1521

NASA CONTRACTOR
REPORT

NASA CR-178818

A COMPUTER CODE FOR THREE-DIMENSIONAL INCOMPRESSIBLE FLOWS USING NONORTHOGONAL BODY-FITTED COORDINATE SYSTEMS

By Y. S. Chen
Universities Space Research Association
Science and Engineering Directorate
Atmospheric Sciences Division
Systems Dynamics Laboratory

Final Report

March 1986

(NASA-CR-178818) A COMPUTER CODE FOR THREE-DIMENSIONAL INCOMPRESSIBLE FLOWS USING NONORTHOGONAL BODY-FITTED COORDINATE SYSTEMS Final Report (Universities Space Research Association) 86 p HC A05/MF A01 CSCL 20D G3/34

N86-26544

Unclas
42915

Prepared for
NASA-Marshall Space Flight Center
Marshall Space Flight Center, Alabama 35812

TECHNICAL REPORT STANDARD TITLE PAGE

1. REPORT NO. NASA CR-178818	2. GOVERNMENT ACCESSION NO.	3. RECIPIENT'S CATALOG NO.
4. TITLE AND SUBTITLE A Computer Code for Three-Dimensional Incompressible Flows Using Nonorthogonal Body-Fitted Coordinate Systems		5. REPORT DATE March 1986
		6. PERFORMING ORGANIZATION CODE
7. AUTHOR(S) *Y. S. Chen		8. PERFORMING ORGANIZATION REPORT #
9. PERFORMING ORGANIZATION NAME AND ADDRESS George C. Marshall Space Flight Center Marshall Space Flight Center, Alabama 35812		10. WORK UNIT NO.
		11. CONTRACT OR GRANT NO. NAS8-35918
12. SPONSORING AGENCY NAME AND ADDRESS National Aeronautics and Space Administration Washington, D.C. 20546		13. TYPE OF REPORT & PERIOD COVERED Final Contractor Report
		14. SPONSORING AGENCY CODE

15. SUPPLEMENTARY NOTES *Universities Space Research Association

Prepared by Atmospheric Sciences Division, Systems Dynamics Laboratory, Science and Engineering Directorate.

16. ABSTRACT

In this report, a numerical method for solving the equations of motion of three-dimensional incompressible flows in nonorthogonal body-fitted coordinate (BFC) systems has been developed. The equations of motion are transformed to a generalized curvilinear coordinate system from which the transformed equations are discretized using finite difference approximations in the transformed domain. The hybrid scheme is used to approximate the convection terms in the governing equations. Solutions of the finite difference equations are obtained iteratively by using a pressure-velocity correction algorithm (SIMPLE-C). Numerical examples of two- and three-dimensional, laminar and turbulent flow problems are employed to evaluate the accuracy and efficiency of the present computer code. The user's guide and computer program listing of the present code are also included.

17. KEY WORDS Navier-Stokes Solver; Incompressible; Curvilinear Coordinate; Computer Code	18. DISTRIBUTION STATEMENT Unclassified-Unlimited

19. SECURITY CLASSIF. (of this report)	20. SECURITY CLASSIF. (of this page)	21. NO. OF PAGES	22. PRICE
Unclassified	Unclassified	87	NTIS

MSFC - Form 3292 (May 1969)

For sale by National Technical Information Service, Springfield, Virginia 22151

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

NOMENCLATURE

Symbol	Definition
A	link coefficient of finite difference equation
A_p	link coefficient of the node at the center of a control volume
A^u	link coefficient for the u-equation
A^v	link coefficient for the v-equation
A^w	link coefficient for the w-equation
A^o	link coefficient for time marching scheme
C_v	specific heat constant at constant volume
C_1	turbulence model constant, = 1.44
C_2	turbulence model constant, = 1.92
C_μ	turbulence model constant, = 0.09
D	diffusion coefficient for the pressure correction equation
e	internal energy per unit mass (jour/kg)
J	Jacobian of the metric transformation
k	turbulence kinetic energy (m^2/S^2)
k'	thermal conductivity of the fluid
P	pressure in the fluid (N/m^2)
P_r	production term for the turbulent kinetic energy
Q	energy added per unit volume ($jour/m^3$)
S	source term
S_u	source term of the u-equation
S_v	source term of the v-equation
S_w	source term of the w-equation
T	temperature (°K)
t	time (sec)
u	velocity in x direction

v	velocity in y direction
w	velocity in z direction
X	X-coordinate (m)
Y	Y-coordinate (m)
Z	Z-coordinate (m)

Greek

Γ	diffusion coefficient
ε	turbulent kinetic energy dissipation rate (m^2/s^3)
Δ	difference operator
ϕ	variable of general transport equation
ϕ^o	solution at the previous time level
σ_k	turbulence model constant, = 1.0
σ_ε	turbulence model constant, = 1.3
ξ	curvilinear coordinate
ζ	curvilinear coordinate
μ	molecular viscosity ($N\text{-}S/m^2$)
μ_t	turbulent eddy viscosity ($N\text{-}S/m^2$)
μ_{eff}	effective viscosity ($N\text{-}S/m^2$)
ρ	density (kg/m^3)
$\sum_i$	summation over all values around a grid node P
η	curvilinear coordinate

Subscript

i	index of all possible values
ref	reference value
max	maximum quantity

Superscript

o	previous time level solution
*	current solution
'	correction quantity

CONTRACTOR REPORT

A COMPUTER CODE FOR THREE-DIMENSIONAL INCOMPRESSIBLE FLOWS USING NONORTHOGONAL BODY-FITTED COORDINATE SYSTEMS

INTRODUCTION

With the currently increasing computer capability and various flow solvers developed, numerical simulations of three-dimensional incompressible flow problems using Reynolds-average Navier-Stokes equations are now becoming more feasible in many engineering design and analysis applications. In many real world flow problems, the boundary geometries are complex such that it is more accurate to describe the geometries using body-fitted coordinate (BFC) systems. Especially for internal flow problems with complex geometries such as those of the hot gas manifold (HGM) of the Space Shuttle Main Engine (SSME), the use of nonorthogonal BFC systems for numerical solutions can be beneficial in many aspects. It is not only the boundary geometries that can be represented more closely using BFC systems, but also grid-refined solutions can be obtained without increasing an excessive amount in computer memory. In addition, once a particular flow problem has been set up, the redesign or optimization process of the boundary shapes can be performed very easily using BFC systems.

Several numerical methods [1,2,3,4,5,6] has been developed for solving the incompressible Navier-Stokes equations in 3-D BFC systems. The main difference between these methods lies in the way of finding a pressure field such that the flowfield can be as close to divergence-free as possible (i.e. to satisfy the mass conservation equation). This is the main feature and difficulty of solving the incompressible flow problems. Numerical methods of References 1, 2 and 3, for instance, have employed the pseudocompressibility approach and time-iterative scheme to generate the pressure field so that the continuity equation is satisfied when a steady state solution is reached. In these methods, artificial smoothing techniques must be used to obtain a strong coupling between the velocity and pressure fields. Methods of References 4, 5 and 6, on the other hand, have utilized a successive pressure-velocity correction scheme by using a Poisson's equation for pressure correction derived approximately form the continuity and momentum equations. For these latter methods, grid staggering between the velocity vectors and the pressure nodes must be used to ensure stability of the numerical solutions.

There are several possible methods of grid staggering associated with different features in solving the pressure correction equation. These grid staggering methods were discussed in Reference 6, from which one of the methods was shown to be the most promising arrangement (i.e. with the velocity vectors located at the faces of a volume which contains the pressure and other scalars at its center). But, this method has one drawback, that the velocity components are solved using different control volumes. It is for this reason that a grid staggering system similar to the one used by Vanka et al. [4] is developed in the present study. The present method of grid staggering and pressure correction equation that was described by Vanka [4[and Maliska [6]. Also, using the present method, the same control volume is used for the velocity components and scalar quantities.

In the following sections, basic elements for establishing the present computer code for solving the curvilinear Navier-Stokes equations in three-dimensional space (CNS3D) will be described. These are followed by a series of standard numerical examples used to evaluate the accuracy and efficiency of the present numerical method. The numerical examples include laminar flow driven-cavity problem, cases of laminar and turbulent flows over backward-facing steps, and 3-D laminar flows inside a 90-deg-bend square duct. Applications of the present code to the internal flow problems of SSME will be included in future publications.

A user's guide to the present CNS3D code is provided in Appendix A. Appendix B contains a list and definitions of all the major fortran symbols used in the computer program which is listed in Appendix C.

TRANSFORMATION OF THE EQUATIONS OF MOTION

For incompressible Newtonian fluid, the continuity, momentum and energy equations can be written as:

$$U_t + E_x + F_y + G_z = S \tag{1}$$

where (x,y,z) represent the Cartesian coordinates, and

$$U = \begin{Bmatrix} \rho \\ \rho u \\ \rho v \\ \rho w \\ \rho e - Q \end{Bmatrix} \qquad E = \begin{Bmatrix} \rho u \\ \rho uu - \mu u_x \\ \rho uv - \mu v_x \\ \rho uw - \mu w_x \\ \rho ue - k' T_x \end{Bmatrix}$$

$$F = \begin{Bmatrix} \rho v \\ \rho vu - \mu u_y \\ \rho vu - \mu v_y \\ \rho vw - \mu w_y \\ \rho ve - k' T_y \end{Bmatrix} \qquad G = \begin{Bmatrix} \rho w \\ \rho wu - \mu u_z \\ \rho wv - \mu w_z \\ \rho ww - \mu w_z \\ \rho we - k' T_z \end{Bmatrix}$$

$$
S = \left\{ \begin{array}{l}
0 \\
(\mu u_x)_x + (\mu v_x)_y + (\mu w_x)_z - P_x \\
(\mu u_y)_x + (\mu v_y)_y + (\mu w_y)_z - P_y \\
(\mu u_z)_x + (\mu v_z)_y + (\mu w_z)_z - P_z \\
\mu [2(u_x{}^2 + v_y{}^2 + w_z{}^2) + (v_x + u_y)^2 + (w_y + v_z)^2 + (u_z + w_x)^2 \\
\qquad\qquad\qquad\qquad\qquad\qquad - \frac{2}{3} (u_x + v_y + w_z)^2]
\end{array} \right\}
$$

e = the internal energy per unit mass = $C_v T$ for perfect gas

Q = energy added per unit volume

k' = thermal conductivity of the fluid .

Equation (1) is transformed to a general curvilinear coordinate system (ξ, η, ζ), which results in equation (2).

$$
\begin{aligned}
& U_t + E_\xi \, \xi_x + E_\eta \, \eta_x + E_\zeta \, \zeta_x + F_\xi \, \xi_y + F_\eta \, \eta_y + F_\zeta \, \zeta_y \\
& \quad + G_\xi \, \xi_z + G_\eta \, \eta_z + G_\zeta \, \zeta_z = S
\end{aligned} \tag{2}
$$

where

$$\xi_x = J(y_\eta \, z_\zeta - y_\zeta \, z_\eta)$$

$$\xi_y = -J(x_\eta \, z_\zeta - x_\zeta \, z_\eta)$$

$$\xi_z = J(x_\eta \, y_\zeta - x_\zeta \, y_\eta)$$

$$\eta_x = -J(y_\xi \, z_\zeta - y_\zeta \, z_\xi)$$

$$\eta_y = J(x_\xi \, z_\zeta - x_\zeta \, z_\xi)$$

$$\eta_z = -J(x_\xi \, y_\zeta - x_\zeta \, y_\xi)$$

$$\zeta_x = J(y_\xi z_\eta - y_\eta z_\xi)$$

$$\zeta_y = -J(x_\xi z_\eta - x_\eta z_\xi)$$

$$\zeta_z = J(x_\xi y_\eta - x_\eta y_\xi)$$

$$J = 1/[x_\xi(y_\eta z_\zeta - y_\zeta z_\eta) - x_\eta(y_\xi z_\zeta - y_\zeta z_\xi) + x_\zeta(y_\xi z_\eta - y_\eta z_\xi)] \quad .$$

The transformation coefficients, ξ_x, ξ_y, ξ_z, η_x, η_y, η_z, ζ_x, ζ_y, and ζ_z, are computed numerically using second order central differencing. In the transformed domain, the grid sizes (i.e., $\Delta\xi$, $\Delta\eta$, and $\Delta\zeta$) are set to be unity. This simplifies the calculation of the transformation coefficients.

For turbulent flow computations, the present code has employed the standard k-ε turbulence model [7] to provide the turbulent eddy viscosity μ_t. The standard k-ε turbulence model (which consists of a turbulent kinetic energy equation, k-equation, and a turbulent kinetic energy dissipation rate equation, ε-equation) is given as:

$$(\rho k)_t + \left(\rho u_i k - \frac{\mu_{eff}}{\sigma_k} k_{x_i}\right)_{x_i} = \rho(P_r - \varepsilon) \qquad (3)$$

$$(\rho \varepsilon)_t + \left(\rho u_i k - \frac{\mu_{eff}}{\sigma_\varepsilon} \varepsilon_{x_i}\right)_{x_i} = \rho \frac{\varepsilon}{k} (C_1 P_r - C_2 \varepsilon) \qquad (4)$$

where the effective viscosity μ_{eff} is calculated from:

$$\mu_{eff} = \mu + \mu_t = \mu + \rho\, C_\mu\, k^2/\varepsilon$$

and the turbulent kinetic energy production term, P_r, is defined as:

$$P_r = C_\mu \frac{k^2}{\varepsilon} [(u_y + v_x)^2 + (v_z + w_y)^2 + (w_x + u_z)^2 + 2(u_x{}^2 + v_y{}^2 + w_z{}^2)]$$

The turbulence model constants are:

$$C_\mu = 0.09 \quad , \quad \sigma_k = 1.0 \quad , \quad \sigma_\varepsilon = 1.3$$

$$C_1 = 1.44 \quad , \quad C_2 = 1.92 \quad .$$

Also, the molecular viscosity μ in equation (1) is replaced by the effective viscosity μ_{eff} for turbulent flow cases.

In order to save the computational efforts, the widely used wall function approach [8] is employed to provide the near wall boundary conditions for the momentum and energy equations and the k-ε turbulence model. This approach avoids the requirement of integrating the governing equations up to the wall which requires a large number of additional grid points near the wall.

Equations (2), (3), and (4) form a closed set of nonlinear partial differential equations governing the fluid motion. This set of equations are to be solved by means of finite difference approximations which are performed in the transformed domain. For treating the convection terms, the hybrid scheme [9] is employed for simplicity (although other more elaborate schemes such as central differencing plus artificial dissipation scheme, QUICK scheme, or skew upwind differencing scheme, etc. can be implemented [10]). These are described in the following sections.

DISCRETIZATION OF THE EQUATIONS OF MOTION

In this section, finite difference approximations are used to discretize the governing equations, equations (2), (3), and (4). Second-order central differencing is used for the diffusion terms and the source terms. The hybrid differencing scheme [9] is employed to approximate the convection terms in the governing equations. The finite difference discretizations are performed in the transformed domain. The solution procedure for the discretized equations using a velocity-pressure correction algorithm (SIMPLE-C) of References 11 and 12 will be described in the next section.

The governing equations of motion can be represented by the following model transport equation in which ϕ denotes all the dependent variables respectively and Γ is the diffusion coefficient..

$$
\begin{aligned}
(\rho\phi)_t &+ [\rho u\phi - \Gamma(\phi_\xi\, \xi_x + \phi_\eta\, \eta_x + \phi_\zeta\, \zeta_x)]_\xi\, \xi_x \\
&+ [\rho u\phi - \Gamma(\phi_\xi\, \xi_x + \phi_\eta\, \eta_x + \phi_\zeta\, \zeta_x)]_\eta\, \eta_x \\
&+ [\rho u\phi - \Gamma(\phi_\xi\, \xi_x + \phi_\eta\, \eta_x + \phi_\zeta\, \zeta_x)]_\zeta\, \zeta_x \\
&+ [\rho v\phi - \Gamma(\phi_\xi\, \xi_y + \phi_\eta\, \eta_y + \phi_\zeta\, \zeta_y)]_\xi\, \xi_y \\
&+ [\rho v\phi - \Gamma(\phi_\xi\, \xi_y + \phi_\eta\, \eta_y + \phi_\zeta\, \zeta_y)]_\eta\, \eta_y \\
&+ [\rho v\phi - \Gamma(\phi_\xi\, \xi_y + \phi_\eta\, \eta_y + \phi_\zeta\, \zeta_y)]_\zeta\, \zeta_y \\
&+ [\rho w\phi - \Gamma(\phi_\xi\, \xi_z + \phi_\eta\, \eta_z + \phi_\zeta\, \zeta_z)]_\xi\, \xi_z \\
&+ [\rho w\phi - \Gamma(\phi_\xi\, \xi_z + \phi_\eta\, \eta_z + \phi_\zeta\, \zeta_z)]_\eta\, \eta_z \\
&+ [\rho w\phi - \Gamma(\phi_\xi\, \xi_z + \phi_\eta\, \eta_z + \phi_\zeta\, \zeta_z)]_\zeta\, \zeta_z = S \quad . \qquad (5)
\end{aligned}
$$

Discretization of equation (5) is performed using finite difference approximations in the transformed domain. The second order central differencing is used for approximating the diffusion terms. For the convection terms, the hybrid differencing scheme [9] is employed (i.e., using central differencing for cell Peclet number less than or equal to 2 and switching to upwind differencing when the cell Peclet number is greater than 2). The finite difference equation is arranged by collecting terms according to the grid nodes around a control volume as shown in Figure 1. The final expression is given by equation (6) in which A represents the link coefficients between grid nodes P, E, W, N, S, T, B, NE, NW, NT, NB, SE, SW, ST, SB, ET, EB, WT, and WB as shown in Figure 1.

$$A_P \phi_P = A_E \phi_E + A_W \phi_W + A_N \phi_N + A_S \phi_S + A_T \phi_T + A_B \phi_B + S_1 \tag{6}$$

where

$$S_1 = S + A_P^\circ \phi_P^\circ + A_{NE} \phi_{NE} + A_{NW} \phi_{NW} + A_{NT} \phi_{NT} + A_{NB} \phi_{NB}$$
$$+ A_{SE} \phi_{SE} + A_{SW} \phi_{SW} + A_{ST} \phi_{ST} + A_{SB} \phi_{SB}$$
$$+ A_{ET} \phi_{ET} + A_{EB} \phi_{EB} + A_{WT} \phi_{WT} + A_{WB} \phi_{WB}$$

$$A_P = A_E + A_W + A_N + A_S + A_T + A_B + A_P^\circ$$

$$A_P^\circ = \rho_P^\circ / \Delta t \quad .$$

The subscript ° denotes the solution at the previous time level. A fully implicit formulation is employed for solving the time dependent transient problems.

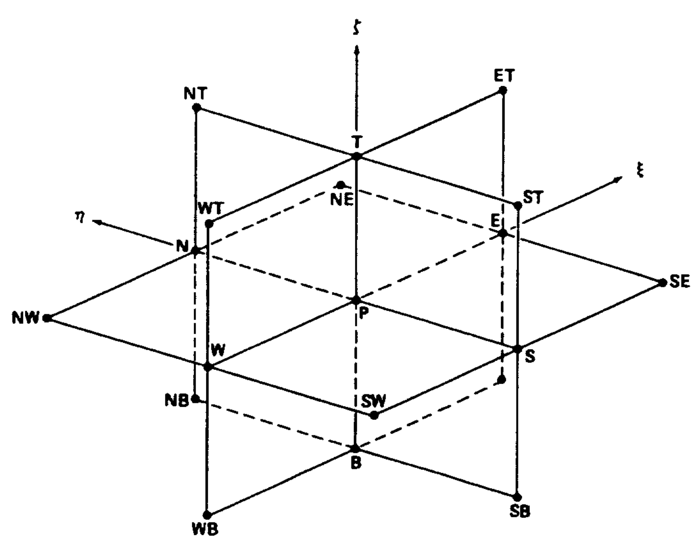


Figure 1. Three-dimensional grid structure and labeling around a grid node P.

Thus, the nonlinear equations of motion are approximated by a system of linear algebraic equations which have the form of equation (6). Only one program subroutine is designed to calculate the link coefficients and the source terms. The number of algebraic equations depends on the number of interior grid points. For a grid size of 10 x 10 x 10 the number of algebraic equations to be solved would be around 512. This large system of equations are preferred to be solved by some iterative methods, such as Gauss-Seidel iteration, line-underrelaxation method [13] or Stone's method [14], etc., rather than using direct methods such as Gaussian elimination method. Only a few (6 to 10) iterations through the whole computational domain are needed and a complete convergence of the system of algebraic equations is not required. Since equation (6) is only a linearized version of the governing equations which are nonlinear and coupled in nature, solutions of the equations of motion must be obtained through global iterations among the equations. A tentative solution to equation (6) will not affect the final results significantly. On the other hand, if too many iterations are used to get a better solution of equation (6), then a great deal of computing time would be virtually wasted. However, the above argument can not be applied when the pressure correction equation (which will be derived in the next section) is solved. Since during each global iteration it is desirable to retain a divergence-free velocity field, better solution of the pressure correction equation would in effect promote the convergence of the whole numerical scheme. Therefore, more iterations are usually used to solve the pressure correction equation.

SOLUTION PROCEDURES

The governing equations used in the present analysis are nonlinear and strongly coupled. Iterative procedures are employed to drive the equations to a converged solution. It is particularly important for incompressible flow to make the flow field satisfy the continuity equation and the momentum equations at the same time. This requires a correct pressure field associated with a divergence-free velocity field. A velocity-pressure correction procedure is developed in the present study to drive the pressure field and the velocity field to be divergence free. This kind of procedure requires grid staggering between the velocity components and the locations where the pressure is estimated and stored such that the velocity field and the pressure field will not be uncoupled.

In the present study, staggering grid systems as shown in Figure 2 (for 2-D case) are used. The velocity components, u and v, are solved and stored at the grid nodes and the pressure, p, is located at the corners of the control volume of u and v. In this way, solutions of u and v can be solved using the same control volume and coupling between u, v and p can also be enforced. To estimate the pressure field, a pressure correction equation is derived approximately from the discretized momentum and continuity equations. The velocity and pressure fields are then corrected using the solutions of the pressure correction equation.

First, the finite difference momentum equations (for u, v and w) can be written as:

$$A_P^u u_P^* = \sum_i A_i^u u_i^* = P_x^* + S_u \tag{8a}$$

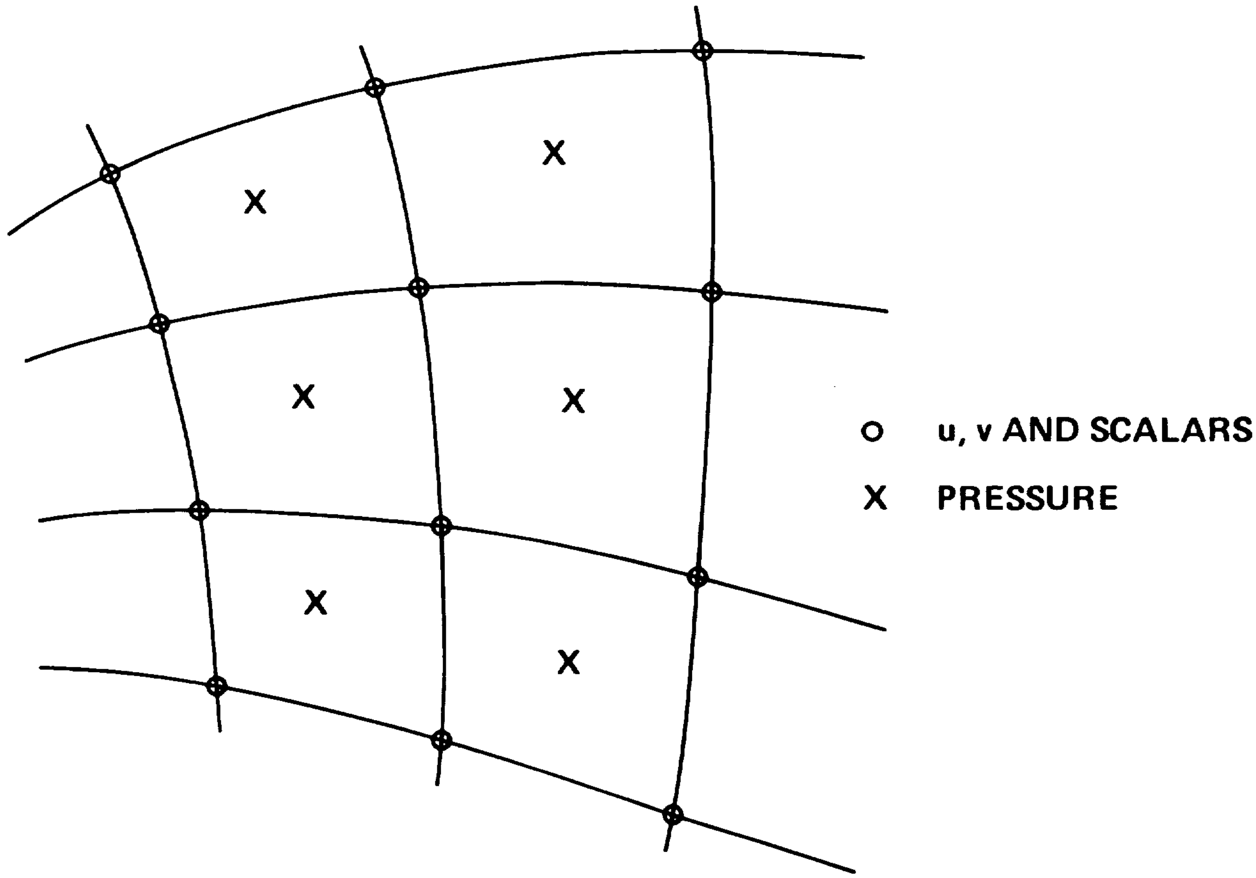


Figure 2. Locations where the variables are stored (staggering grids are used).

$$A_P^{\,v}\ v_P^{\,*} = \sum_i A_i^{\,v}\ v_i^{\,*} = P_y^{\,*} + S_v \tag{8b}$$

$$A_P^{\,w}\ w_P^{\,*} = \sum_i A_i^{\,w}\ w_i^{\,*} - P_z^{\,*} + S_w \tag{8c}$$

where u*, v*, w*, and p* represent the solutions of equations (8a) and (8b). To satisfy the continuity equation the velocities and pressure are corrected according to the following relations:

$$u = u^* + u' \tag{9a}$$

$$v = v^* + v' \tag{9b}$$

$$w = w^* + w' \tag{9c}$$

$$P = P^* + P' \tag{9d}$$

A new set of momentum equations can be constructed approximately using the divergence-free flow field, u, v, w, and p:

$$A_P^u u_P = \sum_i A_i^u u_i - P_x + S_u \tag{10a}$$

$$A_P^v v_P = \sum_i A_i^v v_i - P_y + S_v \tag{10b}$$

$$A_P^w w_P = \sum_i A_i^w w_i - P_z + S_w \tag{10c}$$

By subtracting equations (8a) through (8c) from equations (10a) through (10c), respectively, the following equations result:

$$A_P^u u_P' = \sum_i A_i^u u_i' - P_x' \tag{11a}$$

$$A_P^v v_p' = \sum_i A_i^v v_i' - P_y' \tag{11b}$$

$$A_P^w w_P' = \sum_i A_i^w w_i' - P_z' \tag{11c}$$

According to SIMPLE-C algorithm [11], equations (11a) through (11c) are rearranged to be:

$$(A_P^u - \sum_i A_i^u)\, u_P' = \sum_i A_i^u (u_i' - u_P') - P_x' \tag{12a}$$

$$(A_P^v - \sum_i A_i^v)\, v_P' = \sum_i A_i^v (v_i' - v_P') - P_y' \tag{12b}$$

$$(A_P^w - \sum_i A_i^w)\, w_P' = \sum_i A_i^w (w_i' - w_P') - P_z' \tag{12c}$$

The first terms on the right-hand side of equations (12a) through (12c) are neglected to simplify the formulation. Thus,

$$u_P' = -\left(\frac{1}{A_P^u - \sum_i A_i^u}\right) P_x' = - D_u P_x' \tag{13a}$$

$$v_P' = -\left(\frac{1}{A_P^v - \sum_i A_i^v}\right) P_y' = - D_v P_y' \tag{13b}$$

$$w_P' = -\left(\frac{1}{A_P^w - \sum_i A_i^w}\right) P_z' = - D_w P_z' \quad . \tag{13c}$$

Using the decompositions of equations (9a) through (9c), the continuity equation can be written as:

$$u_x + v_y + w_z = (u_x^* + v_y^* + w_z^*) + (u_x' + v_y' + w_z') = 0 \quad . \tag{14}$$

Substituting equations (13a) through (13c) into equation (14), the following pressure correction equation can be obtained:

$$-[(D_u P_x')_x + (D_v P_y')_y + (D_w P_z')_z] = -(u_x^* + v_y^* + w_z^*) \quad . \tag{15}$$

Equation (15) is a Poisson's equation with the source term equal to the local divergence of the flow field. To enforce the coupling between the velocity and pressure fields, the source term of equation (15) is first evaluated at the control volumes centered between the velocity nodes as shown in Figure 3. An averaged source term is then calculated at the cell center of p node for solving equation (15). In this way, the difficulties in solving the pressure correction equation, as described by Vanka [4] and Maliska [6], are eliminated. Coupling between the velocity and the pressure field is also assured.

According to the above analyses, the present numerical method contains the following solution steps:

1) Guess initial velocity and pressure field.

2) Solve for the velocity field using equations (8a) through (8c).

3) Solve for other scalar transport equations.

4) Solve the pressure correction equation, equation (15).

