C2*DEN(INOD)*DE(INOD)/TMDK
      SU(INOD)=SUTM*TJO(INOD)+SUK(INOD)*DE(INOD)+SU(INOD)
  65  CONTINUE
C
C-----MODIFY WALL BOUNDARY CONDITIONS THRU WALL FUNCTIONS
C
      IF(IG .NE. 2) GO TO 410
C
C-----EVALUATE WALL BOUNDARY CONDITIONS USING WALL FUNCTIONS
C
      DO 150 III=1,IITO
      I=IBC(III)
      J=JBC(III)
      K=KBC(III)
```

```
      INOD= I + (K-1)*L + (J-1)*L*N
C
C-----NORTH/SOUTH ETC ARE SAME FOR E
C-----E
C
      TERM=CMU2/(CK*YN(III))
      SU(INOD)=HINUM*TERM*DK(INOD)**1.5
      SP(INOD)=-HINUM
 150  CONTINUE
C
 410  CONTINUE
C
C-----EAST OUT
C
      I=IT
      DO 200 J=2,MT
      DO 200 K=2,NT
      INOD= I + (K-1)*L + (J-1)*L*N
      AE(INOD)=0.0
 200  CONTINUE
      DO 501 I=IS-1,IT+1
      DO 501 J=JS-1,JT+1
      DO 501 K=KS-1,KT+1
      INOD= I + (K-1)*L + (J-1)*L*N
 501  F1(INOD)=0.0
C
C-----LINK COEFF. ASSEMBLY AND BLOCKAGES
C
      DO 500 I=IS,IT
      DO 500 J=JS,JT
      DO 500 K=KS,KT
C
      INOD= I  +(K-1)  *L +(J-1) *L*N
      IECC= I+1+(K-1)  *L+(J-1)  *L*N
      IWCC= I-1+(K-1)  *L+(J-1)  *L*N
      ICNC= I  +(K-1)  *L+(J)    *L*N
      ICSC= I  +(K-1)  *L+(J-2)  *L*N
      ICCT= I  +(K)    *L+(J-1)  *L*N
      ICCB= I  +(K-2)  *L+(J-1)  *L*N
C
      ANAB=AE(INOD)+AW(INOD)+AN(INOD)+AS(INOD)+AT(INOD)+
     .  AB(INOD)+AP0(INOD)
      AP(INOD)=ANAB-SP(INOD)
      PDUV=1.0
      SCAL=1.0
      IF(IJLO(INOD).NE.0) SCAL=SMNUM
      IF(MC(INOD).GE.1) THEN
        AP(INOD)=ALK
        AN(INOD)=0.0
        AS(INOD)=0.0
        AE(INOD)=0.0
        AW(INOD)=0.0
        AT(INOD)=0.0
        AB(INOD)=0.0
        SU(INOD)=DE(INOD)
        PDUV=0.0
      ENDIF
C
C-----UNDER-RELAXATION
C
```

```
      P1=AMAX1(SMNUM,(AP(INOD)/0.8-ANAB))
      AP(INOD)=AP(INOD)/ALK
      RD=-AP(INOD)*DE(INOD)+AE(INOD)*DE(IECC)+AW(INOD)
     1 *DE(IWCC)+AN(INOD)*DE(ICNC)+AS(INOD)*DE(ICSC)
     2          +AT(INOD)*DE(ICCT)+AB(INOD)*DE(ICCB)
     3 +PDUV*(1.0-ALDE)*AP(INOD)*DE(INOD)+SU(INOD)
      SU(INOD)=RD
      ERRDE=ERRDE+ABS(RD*SCAL)
C
 500  CONTINUE
C
C-----LINEAR EQUATIONS SLOVER
C
      CALL LINERX(1,ISWK,IS,JS,KS,IT,JT,KT,L,M,N,F1)
C
C-----CALCULATE MAXIMUM CORRECTION OF CURRENT ITERATION
C
      DO 555 I=IS,IT
      DO 555 J=JS,JT
      DO 555 K=KS,KT
      INOD= I  +(K-1)  *L +(J-1) *L*N
      DE(INOD)=DE(INOD)+F1(INOD)
 555  CONTINUE
      RETURN
      END
C
C#############################################################
      SUBROUTINE IVA4(IA,IB,IC,ID, JA,JB,JC,JD)
C#############################################################
C
      IA=JA
      IB=JB
      IC=JC
      ID=JD
      RETURN
      END
C
C#############################################################
      SUBROUTINE RVA4(PA,PB,PC,PD, QA,QB,QC,QD)
C#############################################################
C
      PA=QA
      PB=QB
      PC=QC
      PD=QD
      RETURN
      END
C
C#################################################################
      SUBROUTINE LINERX(ISOL,ISWF,IS,JS,KS,IT,JT,KT,L,M,N,F)
C#################################################################
C
      DIMENSION A(90),B(90),C(90),D(90),F(46656)
      COMMON
     1/COEF/ AP(46656),SU(46656),SP(46656),SUK(46656),
     2   AE(46656),AW(46656),AN(46656),
     3   AS(46656),AT(46656),AB(46656),AP0(46656)
C
C-----LINE-RELAXATION USING TDMA
C
```