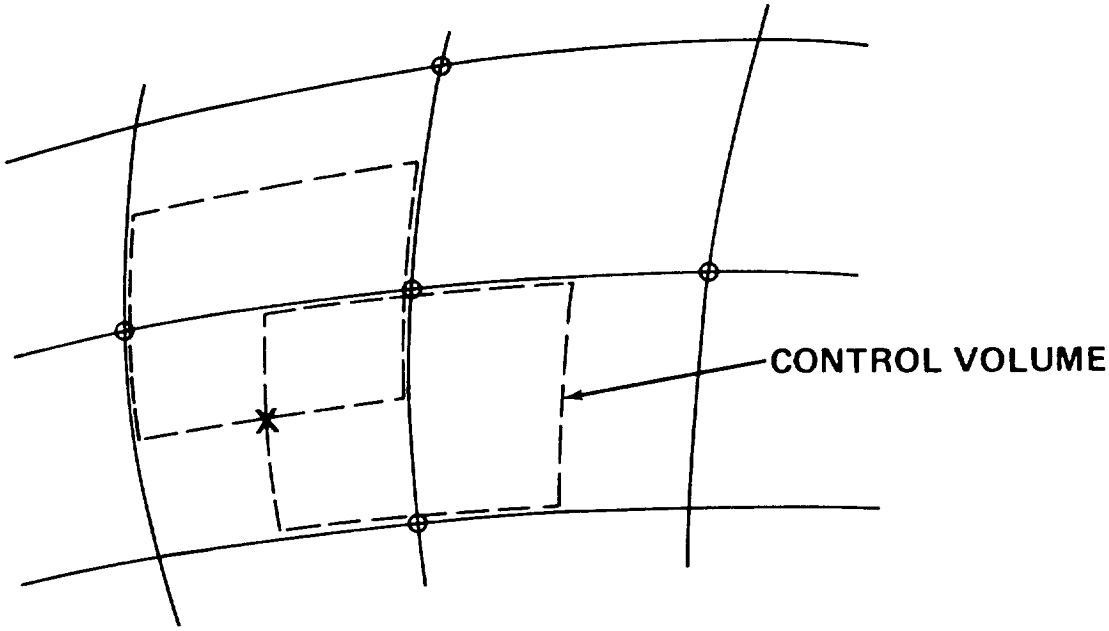


Figure 3. Control volumes where the mass conservation is evaluated for solving the pressure correction equation.

5) Correct the velocity and pressure fields using equations (13a) through (13c) and equation (9d).

6) Go back to step (2) until solution converges.

A converged solution is obtained when the following criterion is met:

$$\text{Error} \equiv (|\Delta u|_{max} + |\Delta v|_{max} + |\Delta w|_{max}) \;/\; U_{ref} + |P'|_{max}/\rho U_{ref}^2$$

$$\leq 3 \times 10^{-4} \quad ,$$

where Δu, Δv, and Δw represent velocity changes during each iteration due to the solutions of the momentum equations.

In solving the momentum equations in step (b) above, underrelaxation factor of about 0.6 is recommended. With this, A_P's in equations (8a) through (8c) are modified according to the underrelaxation factor. For the correction of velocity field, no underrelaxation is required. But the correction of pressure field should be underrelaxed slightly (around 0.9) when the grid nonorthogonality is strong. This is different from that suggested by References 11 and 12 (which recommend no underrelaxation for pressure correction).

NUMERICAL EXAMPLES

In this section, several numerical examples are employed to demonstrate the efficiency and accuracy of the present numerical method. To serve this purpose, 2-D and 3-D, laminar and turbulent flow cases are included. These cases are: (a) 2-D laminar driven square-cavity flows; (b) 2-D laminar flows over a backward-facing step; (c) 2-D turbulent flows over a backward-facing step; (d) 3-D developing laminar flow inside a 90-deg-bend square duct. Detailed descriptions and results of the computation of the above cases are included as follows.

A. 2-D Laminar Driven Square-Cavity Flows

The first test case is concerning laminar recirculating flows inside a square cavity. Only one side of the walls is moving at a constant speed tangent to that wall. This case has been studied extensively by Burggraf [15] and has often been used as one of the standard testing cases for numerical methods in solving the incompressible Navier-Stokes equations. Physical geometry and wall boundary conditions are illustrated in Figure 4. Reynolds number of the flow (based on the cavity size and the moving wall velocity) studied in the present analysis is 400. Two different mesh systems, as shown in Figure 5, are used to study the effect of grid non-orthogonality on the accuracy of the present method. The grid system of Figure 5(a) is uniform and orthogonal while the grid system of Figure 5(b) is non-uniform and non-orthogonal.

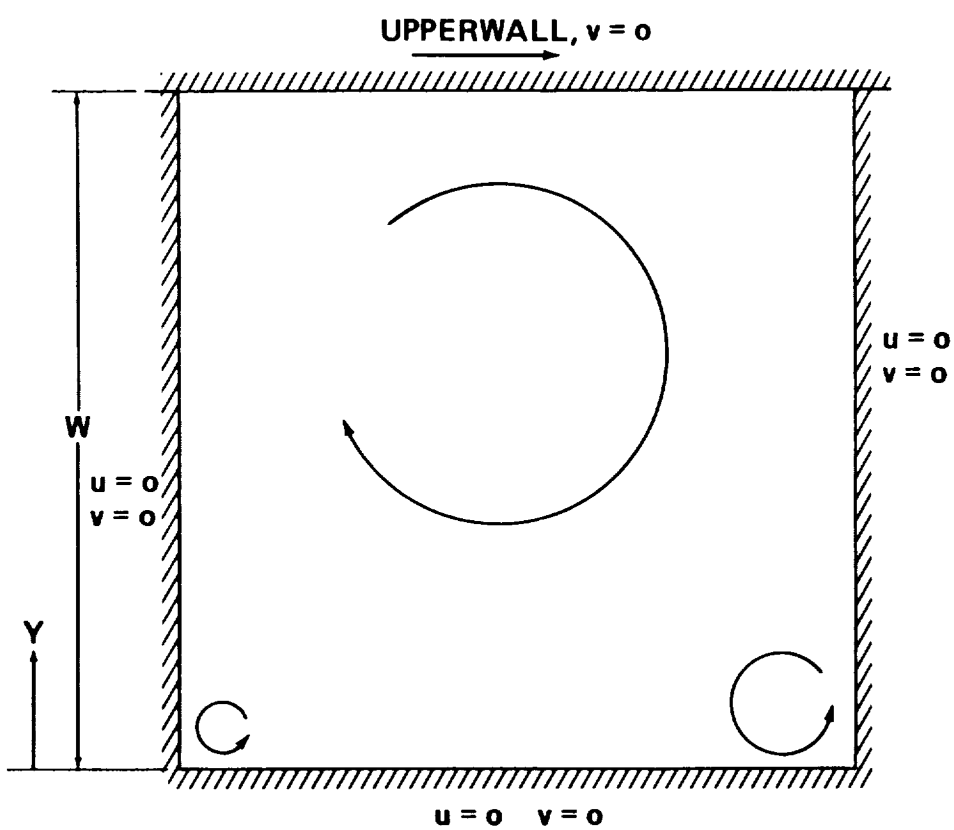
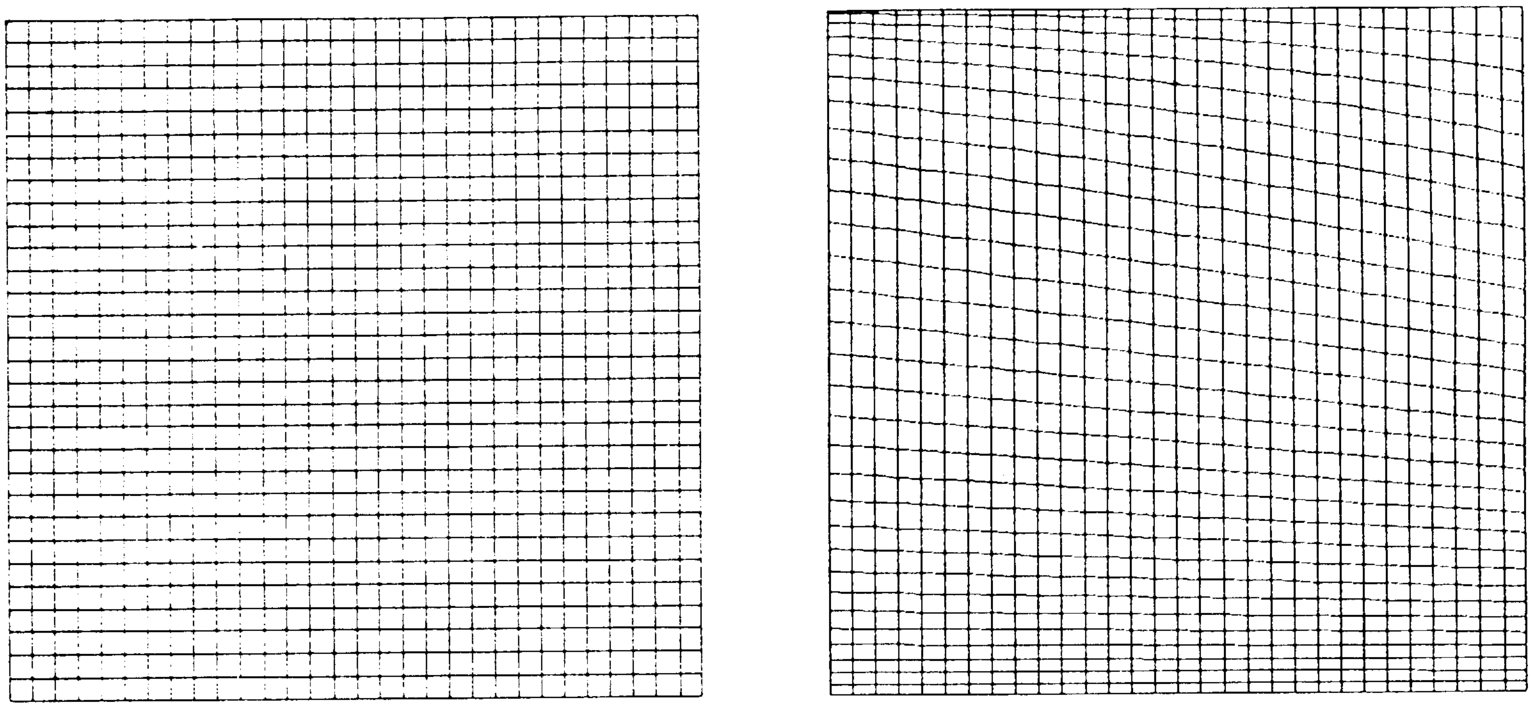


Figure 4. Physical geometry and wall boundary conditions for laminar flows inside a wall-driven square cavity.



(a) (b)

Figure 5. Mesh systems used for driven cavity problem. (a) Uniform and orthogonal grid. (b) Nonuniform and nonorthogonal grid.

Results of the computations are shown in Figures 6 and 7. Velocity vector plots of the predicted flow fields are compared in Figure 6 for the mesh systems shown in Figure 5. Detailed comparisons of the predicted velocity profiles along the mid-section of the cavity are illustrated in Figure 7. Predicted results of Burggraf [15] are also included. Good agreements between the present calculations and those

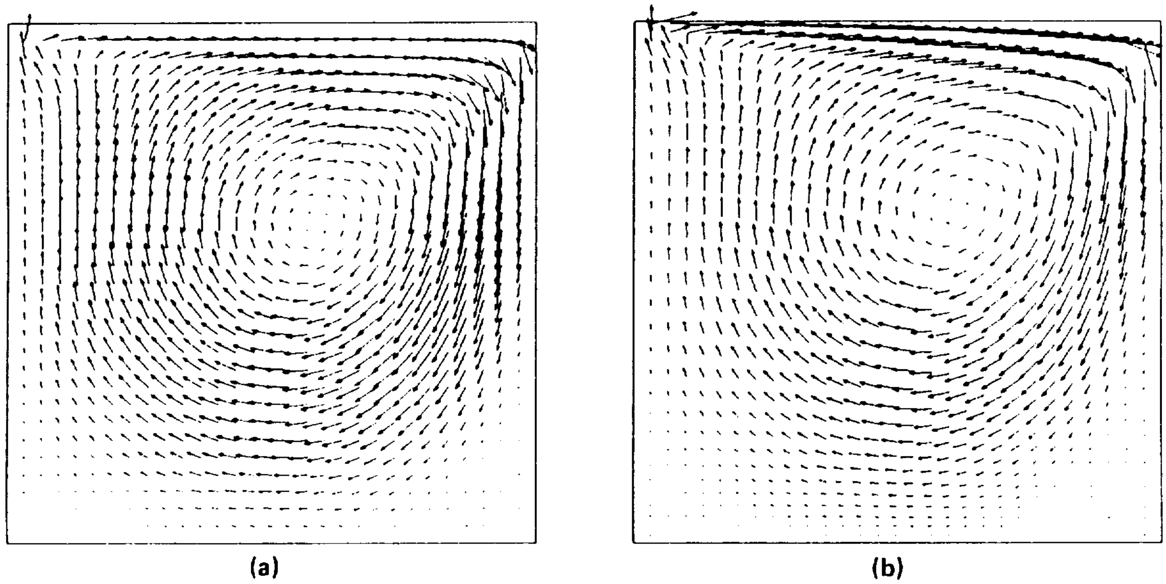


Figure 6. Velocity vector plots. (a) Orthogonal grid. (b) Nonorthogonal grid.

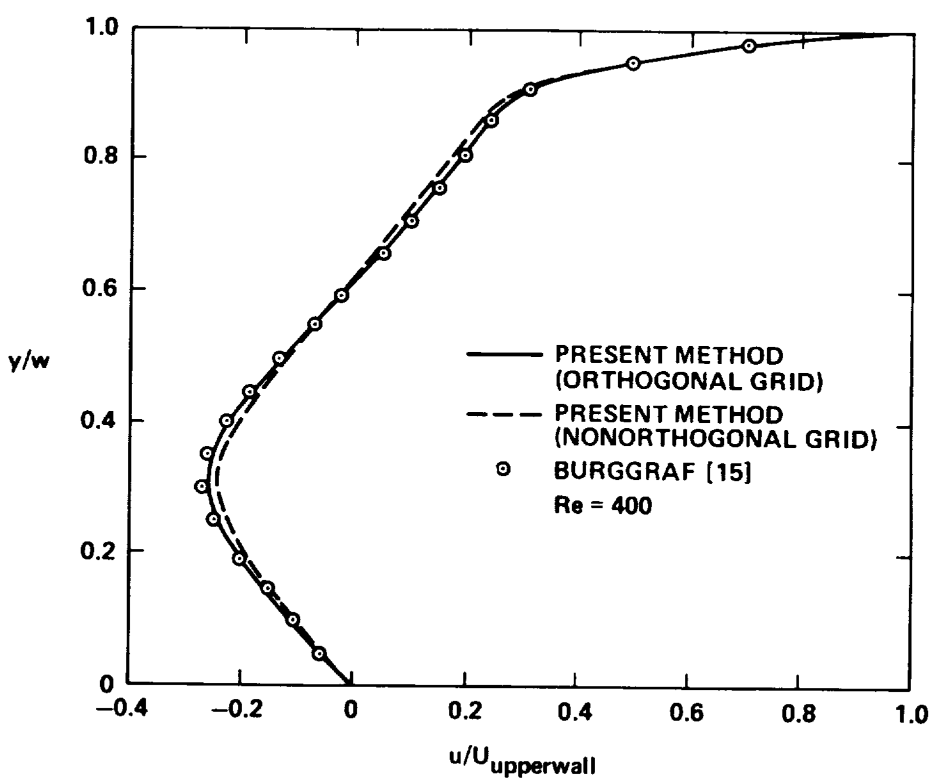


Figure 7. Comparisons of velocity profiles along the mid-section of the square cavity.

of Burggraf [15] are also included. Good agreements between the present calculations and those of Burggraf are shown in Figure 7. Discrepancies between the present predictions and Burggraf's results are mainly due to the hybrid differencing scheme used in the present method. The upwind part of the hybrid scheme produces large numerical diffusion which tends to reduce the strength of the vorticity inside the cavity. Effects of differencing schemes in approximating the convection terms on the predicted results will be studied in the next test case.

Convergence history of the computation of the present case using uniform grids is given in Figure 8 which shows that the present numerical method is quite different. Almost identical convergence rates were found for the non-orthogonal case.

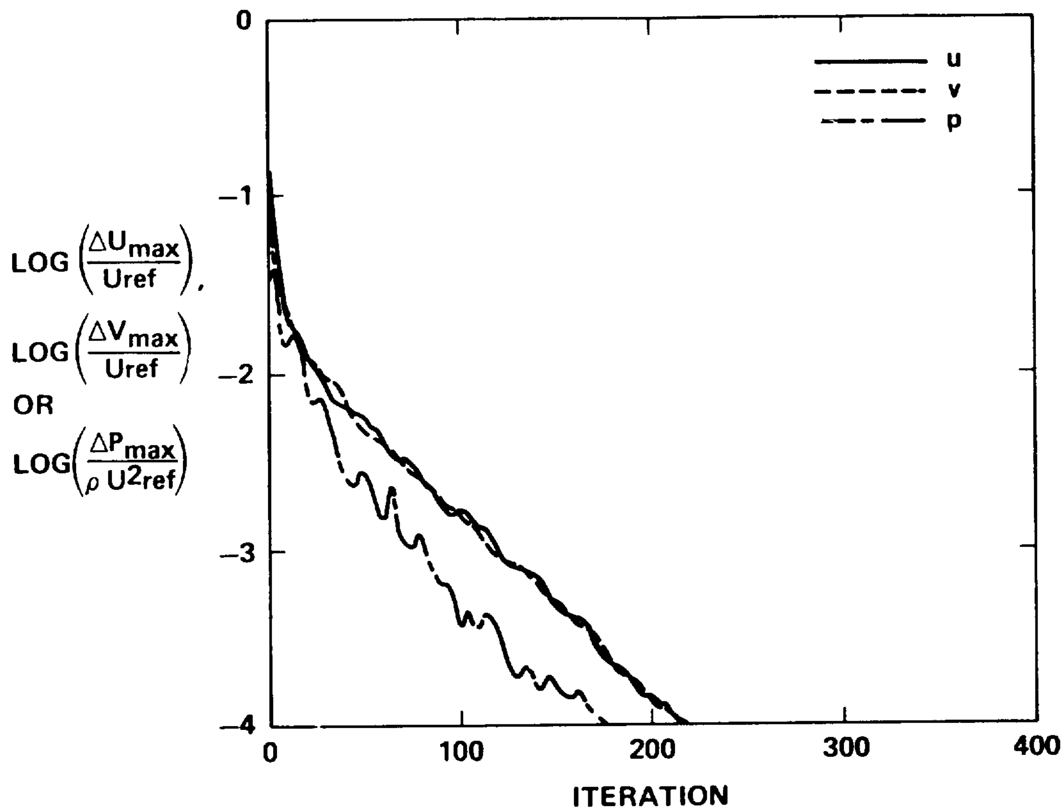


Figure 8. Convergence history for the driven cavity problem, Re = 400.

B. 2-D Laminar Flows Over a Backward-facing Step

This test case concerns 2-D laminar recirculating flows over a backward-facing step with 1:2 expansion ratio. The dependence of the size of the recirculation region (characterized by the reattachment length) on the Reynolds number (based on the inlet bulk velocity and twice of the inlet channel width) of the flow is of major concern. The physical domain and boundary conditions are illustrated in Figure 9 in which a fully developed laminar flow velocity profile is imposed at the flow entrance. A non-uniform grid of 45 x 45 was used for numerical computations. Several cases with different Reynolds numbers from 100 to 800 have been studied. An experimental and theoretical study about this problem, which results will be used as the basis of data comparisons, has been provided by Amaly et al. [16].

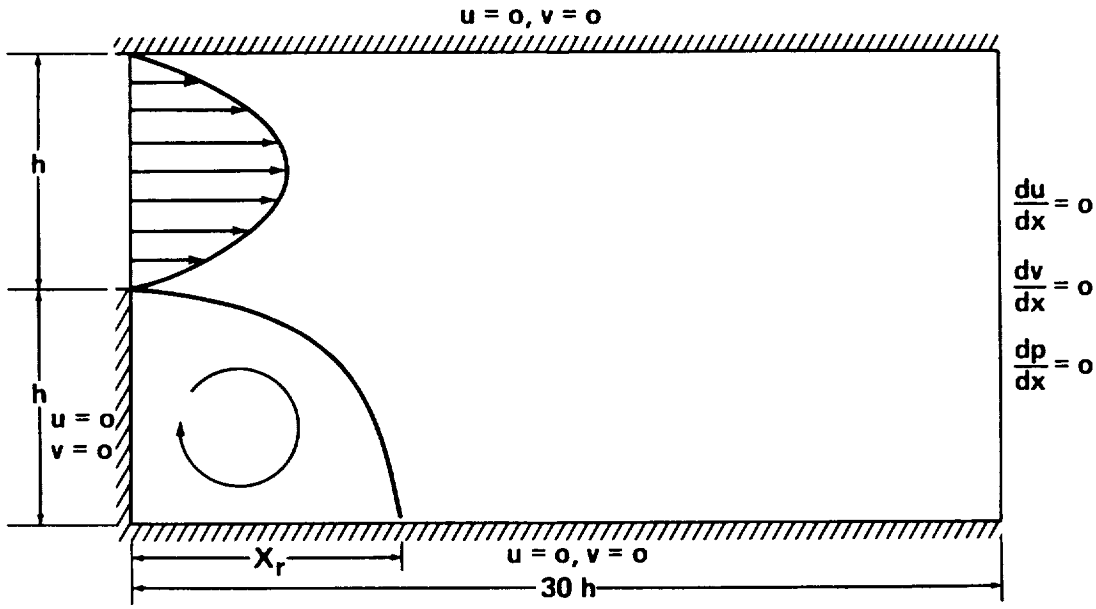


Figure 9. Physical geometry and boundary conditions of laminar flows over a backward-facing step (1:2 expansion).

To save computational efforts, the solution of one case with Reynolds number 100 is obtained in the first run. Then, a series of cases with increasing Reynolds numbers (i.e., 100, 200, 300, 400, 600, and 800) are calculated using the preceding results of lower Reynolds number as the initial guesses of the flow field. In this way, an average of 500 iterations for each case were needed to obtain converged solutions.

Two different differencing schemes in approximating the convection terms are employed to demonstrate the effects of the differencing schemes on the predictions. One of the schemes is the widely used hybrid scheme [9]. The other scheme employs the central differencing scheme plus an artificial dissipation term used to stabilize the solution which is similar to the one used by Rhie [17]. The artificial dissipation term becomes effective only when the cell Peclet number (or cell Reynolds number) exceeds 10.

Results of the present predictions using two different differencing schemes are compared with the experimental measurements [16] and other predictions as shown in Figure 10. It can be seen clearly from Figure 10 that the present method with hybrid scheme gives results similar to those predicted by TEACH code [16] while the present method with central differencing and artificial dissipation reveals predictions close to those predicted by INS3D [18] and the method of Kim and Moin [19]. This is reasonable since the TEACH code and the present method (with the first scheme) use the hybrid scheme which introduces large numerical dissipation by its upwind part (for cell Peclet number greater than 2). This tends to reduce the reattachment length for Reynolds number greater than 400. The second scheme, which is similar to the ones used in INS3D and the method of Kim and Moin, has the numerical accuracy close to second order by setting the artificial dissipation to be as small as the solution stability permits such that better accuracy of the predictions is expected.

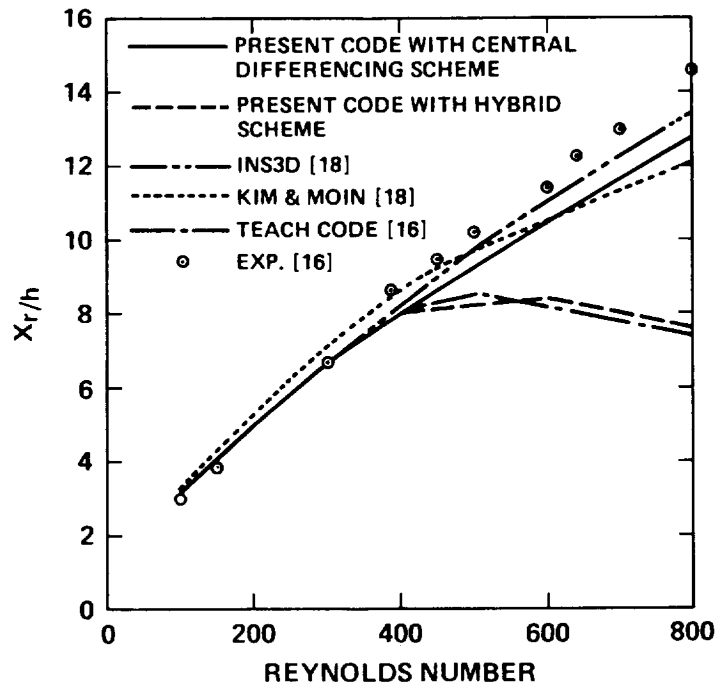


Figure 10. Reattachment length versus Reynolds number for laminar flows over a backward-facing step (1:2 expansion).

Stream function plots of the predictions using the two differencing schemes for Reynolds number 600 are compared in Figure 11. It is shown in Figure 11 that the second scheme gives a smooth shape of the recirculation zone while the hybrid scheme gives a sudden change in the shape of the recirculation region upstream of the reattachment point. Also, larger sizes of the separation regions on the step side wall and along the upper wall are predicted using the second scheme.

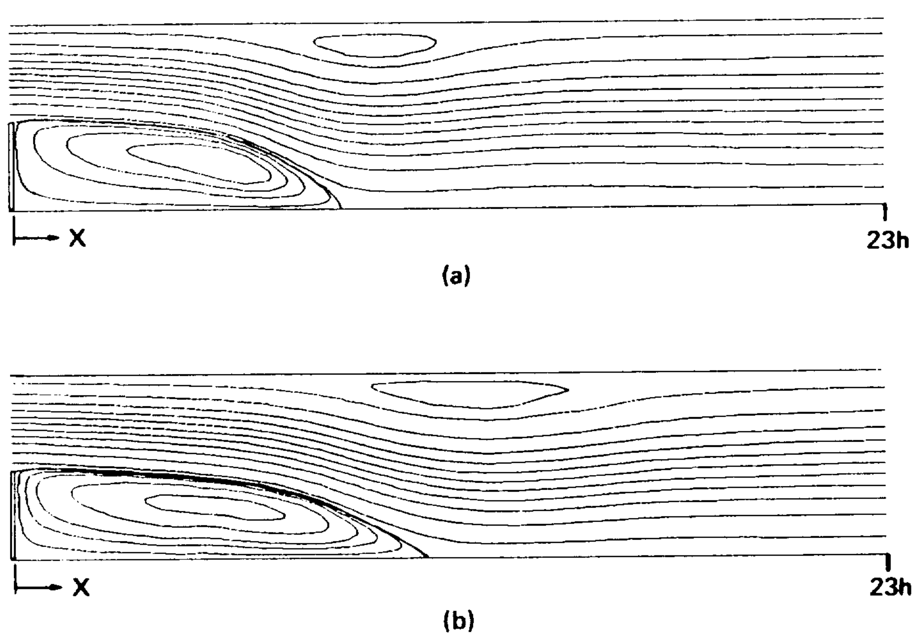


Figure 11. Streamline plots for laminar flow over a backward-facing step (1:2 expansion). (a) Hybrid Scheme. (b) Central differencing plus artificial dissipation scheme.

C. 2-D Turbulent Flows Over a Backward-Facing Step

In order to demonstrate the applicability of the present method to turbulent flow case, one of the standard test cases presented in the Stanford Conference [20] is selected here (i.e., turbulent flow over a 2:3 expansion backward-facing step). The standard k-ε turbulence model was used to provide the eddy viscosity for the transport equations. The physical geometry and boundary conditions imposed are shown in Figure 12. The calculation domain extends upstream of the expansion plane by 4 step heights and downstream of the expansion plane by 30 step heights to assure a fully developed velocity profile at the exit. A uniform velocity profile is located at the inlet plane. A 45 x 42 grid was used in the computation. 300 iterations were required to obtain converged solutions. Only hybrid differencing schemes were used in this case.

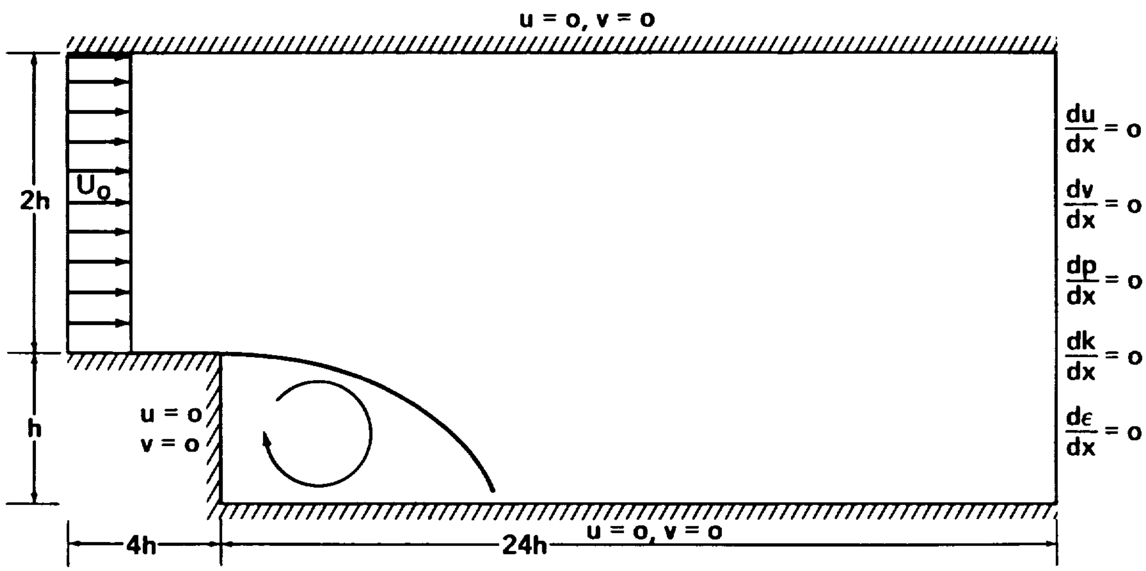


Figure 12. Physical geometry and boundary conditions of turbulent flows over a backward-facing step (2:3 expansion).

Results of the computation are shown in Figures 13, 14, and 15. These results are compared with the experimental measurements [20]. The under-prediction of the reattachment length is mainly due to the fast development of the mixing layer downstream of the expansion plane which is the characteristics of the standard k-ε turbulence model. Numerical diffusion provided by the hybrid scheme also contributes some part to the discrepancies between the predictions and measurements.

D. Developing Laminar Flow Inside a 90-Deg-Bend Square Duct

This test case simulates a three-dimensional developing laminar flow inside a 90-deg-bend square duct as illustrated in Figure 16(a). The symmetry plane is located at $z = 0$ where the symmetric boundary conditions are imposed. A fully developed velocity profile of laminar flow inside a straight square duct is prescribed at the entrance which is 2.8 duct widths upstream of the bend, A zero pressure

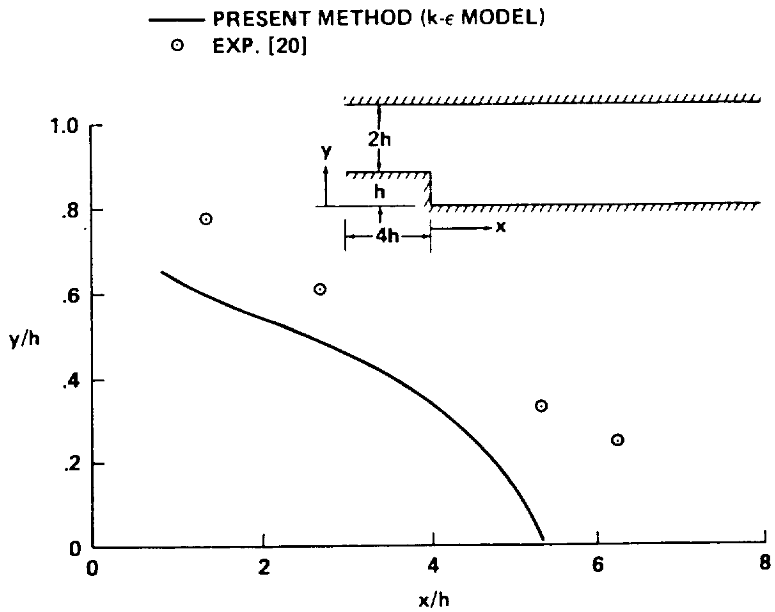


Figure 13. Locus of flow reversal inside the recirculation region for turbulent flow over a backward-facing step (2:3 expansion).

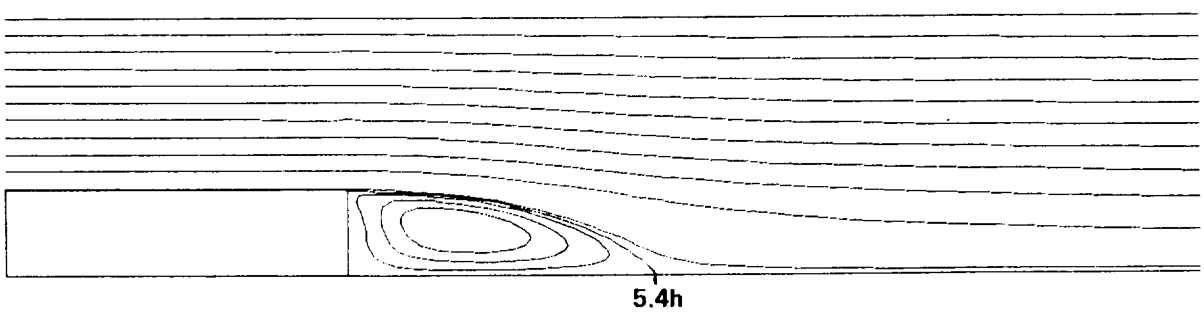


Figure 14. Stream line pattern of turbulent flow over a backward-facing step with 2:3 expansion ratio.

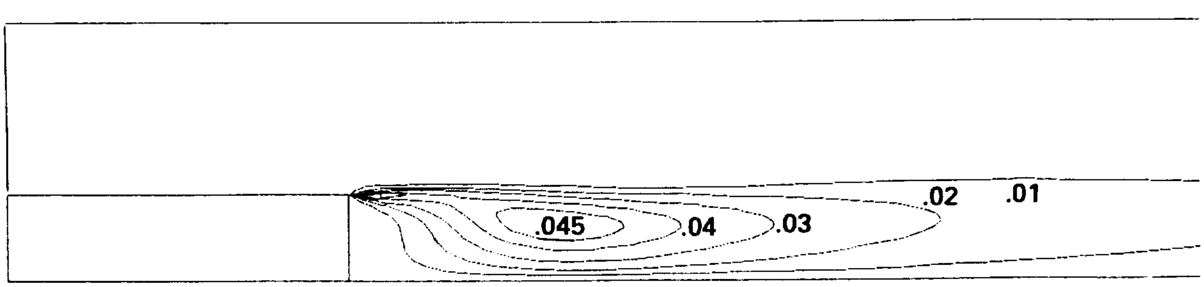


Figure 15. Contours of turbulent kinetic energy (k/U_o^2) of turbulent flow over a backward-facing step with 2:3 expansion ratio.

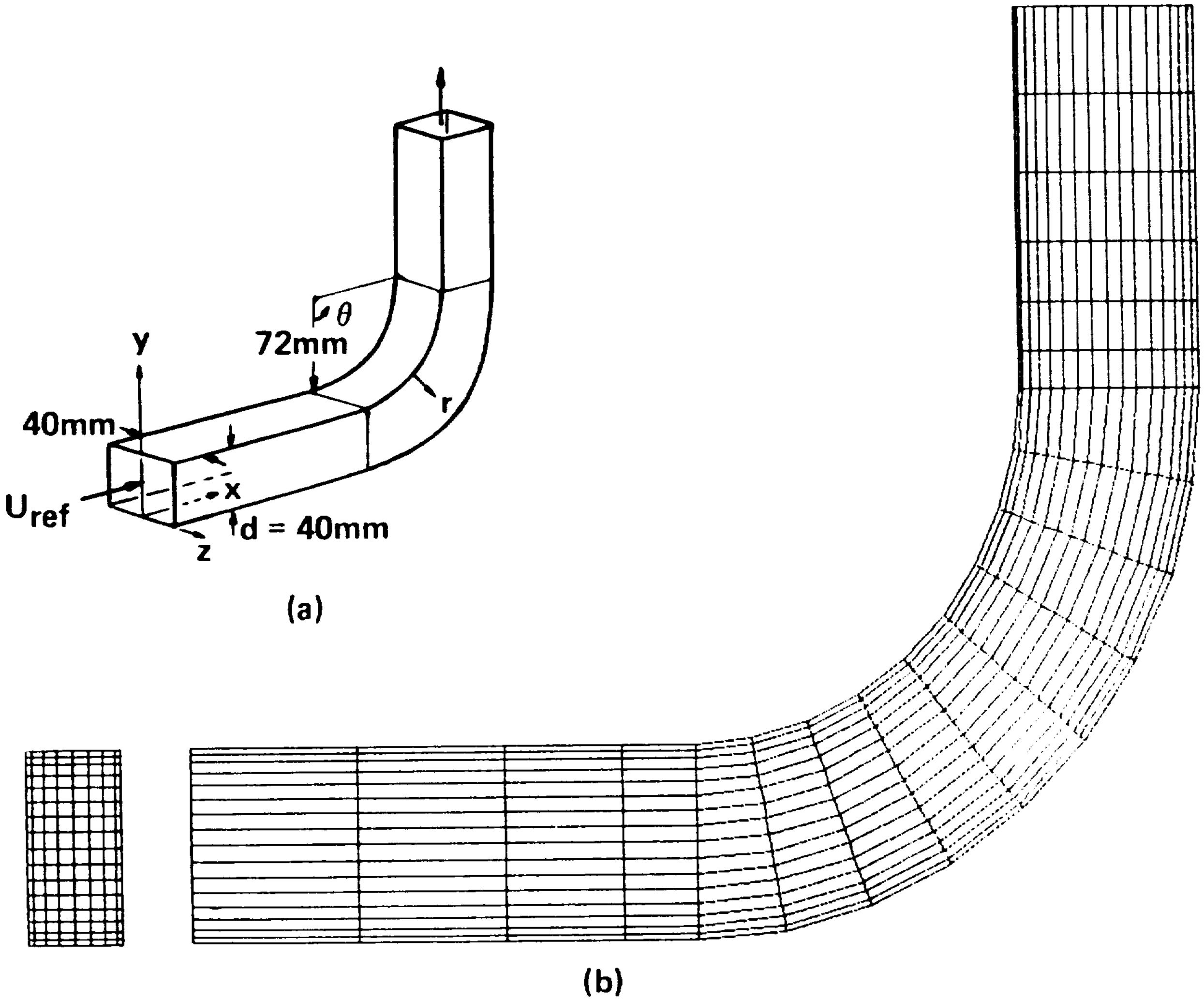


Figure 16. Geometry and mesh system of a 90-deg-bend square duct developing laminar flow problem.

gradient exit (which is 4.5 duct widths downstream of the bend) boundary condition is imposed. The Reynolds number of the flow (based on the duct hydrolic diameter and the inlet bulk velocity) is 790. A 21 x 18 x 10 grid was used for numerical computations. The front view and side view of the mesh system are illustrated in Figure 16(b). Experimental measurements of Humphrey et al. [21] are used for data comparisons.

Velocity vector plots on three sections along the main flow directions (i.e., on x-y plane) are shown in Figure 17. Secondary flow patterns at several stations across the bend are illustrated in Figure 18. These results are very similar to those obtained by Vanka [22] and Rhie [23]. Grid sizes of 50 x 22 x 15 and 58 x 15 x 11 were used by Rhie and Vanka, respectively. The present investigation, using only less than half of their grid numbers, gives highly encouraging results. Detailed comparisons between the measured and the predicted main velocity profiles are given in Figure 19.

With the above successful numerical simulations, it is believed that the present numerical method can be applied to general fluid dynamics problems with good numerical accuracy and efficiency.

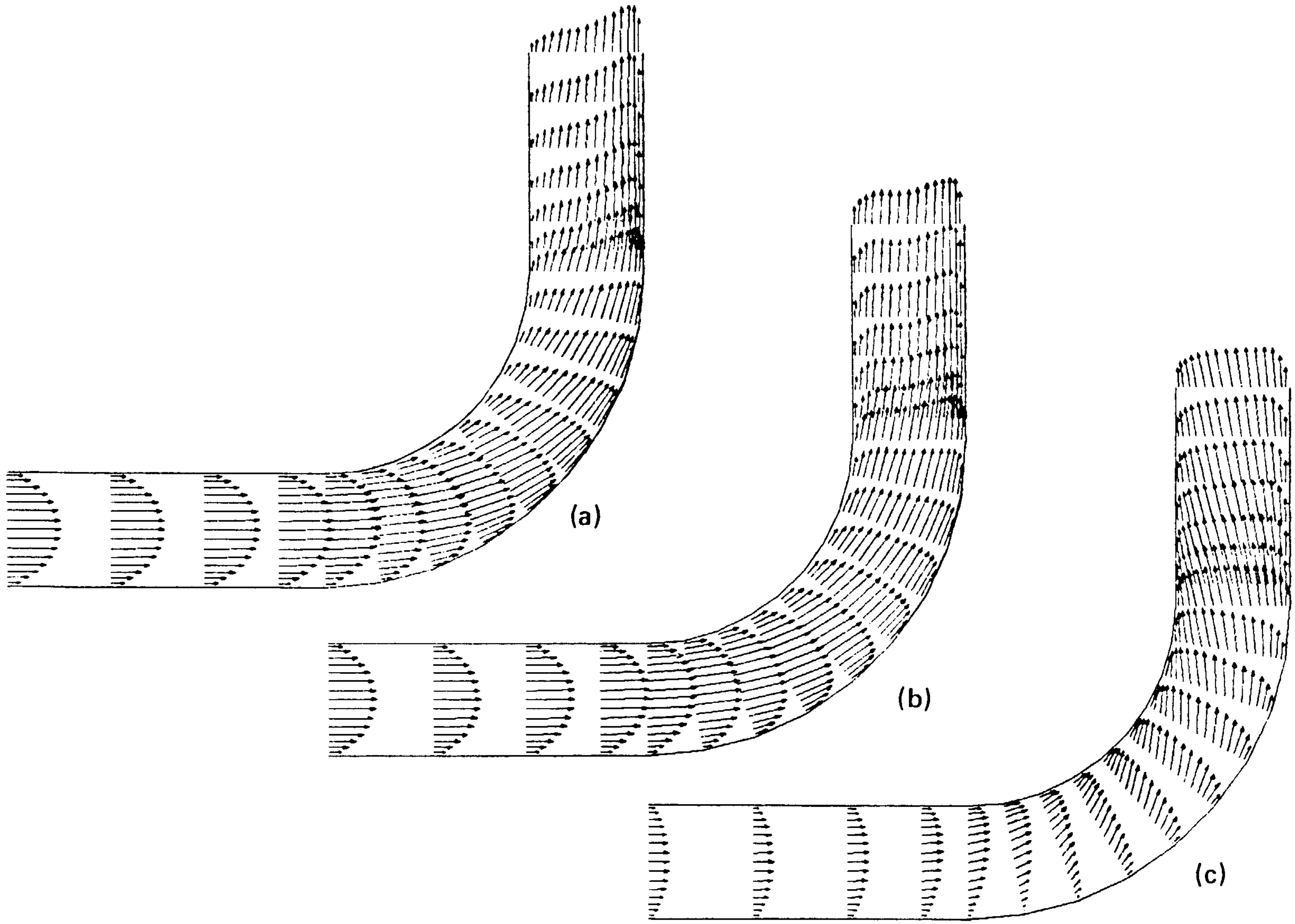


Figure 17. Primary velocity patterns of laminar flow inside a 90-deg-bend square duct. (a) $z/d = 0.0$. (b) $z/d = 0.25$. (c) $z/d = 0.48$.

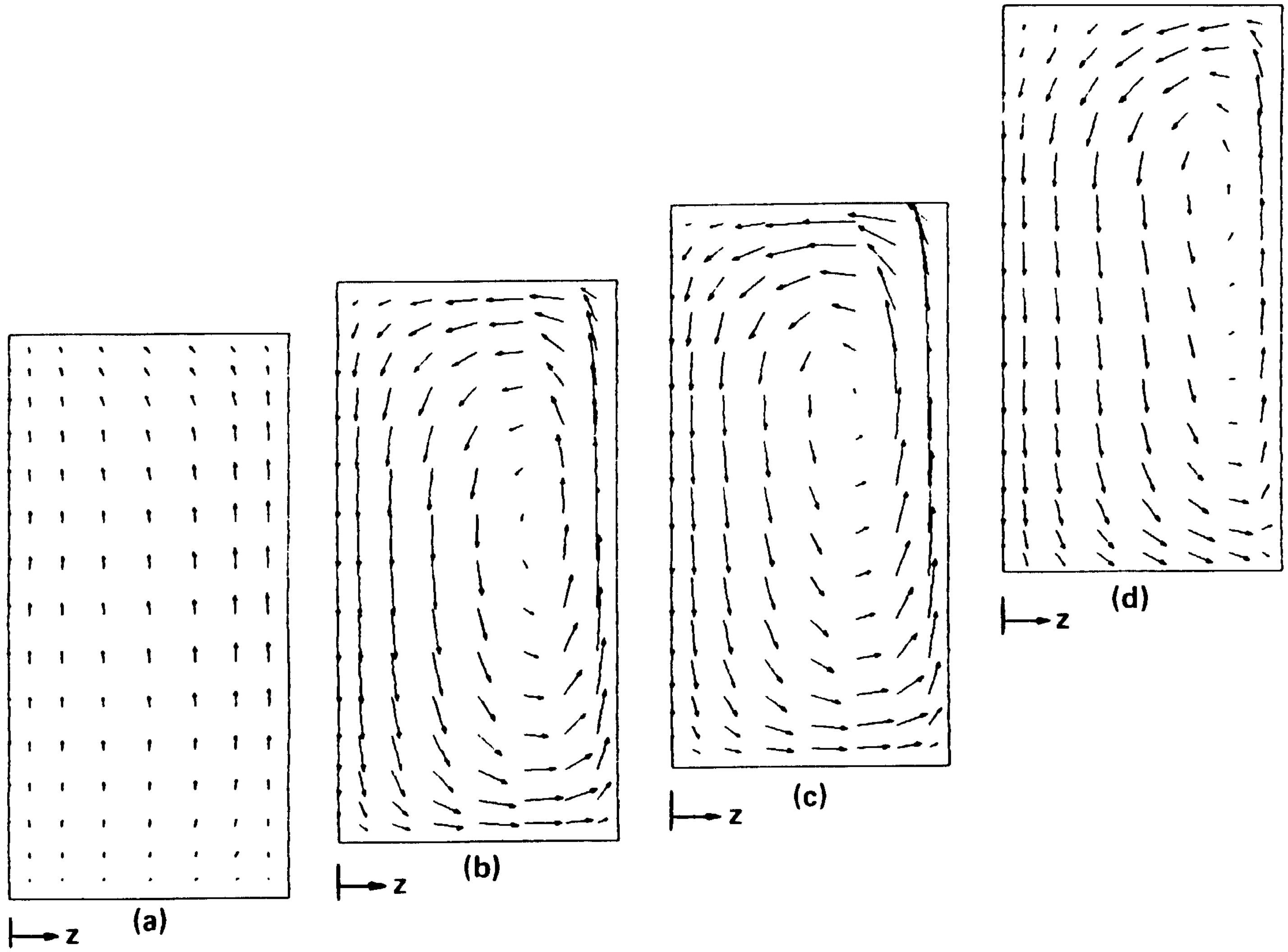


Figure 18. Secondary velocity patterns of laminar flow inside a 90-deg bend.
(a) θ = 0 deg. (b) θ = 30 deg. (c) θ = 60 deg. (d) θ = 90 deg.

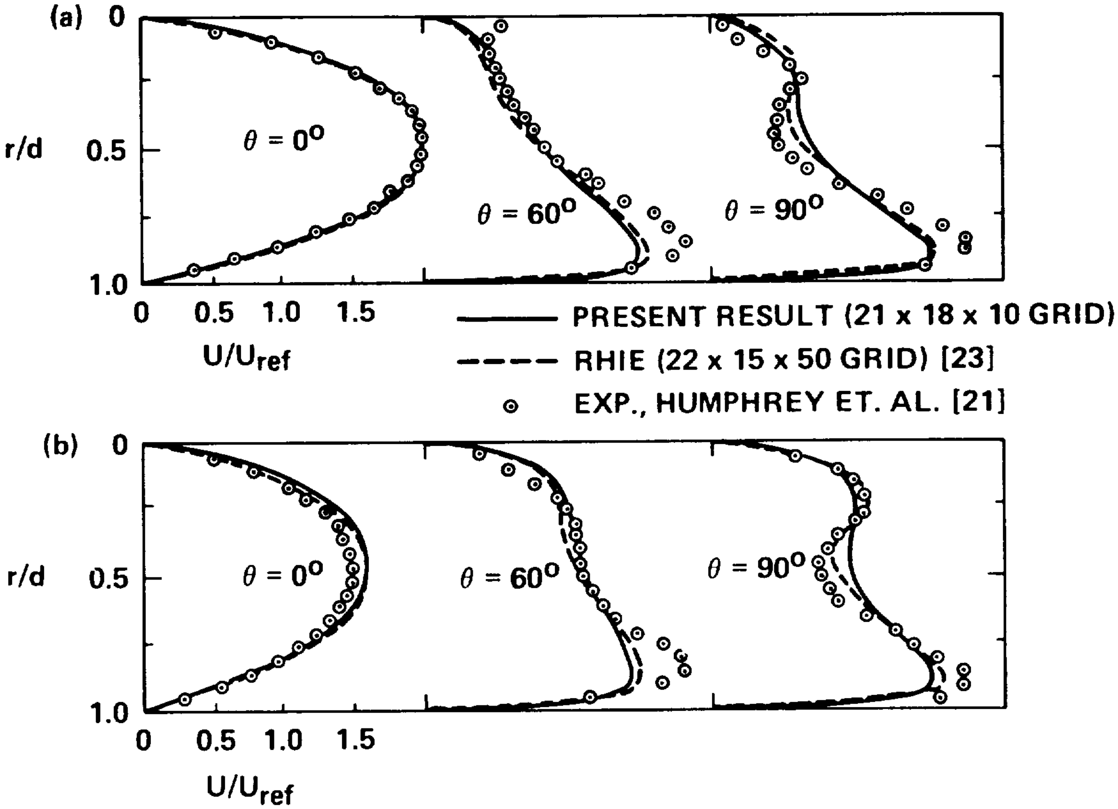


Figure 19. Primary velocity profiles for a 3-D 90-deg-bend square duct. (a) z/d = 0.0. (b) z/d = 0.25.

CONCLUSIONS

A numerical method for solving the steady or transient incompressible Navier-Stokes equations in three-dimensional body-fitted coordinate systems has been developed. In the present paper, the basic numerical algorithms and grid arrangements have been described in detail. A brief user's guide to the present computer code (CNS3D) has been included in Appendix A. A program listing has also been attached in Appendix C.

Several numerical testing examples of 2-D and 3-D, laminar and turbulent flow problems included in the present work have demonstrated that the present computer code is efficient and robust, and can be used as a reliable tool for engineering design and analysis applications. Applications of the present code to the internal turbulent flow problems of the SSME will be presented in the future publications.

REFERENCES

1. Chorin, A. J.: A Numerical Method for Solving Incompressible Viscous Flow Problems. Journal of Comp. Physics, Vol. 2, 1967, pp. 12-26.

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

3. Kwak, D., Chang, J. L. C., Shanks, S. P., and Chakravarthy, S.: An Incompressible Navier-Stokes Flow Solver in Three-Dimensional Curvilinear Coordinate Systems Using Primitive Variables. AIAA Paper 80-0253, Reno, Nev., Jan. 1984.

4. Vanka, S. P., Chen, B. C.-J., and Sha, W. T.: A Semi-Implicit Calculation Procedure for Flows Described in Boundary-Fitted Coordinates Systems. Numerical Heat Transfer, Vol. 3, 1980, pp. 1-19.

5. Shyy, W., Tong, S. S., and Correa, S. M.: Numerical Recirculating Flow Calculation Using a Body-Fitted Coordinate System. Numerical Heat Transfer, Vol. 8, 1985, pp. 99-113.

6. Maliska, C. R., and Raithby, G. D.: A Method for Computing Three Dimensional Flows Using Non-Orthogonal Boundary-Fitted Coordinates. International J. for Numerical Methods in Fluids, Vol. 4, 1984, pp. 519-537.

7. Launder, B. E., and Spalding, D. B.: Mathematical Models of Turbulence. Academic Press, London, 1972.

8. Launder, B. E., and Spalding, D. B.: The Numerical Computation of Turbulent Flows. Comp. Meth. Appl. Mech. Engr., Vol. 3, 1974, pp. 269-289.

9. Patankar, S. V.: Numerical Heat Transfer and Fluid Flow. McGraw-Hill Book Company, 1980.

10. Syed, Saadat A., Chiappetta, L. M., and Gosman, A. D.: Error Reduction Program - Final Report. NASA CR-174776, January 11, 1985.

11. Van Doormal, J. P., and Raithby, G. D.: Enhancements of the SIMPLE Method for Predicting Incompressible Fluid Flows. Numerical Heat Transfer, Vol. 7, 1984, pp. 147-163.

12. Latimer, B. R. and Pollard, A.: Comparison of Pressure-Velocity Coupling Solution Algorithms. Numerical Heat Transfer, Vol. 8, 1985, pp. 635-652.

13. Anderson, D. A., Tannehill, J. C., and Pletcher, R. H.: Computational Fluid Mechanics and Heat Transfer. McGraw-Hill Book Company, 1984.

14. Stone, H. L.: Iterative Solution of Implicit Approximations of Multidimensional Partial Differential Equations. SIAM J. Numer. Anal., Vol. 5, pp. 530-558.

15. Burggraf, O. R.: Analytical and Numerical Studies of the Structure of Steady Separated Flows. J. Fluid Mech., Vol. 24, Part 1, 1966, pp. 113-151.

16. Armaly, B. F., Durst, F., Pereira, J. C. F., and Schönung, B.: Experimental and Theoretical Investigation of Backward-Facing Step Flow. J. Fluid Mech., Vol. 127, 1983, pp. 473-496.

17. Rhie, C. M.: A Pressure Based Navier-Stokes Solver Using the Multigrid Method. AIAA Paper 86-0207, Reno, Nev., January 1986.

18. Rogers, S. E., Kwak, D., and Kaul, U.: On the Accuracy of the Pseudocompressibility Method in Solving the Incompressible Navier-Stokes Equations. AIAA Paper 85-1689, AIAA 18th Fluid Dynamics and Plasmadynamics and Laser Conference, July 16-18, 1985, Cincinnati, Ohio.

19. Kim. J. and Moin, P.: Application of a Fractional-Step Method to Incompressible Navier-Stokes Equations. J. of Comp. Phy., Vol. 59, 1985, pp. 308-323.

20. Kline, S. J., Cantwell, B. J., and Lilley, G. M. (Ed.): The 1980-1981 AFOSR-HTTM-Stanford Conference on Complex Turbulent Flows. Stanford University, I, II and III.

21. Humphrey, J. A. C., Taylor, A. M. K., and Whitelaw, J. H.: Laminar Flow in a Square Duct of Strong Curvature. J. Fluid Mech., Vol. 83, Part 3, 1977, p. 509.

22. Vanka, S. P.: Block-Implicit Calculations of Three-Dimensional Laminar Flow in Strongly Curved Ducts. AIAA J., Vol. 23, No. 12, December 1985, p. 1989.

23. Rhie, C. M.: A Three-Dimensional Passage Flow Analysis Method Aimed at Centrifugal Impellers. Computers and Fluids, Vol. 13, No. 4, 1985, pp. 443-460.

24. Chieng, C. C. and Launder, B. E.: On the Calculation of Turbulent Heat Transport Downstream from an Abrupt Pipe Expansion. Numerical Heat Transfer, Vol. 3, 1980, pp. 189-207.

25. Amano, R. S.: Development of a Turbulence Near-Wall Model and Its Application to Separated and Reattached Flows. Numerical Heat Transfer, Vol. 7, 1984, pp. 59-75.

26. Chen, Y. S.: Applications of a New Wall Function to Turbulent Flow Computations. AIAA Paper 86-0438, Reno, Nev., January 1986.

APPENDIX A

COMPUTER CODE STRUCTURES AND USER'S GUIDE

The global structure of the present computer code (CNS3D) can be represented by a flow chart, shown in Figure A-1. The user is referred to Appendix C for detailed information. First, the program requests inputs, from logic unit 5 (LU = 5), of program control parameters that specify the maximum number of iterations, the type of flow (i.e., laminar or turbulent), number of iterations for solving the pressure correction equation (typically 10), and underrelaxation factors for solving the transport equations, etc. This is followed by the definitions of all the program constants including turbulence model constants (these constants are subject to change according to the user's specific flow problem). Next, the program asks for inputs of the initial flow field guess from a restart file (LU = 8) which contains the grid system coordinates and flow field data that may be created by the user (including grid generation) or obtained from the previous solutions. Format of this data file is also subject to change according to the user's preference. Next, wall boundary control parameters, boundary grid normal distance to the wall, and wall boundary direction cosine are calculated in subroutine DIRCOS. Subroutine TRANF is then invoked to obtain the grid transformation coefficients. Before the solution procedure starts, the inlet mass flow rate is calculated which will be used to control the outlet mass flow rate to enhance mass conservation. The solution procedures consist of a series of subroutine calls to SOLVEQ starting from the solutions of the velocity vectors, u, v, and w, and then the solutions of scalar quantities (including the energy equation and the turbulence model equations) and finally the solution of the pressure correction equation to update the velocity and pressure field such that a divergence-free flow field can be retained.

After each global iteration of the solution procedures, the numerical of iterations and the maximum flow field corrections are checked with the initial settings. If the convergence criterion is satisfied or the number of iterations reaches the prescribed value then the solution procedures stop and the flow field solutions will be written on the pre-assigned disc file (LU = 7).

For instance, if a steady-state laminar flow problem (Reynolds number of 600) is of interest and a converged solution is expected within 300 iterations and the number of iterations for solving the pressure correction equation is 10 and the underrelaxation factors are 0.5 and 0.95 for transport equations and pressure correction equation, respectively, the first inputs from LU = 5 would be:

Line								
1.	300	1	10	1				
2.	0.5	0.5	0.5	0.95	0.5	0.5	0.5	0.5
3.	600.	0.0						

In the second input sequence (i.e., from restart file), the program reads in L x M x N lines of data records. See Figure A-2 for grid structures. Notice that the program requires variable dimensions of (L+1, M+1, N+1) for solving the pressure correction equation. It is important to check the COMMON table for proper variable dimensions.