```
      DO 90 ISW=1,ISWF
      ERRF=0.0
C-----------------------------------------------
      A(JS-1)=0.0
      DO 120 I=IS,IT
      DO 120 K=KS,KT
      INODJS= I  +(K-1)  *L +(JS-2) *L*N
      C(JS-1)=F(INODJS)
      DO 121 J=JS,JT
C
      INOD= I  +(K-1)  *L +(J-1) *L*N
      IECC= I+1+(K-1)  *L+(J-1)  *L*N
      IWCC= I-1+(K-1)  *L+(J-1)  *L*N
      ICCT= I  +(K)    *L+(J-1)  *L*N
      ICCB= I  +(K-2)  *L+(J-1)  *L*N
C
      A(J)=AN(INOD)
      B(J)=AS(INOD)
      C(J)=SU(INOD)+AE(INOD)*F(IECC)+AW(INOD)*F(IWCC)+
     1  AT(INOD)*F(ICCT)+AB(INOD)*F(ICCB)
      D(J)=AP(INOD)
      TERM=1.0/(D(J)-B(J)*A(J-1))
      A(J)=A(J)*TERM
 121  C(J)=(C(J)+B(J)*C(J-1))*TERM
      DO 122 JX=JS,JT
      J=JS+JT-JX
      INOD= I  +(K-1)  *L +(J-1) *L*N
      ICNC= I  +(K-1)  *L+ (J)    *L*N
 122  F(INOD)=A(J)*F(ICNC)+C(J)
 120  CONTINUE
C-----------------------------------------------
      A(KS-1)=0.0
      DO 110 I=IS,IT
      DO 110 J=JS,JT
      INODKS= I  +(KS-2)  *L +(J-1) *L*N
      C(KS-1)=F(INODKS)
      DO 111 K=KS,KT
C
      INOD= I  +(K-1)  *L +(J-1) *L*N
      IECC= I+1+(K-1)  *L+(J-1)  *L*N
      IWCC= I-1+(K-1)  *L+(J-1)  *L*N
      ICNC= I  +(K-1)  *L+(J)    *L*N
      ICSC= I  +(K-1)  *L+(J-2)  *L*N
C
      A(K)=AT(INOD)
      B(K)=AB(INOD)
      C(K)=SU(INOD)+AE(INOD)*F(IECC)+AW(INOD)*F(IWCC)+
     1  AN(INOD)*F(ICNC)+AS(INOD)*F(ICSC)
      D(K)=AP(INOD)
      TERM=1.0/(D(K)-B(K)*A(K-1))
      A(K)=A(K)*TERM
 111  C(K)=(C(K)+B(K)*C(K-1))*TERM
      DO 112 KK=KS,KT
      K=KS+KT-KK
      INOD= I  +(K-1)  *L +(J-1) *L*N
      ICCT= I  +(K)    *L+(J-1)  *L*N
 112  F(INOD)=A(K)*F(ICCT)+C(K)
 110  CONTINUE
C-----------------------------------------------
      A(IS-1)=0.0
```

```
      DO 130 J=JS,JT
      DO 130 K=KS,KT
      INODIS= IS-1 +(K-1)  *L +(J-1) *L*N
      C(IS-1)=F(INODIS)
      DO 131 I=IS,IT
C
      INOD= I  +(K-1)  *L +(J-1) *L*N
      IECC= I+1+(K-1)  *L+(J-1)  *L*N
      IWCC= I-1+(K-1)  *L+(J-1)  *L*N
      ICNC= I  +(K-1)  *L+(J)    *L*N
      ICSC= I  +(K-1)  *L+(J-2)  *L*N
      ICCT= I  +(K)    *L+(J-1)  *L*N
      ICCB= I  +(K-2)  *L+(J-1)  *L*N
C
      A(I)=AE(INOD)
      B(I)=AW(INOD)
      C(I)=SU(INOD)+AN(INOD)*F(ICNC)+AS(INOD)*F(ICSC)+
     1  AT(INOD)*F(ICCT)+AB(INOD)*F(ICCB)
      D(I)=AP(INOD)
      ERRF=ERRF+ABS(C(I)-D(I)*F(INOD)+A(I)*F(IECC)+B(I)*F(IWCC))
      TERM=1.0/(D(I)-B(I)*A(I-1))
      A(I)=A(I)*TERM
 131  C(I)=(C(I)+B(I)*C(I-1))*TERM
      DO 132 II=IS,IT
      I=IS+IT-II
      INOD= I  +(K-1)  *L +(J-1) *L*N
      IECC= I+1+(K-1)  *L+(J-1)  *L*N
      F(INOD)=A(I)*F(IECC)+C(I)
 132  CONTINUE
 130  CONTINUE
      IF(ISW.EQ.1) ERRF1=AMAX1(ERRF,0.000001)
      IF((ERRF/ERRF1).LE.1.E-01) RETURN
  90  CONTINUE
      RETURN
      END
C
```