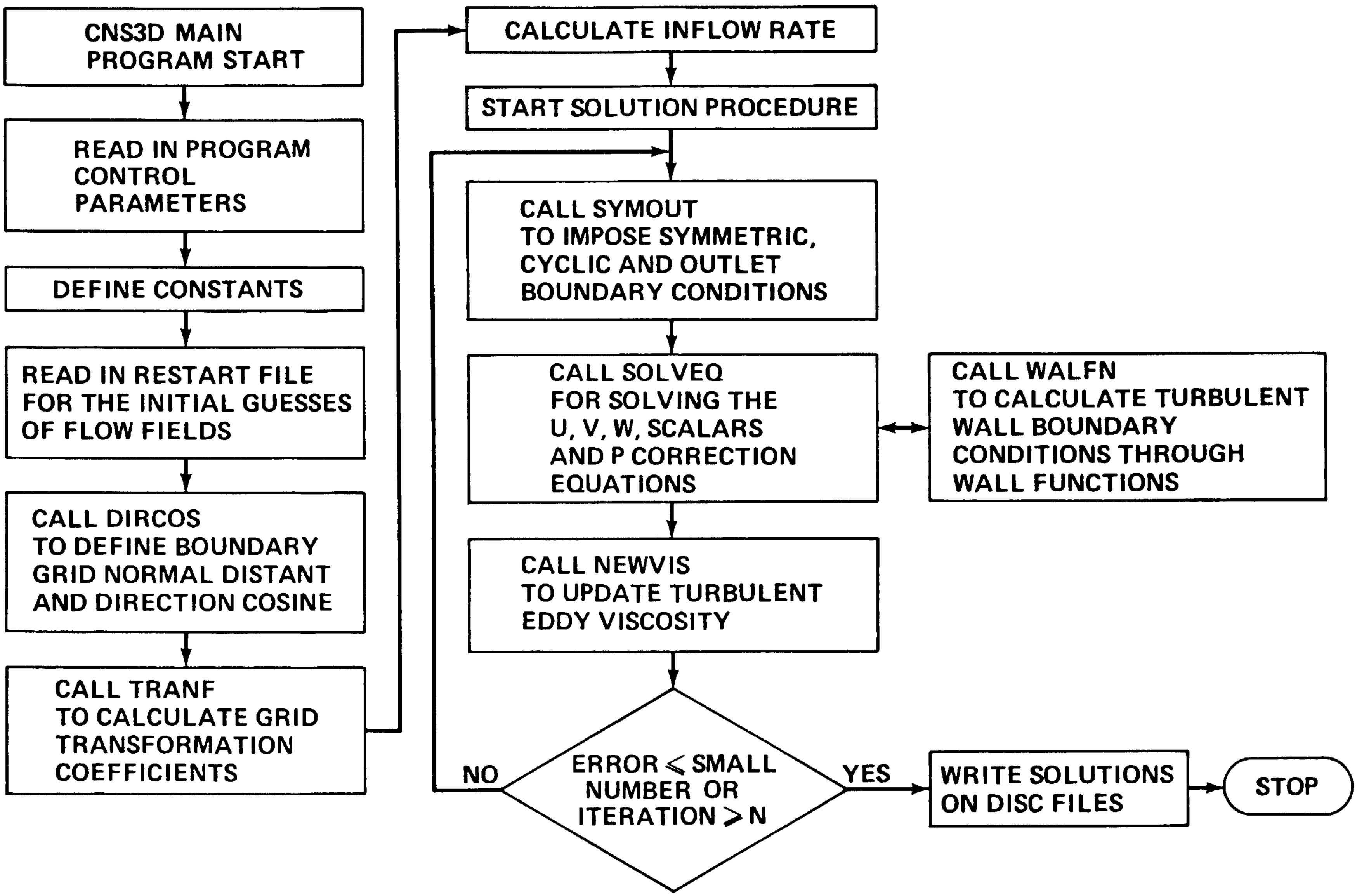


Figure A-1. Global structure of the present computer program CNS3D.

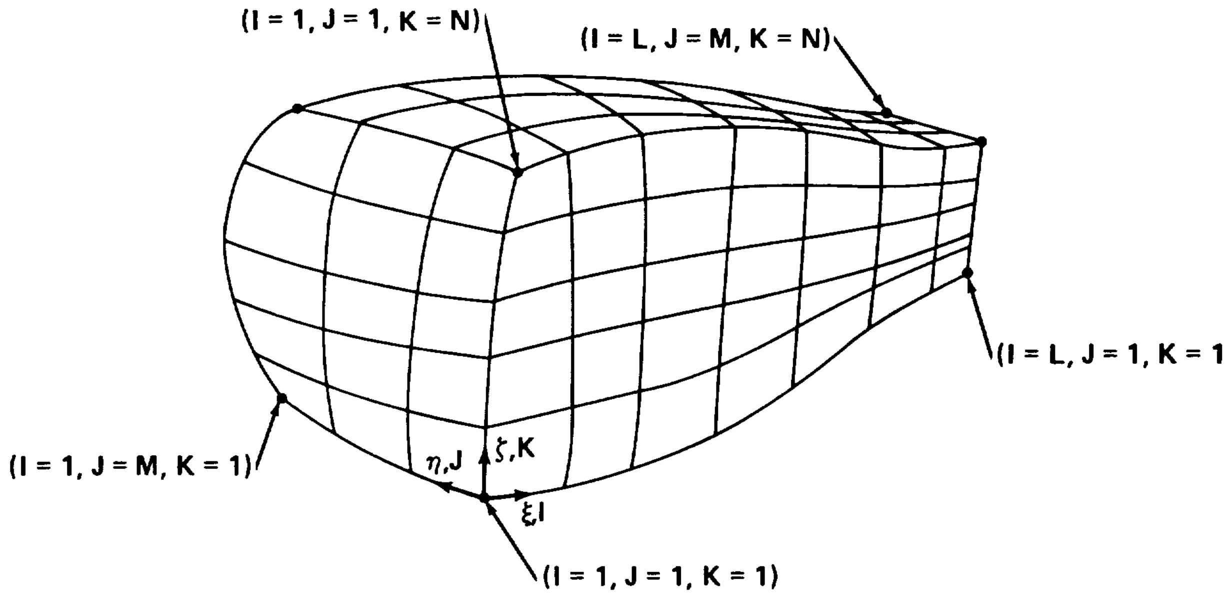


Figure A-2. Grid mesh structures for 3-D calculations.

If the flow problem involves symmetric or cyclic boundary conditions, then the user can look into the subroutine SYMOUT to specify the appropriate boundary conditions (the conditions shown in the program listing of Appendix C are for symmetric boundary conditions at K = 1). For cyclic boundary conditions at K = 1 and K = N, data at K = 2 and K = N-1 can be used to obtain boundary conditions at K = 1 and K = N by requiring same gradients across K = 1 and K = N. This method is simple but will lag the boundary conditions by one iteration. A direct method without lagging the boundary conditions can also be employed by modifying the subroutine of linear algebra solver LINERX such that the boundary conditions can be part of the solution of the TDMA (tridiagonal matrix) solver.

In case of incorporating different wall functions for turbulent flow problems (e.g., References 24, 25, and 26), subroutine BOUNC and WALFN can be modified according to the user's method of wall treatments. The set of wall functions given in the program listing of Appendix C are derived from the conventional wall law and the equilibrium turbulent kinetic energy relations [8].

When additional source terms are to be added to the transport equations due to flow problem requirements, modifications to the source term calculation section in the subroutine SOLVEQ can be carried out. Notice that in the subroutine SOLVEQ source terms for the velocities v and w are included in the u-source section. Purpose of this is to save some computing time since these source terms use similar calculation routines.

Some times it is required to solve more transport equations other than the basic ones included in Appendix C. To modify the program to incorporate more equations, several changes are necessary. First, new variables must be added to the COMMON table (this can be easily done through the computer editor session). Then, new source term sections are added in the SOLVEQ subroutine. Finally, subroutines WALFN and SYMOUT are modified to incorporate the new variables into the boundary condition setting routines.

APPENDIX B

LIST OF FORTRAN SYMBOLS

A(K)	=	Matrix elements of a tridiagonal matrix
AB(I,J,K)	=	Link coefficients through the bottom face of a control volume
AE(I,J,K)	=	Link coefficients through the east face of a control volume
ALC	=	Underrelaxation factor for symmetry or cyclic boundary conditions
ALE	=	Underrelaxation factor for the ε-equation
ALK	=	Underrelaxation factor for the k-equation
ALP	=	Underrelaxation factor for the pressure correction equation
ALU	=	Underrelaxation factor for the u-equation
ALV	=	Underrelaxation factor for the v-equation
ALVIS	=	Underrelaxation factor for the effective viscosity
ALW	=	Underrelaxation factor for the w-equation
AN(I,J,K)	=	Link coefficient through the north face of a control volume
ANAB	=	Sum of the link coefficients at all faces
ANV1(I)	=	Modified wall boundary link coefficient for v-equation
ANW1(I)	=	Modified wall boundary link coefficient for w-equation
AP(I,J,K)	=	Sum of the link coefficients around a control volume
APO(I,J,K)	=	Link coefficients in time marching direction
ARDEN	=	Area times density across a section in physical domain
AREA	=	Area of a section in physical domain
AS(I,J,K)	=	Link coefficients through the south face of a control volume
AT(I,J,K)	=	Link coefficients through the top face of a control volume
AW(I,J,K)	=	Link coefficients through the west face of a control volume
B(K)	=	Matrix elements of a tridiagonal matrix
BB(I,J,K)	=	Coefficients in Stone's partial factorization technique
BOUNC	=	Subroutine for getting turbulent wall boundary conditions through wall functions

C(K)	= Matrix elements of a tridiagonal matrix
C1	= Turbulence model constant, = 1.44
C2	= Turbulence model constant, = 1.92
CB	= Convective flux through the bottom face of a control volume
CE	= Convective flux through the east face of a control volume
CK	= Von Karman constant, = 0.4
CMU	= Turbulence model constant, = 0.09
CMU1	= CMU**0.25
CMU2	= CMU**0.75
CN	= Convective flux through the north face of a control volume
CS	= Convective flux through the south face of a control volume
CT	= Convective flux through the top face of a control volume
CW	= Convective flux through the west face of a control volume
CX(I,J,K)	= Grid transformation coefficient, ξ_x
CY(I,J,K)	= Grid transformation coefficient, ξ_y
CZ(I,J,K)	= Grid transformation coefficient, ξ_z
D(K)	= Matrix elements of a tridiagonal matrix
DDB	= Diffusive flux through the bottom face of a control volume
DDE	= Diffusive flux through the east face of a control volume
DDN	= Diffusive flux through the north face of a control volume
DDS	= Diffusive flux through the south face of a control volume
DDT	= Diffusive flux through the top face of a control volume
DDW	= Diffusive flux through the west face of a control volume
DE(I,J,K)	= Turbulent kinetic energy dissipation rate, ε
DEO(I,J,K)	= DE at the previous time level
DEN(I,J,K)	= Density of the fluid
DENO(I,J,K)	= DEN at the previous time level
DENC	= Density at the center of a surface

DENIN = Initial value of density of the fluid

DIRCOS = Subroutine for calculating the boundary grid sizes and direction cosines

DITM = Wall boundary average value of dissipation rate

DK(I,J,K) = Turbulent kinetic energy, k

DKO(I,J,K) = DK at the previous time level

DTT = Time step size, Δt

DU(I,J,K) = Diffusive coefficient for the p'-equation

DV(I,J,K) = Diffusive coefficient for the p'-equation

DW(I,J,K) = Diffusive coefficient for the p'-equation

E = Wall law constant, = 9.01069

EREXT = Convergence criterion tolerance

ERRE = Maximum correction in ε

ERRF = Maximum correction of a variable

ERRK = Maximum correction in k

ERRM = Maximum correction in p

ERRU = Maximum correction in u

ERRV = Maximum correction in v

ERRW = Maximum correction in w

EX(I,J,K) = Grid transformation coefficient, η_x

EY(I,J,K) = Grid transformation coefficient, η_y

EZ(I,J,K) = Grid transformation coefficient, η_z

F(I,J,K) = Tentative variable of the transport equations

FO(I,J,K) = F at the previous time level

F1(I,J,K) = Variable quantity at the previous iteration step

FLOW = Outlet mass flow rate

FLOWIN = Inlet mass flow rate

GEN(I,J,K) = Turbulent kinetic energy production rate

HINUM	= Large number, = 1.E30
I	= Index along the ξ grid lines
IBC(I)	= Boundary grid index
IE	= Index assigned for the transport equations
IG	= Problem control parameter, =1 for laminar flow and =2 for turbulent flow
IITO	= Total number of wall boundary grids
IITY	= Boundary grid face type
IJLO(I,J,K)	= Boundary grid sequential order
INIT	= Subroutine for initializing variables
INPRO	= Logical parameter for updating the effective viscosity
INSOE	= Logical parameter for solving the ε-equation
INSOK	= Logical parameter for solving the k-equation
INSOP	= Logical parameter for solving the p'-equation
INSOT	= Logical parameter for solving the T-equation
INSOU	= Logical parameter for solving the u-equation
INSOV	= Logical parameter for solving the v-equation
INSOW	= Logical parameter for solving the w-equation
IS	= Starting value of I of the solution domain
ISWE	= Number of sweeps for solving the ε-equation
ISWK	= Number of sweeps for solving the k-equation
ISWP	= Number of sweeps for solving the p'-equation
ISWU	= Number of sweeps for solving the u-equation
ISWV	= Number of sweeps for solving the v-equation
ISWW	= Number of sweeps for solving the w-equation
IT	= Last value of I of the solution domain
ITT	= Number of time steps
J	= Index along the η grid lines

JBC(I)	=	Boundary grid index
JS	=	Starting value of J of the solution domain
JT	=	Last value of J of the solution domain
K	=	Index along the ζ grid lines
KBC(I)	=	Boundary grid index
KS	=	Starting value of K of the solution domain
KT	=	Last value of K of the solution domain
L	=	Maximum dimension of grid system in I direction
LO	=	L + 1
L1	=	Starting point of blockage region in I direction
L2	=	Last point of blockage region in I direction
LINERX	=	Subroutine for solving algebraic equations
LT	=	L - 1
M	=	Maximum dimension of grid system in J direction
MO	=	M + 1
M1	=	Starting point of blockage region in J direction
M2	=	Last point of blockage region in J direction
MC(I,J,K)	=	Wall blockage region control parameter
MT	=	M - 1
N	=	Maximum dimension of grid system in K direction
NO	=	N + 1
N1	=	Starting point of blockage region in K direction
N2	=	Last point of blockage region in K direction
NEWVIS	=	Subroutine for updating the effective viscosity
NLIMT	=	Limit of maximum number of iterations
NT	=	N - 1
P	=	Static pressure (relative)
PCXI	=	Pressure gradient, P_ξ

PDUV = Blockage control parameter for link coefficients

PEDA = Pressure gradient, p_{η}

PP = Pressure correction, p'

PPBLK = Global pressure correction

PSCI = Pressure gradient, P_{ζ}

PTA = Wall boundary source term for the momentum equations

PW = Wall value control parameter

RENL = Reynolds number of the fluid

SIGE = Turbulence model constant, = 1.3

SIGK = Turbulence model constant, = 1.0

SINX(I) = Wall boundary direction cosine

SINY(I) = Wall boundary direction cosine

SINZ(I) = Wall boundary direction cosine

SMNUM = Small number, 1.E-30

SOC1 = Source term due to shear stress

SOC2 = Source term due to shear stress

SOC3 = Source term due to shear stress

SOLVEQ = Subroutine for solving general transport equation

SP(I,J,K) = Linear part of the source term

SPK(I,J,K) = Secondary linear part of the source term

SU(I,J,K) = Constant part of the source term

SUK(I,J,K) = Secondary constant part of the source term

SX(I,J,K) = Grid transformation coefficient, ζ_x

SY(I,J,K) = Grid transformation coefficient, ζ_y

SYMOUT = Subroutine for setting flow boundary conditions

SZ(I,J,K) = Grid transformation coefficient, ζ_z

TAUN(I) = Wall shear stress

TIMT	= Total time
TJO(I,J,K)	= Jacobian of metric transformation
TM(I,J,K)	= Temperature
TMO(I,J,K)	= TM at the previous time level
TMULT	= Wall shear stress
TRANF	= Subroutine for calculating the grid transformation coefficients
TXXE(I,J,K)	= Metric coefficient for east face diffusive flux
TXXW(I,J,K)	= Metric coefficient for west face diffusive flux
TXYN(I,J,K)	= Metric coefficient for north face diffusive flux
TXYS(I,J,K)	= Metric coefficient for south face diffusive flux
TXZT(I,J,K)	= Metric coefficient for top face diffusive flux
TXZB(I,J,K)	= Metric coefficient for bottom face diffusive flux
TYYN(I,J,K)	= Metric coefficient for north face diffusive flux
TYYS(I,J,K)	= Metric coefficient for south face diffusive flux
TYXE(I,J,K)	= Metric coefficient for east face diffusive flux
TYXW(I,J,K)	= Metric coefficient for west face diffusive flux
TYZT(I,J,K)	= Metric coefficient for top face diffusive flux
TYZB(I,J,K)	= Metric coefficient for bottom face diffusive flux
TZZT(I,J,K)	= Metric coefficient for top face diffusive flux
TZZB(I,J,K)	= Metric coefficient for bottom face diffusive flux
TZXE(I,J,K)	= Metric coefficient for east face diffusive flux
TZXW(I,J,K)	= Metric coefficient for west face diffusive flux
TZYN(I,J,K)	= Metric coefficient for north face diffusive flux
TZYS(I,J,K)	= Metric coefficient for south face diffusive flux
U(I,J,K)	= U-velocity
UO(I,J,K)	= U at the previous time level
UC	= Velocity at the center of a surface
UCXI	= U-velocity gradient, u_ξ

UEDA	= U-velocity gradient, u_η
UINC	= Velocity correction at outlet plane
USCI	= U-velocity gradient, u_ζ
UX	= U-velocity gradient, u_x
UY	= U-velocity gradient, u_y
UZ	= U-velocity gradient, u_z
V(I,J,K)	= V-velocity
VO(I,J,K)	= V at the previous time level
VISC	= Molecular viscosity, μ
VISE(I,J,K)	= Effective viscosity, μ_{eff}
VCXI	= V-velocity gradient, v_ξ
VEDA	= V-velocity gradient, v_η
VSCI	= V-velocity gradient, v_ζ
VX	= V-velocity gradient, v_x
VY	= V-velocity gradient, v_y
VZ	= V-velocity gradient, v_z
W(I,J,K)	= W-velocity
WO(I,J,K)	= W at the previous time level
WALLFN	= Subroutine for calculating the wall functions
WALVAL	= Subroutine for assigning wall values
WCXI	= W-velocity gradient, w_ξ
WEDA	= W-velocity gradient, w_η
WSCI	= W-velocity gradient, w_ζ
WX	= W-velocity gradient, w_x
WY	= W-velocity gradient, w_y
WZ	= W-velocity gradient, w_z
X(I,J,K)	= X-coordinate

Y(I,J,K)	= Y-coordinate
YN(I)	= Wall normal distance from the last grid
YN1(I)	= Wall grid volume size
YPLN(I)	= Nondimensionalized YN, $y^+ = u_\tau y/\nu$
Z(I,J,K)	= Z-coordinate

APPENDIX C

PROGRAM LISTING

```
FORTRAN VIID: LICENSED RESTRICTED RIGHTS AS STATED IN LICENSE  *****************, SEE DOCUMENTATION PACKAGE, 04-101M99.
000000I         PROGRAM  CNS3D
            C       BY: Y. S. CHEN         CN: 8/13/1985
            C
            C*****[ CURVILINEAR N-S CODE FOR 3-D INCOMPRESSIBLE FLOWS ]***********
            C
000006I         COMMON
               1/VAR/U(21,18,10),V(21,18,10),P(21,18,10),DK(21,18,10),
               2  DE(21,18,10),ERRU,ERRV,ERRM,ERRK,ERRE,ERRW,
               3  PF(21,18,10),W(21,18,10),TM(21,18,10)
               1/PRCP/ VISE(21,18,10),DEN(21,18,10),VISC,DENIN,FLOWIN
               1/PCCR/ DU(21,18,10),DV(21,18,10),DW(21,18,10)
               1/TUR/ SIGK,SIGE,CMU,C1,C2,CMU1,CMU2,E,CK,HINUM,SMNUM,ANV1(800),
               2 YN(800),YN1(800),SINX(800),SINY(800),SINZ(800),ANW1(800),
               3 YPLN(800),TAUN(800),IEC(800),JBC(800),KBC(800),IITY(800),
               4 TALW(800),GEN(21,18,10),MC(21,18,10),IJLO(21,18,10),IITO
               1/COEF/ AP(21,18,10),SU(21,18,10),SP(21,18,10),SUK(21,18,10),
               2    SPK(21,18,10),AE(21,18,10),AW(21,18,10),AN(21,18,10),
               3    AS(21,18,10),AT(21,18,10),AB(21,18,10),APC(21,18,10)
000006I         COMMON
               1/TRAN/ X(21,18,10),Y(21,18,10),Z(21,18,10),TJO(21,18,10),
               2    CX(21,18,10),CY(21,18,10),CZ(21,18,10),
               3    EX(21,18,10),EY(21,18,10),EZ(21,18,10),
               3    SX(21,18,10),SY(21,18,10),SZ(21,18,10)
               1/LIMT/ L,M,LT,MT,L1,L2,M1,M2,LO,MO,ISWU,ISWV,ISWP,ISWK,ISWE,
               2    ALU,ALV,ALP,ALK,ALE,ALVIS,ALW,N,N1,N2,NO,ISWW,IG,NT,ALC,DTT
000006I         COMMON
               1/TTRAN/TXXE(21,18,10),TXXW(21,18,10),TYYN(21,18,10),
               2    TYYS(21,18,10),TZZT(21,18,10),TZZB(21,18,10),
               3    TYXE(21,18,10),TYXW(21,18,10),TYZT(21,18,10),
               4    TYZS(21,18,10),TXYN(21,18,10),TXYS(21,18,10),
               5    TXZT(21,18,10),TXZB(21,18,10),TZXE(21,18,10),
               6    TZXW(21,18,10),TZYN(21,18,10),TZYS(21,18,10)
               1/UNSTDY/UO(21,18,10),VO(21,18,10),WO(21,18,10),DKO(21,18,10),
               2    CEO(21,18,10),DENO(21,18,10),TMO(21,18,10)
000006I         LOGICAL INSCU,INSOV,INSCP,INSOK,INSOE,INPRO,INSOW,INSOT
            C*****[ INPUT DATA GUIDE ]******************************************
            C
            C  NLIMT : MAXIMUM NO. OF ITERATIONS LIMIT
            C
            C  IG    = 1 : LAMINAR
            C          2 : TURBULENT (K-E MODEL)
            C
            C  ISWP  : NO. OF SWEEPS FOR SOLVING THE P' EQUATION (PP).
            C
            C  ITT   : TOTAL NO. OF TIME STEPS.
            C
            C  ALU,ALV,ALW,ALP,ALK,ALE,ALVIS,ALC : UNDER-RELAXATION FACTORS
            C
            C  RENL  : REFERENCE REYNOLDS NUMBER.
            C
            C  DTT   : TIME STEP FOR UNSTEADY PROBLEMS.
            C
            C*******************************************************************
            C-----INPUT DATA (PROBLEM CONTROL SETTING)
000006I         READ(5,100) NLIMT,IG,ISWP,ITT
000030I         READ(5,200) ALU,ALV,ALW,ALP,ALK,ALE,ALVIS,ALC
00006CI         READ(5,200) RENL,DTT
```

```
          C-----CONSTANTS
000090I         EREXT=1.E-3                                                   4
00009CI         ISWL=7                                                        5
0000A4I         ISWV=7                                                        6
0000ACI         ISWW=7                                                        7
0000B4I         ISWK=5                                                        8
0000BCI         ISWE=5                                                        9
0000C4I         DENIN=1.0                                                    10
0000D0I         VISC=1./RENL                                                 11
0000E2I         SIGL=1.C                                                     12
0000EEI         SIGK=1.C                                                     13
0000FAI         SIGE=1.3                                                     14
000106I         CMU=0.09                                                     15
000112I         C1=1.43                                                      16
00011EI         C2=1.92                                                      17
00012AI         E=9.01C69                                                    18
000136I         CK=C.4                                                       19
000142I         PI=3.141592654                                               20
00014EI         HINUM=1.E30                                                  21
00015AI         SMNUM=1.E-30                                                 22
000166I         CMU1=CMU**0.25                                               23
00017EI         CMU2=CMU**0.75                                               24
000196I         INSCU=.TRUE.                                                 25
00019EI         INSCV=.TRUE.                                                 26
0001A6I         INSCP=.TRUE.                                                 27
0001AEI         INSCK=.TRUE.                                                 28
0001B6I         INSCE=.TRUE.                                                 29
0001BEI         INPRO=.TRUE.                                                 30
0001C6I         INSCW=.TRUE.                                                 31
0001CEI         INSCT=.FALSE.                                                32
0001D6I         IF(IG .EQ. 2) GC TO 10                                       33
0001ECI         INSCK=.FALSE.                                                35
0001F4I         INSCE=.FALSE.                                                36
0001FCI         INPRO=.FALSE.                                                37
000204I     10  CONTINUE                                                     38
          C*****[READ IN INITIAL FLOW FIELDS FROM RESTART FILE (LU = 8)]********
000204I         REAC(8,100) L,M,N,L1,L2,M1,M2,N1,N2                          39
000244I         LO=L+1                                                       40
000252I         MO=M+1                                                       41
000260I         NO=N+1                                                       42
00026EI         LT=L-1                                                       43
00027CI         MT=M-1                                                       44
00028AI         NT=N-1                                                       45
          C-----INITIALIZE VARIABLES
000298I         CALL INIT                                                    46
          C-----RESTART FILE
0002A0I         DO 50 K=1,N                                                  47
0002B4I         DO 50 I=1,L                                                  48
0002C8I         REAC(8,400)                                                  49
0002E4I         DO 50 J=1,M                                                  50
0002F8I         REAC(8,500) X(I,J,K),Y(I,J,K),Z(I,J,K),U(I,J,K),V(I,J,K),    51
               1 W(I,J,K),P(I,J,K),TM(I,J,K),DK(I,J,K),DE(I,J,K),
               2 VISETM,DENTM
000538I         UO(I,J,K)=U(I,J,K)                                           52
000604I         VO(I,J,K)=V(I,J,K)                                           53
000650I         WO(I,J,K)=W(I,J,K)                                           54
00069CI         TMO(I,J,K)=TM(I,J,K)                                         55
```

```
0006E8I         DKO(I,J,K)=DK(I,J,K)                                                56
000734I         DEO(I,J,K)=DE(I,J,K)                                                57
000780I         DENC(I,J,K)=DEN(I,J,K)                                              58
0007CCI    50   CONTINUE                                                            59
          C*********************************************************************
          C-----GET BOUNDARY CONTROL PARAMETERS
000814I         CALL DIRCOS                                                         60
          C-----SET BOUNDARY TURBULENCE QUANTITIES TO ZERO
00081CI         DO 121 I=1,L                                                        61
000830I         DO 121 J=1,M                                                        62
000844I         DO 121 K=1,N                                                        63
000858I         IF(MC(I,J,K) .NE. 0) GO TO 122                                      64
00088EI         GO TO 121                                                           66
000894I   122   DK(I,J,K)=0.0                                                       67
0008C0I         DE(I,J,K)=0.0                                                       68
0008ECI         U(I,J,K)=0.0                                                        69
000918I         V(I,J,K)=0.0                                                        70
000944I         W(I,J,K)=0.0                                                        71
000970I   121   CONTINUE                                                            72
          C-----CALCULATE GRID TRANSFORMATION COEFFICIENTS
000988I         CALL TRANF                                                          73
          C-----TURBULENT VISCOSITY
0009C0I         IF(INPRC) CALL NEWVIS                                               74
          C-----CALCULATE INLET MASS FLOW RATE
0009D4I         FLOWIN=0.0                                                          76
0009E0I         I=1                                                                 77
0009E8I         DO 45 J=2,M                                                         78
0009FCI         DO 45 K=2,N                                                         79
000A10I         UC=(U(I,J,K)+U(I,J-1,K)+U(I,J,K-1)+U(I,J-1,K-1))*0.25               80
000AC0I         DENC=(DEN(I,J,K)+DEN(I,J-1,K)+DEN(I,J,K-1)+DEN(I,J-1,K-1))*0.25     81
000B70I         P1=(X(I,J,K)+X(I,J,K-1)-X(I,J-1,K)-X(I,J-1,K-1))*0.5                82
000C20I         P2=(Y(I,J,K)+Y(I,J,K-1)-Y(I,J-1,K)-Y(I,J-1,K-1))*0.5                83
000CD0I         P3=(Z(I,J,K)+Z(I,J,K-1)-Z(I,J-1,K)-Z(I,J-1,K-1))*0.5                84
000D80I         Q1=(X(I,J,K)+X(I,J-1,K)-X(I,J,K-1)-X(I,J-1,K-1))*0.5                85
000E30I         Q2=(Y(I,J,K)+Y(I,J-1,K)-Y(I,J,K-1)-Y(I,J-1,K-1))*0.5                86
000EE0I         Q3=(Z(I,J,K)+Z(I,J-1,K)-Z(I,J,K-1)-Z(I,J-1,K-1))*0.5                87
000F90I         AREA=SQRT(P1*P1+P2*P2+P3*P3)*SQRT(Q1*Q1+Q2*Q2+Q3*Q3)                88
00102CI         FLOWIN=FLOWIN+DENC*AREA*UC                                          89
00104AI    45   CONTINUE                                                            90
00107AI         ITC=1                                                               91
          C-----TRANSIENT PROCESS
001032I     2   CONTINUE                                                            92
001082I         CALL SYMOUT(3,1,2,L,2,M,2,N)                                        93
0010DCI         ITER=1                                                              94
          C-----SOLUTION PROCEDURES START
0010E4I     1   CONTINUE                                                            95
0010E4I         CALL SYMOUT(1,1,2,LT,2,MT,2,NT)                                     96
001140I         IF(INSCU) CALL SOLVEQ(1,ISWU,ALU,SIGU,ERRU,U,UC)                    97
001184I         IF(INSCV) CALL SOLVEQ(2,ISWV,ALV,SIGU,ERRV,V,VO)                    99
0011C8I         IF(INSCW) CALL SOLVEQ(3,ISWW,ALW,SIGU,ERRW,W,WO)                   101
00120CI         IF(INSCT) CALL SOLVEQ(4,ISWW,ALW,SIGU,ERRW,TM,TMO)                 103
001250I         IF(INSCK) CALL SOLVEQ(5,ISWK,ALK,SIGK,ERRK,DK,DKO)                 105
001294I         IF(INSOE) CALL SOLVEQ(6,ISWE,ALE,SIGE,ERRE,DE,DEO)                 107
0012D8I         IF(INSCP) CALL SOLVEQ(0,ISWP,ALP,SIGU,ERRM,PP,PP)                  109
00131CI         IF(INPRC) CALL NEWVIS                                              111
          C-----CONVERGENCE CHECK
001330I         WRITE(6,300) ITER,ERRU,ERRV,ERRW,ERRM,ERRK,ERRE,U(7,2,6)           113
```

```
172  00136CI          ERRMAX=ERRM+ERRU+ERRV+ERRW                                          172      114
173  00138AI          IF(ITER .GE. 20 .AND. ERRMAX .GT. 1.E03) GO TO 99                  173      115
174  001382I          IF(ITER .GE. NLIMT .OR. ERRMAX .LE. EREXT) GO TO 99                174      117
175  0013DCI          ITER=ITER+1                                                         175      119
176  0013EAI          GO TO 1                                                             176      120
177             C-----PRINT OUT SOLUTIONS                                                 177
178  0013EEI     99   CONTINUE                                                            178      121
179  0013EEI          WRITE(7,100) L,M,N,L1,L2,M1,M2,N1,N2                               179      122
180  00142CI          DO 901 K=1,N                                                        180      123
181  001440I          DO 901 I=1,L                                                        181      124
182  001454I          WRITE(7,400)                                                        182      125
183  001470I          DO 902 J=1,M                                                        183      126
184  001484I          XV=X(I,J,K)                                                         184      127
185  00148DI          YV=Y(I,J,K)                                                         185      128
186  0014DCI          ZV=Z(I,J,K)                                                         186      129
187  00150BI          WRITE(7,500) XV,YV,ZV,U(I,J,K),V(I,J,K),W(I,J,K),P(I,J,K),         187      130
188                1         TM(I,J,K),DK(I,J,K),DE(I,J,K),VISE(I,J,K),DEN(I,J,K)         188
189  0016E3I    902   CONTINUE                                                            189      131
190  00170CI    901   CONTINUE                                                            190      132
191  001730I          TIMT=DTT*ITO                                                        191      133
192  001744I          WRITE(7,300) ITO,TIMT                                               192      134
193  00176BI          IF(ITO .GE. ITT .OR. DTT .EQ. 0.0) GO TO 999                       193      135
194  001792I          ITO=ITO+1                                                           194      137
195  0017A0I          GO TO 2                                                             195      138
196  0017A4I    999   CONTINUE                                                            196      139
197  0017A4I    100   FORMAT(9I5)                                                         197
198  0017B0I    200   FORMAT(11F7.4)                                                      198
199  00175EI    300   FORMAT(1X,I5,7E10.2)                                                199
200  0017C0I    400   FORMAT(//)                                                          200
201  0017DAI    500   FORMAT(3F8.4,3E12.4,2X,6E11.4)                                      201
202  0017F6I          STOP                                                                202      140
203  0017FEI          END                                                                 203      141

NO  ERRORS:F7D  R05-01.0C  MAINPROG    CNS3C     02/21/86  09:45:56  TABLE SPACE:  11 KB
STATEMENT BUFFER:   20 LINES/1321 BYTES    STACK SPACE: 181 WORDS
SINGLE PRECISION FLOATING PT SUPPORT REQUIRED FOR EXECUTION
```

```
000000I          SUBROUTINE DIRCCS
000004I          COMMON
                 1/TUR/ SIGK,SIGE,CMU,C1,C2,CMU1,CMU2,E,CK,HINUM,SMNUM,ANV1(800),
                 2 YN(800),YN1(800),SINX(800),SINY(800),SINZ(800),ANW1(800),
                 3 YPLN(800),TAUN(800),IBC(800),JBC(800),KBC(800),IITY(800),
                 4 TAUW(800),GEN(21,18,10),MC(21,18,10),IJLO(21,18,10),IITO
                 1/TRAN/ X(21,18,10),Y(21,18,10),Z(21,18,10),TJO(21,18,10),
                 2   CX(21,18,10),CY(21,18,10),CZ(21,18,10),
                 3   EX(21,18,10),EY(21,18,10),EZ(21,18,10),
                 3   SX(21,18,10),SY(21,18,10),SZ(21,18,10)
                 1/LIMT/ L,M,LT,MT,L1,L2,M1,M2,LC,MO,ISWU,ISWV,ISWP,ISWK,ISWE,
                 2   ALU,ALV,ALP,ALK,ALE,ALVIS,ALW,N,N1,N2,NO,ISWW,IG,NT,ALC,DTT
         C-----SET DOMAIN BLOCKAGE CONTROL PARAMETER
         C-----SCALAR BLOCKAGE :   MC(I,J,K)=1
         C-----PRESSURE BLOCKAGE : MC(I,J,K)=2
000004I          DO 10 I=1,LO                                                   1
000018I          DO 10 J=1,MO                                                   2
00002CI          DO 10 K=1,NC                                                   3
000040I          MC(I,J,K)=0                                                    4
000068I          IF(J.EQ.1.OR.J.EQ.M.OR.K.EQ.N) MC(I,J,K)=1                     5
0000C4I          IF(I.GE.L1.AND.I.LE.L2.AND.J.GE.M1.AND.J.LE.M2.AND.            7
                1   K.GE.N1.AND.K.LE.N2) MC(I,J,K)=1
000158I          IF(I.GT.L1.AND.I.LE.L2.AND.J.GT.M1.AND.J.LE.M2.AND.            9
                1   K.GT.N1.AND.K.LE.N2) MC(I,J,K)=2
         C-----ADD BLOCKAGES AS NEEDED HERE
0001ECI     10   CONTINUE                                                       11
         C-----CALCULATE BOUNDARY GRID SIZES AND ORIENTATIONS
000234I          III=1                                                          12
00023CI          DO 30 I=2,LT                                                   13
000250I          DO 30 J=2,MT                                                   14
000264I          DO 30 K=2,NT                                                   15
000278I          IF(MC(I,J,K) .NE. 0) GO TO 30                                  16
0002AEI          MCT=MC(I+1,J,K)+MC(I-1,J,K)+MC(I,J+1,K)+MC(I,J-1,K)+           18
                1    MC(I,J,K+1)+MC(I,J,K-1)
0003ACI          IF(MCT .EQ. 0) GO TO 30                                        19
0003C2I          IF(MC(I,J+1,K) .EQ. 0) GO TO 2                                 21
         C-----NORTH
0003FCI          IBC(III)=I                                                     23
000410I          JBC(III)=J                                                     24
000424I          KBC(III)=K                                                     25
000438I          IITY(III)=1                                                    26
000448I          I1=I+1                                                         27
000456I          I2=I-1                                                         28
000464I          K1=K+1                                                         29
000472I          K2=K-1                                                         30
000480I          IF(I .EQ. L1) I2=I                                             31
00049EI          IF(I .EQ. L2) I1=I                                             33
0004BCI          IF(K .EQ. N1) K2=K                                             35
0004DAI          IF(K .EQ. N2) K1=K                                             37
0004F8I          J1=J-1                                                         39
000506I          J2=J-2                                                         40
000514I          P1=(Y(I1,J,K1)-Y(I1,J,K2))*(Z(I1,J,K2)-Z(I2,J,K))-             41
                1   (Z(I1,J,K1)-Z(I1,J,K2))*(Y(I1,J,K2)-Y(I2,J,K))
000650I          P2=(Z(I1,J,K1)-Z(I1,J,K2))*(X(I1,J,K2)-X(I2,J,K))-             42
                1   (X(I1,J,K1)-X(I1,J,K2))*(Z(I1,J,K2)-Z(I2,J,K))
00078CI          P3=(X(I1,J,K1)-X(I1,J,K2))*(Y(I1,J,K2)-Y(I2,J,K))-             43
                1   (Y(I1,J,K1)-Y(I1,J,K2))*(X(I1,J,K2)-X(I2,J,K))
```

```
 58  0008C8I         PQ=SQRT(P1*P1+P2*P2+P3*P3)                              261   44
 59  000912I         P1=P1/PQ                                                262   45
 60  000924I         P2=P2/PQ                                                263   46
 61  000936I         P3=P3/PQ                                                264   47
 62  000948I         R1=(1.-P1**2)                                           265   48
 63  000964I         R2=(1.-P2**2)                                           266   49
 64  000980I         R3=(1.-P3**2)                                           267   50
 65  00099CI         SINX(III)=SQRT(R1)                                      268   51
 66  0009C2I         SINY(III)=SQRT(R2)                                      269   52
 67  0009E6I         SINZ(III)=SQRT(R3)                                      270   53
 68  000A0AI         Q1=X(I,J,K)-X(I,J1,K)                                   271   54
 69  000A5CI         Q2=Y(I,J,K)-Y(I,J1,K)                                   272   55
 70  000AAEI         Q3=Z(I,J,K)-Z(I,J1,K)                                   273   56
 71  000B00I         AA=SQRT((Q1-P1)**2+(Q2-P2)**2+(Q3-P3)**2)               274   57
 72  000B7EI         CC=1.0                                                  275   58
 73  000B8AI         BB=SQRT(Q1*Q1+Q2*Q2+Q3*Q3)                              276   59
 74  000BD6I         COTH=(BB*BB+CC*CC-AA*AA)/(2*BB*CC)                      277   60
 75  000C18I         YN(III)=BB*ABS(COTH)                                    278   61
 76  000C44I         Q1=X(I,J,K)-X(I,J2,K)                                   279   62
 77  000C96I         Q2=Y(I,J,K)-Y(I,J2,K)                                   280   63
 78  000CE8I         Q3=Z(I,J,K)-Z(I,J2,K)                                   281   64
 79  000D3AI         BB=SQRT(Q1*Q1+Q2*Q2+Q3*Q3)                              282   65
 80  000D86I         AA=SQRT((Q1-P1)**2+(Q2-P2)**2+(Q3-P3)**2)               283   66
 81  000E02I         COTH=(BB*BB+CC*CC-AA*AA)/(2*BB*CC)                      284   67
 82  000E44I         YN1(III)=(BB*ABS(COTH)+YN(III))*0.5                     285   68
 83  000E84I         IJLC(I,J,K)=III                                         286   69
 84  000EB0I         III=III+1                                               287   70
 85  000EBEI     2   CONTINUE                                                288   71
 86  000EBEI         IF(MC(I,J-1,K) .EQ. 0) GO TO 3                          289   72
 87             C-----SOUTH                                                  290
 88  000EF8I         IBC(III)=I                                              291   74
 89  000F0CI         JBC(III)=J                                              292   75
 90  000F20I         KBC(III)=K                                              293   76
 91  000F34I         IITY(III)=2                                             294   77
 92  000F44I         I1=I+1                                                  295   78
 93  000F52I         I2=I-1                                                  296   79
 94  000F60I         K1=K+1                                                  297   80
 95  000F6EI         K2=K-1                                                  298   81
 96  000F7CI         J1=J+1                                                  299   82
 97  000F8AI         J2=J+2                                                  300   83
 98  000F98I         P1=(Y(I1,J,K1)-Y(I1,J,K2))*(Z(I1,J,K2)-Z(I2,J,K))-      301   84
 99                 1   (Z(I1,J,K1)-Z(I1,J,K2))*(Y(I1,J,K2)-Y(I2,J,K))       302
100  001004I         P2=(Z(I1,J,K1)-Z(I1,J,K2))*(X(I1,J,K2)-X(I2,J,K))-      303   85
101                 1   (X(I1,J,K1)-X(I1,J,K2))*(Z(I1,J,K2)-Z(I2,J,K))       304
102  001210I         P3=(X(I1,J,K1)-X(I1,J,K2))*(Y(I1,J,K2)-Y(I2,J,K))-      305   86
103                 1   (Y(I1,J,K1)-Y(I1,J,K2))*(X(I1,J,K2)-X(I2,J,K))       306
104  00134CI         PQ=SQRT(P1*P1+P2*P2+P3*P3)                              307   87
105  001396I         P1=P1/PQ                                                308   88
106  0013A8I         P2=P2/PQ                                                309   89
107  0013BAI         P3=P3/PQ                                                310   90
108  0013CCI         R1=(1.-P1**2)                                           311   91
109  0013E8I         R2=(1.-P2**2)                                           312   92
110  001404I         R3=(1.-P3**2)                                           313   93
111  001420I         SINX(III)=SQRT(R1)                                      314   94
112  001446I         SINY(III)=SQRT(R2)                                      315   95
113  00146AI         SINZ(III)=SQRT(R3)                                      316   96
114  00148EI         Q1=X(I,J,K)-X(I,J1,K)                                   317   97
```

```
115  0014E0I        Q2=Y(I,J,K)-Y(I,J1,K)                                     318    98
116  001532I        Q3=Z(I,J,K)-Z(I,J1,K)                                     319    99
117  001584I        AA=SQRT((Q1-P1)**2+(Q2-P2)**2+(Q3-P3)**2)                 320   100
118  001602I        CC=1.0                                                    321   101
119  00160EI        BB=SQRT(Q1*Q1+Q2*Q2+Q3*Q3)                                322   102
120  00165AI        COTH=(BB*BB+CC*CC-AA*AA)/(2*BB*CC)                        323   103
121  00169CI        YN(III)=BB*ABS(COTH)                                      324   104
122  0016C8I        Q1=X(I,J,K)-X(I,J2,K)                                     325   105
123  00171AI        Q2=Y(I,J,K)-Y(I,J2,K)                                     326   106
124  00176CI        Q3=Z(I,J,K)-Z(I,J2,K)                                     327   107
125  0017BEI        BB=SQRT(Q1*Q1+Q2*Q2+Q3*Q3)                                328   108
126  00180AI        AA=SQRT((Q1-P1)**2+(Q2-P2)**2+(Q3-P3)**2)                 329   109
127  001866I        COTH=(BB*BB+CC*CC-AA*AA)/(2*BB*CC)                        330   110
128  0018C8I        YN1(III)=(BB*ABS(COTH)+YN(III))*0.5                       331   111
129  001903I        IJLC(I,J,K)=III                                           332   112
130  001934I        III=III+1                                                 333   113
131  001942I      3 CONTINUE                                                  334   114
132  001942I        IF(MC(I+1,J,K) .EQ. 0) GO TO 4                            335   115
133           C-----EAST                                                      336
134  00197CI        IBC(III)=I                                                337   117
135  001990I        JBC(III)=J                                                338   118
136  0019A4I        KBC(III)=K                                                339   119
137  0019B8I        IITY(III)=3                                               340   120
138  0019C8I        J1=J+1                                                    341   121
139  0019D6I        J2=J-1                                                    342   122
140  0019E4I        K1=K+1                                                    343   123
141  0019F2I        K2=K-1                                                    344   124
142  001A00I        IF(J .EQ. M1) J2=J                                        345   125
143  001A1EI        IF(J .EQ. M2) J1=J                                        346   127
144  001A3CI        IF(K .EQ. N1) K2=K                                        347   129
145  001A5AI        IF(K .EQ. N2) K1=K                                        348   131
146  001A78I        I1=I-1                                                    349   133
147  001A86I        I2=I-2                                                    350   134
148  001A94I        P1=(Y(I,J1,K1)-Y(I,J1,K2))*(Z(I,J1,K2)-Z(I,J2,K))-        351   135
149                1   (Z(I,J1,K1)-Z(I,J1,K2))*(Y(I,J1,K2)-Y(I,J2,K))         352
150  001B00I        P2=(Z(I,J1,K1)-Z(I,J1,K2))*(X(I,J1,K2)-X(I,J2,K))-        353   136
151                1   (X(I,J1,K1)-X(I,J1,K2))*(Z(I,J1,K2)-Z(I,J2,K))         354
152  001D0CI        P3=(X(I,J1,K1)-X(I,J1,K2))*(Y(I,J1,K2)-Y(I,J2,K))-        355   137
153                1   (Y(I,J1,K1)-Y(I,J1,K2))*(X(I,J1,K2)-X(I,J2,K))         356
154  001E48I        PQ=SQRT(P1*P1+P2*P2+P3*P3)                                357   138
155  001E92I        P1=P1/PQ                                                  358   139
156  001EA4I        P2=P2/PQ                                                  359   140
157  001EB6I        P3=P3/PQ                                                  360   141
158  001EC8I        R1=(1.-P1**2)                                             361   142
159  001EE4I        R2=(1.-P2**2)                                             362   143
160  001F00I        R3=(1.-P3**2)                                             363   144
161  001F1CI        SINX(III)=SQRT(R1)                                        364   145
162  001F42I        SINY(III)=SQRT(R2)                                        365   146
163  001F66I        SINZ(III)=SQRT(R3)                                        366   147
164  001F8AI        Q1=X(I,J,K)-X(I1,J,K)                                     367   148
165  001FDCI        Q2=Y(I,J,K)-Y(I1,J,K)                                     368   149
166  00202EI        Q3=Z(I,J,K)-Z(I1,J,K)                                     369   150
167  002080I        AA=SQRT((Q1-P1)**2+(Q2-P2)**2+(Q3-P3)**2)                 370   151
168  0020FEI        CC=1.0                                                    371   152
169  00210AI        BB=SQRT(Q1*Q1+Q2*Q2+Q3*Q3)                                372   153
170  002156I        COTH=(BB*BB+CC*CC-AA*AA)/(2*BB*CC)                        373   154
171  002198I        YN(III)=BB*ABS(COTH)                                      374   155
```

```
0021C4I        Q1=X(I,J,K)-X(I2,J,K)                                             375    156
002216I        Q2=Y(I,J,K)-Y(I2,J,K)                                             376    157
002268I        Q3=Z(I,J,K)-Z(I2,J,K)                                             377    158
00228AI        BB=SQRT(Q1*Q1+Q2*Q2+Q3*Q3)                                        378    159
002306I        AA=SQRT((Q1-P1)**2+(Q2-P2)**2+(Q3-P3)**2)                         379    160
002382I        COTH=(BB*BB+CC*CC-AA*AA)/(2*BB*CC)                                380    161
0023C4I        YN1(III)=(BB*ABS(COTH)+YN(III))*0.5                               381    162
002404I        IJLC(I,J,K)=III                                                   382    163
002430I        III=III+1                                                         383    164
00243EI      4 CONTINUE                                                          384    165
00243EI        IF(MC(I-1,J,K) .EQ. 0) GO TO 5                                    385    166
            C-----WEST                                                           386
002478I        ISC(III)=I                                                        387    168
00248CI        JBC(III)=J                                                        388    169
0024A0I        KBC(III)=K                                                        389    170
0024B4I        IITY(III)=4                                                       390    171
0024C4I        J1=J+1                                                            391    172
0024D2I        J2=J-1                                                            392    173
0024E0I        K1=K+1                                                            393    174
0024EEI        K2=K-1                                                            394    175
0024FCI        IF(J .EQ. M1) J2=J                                                395    176
00251AI        IF(J .EQ. M2) J1=J                                                396    178
002538I        IF(K .EQ. N1) K2=K                                                397    180
002556I        IF(K .EQ. N2) K1=K                                                398    182
002574I        I1=I+1                                                            399    184
002582I        I2=I+2                                                            400    185
002590I        P1=(Y(I,J1,K1)-Y(I,J1,K2))*(Z(I,J1,K2)-Z(I,J2,K))-                401    186
              1   (Z(I,J1,K1)-Z(I,J1,K2))*(Y(I,J1,K2)-Y(I,J2,K))                 402
0026CCI        P2=(Z(I,J1,K1)-Z(I,J1,K2))*(X(I,J1,K2)-X(I,J2,K))-                403    187
              1   (X(I,J1,K1)-X(I,J1,K2))*(Z(I,J1,K2)-Z(I,J2,K))                 404
002808I        P3=(X(I,J1,K1)-X(I,J1,K2))*(Y(I,J1,K2)-Y(I,J2,K))-                405    188
              1   (Y(I,J1,K1)-Y(I,J1,K2))*(X(I,J1,K2)-X(I,J2,K))                 406
002944I        PQ=SQRT(P1*P1+P2*P2+P3*P3)                                        407    189
00298EI        P1=P1/PQ                                                          408    190
0029A0I        P2=P2/PQ                                                          409    191
0029B2I        P3=P3/PQ                                                          410    192
0029C4I        R1=1.-P1**2                                                       411    193
0029E0I        R2=1.-P2**2                                                       412    194
0029FCI        R3=1.-P3**2                                                       413    195
002A18I        SINX(III)=SQRT(R1)                                                414    196
002A3EI        SINY(III)=SQRT(R2)                                                415    197
002A62I        SINZ(III)=SQRT(R3)                                                416    198
002A86I        Q1=X(I,J,K)-X(I1,J,K)                                             417    199
002AD8I        Q2=Y(I,J,K)-Y(I1,J,K)                                             418    200
002B2AI        Q3=Z(I,J,K)-Z(I1,J,K)                                             419    201
002B7CI        AA=SQRT((Q1-P1)**2+(Q2-P2)**2+(Q3-P3)**2)                         420    202
002BFAI        CC=1.0                                                            421    203
002C06I        BB=SQRT(Q1*Q1+Q2*Q2+Q3*Q3)                                        422    204
002C52I        COTH=(BB*BB+CC*CC-AA*AA)/(2*BB*CC)                                423    205
002C94I        YN(III)=BB*ABS(COTH)                                              424    206
002CC0I        Q1=X(I,J,K)-X(I2,J,K)                                             425    207
002D12I        Q2=Y(I,J,K)-Y(I2,J,K)                                             426    208
002D64I        Q3=Z(I,J,K)-Z(I2,J,K)                                             427    209
002D36I        BB=SQRT(Q1*Q1+Q2*Q2+Q3*Q3)                                        428    210
002E02I        AA=SQRT((Q1-P1)**2+(Q2-P2)**2+(Q3-P3)**2)                         429    211
002E7EI        COTH=(BB*BB+CC*CC-AA*AA)/(2*BB*CC)                                430    212
002EC0I        YN1(III)=(BB*ABS(COTH)+YN(III))*0.5                               431    213
```

```
002F00I        IJLC(I,J,K)=III                                          214
002F2CI        III=III+1                                                215
002F3AI    5   CONTINUE                                                 216
002F3AI        IF(MC(I,J,K+1) .EQ. 0) GO TO 6                           217
         C-----TOP
002F72I        IBC(III)=I                                               219
002F86I        JBC(III)=J                                               220
002F9AI        KBC(III)=K                                               221
002FAEI        IITY(III)=5                                              222
002FBEI        I1=I+1                                                   223
002FCCI        I2=I-1                                                   224
002FDAI        J1=J+1                                                   225
002FE8I        J2=J-1                                                   226
002FF6I        IF(I .EC. L1) I2=I                                       227
003014I        IF(I .EC. L2) I1=I                                       229
003032I        IF(J .EC. M1) J2=J                                       231
003050I        IF(J .EC. M2) J1=J                                       233
00306EI        K1=K-1                                                   235
00307CI        K2=K-2                                                   236
00308AI        P1=(Y(I1,J1,K)-Y(I1,J2,K))*(Z(I1,J2,K)-Z(I2,J,K))-       237
              1   (Z(I1,J1,K)-Z(I1,J2,K))*(Y(I1,J2,K)-Y(I2,J,K))
0031C6I        P2=(Z(I1,J1,K)-Z(I1,J2,K))*(X(I1,J2,K)-X(I2,J,K))-       238
              1   (X(I1,J1,K)-X(I1,J2,K))*(Z(I1,J2,K)-Z(I2,J,K))
003302I        P3=(X(I1,J1,K)-X(I1,J2,K))*(Y(I1,J2,K)-Y(I2,J,K))-       239
              1   (Y(I1,J1,K)-Y(I1,J2,K))*(X(I1,J2,K)-X(I2,J,K))
00343EI        PQ=SQRT(P1*P1+P2*P2+P3*P3)                               240
00348AI        P1=P1/PC                                                 241
00349CI        P2=P2/PC                                                 242
0034AEI        P3=P3/PC                                                 243
0034C0I        R1=1.-P1**2                                              244
0034DCI        R2=1.-P2**2                                              245
0034F8I        R3=1.-P3**2                                              246
003514I        SINX(III)=SQRT(R1)                                       247
00353AI        SINY(III)=SQRT(R2)                                       248
00355EI        SINZ(III)=SQRT(R3)                                       249
003582I        Q1=X(I,J,K)-X(I,J,K1)                                    250
0035D4I        Q2=Y(I,J,K)-Y(I,J,K1)                                    251
003626I        Q3=Z(I,J,K)-Z(I,J,K1)                                    252
003678I        AA=SQRT((Q1-P1)**2+(Q2-P2)**2+(Q3-P3)**2)                253
0036F6I        CC=1.0                                                   254
003702I        BB=SQRT(Q1*Q1+Q2*Q2+Q3*Q3)                               255
00374EI        COTH=(BB*BB+CC*CC-AA*AA)/(2*BB*CC)                       256
003790I        YN(III)=BB*ABS(COTH)                                     257
0037BCI        Q1=X(I,J,K)-X(I,J,K2)                                    258
00380EI        Q2=Y(I,J,K)-Y(I,J,K2)                                    259
003860I        Q3=Z(I,J,K)-Z(I,J,K2)                                    260
0038B2I        BB=SQRT(Q1*Q1+Q2*Q2+Q3*Q3)                               261
0038FEI        AA=SQRT((Q1-P1)**2+(Q2-P2)**2+(Q3-P3)**2)                262
00397AI        COTH=(BB*BB+CC*CC-AA*AA)/(2*BB*CC)                       263
0039BCI        YN1(III)=(BB*ABS(COTH)+YN(III))*0.5                      264
0039FCI        IJLC(I,J,K)=III                                          265
003A28I        III=III+1                                                266
003A36I    6   CONTINUE                                                 267
003A36I        IF(MC(I,J,K-1) .EQ. 0) GO TO 30                          268
         C-----BOTTOM
003A6EI        IBC(III)=I                                               270
003A82I        JBC(III)=J                                               271
```

```
003A96I          KBC(III)=K                                          272
003AAAI          IITY(III)=6                                         273
003ABAI          I1=I+1                                              274
003AC8I          I2=I-1                                              275
003AD6I          J1=J+1                                              276
003AE4I          J2=J-1                                              277
003AF2I          IF(I .EQ. L1) I2=I                                  278
003B10I          IF(I .EQ. L2) I1=I                                  280
003B2EI          IF(J .EQ. M1) J2=J                                  282
003B4CI          IF(J .EQ. M2) J1=J                                  284
003B6AI          K1=K+1                                              286
003B78I          K2=K+2                                              287
003B86I          P1=(Y(I1,J1,K)-Y(I1,J2,K))*(Z(I1,J2,K)-Z(I2,J,K))-  288
                1    (Z(I1,J1,K)-Z(I1,J2,K))*(Y(I1,J2,K)-Y(I2,J,K))
003CC2I          P2=(Z(I1,J1,K)-Z(I1,J2,K))*(X(I1,J2,K)-X(I2,J,K))-  289
                1    (X(I1,J1,K)-X(I1,J2,K))*(Z(I1,J2,K)-Z(I2,J,K))
003DFEI          P3=(X(I1,J1,K)-X(I1,J2,K))*(Y(I1,J2,K)-Y(I2,J,K))-  290
                1    (Y(I1,J1,K)-Y(I1,J2,K))*(X(I1,J2,K)-X(I2,J,K))
003F3AI          PQ=SQRT(P1*P1+P2*P2+P3*P3)                          291
003F86I          P1=P1/PQ                                            292
003F98I          P2=P2/PQ                                            293
003FAAI          P3=P3/PQ                                            294
003FBCI          R1=1.-P1**2                                         295
003FD8I          R2=1.-P2**2                                         296
003FF4I          R3=1.-P3**2                                         297
004010I          SINX(III)=SQRT(R1)                                  298
004036I          SINY(III)=SQRT(R2)                                  299
00405AI          SINZ(III)=SQRT(R3)                                  300
00407EI          Q1=X(I,J,K)-X(I,J,K1)                               301
0040D0I          Q2=Y(I,J,K)-Y(I,J,K1)                               302
004122I          Q3=Z(I,J,K)-Z(I,J,K1)                               303
004174I          AA=SQRT((Q1-P1)**2+(Q2-P2)**2+(Q3-P3)**2)           304
0041F2I          CC=1.0                                              305
0041FEI          BB=SQRT(Q1*Q1+Q2*Q2+Q3*Q3)                          306
00424AI          COTH=(BB*BB+CC*CC-AA*AA)/(2*BB*CC)                  307
00428CI          YN(III)=BB*ABS(COTH)                                308
0042B8I          Q1=X(I,J,K)-X(I,J,K2)                               309
00430AI          Q2=Y(I,J,K)-Y(I,J,K2)                               310
00435CI          Q3=Z(I,J,K)-Z(I,J,K2)                               311
0043AEI          BB=SQRT(Q1*Q1+Q2*Q2+Q3*Q3)                          312
0043FAI          AA=SQRT((Q1-P1)**2+(Q2-P2)**2+(Q3-P3)**2)           313
004476I          COTH=(BB*BB+CC*CC-AA*AA)/(2*BB*CC)                  314
004498I          YN1(III)=(BB*ABS(COTH)+YN(III))*0.5                 315
0044F8I          IJLC(I,J,K)=III                                     316
004524I          III=III+1                                           317
004532I     30   CONTINUE                                            318
004580I          IITO=III-1                                          319
00458EI          WRITE(6,100) LO,MO,NO,IITO                          320
004598I    100   FORMAT(4I5)
0045C4I          RETURN                                              321
0045CCI          END                                                 322

NO  ERRORS:F7D  R05-01.0C  SUBROUTINE  CIRCCS    02/21/86  09:47:58  TABLE SPACE:   6 KB
STATEMENT BUFFER:   20 LINES/1321 BYTES     STACK SPACE: 203 WORDS
SINGLE PRECISION FLOATING PT SUPPORT REQUIRED FOR EXECUTION
```

```
000000I          SUBROUTINE TRANF
000004I          COMMON
               1/VAR/U(21,18,10),V(21,18,10),P(21,18,10),DK(21,18,10),
               2  DE(21,18,10),ERRU,ERRV,ERRM,ERRK,ERRE,ERRW,
               3  PP(21,18,10),W(21,18,10),TM(21,18,10)
               1/PROP/ VISE(21,18,10),DEN(21,18,10),VISC,DENIN,FLOWIN
               1/PCOR/ DU(21,18,10),DV(21,18,10),DW(21,18,10)
               1/TUR/ SIGK,SIGE,CMU,C1,C2,CMU1,CMU2,E,CK,HINUM,SMNUM,ANV1(800),
               2 YN(800),YN1(800),SINX(800),SINY(800),SINZ(800),ANW1(800),
               3 YPLN(800),TAUN(800),IBC(800),JBC(800),KBC(800),IITY(800),
               4 TAUW(800),GEN(21,18,10),MC(21,18,10),IJLO(21,18,10),IITO
               1/COEF/ AP(21,18,10),SU(21,18,10),SP(21,18,10),SUK(21,18,10),
               2    SPK(21,18,10),AE(21,18,10),AW(21,18,10),AN(21,18,10),
               3    AS(21,18,10),AT(21,18,10),AB(21,18,10),APO(21,18,10)
000004I          COMMON
               1/TRAN/ X(21,18,10),Y(21,18,10),Z(21,18,10),TJO(21,18,10),
               2    CX(21,18,10),CY(21,18,10),CZ(21,18,10),
               3    EX(21,18,10),EY(21,18,10),EZ(21,18,10),
               3    SX(21,18,10),SY(21,18,10),SZ(21,18,10)
               1/LIMT/ L,M,LT,MT,L1,L2,M1,M2,LO,MO,ISWU,ISWV,ISWP,ISWK,ISWE,
               2    ALU,ALV,ALP,ALK,ALE,ALVIS,ALW,N,N1,N2,NO,ISWW,IG,NT,ALC,DTT
000004I          COMMON
               1/TTRAN/TXXE(21,18,10),TXXW(21,18,10),TYYN(21,18,10),
               2    TYYS(21,18,10),TZZT(21,18,10),TZZB(21,18,10),
               3    TYXE(21,18,10),TYXW(21,18,10),TYZT(21,18,10),
               4    TYZB(21,18,10),TXYN(21,18,10),TXYS(21,18,10),
               5    TXZT(21,18,10),TXZB(21,18,10),TZXE(21,18,10),
               6    TZXW(21,18,10),TZYN(21,18,10),TZYS(21,18,10)
            C-----CALCULATE GRID TRANSFORMATION COEFFICIENTS
000004I          DO 40 I=2,L                                                   1
000018I          DO 40 J=2,M                                                   2
00002CI          DO 40 K=2,N                                                   3
000040I          P1=(X(I,J,K)+X(I,J-1,K)+X(I,J,K-1)+X(I,J-1,K-1)-X(I-1,J,K)-   4
               1  X(I-1,J-1,K)-X(I-1,J,K-1)-X(I-1,J-1,K-1))*0.25
0001A4I          P2=(X(I,J,K)+X(I-1,J,K)+X(I,J,K-1)+X(I-1,J,K-1)-X(I,J-1,K)-   5
               1  X(I-1,J-1,K)-X(I,J-1,K-1)-X(I-1,J-1,K-1))*0.25
000308I          P3=(X(I,J,K)+X(I-1,J,K)+X(I,J-1,K)+X(I-1,J-1,K)-X(I,J,K-1)-   6
               1  X(I-1,J,K-1)-X(I,J-1,K-1)-X(I-1,J-1,K-1))*0.25
00046CI          Q1=(Y(I,J,K)+Y(I,J-1,K)+Y(I,J,K-1)+Y(I,J-1,K-1)-Y(I-1,J,K)-   7
               1  Y(I-1,J-1,K)-Y(I-1,J,K-1)-Y(I-1,J-1,K-1))*0.25
0005D0I          Q2=(Y(I,J,K)+Y(I-1,J,K)+Y(I,J,K-1)+Y(I-1,J,K-1)-Y(I,J-1,K)-   8
               1  Y(I-1,J-1,K)-Y(I,J-1,K-1)-Y(I-1,J-1,K-1))*0.25
000734I          Q3=(Y(I,J,K)+Y(I-1,J,K)+Y(I,J-1,K)+Y(I-1,J-1,K)-Y(I,J,K-1)-   9
               1  Y(I-1,J,K-1)-Y(I,J-1,K-1)-Y(I-1,J-1,K-1))*0.25
000898I          R1=(Z(I,J,K)+Z(I,J-1,K)+Z(I,J,K-1)+Z(I,J-1,K-1)-Z(I-1,J,K)-   10
               1  Z(I-1,J-1,K)-Z(I-1,J,K-1)-Z(I-1,J-1,K-1))*0.25
0009FCI          R2=(Z(I,J,K)+Z(I-1,J,K)+Z(I,J,K-1)+Z(I-1,J,K-1)-Z(I,J-1,K)-   11
               1  Z(I-1,J-1,K)-Z(I,J-1,K-1)-Z(I-1,J-1,K-1))*0.25
000B60I          R3=(Z(I,J,K)+Z(I-1,J,K)+Z(I,J-1,K)+Z(I-1,J-1,K)-Z(I,J,K-1)-   12
               1  Z(I-1,J,K-1)-Z(I,J-1,K-1)-Z(I-1,J-1,K-1))*0.25
000CC4I          PTR=1.0/(P1*(Q2*R3-Q3*R2)-P2*(Q1*R3-Q3*R1)+P3*(Q1*R2-Q2*R1))  13
000D36I          SU(I,J,K)=ABS(1.0/PTR)                                        14
000D86I          AE(I,J,K)=PTR*(Q2*R3-Q3*R2)                                   15
000DCCI          AW(I,J,K)=-PTR*(P2*R3-P3*R2)                                  16
000E1AI          AN(I,J,K)=PTR*(P2*Q3-P3*Q2)                                   17
000E60I          AS(I,J,K)=-PTR*(Q1*R3-Q3*R1)                                  18
000EAEI          AT(I,J,K)=PTR*(P1*R3-P3*R1)                                   19
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
000EF4I          AB(I,J,K)=-PTR*(P1*Q3-P3*Q1)                                      20
000F42I          DU(I,J,K)=PTR*(Q1*R2-Q2*R1)                                       21
000F88I          DV(I,J,K)=-PTR*(P1*R2-P2*R1)                                      22
000FD6I          DW(I,J,K)=PTR*(P1*Q2-P2*Q1)                                       23
00101CI     40   CONTINUE                                                          24
001064I          CALL WALVAL(1.0,2,L,2,M,2,N,AE)                                   25
0010BCI          CALL WALVAL(1.0,2,L,2,M,2,N,AW)                                   26
001114I          CALL WALVAL(1.0,2,L,2,M,2,N,AN)                                   27
00116CI          CALL WALVAL(1.0,2,L,2,M,2,N,AS)                                   28
0011C4I          CALL WALVAL(1.0,2,L,2,M,2,N,AT)                                   29
00121CI          CALL WALVAL(1.0,2,L,2,M,2,N,AB)                                   30
001274I          CALL WALVAL(1.0,2,L,2,M,2,N,DU)                                   31
0012CCI          CALL WALVAL(1.0,2,L,2,M,2,N,DV)                                   32
001324I          CALL WALVAL(1.0,2,L,2,M,2,N,DW)                                   33
00137CI          CALL WALVAL(1.0,2,L,2,M,2,N,SU)                                   34
0013D4I          DO 80 I=1,L                                                       35
0013E8I          DO 80 J=1,M                                                       36
0013FCI          DO 80 K=1,N                                                       37
001410I          CX(I,J,K)=(AE(I,J,K)+AE(I,J+1,K)+AE(I,J,K+1)+AE(I,J+1,K+1)+       38
                1 AE(I+1,J,K)+AE(I+1,J+1,K)+AE(I+1,J,K+1)+AE(I+1,J+1,K+1))*0.125
001594I          CY(I,J,K)=(AW(I,J,K)+AW(I,J+1,K)+AW(I,J,K+1)+AW(I,J+1,K+1)+       39
                1 AW(I+1,J,K)+AW(I+1,J+1,K)+AW(I+1,J,K+1)+AW(I+1,J+1,K+1))*0.125
001718I          CZ(I,J,K)=(AN(I,J,K)+AN(I,J+1,K)+AN(I,J,K+1)+AN(I,J+1,K+1)+       40
                1 AN(I+1,J,K)+AN(I+1,J+1,K)+AN(I+1,J,K+1)+AN(I+1,J+1,K+1))*0.125
00189CI          EX(I,J,K)=(AS(I,J,K)+AS(I,J+1,K)+AS(I,J,K+1)+AS(I,J+1,K+1)+       41
                1 AS(I+1,J,K)+AS(I+1,J+1,K)+AS(I+1,J,K+1)+AS(I+1,J+1,K+1))*0.125
001A20I          EY(I,J,K)=(AT(I,J,K)+AT(I,J+1,K)+AT(I,J,K+1)+AT(I,J+1,K+1)+       42
                1 AT(I+1,J,K)+AT(I+1,J+1,K)+AT(I+1,J,K+1)+AT(I+1,J+1,K+1))*0.125
001BA4I          EZ(I,J,K)=(AB(I,J,K)+AB(I,J+1,K)+AB(I,J,K+1)+AB(I,J+1,K+1)+       43
                1 AB(I+1,J,K)+AB(I+1,J+1,K)+AB(I+1,J,K+1)+AB(I+1,J+1,K+1))*0.125
001D28I          SX(I,J,K)=(DU(I,J,K)+DU(I,J+1,K)+DU(I,J,K+1)+DU(I,J+1,K+1)+       44
                1 DU(I+1,J,K)+DU(I+1,J+1,K)+DU(I+1,J,K+1)+DU(I+1,J+1,K+1))*0.125
001EACI          SY(I,J,K)=(DV(I,J,K)+DV(I,J+1,K)+DV(I,J,K+1)+DV(I,J+1,K+1)+       45
                1 DV(I+1,J,K)+DV(I+1,J+1,K)+DV(I+1,J,K+1)+DV(I+1,J+1,K+1))*0.125
002030I          SZ(I,J,K)=(DW(I,J,K)+DW(I,J+1,K)+DW(I,J,K+1)+DW(I,J+1,K+1)+       46
                1 DW(I+1,J,K)+DW(I+1,J+1,K)+DW(I+1,J,K+1)+DW(I+1,J+1,K+1))*0.125
002194I          TJO(I,J,K)=(SU(I,J,K)+SU(I,J+1,K)+SU(I,J,K+1)+SU(I,J+1,K+1)+      47
                1 SU(I+1,J,K)+SU(I+1,J+1,K)+SU(I+1,J,K+1)+SU(I+1,J+1,K+1))*0.125
002338I     80   CONTINUE                                                          48
00238OI          DO 200 I=1,LO                                                     49
002394I          DO 200 J=1,MO                                                     50
0023A8I          DO 200 K=1,NO                                                     51
0023BCI          DU(I,J,K)=0.0                                                     52
0023E8I    200   CONTINUE                                                          53
002430I          DO 160 I=2,LT                                                     54
002444I          DO 160 J=2,MT                                                     55
002458I          DO 160 K=2,NT                                                     56
00246CI          CXE=(CX(I+1,J,K)+CX(I,J,K))*0.5                                   57
0024C8I          CXW=(CX(I-1,J,K)+CX(I,J,K))*0.5                                   58
002524I          CXN=(CX(I,J+1,K)+CX(I,J,K))*0.5                                   59
002580I          CXS=(CX(I,J,K)+CX(I,J-1,K))*0.5                                   60
0025DCI          CXT=(CX(I,J,K+1)+CX(I,J,K))*0.5                                   61
002636I          CXB=(CX(I,J,K)+CX(I,J,K-1))*0.5                                   62
002690I          CYE=(CY(I+1,J,K)+CY(I,J,K))*0.5                                   63
0026ECI          CYW=(CY(I,J,K)+CY(I-1,J,K))*0.5                                   64
002748I          CYN=(CY(I,J+1,K)+CY(I,J,K))*0.5                                   65
0027A4I          CYS=(CY(I,J,K)+CY(I,J-1,K))*0.5                                   66
```

```
002800I   CYT=(CY(I,J,K+1)+CY(I,J,K))*0.5                                          654   67
00285AI   CYB=(CY(I,J,K)+CY(I,J,K-1))*0.5                                          655   68
0028B4I   CZE=(CZ(I+1,J,K)+CZ(I,J,K))*0.5                                          656   69
00291OI   CZW=(CZ(I,J,K)+CZ(I-1,J,K))*0.5                                          657   70
00296CI   CZN=(CZ(I,J+1,K)+CZ(I,J,K))*0.5                                          658   71
0029C8I   CZS=(CZ(I,J,K)+CZ(I,J-1,K))*0.5                                          659   72
002A24I   CZT=(CZ(I,J,K+1)+CZ(I,J,K))*0.5                                          660   73
002A7EI   CZB=(CZ(I,J,K)+CZ(I,J,K-1))*0.5                                          661   74
002AD8I   EXE=(EX(I+1,J,K)+EX(I,J,K))*0.5                                          662   75
002B34I   EXW=(EX(I,J,K)+EX(I-1,J,K))*0.5                                          663   76
002B90I   EXN=(EX(I,J+1,K)+EX(I,J,K))*0.5                                          664   77
002BECI   EXS=(EX(I,J,K)+EX(I,J-1,K))*0.5                                          665   78
002C48I   EXT=(EX(I,J,K+1)+EX(I,J,K))*0.5                                          666   79
002CA2I   EXB=(EX(I,J,K)+EX(I,J,K-1))*0.5                                          667   80
002CFCI   EYE=(EY(I+1,J,K)+EY(I,J,K))*0.5                                          668   81
002D58I   EYW=(EY(I,J,K)+EY(I-1,J,K))*0.5                                          669   82
002DB4I   EYN=(EY(I,J+1,K)+EY(I,J,K))*0.5                                          670   83
002E10I   EYS=(EY(I,J,K)+EY(I,J-1,K))*0.5                                          671   84
002E6CI   EYT=(EY(I,J,K+1)+EY(I,J,K))*0.5                                          672   85
002EC6I   EYB=(EY(I,J,K)+EY(I,J,K-1))*0.5                                          673   86
002F20I   EZE=(EZ(I+1,J,K)+EZ(I,J,K))*0.5                                          674   87
002F7CI   EZW=(EZ(I,J,K)+EZ(I-1,J,K))*0.5                                          675   88
002FD8I   EZN=(EZ(I,J+1,K)+EZ(I,J,K))*0.5                                          676   89
003034I   EZS=(EZ(I,J,K)+EZ(I,J-1,K))*0.5                                          677   90
003090I   EZT=(EZ(I,J,K+1)+EZ(I,J,K))*0.5                                          678   91
0030EAI   EZB=(EZ(I,J,K)+EZ(I,J,K-1))*0.5                                          679   92
003144I   SXE=(SX(I+1,J,K)+SX(I,J,K))*0.5                                          680   93
0031A0I   SXW=(SX(I,J,K)+SX(I-1,J,K))*0.5                                          681   94
0031FCI   SXN=(SX(I,J+1,K)+SX(I,J,K))*0.5                                          682   95
003258I   SXS=(SX(I,J,K)+SX(I,J-1,K))*0.5                                          683   96
0032B4I   SXT=(SX(I,J,K+1)+SX(I,J,K))*0.5                                          684   97
00330EI   SXB=(SX(I,J,K)+SX(I,J,K-1))*0.5                                          685   98
003368I   SYE=(SY(I+1,J,K)+SY(I,J,K))*0.5                                          686   99
0033C4I   SYW=(SY(I,J,K)+SY(I-1,J,K))*0.5                                          687   100
003420I   SYN=(SY(I,J+1,K)+SY(I,J,K))*0.5                                          688   101
00347CI   SYS=(SY(I,J,K)+SY(I,J-1,K))*0.5                                          689   102
0034D8I   SYT=(SY(I,J,K+1)+SY(I,J,K))*0.5                                          690   103
003532I   SYB=(SY(I,J,K)+SY(I,J,K-1))*0.5                                          691   104
00358CI   SZE=(SZ(I+1,J,K)+SZ(I,J,K))*0.5                                          692   105
0035E8I   SZW=(SZ(I,J,K)+SZ(I-1,J,K))*0.5                                          693   106
003644I   SZN=(SZ(I,J+1,K)+SZ(I,J,K))*0.5                                          694   107
0036A0I   SZS=(SZ(I,J,K)+SZ(I,J-1,K))*0.5                                          695   108
0036FCI   SZT=(SZ(I,J,K+1)+SZ(I,J,K))*0.5                                          696   109
003756I   SZB=(SZ(I,J,K)+SZ(I,J,K-1))*0.5                                          697   110
0037B0I   TXXE(I,J,K)=CXE*CX(I,J,K)+CYE*CY(I,J,K)+CZE*CZ(I,J,K)                    698   111
00385EI   TXXW(I,J,K)=CXW*CX(I,J,K)+CYW*CY(I,J,K)+CZW*CZ(I,J,K)                    699   112
00390CI   TYYN(I,J,K)=EXN*EX(I,J,K)+EYN*EY(I,J,K)+EZN*EZ(I,J,K)                    700   113
0039BAI   TYYS(I,J,K)=EXS*EX(I,J,K)+EYS*EY(I,J,K)+EZS*EZ(I,J,K)                    701   114
003A68I   TZZT(I,J,K)=SXT*SX(I,J,K)+SYT*SY(I,J,K)+SZT*SZ(I,J,K)                    702   115
003B16I   TZZB(I,J,K)=SXB*SX(I,J,K)+SYB*SY(I,J,K)+SZB*SZ(I,J,K)                    703   116
003BC4I   TYXE(I,J,K)=(EXE*CX(I,J,K)+EYE*CY(I,J,K)+EZE*CZ(I,J,K))*0.25             704   117
003C78I   TYXW(I,J,K)=(EXW*CX(I,J,K)+EYW*CY(I,J,K)+EZW*CZ(I,J,K))*0.25             705   118
003D2CI   TYZT(I,J,K)=(EXT*SX(I,J,K)+EYT*SY(I,J,K)+EZT*SZ(I,J,K))*0.25             706   119
003DE0I   TYZB(I,J,K)=(EXB*SX(I,J,K)+EYB*SY(I,J,K)+EZB*SZ(I,J,K))*0.25             707   120
003E94I   TXYN(I,J,K)=(CXN*EX(I,J,K)+CYN*EY(I,J,K)+CZN*EZ(I,J,K))*0.25             708   121
003F48I   TXYS(I,J,K)=(CXS*EX(I,J,K)+CYS*EY(I,J,K)+CZS*EZ(I,J,K))*0.25             709   122
003FFCI   TXZT(I,J,K)=(CXT*SX(I,J,K)+CYT*SY(I,J,K)+CZT*SZ(I,J,K))*0.25             710   123
```

```
172  004080I          TXZB(I,J,K)=(CXB*SX(I,J,K)+CYB*SY(I,J,K)+CZB*SZ(I,J,K))*0.25          711   124
173  004164I          TZXE(I,J,K)=(SXE*CX(I,J,K)+SYE*CY(I,J,K)+SZE*CZ(I,J,K))*0.25          712   125
174  004218I          TZXW(I,J,K)=(SXW*CX(I,J,K)+SYW*CY(I,J,K)+SZW*CZ(I,J,K))*0.25          713   126
175  0042CCI          TZYN(I,J,K)=(SXN*EX(I,J,K)+SYN*EY(I,J,K)+SZN*EZ(I,J,K))*0.25          714   127
176  004380I          TZYS(I,J,K)=(SXS*EX(I,J,K)+SYS*EY(I,J,K)+SZS*EZ(I,J,K))*0.25          715   128
177  004434I    160   CONTINUE                                                              716   129
178  00447CI          CALL WALVAL(1.0,2,LT,2,MT,2,NT,TXXE)                                  717   130
179  0044D4I          CALL WALVAL(1.0,2,LT,2,MT,2,NT,TXXW)                                  718   131
180  00452CI          CALL WALVAL(1.C,2,LT,2,MT,2,NT,TYYN)                                  719   132
181  004584I          CALL WALVAL(1.0,2,LT,2,MT,2,NT,TYYS)                                  720   133
182  0045DCI          CALL WALVAL(1.0,2,LT,2,MT,2,NT,TZZT)                                  721   134
183  004634I          CALL WALVAL(1.0,2,LT,2,MT,2,NT,TZZB)                                  722   135
184  00468CI          CALL WALVAL(1.0,2,LT,2,MT,2,NT,TYXE)                                  723   136
185  0046E4I          CALL WALVAL(1.0,2,LT,2,MT,2,NT,TYXW)                                  724   137
186  00473CI          CALL WALVAL(1.0,2,LT,2,MT,2,NT,TYZT)                                  725   138
187  004794I          CALL WALVAL(1.0,2,LT,2,MT,2,NT,TYZB)                                  726   139
188  0047ECI          CALL WALVAL(1.0,2,LT,2,MT,2,NT,TXYN)                                  727   140
189  004844I          CALL WALVAL(1.0,2,LT,2,MT,2,NT,TXYS)                                  728   141
190  00489CI          CALL WALVAL(1.0,2,LT,2,MT,2,NT,TXZT)                                  729   142
191  0048F4I          CALL WALVAL(1.0,2,LT,2,MT,2,NT,TXZB)                                  730   143
192  00494CI          CALL WALVAL(1.0,2,LT,2,MT,2,NT,TZXE)                                  731   144
193  0049A4I          CALL WALVAL(1.0,2,LT,2,MT,2,NT,TZXW)                                  732   145
194  0049FCI          CALL WALVAL(1.0,2,LT,2,MT,2,NT,TZYN)                                  733   146
195  004A54I          CALL WALVAL(1.0,2,LT,2,MT,2,NT,TZYS)                                  734   147
196  004AACI          RETURN                                                                735   148
197  004AB4I          END                                                                   736   149

NO  ERRORS:F7D  R05-01.0C  SUBROUTINE  TRANF     02/21/86  09:50:44  TABLE SPACE:  10 KB
STATEMENT BUFFER:   20 LINES/1321 BYTES    STACK SPACE: 199 WORDS
SINGLE PRECISION FLOATING PT SUPPORT REQUIRED FOR EXECUTION
```

```
0000000I         SUBROUTINE INIT                                                     737
0000004I         COMMON                                                              738
                1/VAR/U(21,18,10),V(21,18,10),P(21,18,10),DK(21,18,10),             739
                2  DE(21,18,10),ERRU,ERRV,ERRM,ERRK,ERRE,ERRW,                       740
                3  PF(21,18,10),W(21,18,10),TM(21,18,10)                            741
                1/PRCP/ VISE(21,18,10),DEN(21,18,10),VISC,DENIN,FLOWIN               742
                1/PCOR/ DU(21,18,10),DV(21,18,10),DW(21,18,10)                       743
0000004I         COMMON                                                              744
                1/LIMT/ L,M,LT,MT,L1,L2,M1,M2,LO,MO,ISWU,ISWV,ISWP,ISWK,ISWE,        745
                2    ALU,ALV,ALP,ALK,ALE,ALVIS,ALW,N,N1,N2,NO,ISWW,IG,NT,ALC,DTT     746
          C-----INITIALIZE VARIABLES                                                 747
0000004I         DO 10 I=1,LO                                                        748     1
0000018I         DO 10 J=1,MO                                                        749     2
000002CI         DO 10 K=1,NO                                                        750     3
0000040I         U(I,J,K)=0.0                                                        751     4
000006CI         V(I,J,K)=0.0                                                        752     5
0000098I         W(I,J,K)=0.0                                                        753     6
00000C4I         P(I,J,K)=0.0                                                        754     7
00000F0I         PP(I,J,K)=0.0                                                       755     8
000011CI         DK(I,J,K)=0.0                                                       756     9
0000146I         DE(I,J,K)=0.0                                                       757    10
0000174I         DEN(I,J,K)=DENIN                                                    758    11
00001A0I         VISE(I,J,K)=VISC                                                    759    12
00001CCI     10  CONTINUE                                                            760    13
0000214I         RETURN                                                              761    14
000021AI         END                                                                 762    15

NO  ERRORS:F70  R05-01.0C  SUBROUTINE  INIT      02/21/86  09:51:07  TABLE SPACE:   3 KB
STATEMENT BUFFER:   20 LINES/1321 BYTES    STACK SPACE: 126 WORDS
SINGLE PRECISION FLOATING PT SUPPORT REQUIRED FOR EXECUTION
```

```
000000I          SUBROUTINE NEWVIS                                                    763
000004I          COMMON                                                               764
                1/VAR/U(21,18,10),V(21,18,10),P(21,18,10),DK(21,18,10),              765
                2  DE(21,18,10),ERRU,ERRV,ERRM,ERRK,ERRE,ERRW,                       766
                3  PP(21,18,10),W(21,18,10),TM(21,18,10)                             767
                1/PROP/ VISE(21,18,10),DEN(21,18,10),VISC,DENIN,FLOWIN               768
                1/PCOR/ CU(21,18,10),DV(21,18,10),DW(21,18,10)                       769
                1/TUR/ SIGK,SIGE,CMU,C1,C2,CMU1,CMU2,E,CK,HINUM,SMNUM,ANV1(800),     770
                2 YN(800),YN1(800),SINX(800),SINY(800),SINZ(800),ANW1(800),          771
                3 YPLN(800),TAUN(800),IBC(800),JBC(800),KBC(800),IITY(800),          772
                4 TALW(800),GEN(21,18,10),MC(21,18,10),IJLO(21,18,10),IITO           773
000004I          COMMON                                                               774
                1/LIMT/ L,M,LT,MT,L1,L2,M1,M2,LO,MO,ISWU,ISWV,ISWP,ISWK,ISWE,        775
                2    ALU,ALV,ALP,ALK,ALE,ALVIS,ALW,N,N1,N2,NO,ISWW,IG,NT,ALC,DTT     776
          C-----EVALUATE TURBULENT VISCOSITY                                          777
000004I          DO 10 I=1,L                                                          778      1
000018I          DO 10 J=1,M                                                          779      2
00002CI          DO 10 K=1,N                                                          780      3
000040I          IF(DK(I,J,K) .LE. SMNUM) DK(I,J,K)=SMNUM                             781      4
00009EI          IF(DE(I,J,K) .LE. SMNUM) DE(I,J,K)=SMNUM                             782      6
0000FCI          IF(DE(I,J,K) .LE. SMNUM) GO TO 12                                    783      8
000134I          TURVIS=DEN(I,J,K)*CMU*DK(I,J,K)**2/DE(I,J,K)+VISC                    784     10
0001C2I          GO TO 14                                                             785     11
0001C8I       12 TURVIS=VISC                                                          786     12
0001D4I       14 CONTINUE                                                             787     13
0001D4I          VISE(I,J,K)=VISE(I,J,K)+ALVIS*(TURVIS-VISE(I,J,K))                   788     14
000252I       10 CONTINUE                                                             789     15
00029AI          RETURN                                                               790     16
0002A0I          END                                                                  791     17

NO  ERRORS:F70  R05-01.0C  SUBROUTINE  NEWVIS    02/21/86  09:51:23  TABLE SPACE:   5 KB
STATEMENT BUFFER:   20 LINES/1321 BYTES    STACK SPACE: 203 WORDS
SINGLE PRECISION FLOATING PT SUPPORT REQUIRED FOR EXECUTION
```

```
000000I         SUBROUTINE SOLVEQ(IE,ISWF,ALF,SIGF,ERRF,F,F0)                    792
000004I         DIMENSION F(21,18,10),F1(21,18,10),F0(21,18,10)                  793
000004I         COMMON                                                           794
               1/VAR/U(21,18,10),V(21,18,10),P(21,18,10),DK(21,18,10),           795
               2  DE(21,18,10),ERRU,ERRV,ERRM,ERRK,ERRE,ERRW,                     796
               3  PP(21,18,10),W(21,18,10),TM(21,18,10)                          797
               1/PRCP/ VISE(21,18,10),DEN(21,18,10),VISC,DENIN,FLOWIN            798
               1/PCCR/ CU(21,18,10),DV(21,18,10),DW(21,18,10)                    799
               1/TUR/ SIGK,SIGE,CMU,C1,C2,CMU1,CMU2,E,CK,HINUM,SMNUM,ANV1(800),  800
               2 YN(800),YN1(800),SINX(800),SINY(800),SINZ(800),ANW1(800),       801
               3 YPLN(800),TAUN(800),IBC(800),JBC(800),KBC(800),IITY(800),       802
               4 TAUW(800),GEN(21,18,10),MC(21,18,10),IJLO(21,18,10),IITO        803
               1/COEF/ AP(21,18,10),SU(21,18,10),SP(21,18,10),SUK(21,18,10),     804
               2    SPK(21,18,10),AE(21,18,10),AW(21,18,10),AN(21,18,10),        805
               3    AS(21,13,10),AT(21,18,10),AB(21,18,10),APO(21,18,10)         806
000004I         COMMON                                                           807
               1/TRAN/ X(21,18,10),Y(21,18,10),Z(21,18,10),TJO(21,18,10),        808
               2    CX(21,18,10),CY(21,18,10),CZ(21,18,10),                      809
               3    EX(21,18,10),EY(21,18,10),EZ(21,18,10),                      810
               3    SX(21,18,10),SY(21,18,10),SZ(21,18,10)                       811
               1/LIMT/ L,M,LT,MT,L1,L2,M1,M2,LO,MO,ISWU,ISWV,ISWP,ISWK,ISWE,     812
               2    ALU,ALV,ALP,ALK,ALE,ALVIS,ALW,N,N1,N2,NO,ISWW,IG,NT,ALC,DTT  813
000004I         COMMON                                                           814
               1/TTRAN/TXXE(21,18,10),TXXW(21,18,10),TYYN(21,18,10),             815
               2    TYYS(21,13,10),TZZT(21,18,10),TZZB(21,18,10),                816
               3    TYXE(21,18,10),TYXW(21,18,10),TYZT(21,18,10),                817
               4    TYZB(21,18,10),TXYN(21,18,10),TXYS(21,18,10),                818
               5    TXZT(21,18,10),TXZB(21,18,10),TZXE(21,18,10),                819
               6    TZXW(21,18,10),TZYN(21,18,10),TZYS(21,18,10)                 820
          C------TRANSPORT EQUATIONS LINERAIZATION AND SOLVER                    821
000004I         ERRF=0.0                                                         822    1
003330I         PI=3.141592654                                                   823    2
          C-----PRESSURE CORRECTION SOLVER STARTS FROM 10                        824
00383CI         IF(IE .EQ. 0) GO TO 10                                           825    3
          C-----U, V, W, TM, K & E EQUATIONS                                     826
003852I         IS=2                                                             827    5
00385AI         IT=LT                                                            828    6
003866I         JS=2                                                             829    7
003B6EI         JT=MT                                                            830    8
00387AI         KS=2                                                             831    9
003382I         KT=NT                                                            832   10
00388EI         GO TO (21,29,29,29,29,21,21,21), IE                              833   11
0038CAI      21 CONTINUE                                                         834   12
0038CAI         DO 22 I=IS,IT                                                    835   13
003BE2I         DO 22 J=JS,JT                                                    836   14
0038FAI         DO 22 K=KS,KT                                                    837   15
003C12I      22 F1(I,J,K)=VISE(I,J,K)/SIGF                                       838   16
          C-----EVALUATE LINK COEFF. AND SOURCE TERMS                            839
003CACI         DO 20 I=IS,IT                                                    840   17
003CC4I         DO 20 J=JS,JT                                                    841   18
003CDCI         DO 20 K=KS,KT                                                    842   19
003CF4I         GAE=0.5*(F1(I+1,J,K)+F1(I,J,K))                                  843   20
003D50I         GAW=0.5*(F1(I-1,J,K)+F1(I,J,K))                                  844   21
003DACI         GAN=0.5*(F1(I,J+1,K)+F1(I,J,K))                                  845   22
003E08I         GAS=0.5*(F1(I,J-1,K)+F1(I,J,K))                                  846   23
003E64I         GAT=0.5*(F1(I,J,K+1)+F1(I,J,K))                                  847   24
003EBEI         GAB=0.5*(F1(I,J,K-1)+F1(I,J,K))                                  848   25
```

```
003F18I        DENE=0.5*(DEN(I+1,J,K)+DEN(I,J,K))                          26
003F74I        DENW=0.5*(DEN(I-1,J,K)+DEN(I,J,K))                          27
003FD0I        DENN=0.5*(DEN(I,J+1,K)+DEN(I,J,K))                          28
00402CI        DENS=0.5*(DEN(I,J-1,K)+DEN(I,J,K))                          29
004088I        DENT=0.5*(DEN(I,J,K+1)+DEN(I,J,K))                          30
0040E2I        DENB=0.5*(DEN(I,J,K-1)+DEN(I,J,K))                          31
00413CI        UE=C.5*(U(I+1,J,K)+U(I,J,K))                                32
004198I        UW=C.5*(U(I-1,J,K)+U(I,J,K))                                33
0041F4I        UN=C.5*(U(I,J+1,K)+U(I,J,K))                                34
004250I        US=C.5*(U(I,J-1,K)+U(I,J,K))                                35
0042ACI        UT=C.5*(U(I,J,K+1)+U(I,J,K))                                36
004306I        UB=0.5*(U(I,J,K-1)+U(I,J,K))                                37
004360I        VE=0.5*(V(I+1,J,K)+V(I,J,K))                                38
00433CI        VW=C.5*(V(I-1,J,K)+V(I,J,K))                                39
004418I        VN=C.5*(V(I,J+1,K)+V(I,J,K))                                40
004474I        VS=C.5*(V(I,J-1,K)+V(I,J,K))                                41
0044D0I        VT=C.5*(V(I,J,K+1)+V(I,J,K))                                42
00452AI        VB=C.5*(V(I,J,K-1)+V(I,J,K))                                43
004584I        WE=C.5*(W(I+1,J,K)+W(I,J,K))                                44
0045E0I        WW=C.5*(W(I-1,J,K)+W(I,J,K))                                45
00463CI        WN=C.5*(W(I,J+1,K)+W(I,J,K))                                46
004698I        WS=C.5*(W(I,J-1,K)+W(I,J,K))                                47
0046F4I        WT=C.5*(W(I,J,K+1)+W(I,J,K))                                48
00474EI        WB=C.5*(W(I,J,K-1)+W(I,J,K))                                49
0047A8I        CE=DENE*(UE*CX(I,J,K)+VE*CY(I,J,K)+WE*CZ(I,J,K))            50
00483CI        CW=DENW*(UW*CX(I,J,K)+VW*CY(I,J,K)+WW*CZ(I,J,K))            51
0048D0I        CN=DENN*(UN*EX(I,J,K)+VN*EY(I,J,K)+WN*EZ(I,J,K))            52
004964I        CS=DENS*(US*EX(I,J,K)+VS*EY(I,J,K)+WS*EZ(I,J,K))            53
0049F8I        CT=DENT*(UT*SX(I,J,K)+VT*SY(I,J,K)+WT*SZ(I,J,K))            54
004A8CI        CB=DENB*(UB*SX(I,J,K)+VB*SY(I,J,K)+WB*SZ(I,J,K))            55
004B20I        DDE=GAE*TXXE(I,J,K)+GAN*TXYN(I,J,K)-GAS*TXYS(I,J,K)+        56
              1     GAT*TXZT(I,J,K)-GAB*TXZB(I,J,K)
004C0AI        DDW=GAW*TXXW(I,J,K)+GAS*TXYS(I,J,K)-GAN*TXYN(I,J,K)-        57
              1     GAT*TXZT(I,J,K)+GAB*TXZB(I,J,K)
004CF4I        DDN=GAN*TYYN(I,J,K)+GAE*TYXE(I,J,K)-GAW*TYXW(I,J,K)+        58
              1     GAT*TYZT(I,J,K)-GAB*TYZB(I,J,K)
004DDEI        DDS=GAS*TYYS(I,J,K)+GAW*TYXW(I,J,K)-GAE*TYXE(I,J,K)-        59
              1     GAT*TYZT(I,J,K)+GAB*TYZB(I,J,K)
004EC8I        DDT=GAT*TZZT(I,J,K)+GAE*TZXE(I,J,K)-GAW*TZXW(I,J,K)+        60
              1     GAN*TZYN(I,J,K)-GAS*TZYS(I,J,K)
004FB2I        DDB=GAB*TZZB(I,J,K)+GAW*TZXW(I,J,K)-GAE*TZXE(I,J,K)-        61
              1     GAN*TZYN(I,J,K)+GAS*TZYS(I,J,K)
00509CI        CPO=ABS(CE-CW+CN-CS+CT-CB)                                  62
0050E2I        AE(I,J,K)=(AMAX1(ABS(0.5*CE),DDE)-0.5*CE)*TJO(I,J,K)        63
005186I        AW(I,J,K)=(AMAX1(ABS(0.5*CW),DDW)+0.5*CW)*TJO(I,J,K)        64
00522AI        AN(I,J,K)=(AMAX1(ABS(0.5*CN),DDN)-0.5*CN)*TJO(I,J,K)        65
0052CEI        AS(I,J,K)=(AMAX1(ABS(0.5*CS),DDS)+0.5*CS)*TJO(I,J,K)        66
005372I        AT(I,J,K)=(AMAX1(ABS(0.5*CT),DDT)-0.5*CT)*TJO(I,J,K)        67
005416I        AB(I,J,K)=(AMAX1(ABS(0.5*CB),DDB)+0.5*CB)*TJO(I,J,K)        68
0054BAI        DDNE=GAE*TYXE(I,J,K)+GAN*TXYN(I,J,K)                        69
00551AI        DDSE=-GAE*TYXE(I,J,K)-GAS*TXYS(I,J,K)                       70
005582I        DDNW=-GAW*TYXW(I,J,K)-GAN*TXYN(I,J,K)                       71
0055EAI        DDSW=GAW*TYXW(I,J,K)+GAS*TXYS(I,J,K)                        72
00564AI        DDET=GAE*TZXE(I,J,K)+GAT*TXZT(I,J,K)                        73
0056AAI        DDEB=-GAE*TZXE(I,J,K)-GAB*TXZB(I,J,K)                       74
005712I        DDWT=-GAW*TZXW(I,J,K)-GAT*TXZT(I,J,K)                       75
00577AI        DDWB=GAW*TZXW(I,J,K)+GAB*TXZB(I,J,K)                        76
```

```
0057DAI         DDNT=GAN*TZYN(I,J,K)+GAT*TYZT(I,J,K)                                        77
00583AI         DDNB=-GAN*TZYN(I,J,K)-GAB*TYZB(I,J,K)                                       78
0058A2I         DDST=-GAS*TZYS(I,J,K)-GAT*TYZT(I,J,K)                                       79
00590AI         DDSB=GAS*TZYS(I,J,K)+GAB*TYZB(I,J,K)                                        80
00596AI         SU(I,J,K)=CPO*F(I,J,K)+DDNE*F(I+1,J+1,K)+DDSE*F(I+1,J-1,K)+                 81
               1          DDNW*F(I-1,J+1,K)+DDSW*F(I-1,J-1,K)+
               2          DDET*F(I+1,J,K+1)+DDEB*F(I+1,J,K-1)+
               3          DDWT*F(I-1,J,K+1)+DDWB*F(I-1,J,K-1)+
               4          DDNT*F(I,J+1,K+1)+DDNB*F(I,J+1,K-1)+
               5          DDST*F(I,J-1,K+1)+DDSB*F(I,J-1,K-1)
005C82I         DV(I,J,K)=CPO*V(I,J,K)+DDNE*V(I+1,J+1,K)+DDSE*V(I+1,J-1,K)+                 82
               1          DDNW*V(I-1,J+1,K)+DDSW*V(I-1,J-1,K)+
               2          DDET*V(I+1,J,K+1)+DDEB*V(I+1,J,K-1)+
               3          DDWT*V(I-1,J,K+1)+DDWB*V(I-1,J,K-1)+
               4          DDNT*V(I,J+1,K+1)+DDNB*V(I,J+1,K-1)+
               5          DDST*V(I,J-1,K+1)+DDSB*V(I,J-1,K-1)
005F4CI         DW(I,J,K)=CPO*W(I,J,K)+DDNE*W(I+1,J+1,K)+DDSE*W(I+1,J-1,K)+                 83
               1          DDNW*W(I-1,J+1,K)+DDSW*W(I-1,J-1,K)+
               2          DDET*W(I+1,J,K+1)+DDEB*W(I+1,J,K-1)+
               3          DDWT*W(I-1,J,K+1)+DDWB*W(I-1,J,K-1)+
               4          DDNT*W(I,J+1,K+1)+DDNB*W(I,J+1,K-1)+
               5          DDST*W(I,J-1,K+1)+DDSB*W(I,J-1,K-1)
006216I         PP(I,J,K)=CPO*DK(I,J,K)+DDNE*DK(I+1,J+1,K)+DDSE*DK(I+1,J-1,K)+              84
               1          DDNW*DK(I-1,J+1,K)+DDSW*DK(I-1,J-1,K)+
               2          DDET*DK(I+1,J,K+1)+DDEB*DK(I+1,J,K-1)+
               3          DDWT*DK(I-1,J,K+1)+DDWB*DK(I-1,J,K-1)+
               4          DDNT*DK(I,J+1,K+1)+DDNB*DK(I,J+1,K-1)+
               5          DDST*DK(I,J-1,K+1)+DDSB*DK(I,J-1,K-1)
0064E0I         SUK(I,J,K)=CPO                                                              85
00650CI         SPK(I,J,K)=-CPO-APC(I,J,K)                                                  86
006564I         UCXI=UE-UW                                                                  87
006576I         UEDA=UN-US                                                                  88
006588I         USCI=UT-UB                                                                  89
00659AI         VCXI=VE-VW                                                                  90
0065ACI         VEDA=VN-VS                                                                  91
0065BEI         VSCI=VT-VB                                                                  92
0065D0I         WCXI=WE-WW                                                                  93
0065E2I         WEDA=WN-WS                                                                  94
0065F4I         WSCI=WT-WB                                                                  95
006606I         UX=UCXI*CX(I,J,K)+UEDA*EX(I,J,K)+USCI*SX(I,J,K)                             96
006694I         UY=UCXI*CY(I,J,K)+UEDA*EY(I,J,K)+USCI*SY(I,J,K)                             97
006722I         UZ=UCXI*CZ(I,J,K)+UEDA*EZ(I,J,K)+USCI*SZ(I,J,K)                             98
0067B0I         VX=VCXI*CX(I,J,K)+VEDA*EX(I,J,K)+VSCI*SX(I,J,K)                             99
00683EI         VY=VCXI*CY(I,J,K)+VEDA*EY(I,J,K)+VSCI*SY(I,J,K)                            100
0068CCI         VZ=VCXI*CZ(I,J,K)+VEDA*EZ(I,J,K)+VSCI*SZ(I,J,K)                            101
00695AI         WX=WCXI*CX(I,J,K)+WEDA*EX(I,J,K)+WSCI*SX(I,J,K)                            102
0069E8I         WY=WCXI*CY(I,J,K)+WEDA*EY(I,J,K)+WSCI*SY(I,J,K)                            103
006A76I         WZ=WCXI*CZ(I,J,K)+WEDA*EZ(I,J,K)+WSCI*SZ(I,J,K)                            104
006B04I         GEN(I,J,K)=VISE(I,J,K)*((UY+VX)**2+(VZ+WY)**2+(WX+UZ)**2+                  105
               1          2*(UX*UX+VY*VY+WZ*WZ))
006BE6I      20 CONTINUE                                                                   106
           C------------------------------------------------------------------------
006C2EI      29 CONTINUE                                                                   107
           C-----CALCULATE SOURCE TERMS
006C2EI         GO TO (1,2,3,4,5,6,7), IE                                                  108
           C-----U-, V-, W-SOURCES
006C56I       1 CONTINUE                                                                   109
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
172  006C66I        DO 15 I=IS,IT                                                    963    110
173  006C7EI        DO 15 J=JS,JT                                                    964    111
174  006C96I        DO 15 K=KS,KT                                                    965    112
175  006CAEI        GAE=0.5*(F1(I+1,J,K)+F1(I,J,K))                                  966    113
176  006D0AI        GAW=0.5*(F1(I-1,J,K)+F1(I,J,K))                                  967    114
177  006D66I        GAN=0.5*(F1(I,J+1,K)+F1(I,J,K))                                  968    115
178  006DC2I        GAS=0.5*(F1(I,J-1,K)+F1(I,J,K))                                  969    116
179  006E1EI        GAT=0.5*(F1(I,J,K+1)+F1(I,J,K))                                  970    117
180  006E78I        GAB=0.5*(F1(I,J,K-1)+F1(I,J,K))                                  971    118
181  006E02I        PCXI=(P(I+1,J,K)+P(I+1,J+1,K)+P(I+1,J,K+1)+P(I+1,J+1,K+1)-       972    119
182                1      P(I,J,K)-P(I,J+1,K)-P(I,J,K+1)-P(I,J+1,K+1))*0.25          973
183  007036I        PEDA=(P(I,J+1,K)+P(I+1,J+1,K)+P(I,J+1,K+1)+P(I+1,J+1,K+1)-       974    120
184                1      P(I,J,K)-P(I+1,J,K)-P(I,J,K+1)-P(I+1,J,K+1))*C.25          975
185  00719AI        PSCI=(P(I,J,K+1)+P(I+1,J,K+1)+P(I,J+1,K+1)+P(I+1,J+1,K+1)-       976    121
186                1      P(I,J,K)-P(I+1,J,K)-P(I,J+1,K)-P(I+1,J+1,K))*C.25          977
187  0072FEI        UCE=U(I+1,J,K)-U(I,J,K)                                          978    122
188  007354I        UCW=U(I,J,K)-U(I-1,J,K)                                          979    123
189  0073AAI        UCN=(U(I+1,J+1,K)+U(I+1,J,K)-U(I-1,J+1,K)-U(I-1,J,K))*0.25       980    124
190  007456I        UCS=(U(I+1,J,K)+U(I+1,J-1,K)-U(I-1,J,K)-U(I-1,J-1,K))*0.25       981    125
191  007522I        UCT=(U(I+1,J,K+1)+U(I+1,J,K)-U(I-1,J,K+1)-U(I-1,J,K))*0.25       982    126
192  0075DAI        UCB=(U(I+1,J,K)+U(I+1,J,K-1)-U(I-1,J,K)-U(I-1,J,K-1))*0.25       983    127
193  007692I        UEE=(U(I+1,J+1,K)+U(I,J+1,K)-U(I+1,J-1,K)-U(I,J-1,K))*0.25       984    128
194  00774EI        UEW=(U(I,J+1,K)+U(I-1,J+1,K)-U(I,J-1,K)-U(I-1,J-1,K))*0.25       985    129
195  00780AI        UEN=U(I,J+1,K)-U(I,J,K)                                          986    130
196  007860I        UES=U(I,J,K)-U(I,J-1,K)                                          987    131
197  007886I        UET=(U(I,J+1,K+1)+U(I,J+1,K)-U(I,J-1,K+1)-U(I,J-1,K))*0.25       988    132
198  00796EI        UEB=(U(I,J+1,K)+U(I,J+1,K-1)-U(I,J-1,K)-U(I,J-1,K-1))*0.25       989    133
199  007A26I        USE=(U(I+1,J,K+1)+U(I,J,K+1)-U(I+1,J,K-1)-U(I,J,K-1))*0.25       990    134
200  007ADAI        USW=(U(I,J,K+1)+U(I-1,J,K+1)-U(I,J,K-1)-U(I-1,J,K-1))*0.25       991    135
201  007B3EI        USN=(U(I,J+1,K+1)+U(I,J,K+1)-U(I,J+1,K-1)-U(I,J,K-1))*0.25       992    136
202  007C42I        USS=(U(I,J,K+1)+U(I,J-1,K+1)-U(I,J,K-1)-U(I,J-1,K-1))*0.25       993    137
203  007CF6I        UST=U(I,J,K+1)-U(I,J,K)                                          994    138
204  007D4AI        USB=U(I,J,K)-U(I,J,K-1)                                          995    139
205  007D9EI        VCE=V(I+1,J,K)-V(I,J,K)                                          996    140
206  007DF4I        VCW=V(I,J,K)-V(I-1,J,K)                                          997    141
207  007E4AI        VCN=(V(I+1,J+1,K)+V(I+1,J,K)-V(I-1,J+1,K)-V(I-1,J,K))*0.25       998    142
208  007F06I        VCS=(V(I+1,J,K)+V(I+1,J-1,K)-V(I-1,J,K)-V(I-1,J-1,K))*0.25       999    143
209  007FC2I        VCT=(V(I+1,J,K+1)+V(I+1,J,K)-V(I-1,J,K+1)-V(I-1,J,K))*0.25      1000    144
210  00807AI        VCB=(V(I+1,J,K)+V(I+1,J,K-1)-V(I-1,J,K)-V(I-1,J,K-1))*0.25      1001    145
211  008132I        VEE=(V(I+1,J+1,K)+V(I,J+1,K)-V(I+1,J-1,K)-V(I,J-1,K))*0.25      1002    146
212  0081EEI        VEW=(V(I,J+1,K)+V(I-1,J+1,K)-V(I,J-1,K)-V(I-1,J-1,K))*0.25      1003    147
213  0082AAI        VEN=V(I,J+1,K)-V(I,J,K)                                         1004    148
214  008300I        VES=V(I,J,K)-V(I,J-1,K)                                         1005    149
215  008356I        VET=(V(I,J+1,K+1)+V(I,J+1,K)-V(I,J-1,K+1)-V(I,J-1,K))*0.25      1006    150
216  00840EI        VEB=(V(I,J+1,K)+V(I,J+1,K-1)-V(I,J-1,K)-V(I,J-1,K-1))*0.25      1007    151
217  0084C6I        VSE=(V(I+1,J,K+1)+V(I,J,K+1)-V(I+1,J,K-1)-V(I,J,K-1))*0.25      1008    152
218  00857AI        VSW=(V(I,J,K+1)+V(I-1,J,K+1)-V(I,J,K-1)-V(I-1,J,K-1))*0.25      1009    153
219  00862EI        VSN=(V(I,J+1,K+1)+V(I,J,K+1)-V(I,J+1,K-1)-V(I,J,K-1))*0.25      1010    154
220  0086E2I        VSS=(V(I,J,K+1)+V(I,J-1,K+1)-V(I,J,K-1)-V(I,J-1,K-1))*0.25      1011    155
221  008796I        VST=V(I,J,K+1)-V(I,J,K)                                         1012    156
222  0087EAI        VSB=V(I,J,K)-V(I,J,K-1)                                         1013    157
223  00883EI        WCE=W(I+1,J,K)-W(I,J,K)                                         1014    158
224  008894I        WCW=W(I,J,K)-W(I-1,J,K)                                         1015    159
225  0088EAI        WCN=(W(I+1,J+1,K)+W(I+1,J,K)-W(I-1,J+1,K)-W(I-1,J,K))*0.25      1016    160
226  0089A6I        WCS=(W(I+1,J,K)+W(I+1,J-1,K)-W(I-1,J,K)-W(I-1,J-1,K))*0.25      1017    161
227  008A62I        WCT=(W(I+1,J,K+1)+W(I+1,J,K)-W(I-1,J,K+1)-W(I-1,J,K))*0.25      1018    162
228  008B1AI        WCB=(W(I+1,J,K)+W(I+1,J,K-1)-W(I-1,J,K)-W(I-1,J,K-1))*0.25      1019    163
```

```
229  0088D2I   WEE=(W(I+1,J+1,K)+W(I,J+1,K)-W(I+1,J-1,K)-W(I,J-1,K))*0.25        1020  164
230  008C8EI   WEW=(W(I,J+1,K)+W(I-1,J+1,K)-W(I,J-1,K)-W(I-1,J-1,K))*0.25        1021  165
231  008D4AI   WEN=W(I,J+1,K)-W(I,J,K)                                           1022  166
232  008DA0I   WES=W(I,J,K)-W(I,J-1,K)                                           1023  167
233  008DF6I   WET=(W(I,J+1,K+1)+W(I,J+1,K)-W(I,J-1,K+1)-W(I,J-1,K))*0.25        1024  168
234  008EAEI   WEB=(W(I,J+1,K)+W(I,J+1,K-1)-W(I,J-1,K)-W(I,J-1,K-1))*0.25        1025  169
235  008F66I   WSE=(W(I+1,J,K+1)+W(I,J,K+1)-W(I+1,J,K-1)-W(I,J,K-1))*0.25        1026  170
236  00901AI   WSW=(W(I,J,K+1)+W(I-1,J,K+1)-W(I,J,K-1)-W(I-1,J,K-1))*0.25        1027  171
237  0090CEI   WSN=(W(I,J+1,K+1)+W(I,J,K+1)-W(I,J+1,K-1)-W(I,J,K-1))*0.25        1028  172
238  009182I   WSS=(W(I,J,K+1)+W(I,J-1,K+1)-W(I,J,K-1)-W(I,J-1,K-1))*0.25        1029  173
239  009236I   WST=W(I,J,K+1)-W(I,J,K)                                           1030  174
240  00928AI   WSB=W(I,J,K)-W(I,J,K-1)                                           1031  175
241  0092DEI   SU(I,J,K)=SU(I,J,K)-PCXI*CX(I,J,K)-PEDA*EX(I,J,K)-PSCI*SX(I,J,K)  1032  176
242  0093B4I   DV(I,J,K)=DV(I,J,K)-PCXI*CY(I,J,K)-PEDA*EY(I,J,K)-PSCI*SY(I,J,K)  1033  177
243  00948AI   DW(I,J,K)=DW(I,J,K)-PCXI*CZ(I,J,K)-PEDA*EZ(I,J,K)-PSCI*SZ(I,J,K)  1034  178
244  009560I   CXE=(CX(I+1,J,K)+CX(I,J,K))*0.5                                   1035  179
245  0095BCI   CXW=(CX(I-1,J,K)+CX(I,J,K))*0.5                                   1036  180
246  009618I   CXN=(CX(I,J+1,K)+CX(I,J,K))*0.5                                   1037  181
247  009674I   CXS=(CX(I,J,K)+CX(I,J-1,K))*0.5                                   1038  132
248  0096D0I   CXT=(CX(I,J,K+1)+CX(I,J,K))*0.5                                   1039  183
249  00972AI   CXB=(CX(I,J,K)+CX(I,J,K-1))*0.5                                   1040  184
250  009784I   EXE=(EX(I+1,J,K)+EX(I,J,K))*0.5                                   1041  185
251  0097E0I   EXW=(EX(I,J,K)+EX(I-1,J,K))*0.5                                   1042  186
252  00983CI   EXN=(EX(I,J+1,K)+EX(I,J,K))*0.5                                   1043  187
253  009898I   EXS=(EX(I,J,K)+EX(I,J-1,K))*0.5                                   1044  188
254  0098F4I   EXT=(EX(I,J,K+1)+EX(I,J,K))*0.5                                   1045  189
255  00994EI   EXB=(EX(I,J,K)+EX(I,J,K-1))*0.5                                   1046  190
256  0099A8I   SXE=(SX(I+1,J,K)+SX(I,J,K))*0.5                                   1047  191
257  009A04I   SXW=(SX(I,J,K)+SX(I-1,J,K))*0.5                                   1048  192
258  009A60I   SXN=(SX(I,J+1,K)+SX(I,J,K))*0.5                                   1049  193
259  009ABCI   SXS=(SX(I,J,K)+SX(I,J-1,K))*0.5                                   1050  194
260  009B18I   SXT=(SX(I,J,K+1)+SX(I,J,K))*0.5                                   1051  195
261  009B72I   SXB=(SX(I,J,K)+SX(I,J,K-1))*0.5                                   1052  196
262  009BCCI   QE=GAE*(UCE*CXE+UEE*EXE+USE*SXE)                                  1053  197
263  009C00I   QW=GAW*(UCW*CXW+UEW*EXW+USW*SXW)                                  1054  198
264  009C34I   QN=GAN*(UCN*CXN+UEN*EXN+USN*SXN)                                  1055  199
265  009C68I   QS=GAS*(UCS*CXS+UES*EXS+USS*SXS)                                  1056  200
266  009C9CI   QT=GAT*(UCT*CXT+UET*EXT+UST*SXT)                                  1057  201
267  009CD0I   QB=GAB*(UCB*CXB+UEB*EXB+USB*SXB)                                  1058  202
268  009D04I   SOC1=CX(I,J,K)*(QE-QW)+EX(I,J,K)*(QN-QS)+SX(I,J,K)*(QT-QB)        1059  203
269  009DA4I   QE=GAE*(VCE*CXE+VEE*EXE+VSE*SXE)                                  1060  204
270  009DD8I   QW=GAW*(VCW*CXW+VEW*EXW+VSW*SXW)                                  1061  205
271  009E0CI   QN=GAN*(VCN*CXN+VEN*EXN+VSN*SXN)                                  1062  206
272  009E40I   QS=GAS*(VCS*CXS+VES*EXS+VSS*SXS)                                  1063  207
273  009E74I   QT=GAT*(VCT*CXT+VET*EXT+VST*SXT)                                  1064  208
274  009EA8I   QB=GAB*(VCB*CXB+VEB*EXB+VSB*SXB)                                  1065  209
275  009EDCI   SOC2=CY(I,J,K)*(QE-QW)+EY(I,J,K)*(QN-QS)+SY(I,J,K)*(QT-QB)        1066  210
276  009F7CI   QE=GAE*(WCE*CXE+WEE*EXE+WSE*SXE)                                  1067  211
277  009FB0I   QW=GAW*(WCW*CXW+WEW*EXW+WSW*SXW)                                  1068  212
278  009FE4I   QN=GAN*(WCN*CXN+WEN*EXN+WSN*SXN)                                  1069  213
279  00A018I   QS=GAS*(WCS*CXS+WES*EXS+WSS*SXS)                                  1070  214
280  00A04CI   QT=GAT*(WCT*CXT+WET*EXT+WST*SXT)                                  1071  215
281  00A080I   QB=GAB*(WCB*CXB+WEB*EXB+WSB*SXB)                                  1072  216
282  00A0B4I   SOC3=CZ(I,J,K)*(QE-QW)+EZ(I,J,K)*(QN-QS)+SZ(I,J,K)*(QT-QB)        1073  217
283  00A154I   SU(I,J,K)=SU(I,J,K)+SOC1+SOC2+SOC3+APO(I,J,K)*FO(I,J,K)          1074  218
284  00A206I   CYE=(CY(I+1,J,K)+CY(I,J,K))*0.5                                   1075  219
285  00A262I   CYW=(CY(I,J,K)+CY(I-1,J,K))*0.5                                   1076  220
```

```
286  00A2BEI     CYN=(CY(I,J+1,K)+CY(I,J,K))*0.5                                  1077   221
287  00A31AI     CYS=(CY(I,J,K)+CY(I,J-1,K))*0.5                                  1078   222
288  00A376I     CYT=(CY(I,J,K+1)+CY(I,J,K))*0.5                                  1079   223
289  00A3D0I     CYB=(CY(I,J,K)+CY(I,J,K-1))*0.5                                  1080   224
290  00A42AI     EYE=(EY(I+1,J,K)+EY(I,J,K))*0.5                                  1081   225
291  00A486I     EYW=(EY(I,J,K)+EY(I-1,J,K))*0.5                                  1082   226
292  00A4E2I     EYN=(EY(I,J+1,K)+EY(I,J,K))*0.5                                  1083   227
293  00A53EI     EYS=(EY(I,J,K)+EY(I,J-1,K))*0.5                                  1084   228
294  00A59AI     EYT=(EY(I,J,K+1)+EY(I,J,K))*0.5                                  1085   229
295  00A5F4I     EYB=(EY(I,J,K)+EY(I,J,K-1))*0.5                                  1086   230
296  00A64EI     SYE=(SY(I+1,J,K)+SY(I,J,K))*0.5                                  1087   231
297  00A6AAI     SYW=(SY(I,J,K)+SY(I-1,J,K))*0.5                                  1088   232
298  00A706I     SYN=(SY(I,J+1,K)+SY(I,J,K))*0.5                                  1089   233
299  00A762I     SYS=(SY(I,J,K)+SY(I,J-1,K))*0.5                                  1090   234
300  00A7BEI     SYT=(SY(I,J,K+1)+SY(I,J,K))*0.5                                  1091   235
301  00A818I     SYB=(SY(I,J,K)+SY(I,J,K-1))*0.5                                  1092   236
302  00A872I     QE=GAE*(UCE*CYE+UEE*EYE+USE*SYE)                                 1093   237
303  00A8A6I     QW=GAW*(UCW*CYW+UEW*EYW+USW*SYW)                                 1094   238
304  00A8DAI     QN=GAN*(UCN*CYN+UEN*EYN+USN*SYN)                                 1095   239
305  00A90EI     QS=GAS*(UCS*CYS+UES*EYS+USS*SYS)                                 1096   240
306  00A942I     QT=GAT*(UCT*CYT+UET*EYT+UST*SYT)                                 1097   241
307  00A976I     QB=GAB*(UCB*CYB+UEB*EYB+USB*SYB)                                 1098   242
308  00A9AAI     SOC1=CX(I,J,K)*(QE-QW)+EX(I,J,K)*(QN-QS)+SX(I,J,K)*(QT-QB)       1099   243
309  00AA4AI     QE=GAE*(VCE*CYE+VEE*EYE+VSE*SYE)                                 1100   244
310  00AA7EI     QW=GAW*(VCW*CYW+VEW*EYW+VSW*SYW)                                 1101   245
311  00AAB2I     QN=GAN*(VCN*CYN+VEN*EYN+VSN*SYN)                                 1102   246
312  00AAE6I     QS=GAS*(VCS*CYS+VES*EYS+VSS*SYS)                                 1103   247
313  00AB1AI     QT=GAT*(VCT*CYT+VET*EYT+VST*SYT)                                 1104   248
314  00AB4EI     QB=GAB*(VCB*CYB+VEB*EYB+VSB*SYB)                                 1105   249
315  00AB82I     SOC2=CY(I,J,K)*(QE-QW)+EY(I,J,K)*(QN-QS)+SY(I,J,K)*(QT-QB)       1106   250
316  00AC22I     QE=GAE*(WCE*CYE+WEE*EYE+WSE*SYE)                                 1107   251
317  00AC56I     QW=GAW*(WCW*CYW+WEW*EYW+WSW*SYW)                                 1108   252
318  00AC8AI     QN=GAN*(WCN*CYN+WEN*EYN+WSN*SYN)                                 1109   253
319  00ACBEI     QS=GAS*(WCS*CYS+WES*EYS+WSS*SYS)                                 1110   254
320  00ACF2I     QT=GAT*(WCT*CYT+WET*EYT+WST*SYT)                                 1111   255
321  00AD26I     QB=GAB*(WCB*CYB+WEB*EYB+WSB*SYB)                                 1112   256
322  00AD5AI     SOC3=CZ(I,J,K)*(QE-QW)+EZ(I,J,K)*(QN-QS)+SZ(I,J,K)*(QT-QB)       1113   257
323  00ADFAI     DV(I,J,K)=DV(I,J,K)+SOC1+SOC2+SOC3                               1114   258
324  00AE58I     CZE=(CZ(I+1,J,K)+CZ(I,J,K))*0.5                                  1115   259
325  00AEB4I     CZW=(CZ(I,J,K)+CZ(I-1,J,K))*0.5                                  1116   260
326  00AF10I     CZN=(CZ(I,J+1,K)+CZ(I,J,K))*0.5                                  1117   261
327  00AF6CI     CZS=(CZ(I,J,K)+CZ(I,J-1,K))*0.5                                  1118   262
328  00AFC8I     CZT=(CZ(I,J,K+1)+CZ(I,J,K))*0.5                                  1119   263
329  00B022I     CZB=(CZ(I,J,K)+CZ(I,J,K-1))*0.5                                  1120   264
330  00B07CI     EZE=(EZ(I+1,J,K)+EZ(I,J,K))*0.5                                  1121   265
331  00B0D8I     EZW=(EZ(I,J,K)+EZ(I-1,J,K))*0.5                                  1122   266
332  00B134I     EZN=(EZ(I,J+1,K)+EZ(I,J,K))*0.5                                  1123   267
333  00B190I     EZS=(EZ(I,J,K)+EZ(I,J-1,K))*0.5                                  1124   268
334  00B1ECI     EZT=(EZ(I,J,K+1)+EZ(I,J,K))*0.5                                  1125   269
335  00B246I     EZB=(EZ(I,J,K)+EZ(I,J,K-1))*0.5                                  1126   270
336  00B2A0I     SZE=(SZ(I+1,J,K)+SZ(I,J,K))*0.5                                  1127   271
337  00B2FCI     SZW=(SZ(I,J,K)+SZ(I-1,J,K))*0.5                                  1128   272
338  00B358I     SZN=(SZ(I,J+1,K)+SZ(I,J,K))*0.5                                  1129   273
339  00B3B4I     SZS=(SZ(I,J,K)+SZ(I,J-1,K))*0.5                                  1130   274
340  00B410I     SZT=(SZ(I,J,K+1)+SZ(I,J,K))*0.5                                  1131   275
341  00B46AI     SZB=(SZ(I,J,K)+SZ(I,J,K-1))*0.5                                  1132   276
342  00B4C4I     QE=GAE*(UCE*CZE+UEE*EZE+USE*SZE)                                 1133   277
```

```
00B4F8I          QW=GAW*(UCW*CZW+UEW*EZW+USW*SZW)                                    278
00B52CI          QN=GAN*(UCN*CZN+UEN*EZN+USN*SZN)                                    279
00B560I          QS=GAS*(UCS*CZS+UES*EZS+USS*SZS)                                    280
00B594I          QT=GAT*(UCT*CZT+UET*EZT+UST*SZT)                                    281
00B5C8I          QB=GAB*(UCB*CZB+UEB*EZB+USB*SZB)                                    282
00B5FCI          SOC1=CX(I,J,K)*(QE-QW)+EX(I,J,K)*(QN-QS)+SX(I,J,K)*(QT-QB)          283
00B69CI          QE=GAE*(VCE*CZE+VEE*EZE+VSE*SZE)                                    284
00B6D0I          QW=GAW*(VCW*CZW+VEW*EZW+VSW*SZW)                                    285
00B704I          QN=GAN*(VCN*CZN+VEN*EZN+VSN*SZN)                                    286
00B738I          QS=GAS*(VCS*CZS+VES*EZS+VSS*SZS)                                    287
00B76CI          QT=GAT*(VCT*CZT+VET*EZT+VST*SZT)                                    288
00B7A0I          QB=GAB*(VCB*CZB+VEB*EZB+VSB*SZB)                                    289
00B7D4I          SOC2=CY(I,J,K)*(QE-QW)+EY(I,J,K)*(QN-QS)+SY(I,J,K)*(QT-QB)          290
00B874I          QE=GAE*(WCE*CZE+WEE*EZE+WSE*SZE)                                    291
00B8A8I          QW=GAW*(WCW*CZW+WEW*EZW+WSW*SZW)                                    292
00B8DCI          QN=GAN*(WCN*CZN+WEN*EZN+WSN*SZN)                                    293
00B910I          QS=GAS*(WCS*CZS+WES*EZS+WSS*SZS)                                    294
00B944I          QT=GAT*(WCT*CZT+WET*EZT+WST*SZT)                                    295
00B978I          QB=GAB*(WCB*CZB+WEB*EZB+WSB*SZB)                                    296
00B9ACI          SOC3=CZ(I,J,K)*(QE-QW)+EZ(I,J,K)*(QN-QS)+SZ(I,J,K)*(QT-QB)          297
00BA4CI          DW(I,J,K)=DW(I,J,K)+SOC1+SOC2+SOC3                                  298
00BAAAI          SU(I,J,K)=SU(I,J,K)*TJO(I,J,K)                                      299
00BB1CI          SP(I,J,K)=SPK(I,J,K)*TJO(I,J,K)                                     300
00BB3EI      15  CONTINUE                                                            301
00BBDCI          GO TO 60                                                            302
            C-----V-SOURCE
00BBE2I       2  CONTINUE                                                            303
00BBE2I          DO 25 I=IS,IT                                                       304
00BBFAI          DO 25 J=JS,JT                                                       305
00BC12I          DO 25 K=KS,KT                                                       306
00BC2AI          SU(I,J,K)=(DV(I,J,K)+APC(I,J,K)*FO(I,J,K))*TJO(I,J,K)               307
00BCEEI      25  CONTINUE                                                            308
00BD36I          GO TO 60                                                            309
            C-----W-SOURCE
00BD3CI       3  CONTINUE                                                            310
00BD3CI          DO 35 I=IS,IT                                                       311
00BD54I          DO 35 J=JS,JT                                                       312
00BD6CI          DO 35 K=KS,KT                                                       313
00BD84I          SU(I,J,K)=(DW(I,J,K)+APO(I,J,K)*FO(I,J,K))*TJO(I,J,K)               314
00BE43I      35  CONTINUE                                                            315
00BE90I          GO TO 60                                                            316
            C-----TM-SOURCE
00BE96I       4  CONTINUE                                                            317
00BE96I          GO TO 60                                                            318
            C-----K-SOURCE
00BE9CI       5  CONTINUE                                                            319
00BE9CI          DO 55 I=IS,IT                                                       320
00BEB4I          DO 55 J=JS,JT                                                       321
00BECCI          DO 55 K=KS,KT                                                       322
00BEE4I          SU(I,J,K)=GEN(I,J,K)+PP(I,J,K)+APO(I,J,K)*FO(I,J,K)                 323
00BFAAI          P1=DEN(I,J,K)**2                                                    324
00BFDEI          SP(I,J,K)=SPK(I,J,K)-CMU*P1*F(I,J,K)/VISE(I,J,K)                    325
00C08AI          SU(I,J,K)=SU(I,J,K)*TJO(I,J,K)                                      326
00C0FCI          SP(I,J,K)=SP(I,J,K)*TJO(I,J,K)                                      327
00C16EI      55  CONTINUE                                                            328
00C186I          GO TO 60                                                            329
            C-----E-SOURCE
```

ORIGINAL PAGE IS OF POOR QUALITY

```
00C18CI      6  CONTINUE                                                          330
00C18CI         DO 65 I=IS,IT                                                     331
00C1D4I         DO 65 J=JS,JT                                                     332
00C1ECI         DO 65 K=KS,KT                                                     333
00C204I         P1=DEN(I,J,K)**2                                                  334
00C238I         SU(I,J,K)=SU(I,J,K)+C1*CMU*GEN(I,J,K)*P1*DK(I,J,K)/               335
               1          VISE(I,J,K)+APO(I,J,K)*FO(I,J,K)
00C35CI         TMDK=DK(I,J,K)+SMNUM                                              336
00C38EI         SP(I,J,K)=SPK(I,J,K)-C2*DEN(I,J,K)*F(I,J,K)/TMDK                  337
00C43AI         SU(I,J,K)=SU(I,J,K)*TJO(I,J,K)                                    338
00C4ACI         SP(I,J,K)=SP(I,J,K)*TJO(I,J,K)                                    339
00C51EI     65  CONTINUE                                                          340
00C566I         GO TO 60                                                          341
             C-----PF-SOURCE
00C56CI      7  CONTINUE                                                          342
00C56CI         DO 75 I=IS,IT                                                     343
00C584I         DO 75 J=JS,JT                                                     344
00C59CI         DO 75 K=KS,KT                                                     345
00C5B4I         SU(I,J,K)=SU(I,J,K)+0.85*GEN(I,J,K)+APO(I,J,K)*FO(I,J,K)          346
00C680I         TMDK=DK(I,J,K)+SMNUM                                              347
00C6B2I         SP(I,J,K)=SPK(I,J,K)-DE(I,J,K)/TMDK                               348
00C72CI         SU(I,J,K)=SU(I,J,K)*TJO(I,J,K)                                    349
00C79EI         SP(I,J,K)=SP(I,J,K)*TJO(I,J,K)                                    350
00C810I     75  CONTINUE                                                          351
00C858I     60  CONTINUE                                                          352
             C-----MODIFY WALL BOUNDARY CONDITIONS THRU WALL FUNCTIONS
00C858I         IF(IG .NE. 2) GO TO 410                                           353
00C86EI         CALL BOUNC(IE,F)                                                  355
00C89CI    410  CONTINUE                                                          356
             C-----SET SYMMETRIC, CYCLIC AND EXIT LINK COEFF.
00C89CI         CALL SYMOUT(2,IE,IS,IT,JS,JT,KS,KT)                               357
             C-----LINK COEFF. ASSEMBLY AND BLOCKAGES
00C8D8I         DO 500 I=IS,IT                                                    358
00C8F0I         DO 500 J=JS,JT                                                    359
00C908I         DO 500 K=KS,KT                                                    360
00C920I         F1(I,J,K)=F(I,J,K)                                                361
00C972I         ANAB=AE(I,J,K)+AW(I,J,K)+AN(I,J,K)+AS(I,J,K)+AT(I,J,K)+           362
               1      AB(I,J,K)+APO(I,J,K)
00CA82I         AP(I,J,K)=ANAB-SP(I,J,K)                                          363
00CAD4I         PDUV=1.0                                                          364
00CAE0I         IF(MC(I,J,K) .LT. 1) GO TO 530                                    365
00CB16I         AP(I,J,K)=ALF                                                     367
00CB42I         AN(I,J,K)=0.0                                                     368
00CB6EI         AS(I,J,K)=0.0                                                     369
00CB9AI         AE(I,J,K)=0.0                                                     370
00CBC6I         AW(I,J,K)=0.0                                                     371
00CBF2I         AT(I,J,K)=0.0                                                     372
00CC1EI         AB(I,J,K)=0.0                                                     373
00CC4AI         SU(I,J,K)=F(I,J,K)                                                374
00CC9CI         PDUV=0.0                                                          375
00CCA8I    530  CONTINUE                                                          376
             C-----UNDER-RELAXATION
00CCA8I         P1=1.2*AP(I,J,K)                                                  377
00CCDAI         AP(I,J,K)=AP(I,J,K)/ALF                                           378
00CD2CI         SU(I,J,K)=SU(I,J,K)+PDUV*(1.0-ALF)*AP(I,J,K)*F(I,J,K)             379
00CDDCI         IF(IE .EQ. 1) DU(I,J,K)=TJO(I,J,K)*PDUV/(P1-ANAB)                 380
00CE4CI    500  CONTINUE                                                          382
```

```
                 C-----LINEAR EQUATIONS SLOVER
00CE94I                DO 550 I=1,ISWF                                                    383
00CEA8I            550 CALL LINERX(1,IS,JS,KS,IT,JT,KT,F)                                 384
                 C-----CALCULATE MAXIMUM CORRECTION OF CURRENT ITERATION
00CF0CI                DO 555 I=IS,IT                                                     385
00CF24I                DO 555 J=JS,JT                                                     386
00CF3CI                DO 555 K=KS,KT                                                     387
00CF54I                P1=ABS(F(I,J,K)-F1(I,J,K))                                         388
00CFCAI                ERRF=AMAX1(ERRF,P1)                                                389
00CFEAI            555 CONTINUE                                                           390
000032I                RETURN                                                             391
                 C-----------------------------------------------------------------------
                 C-----P EQUATION
00003AI             10 CONTINUE                                                           392
00003AI                IS=2                                                               393
000042I                IT=L                                                               394
00004EI                JS=2                                                               395
000056I                JT=M                                                               396
000062I                KS=2                                                               397
00006AI                KT=N                                                               398
                 C-----SOURCE AT PX LOCATIONS
000076I                DO 310 I=IS,IT                                                     399
00008EI                DO 310 J=JS,JT                                                     400
0000A6I                DO 310 K=KS,KT                                                     401
0000BEI                F(I,J,K)=0.0                                                       402
0000F0I                SUK(I,J,K)=0.0                                                     403
00011CI                SPK(I,J,K)=0.0                                                     404
000148I                SP(I,J,K)=0.0                                                      405
000174I                DENE=DEN(I,J,K)                                                    406
0001A0I                DENW=DEN(I-1,J,K)                                                  407
0001D0I                DENN=0.25*(DEN(I,J,K)+DEN(I-1,J,K)+DEN(I,J+1,K)+DEN(I-1,J+1,K))   408
000284I                DENS=0.25*(DEN(I,J,K)+DEN(I-1,J,K)+DEN(I,J-1,K)+DEN(I-1,J-1,K))   409
000338I                DENT=0.25*(DEN(I,J,K)+DEN(I-1,J,K)+DEN(I,J,K+1)+DEN(I-1,J,K+1))   410
0003E8I                DENB=0.25*(DEN(I,J,K)+DEN(I-1,J,K)+DEN(I,J,K-1)+DEN(I-1,J,K-1))   411
000498I                UE=U(I,J,K)                                                        412
0004C4I                UW=U(I-1,J,K)                                                      413
0004F4I                UN=C.25*(U(I,J,K)+U(I-1,J,K)+U(I,J+1,K)+U(I-1,J+1,K))             414
0005A8I                US=C.25*(U(I,J,K)+U(I-1,J,K)+U(I,J-1,K)+U(I-1,J-1,K))             415
00065CI                UT=C.25*(U(I,J,K)+U(I-1,J,K)+U(I,J,K+1)+U(I-1,J,K+1))             416
00070CI                UB=C.25*(U(I,J,K)+U(I-1,J,K)+U(I,J,K-1)+U(I-1,J,K-1))             417
0007BCI                VE=V(I,J,K)                                                        418
0007E8I                VW=V(I-1,J,K)                                                      419
000818I                VN=C.25*(V(I,J,K)+V(I-1,J,K)+V(I,J+1,K)+V(I-1,J+1,K))             420
0008CCI                VS=C.25*(V(I,J,K)+V(I-1,J,K)+V(I,J-1,K)+V(I-1,J-1,K))             421
000980I                VT=C.25*(V(I,J,K)+V(I-1,J,K)+V(I,J,K+1)+V(I-1,J,K+1))             422
000A30I                VB=C.25*(V(I,J,K)+V(I-1,J,K)+V(I,J,K-1)+V(I-1,J,K-1))             423
000AE0I                WE=W(I,J,K)                                                        424
000B0CI                WW=W(I-1,J,K)                                                      425
000B3CI                WN=C.25*(W(I,J,K)+W(I-1,J,K)+W(I,J+1,K)+W(I-1,J+1,K))             426
000BF0I                WS=C.25*(W(I,J,K)+W(I-1,J,K)+W(I,J-1,K)+W(I-1,J-1,K))             427
000CA4I                WT=C.25*(W(I,J,K)+W(I-1,J,K)+W(I,J,K+1)+W(I-1,J,K+1))             428
000D54I                WB=C.25*(W(I,J,K)+W(I-1,J,K)+W(I,J,K-1)+W(I-1,J,K-1))             429
000E04I                CXQ=0.5*(CX(I,J,K)+CX(I-1,J,K))                                    430
000E60I                EXQ=0.5*(EX(I,J,K)+EX(I-1,J,K))                                    431
000EBCI                SXQ=0.5*(SX(I,J,K)+SX(I-1,J,K))                                    432
000F18I                CYQ=0.5*(CY(I,J,K)+CY(I-1,J,K))                                    433
000F74I                EYQ=0.5*(EY(I,J,K)+EY(I-1,J,K))                                    434
```

```
00F00I          SYQ=0.5*(SY(I,J,K)+SY(I-1,J,K))                                      435
00E02CI         CZQ=0.5*(CZ(I,J,K)+CZ(I-1,J,K))                                      436
00E088I         EZQ=0.5*(EZ(I,J,K)+EZ(I-1,J,K))                                      437
00E0E4I         SZQ=0.5*(SZ(I,J,K)+SZ(I-1,J,K))                                      438
00E140I         CE=DENE*(UE*CXQ+VE*CYQ+WE*CZQ)                                       439
00E174I         CW=DENW*(UW*CXQ+VW*CYQ+WW*CZQ)                                       440
00E1A8I         CN=DENN*(UN*EXQ+VN*EYQ+WN*EZQ)                                       441
00E1CCI         CS=DENS*(US*EXQ+VS*EYQ+WS*EZQ)                                       442
00E210I         CT=DENT*(UT*SXQ+VT*SYQ+WT*SZQ)                                       443
00E244I         CB=DENB*(UB*SXQ+VB*SYQ+WB*SZQ)                                       444
00E278I         SUK(I,J,K)=-(CE-CW+CN-CS+CT-CB)*(TJO(I-1,J,K)+TJO(I,J,K))*0.5         445
00E322I    310  CONTINUE                                                             446
          C-----SOURCE AT PY LOCATIONS
00E36AI         DO 311 I=IS,IT                                                       447
00E382I         DO 311 J=JS,JT                                                       448
00E39AI         DO 311 K=KS,KT                                                       449
00E3B2I         DENN=DEN(I,J,K)                                                      450
00E3DEI         DENS=DEN(I,J-1,K)                                                    451
00E40EI         DENE=0.25*(DEN(I,J,K)+DEN(I,J-1,K)+DEN(I+1,J,K)+DEN(I+1,J-1,K))      452
00E4C2I         DENW=0.25*(DEN(I,J,K)+DEN(I,J-1,K)+DEN(I-1,J,K)+DEN(I-1,J-1,K))      453
00E576I         DENT=0.25*(DEN(I,J,K)+DEN(I,J-1,K)+DEN(I,J,K+1)+DEN(I,J-1,K+1))      454
00E626I         DENB=0.25*(DEN(I,J,K)+DEN(I,J-1,K)+DEN(I,J,K-1)+DEN(I,J-1,K-1))      455
00E6D6I         UN=U(I,J,K)                                                          456
00E702I         US=U(I,J-1,K)                                                        457
00E732I         UE=0.25*(U(I,J,K)+U(I,J-1,K)+U(I+1,J,K)+U(I+1,J-1,K))                458
00E7E6I         UW=C.25*(U(I,J,K)+U(I,J-1,K)+U(I-1,J,K)+U(I-1,J-1,K))                459
00E89AI         UT=C.25*(U(I,J,K)+U(I,J-1,K)+U(I,J,K+1)+U(I,J-1,K+1))                460
00E94AI         UB=C.25*(U(I,J,K)+U(I,J-1,K)+U(I,J,K-1)+U(I,J-1,K-1))                461
00E9FAI         VN=V(I,J,K)                                                          462
00EA26I         VS=V(I,J-1,K)                                                        463
00EA56I         VE=0.25*(V(I,J,K)+V(I,J-1,K)+V(I+1,J,K)+V(I+1,J-1,K))                464
00EB0AI         VW=C.25*(V(I,J,K)+V(I,J-1,K)+V(I-1,J,K)+V(I-1,J-1,K))                465
00EBBEI         VT=C.25*(V(I,J,K)+V(I,J-1,K)+V(I,J,K+1)+V(I,J-1,K+1))                466
00EC6EI         VB=C.25*(V(I,J,K)+V(I,J-1,K)+V(I,J,K-1)+V(I,J-1,K-1))                467
00ED1EI         WN=W(I,J,K)                                                          468
00ED4AI         WS=W(I,J-1,K)                                                        469
00ED7AI         WE=C.25*(W(I,J,K)+W(I,J-1,K)+W(I+1,J,K)+W(I+1,J-1,K))                470
00EE2EI         WW=C.25*(W(I,J,K)+W(I,J-1,K)+W(I-1,J,K)+W(I-1,J-1,K))                471
00EEE2I         WT=C.25*(W(I,J,K)+W(I,J-1,K)+W(I,J,K+1)+W(I,J-1,K+1))                472
00EF92I         WB=C.25*(W(I,J,K)+W(I,J-1,K)+W(I,J,K-1)+W(I,J-1,K-1))                473
00F042I         CXQ=C.5*(CX(I,J,K)+CX(I,J-1,K))                                      474
00F09EI         EXQ=0.5*(EX(I,J,K)+EX(I,J-1,K))                                      475
00F0FAI         SXQ=0.5*(SX(I,J,K)+SX(I,J-1,K))                                      476
00F156I         CYQ=0.5*(CY(I,J,K)+CY(I,J-1,K))                                      477
00F1B2I         EYQ=0.5*(EY(I,J,K)+EY(I,J-1,K))                                      478
00F20EI         SYQ=0.5*(SY(I,J,K)+SY(I,J-1,K))                                      479
00F26AI         CZQ=0.5*(CZ(I,J,K)+CZ(I,J-1,K))                                      480
00F2C6I         EZQ=C.5*(EZ(I,J,K)+EZ(I,J-1,K))                                      481
00F322I         SZQ=0.5*(SZ(I,J,K)+SZ(I,J-1,K))                                      482
00F37EI         CE=DENE*(UE*CXQ+VE*CYQ+WE*CZQ)                                       483
00F3B2I         CW=DENW*(UW*CXQ+VW*CYQ+WW*CZQ)                                       484
00F3E6I         CN=DENN*(UN*EXQ+VN*EYQ+WN*EZQ)                                       485
00F41AI         CS=DENS*(US*EXQ+VS*EYQ+WS*EZQ)                                       486
00F44EI         CT=DENT*(UT*SXQ+VT*SYQ+WT*SZQ)                                       487
00F482I         CB=DENB*(UB*SXQ+VB*SYQ+WB*SZQ)                                       488
00F4B6I         SPK(I,J,K)=-(CE-CW+CN-CS+CT-CB)*(TJO(I,J-1,K)+TJO(I,J,K))*0.5         489
00F560I    311  CONTINUE                                                             490
```

```
571            C-----SOURCE AT PZ LOCATIONS                                                          1362
572  00F5A8I          DO 312 I=IS,IT                                                                  1363   491
573  00F5C0I          DO 312 J=JS,JT                                                                  1364   492
574  00F5D8I          DO 312 K=KS,KT                                                                  1365   493
575  00F5F0I          DENT=DEN(I,J,K)                                                                 1366   494
576  00F61CI          DENS=DEN(I,J,K-1)                                                               1367   495
577  00F64AI          DENN=0.25*(DEN(I,J,K)+DEN(I,J,K-1)+DEN(I,J+1,K)+DEN(I,J+1,K-1))                 1368   496
578  00F6FAI          DENS=0.25*(DEN(I,J,K)+DEN(I,J,K-1)+DEN(I,J-1,K)+DEN(I,J-1,K-1))                 1369   497
579  00F7AAI          DENE=0.25*(DEN(I,J,K)+DEN(I,J,K-1)+DEN(I+1,J,K)+DEN(I+1,J,K-1))                 1370   498
580  00F85AI          DENW=0.25*(DEN(I,J,K)+DEN(I,J,K-1)+DEN(I-1,J,K)+DEN(I-1,J,K-1))                 1371   499
581  00F90AI          UT=U(I,J,K)                                                                     1372   500
582  00F936I          UB=U(I,J,K-1)                                                                   1373   501
583  00F964I          UN=C.25*(U(I,J,K)+U(I,J,K-1)+U(I,J+1,K)+U(I,J+1,K-1))                           1374   502
584  00FA14I          US=C.25*(U(I,J,K)+U(I,J,K-1)+U(I,J-1,K)+U(I,J-1,K-1))                           1375   503
585  00FAC4I          UE=C.25*(U(I,J,K)+U(I,J,K-1)+U(I+1,J,K)+U(I+1,J,K-1))                           1376   504
586  00FB74I          UW=C.25*(U(I,J,K)+U(I,J,K-1)+U(I-1,J,K)+U(I-1,J,K-1))                           1377   505
587  00FC24I          VT=V(I,J,K)                                                                     1378   506
588  00FC50I          VB=V(I,J,K-1)                                                                   1379   507
589  00FC7EI          VN=C.25*(V(I,J,K)+V(I,J,K-1)+V(I,J+1,K)+V(I,J+1,K-1))                           1380   508
590  00FD2EI          VS=C.25*(V(I,J,K)+V(I,J,K-1)+V(I,J-1,K)+V(I,J-1,K-1))                           1381   509
591  00FDDEI          VE=C.25*(V(I,J,K)+V(I,J,K-1)+V(I+1,J,K)+V(I+1,J,K-1))                           1382   510
592  00FE8EI          VW=C.25*(V(I,J,K)+V(I,J,K-1)+V(I-1,J,K)+V(I-1,J,K-1))                           1383   511
593  00FF3EI          WT=W(I,J,K)                                                                     1384   512
594  00FF6AI          WB=W(I,J,K-1)                                                                   1385   513
595  00FF98I          WN=C.25*(W(I,J,K)+W(I,J,K-1)+W(I,J+1,K)+W(I,J+1,K-1))                           1386   514
596  010048I          WS=C.25*(W(I,J,K)+W(I,J,K-1)+W(I,J-1,K)+W(I,J-1,K-1))                           1387   515
597  0100F8I          WE=C.25*(W(I,J,K)+W(I,J,K-1)+W(I+1,J,K)+W(I+1,J,K-1))                           1388   516
598  0101A8I          WW=C.25*(W(I,J,K)+W(I,J,K-1)+W(I-1,J,K)+W(I-1,J,K-1))                           1389   517
599  010258I          CXQ=0.5*(CX(I,J,K)+CX(I,J,K-1))                                                 1390   518
600  010282I          EXQ=0.5*(EX(I,J,K)+EX(I,J,K-1))                                                 1391   519
601  01030CI          SXQ=0.5*(SX(I,J,K)+SX(I,J,K-1))                                                 1392   520
602  010366I          CYQ=0.5*(CY(I,J,K)+CY(I,J,K-1))                                                 1393   521
603  0103C0I          EYQ=0.5*(EY(I,J,K)+EY(I,J,K-1))                                                 1394   522
604  01041AI          SYQ=0.5*(SY(I,J,K)+SY(I,J,K-1))                                                 1395   523
605  010474I          CZQ=0.5*(CZ(I,J,K)+CZ(I,J,K-1))                                                 1396   524
606  0104CEI          EZQ=0.5*(EZ(I,J,K)+EZ(I,J,K-1))                                                 1397   525
607  010528I          SZQ=0.5*(SZ(I,J,K)+SZ(I,J,K-1))                                                 1398   526
608  010582I          CE=DENE*(UE*CXQ+VE*CYQ+WE*CZQ)                                                  1399   527
609  0105B6I          CW=DENW*(UW*CXQ+VW*CYQ+WW*CZQ)                                                  1400   528
610  0105EAI          CN=DENN*(UN*EXQ+VN*EYQ+WN*EZQ)                                                  1401   529
611  01061EI          CS=DENS*(US*EXQ+VS*EYQ+WS*EZQ)                                                  1402   530
612  010652I          CT=DENT*(UT*SXQ+VT*SYQ+WT*SZQ)                                                  1403   531
613  010686I          CB=DENB*(UB*SXQ+VB*SYQ+WB*SZQ)                                                  1404   532
614  0106BAI          SP(I,J,K)=-(CE-CW+CN-CS+CT-CB)*(TJO(I,J,K-1)+TJO(I,J,K))*0.5                    1405   533
615  010762I      312 CONTINUE                                                                        1406   534
616            C-----LINK COEFF. AND SOURCE TERM AT CELL CENTER                                       1407
617  0107AAI          DO 100 I=IS,IT                                                                  1408   535
618  0107C2I          DO 100 J=JS,JT                                                                  1409   536
619  0107DAI          DO 100 K=KS,KT                                                                  1410   537
620  0107F2I          II=I-1                                                                          1411   538
621  010800I          JJ=J-1                                                                          1412   539
622  01080EI          KK=K-1                                                                          1413   540
623  01081CI          GAE=(DU(I,J,K)+DU(I,JJ,K)+DU(I,J,KK)+DU(I,JJ,KK))*0.25                          1414   541
624  0108C0I          GAW=(DU(II,J,K)+DU(II,JJ,K)+DU(II,J,KK)+DU(II,JJ,KK))*0.25                      1415   542
625  010964I          GAN=(DU(I,J,K)+DU(II,J,K)+DU(I,J,KK)+DU(II,J,KK))*0.25                          1416   543
626  010A08I          GAS=(DU(I,JJ,K)+DU(II,JJ,K)+DU(I,JJ,KK)+DU(II,JJ,KK))*0.25                      1417   544
627  010AACI          GAT=(DU(I,J,K)+DU(II,J,K)+DU(I,JJ,K)+DU(II,JJ,K))*0.25                          1418   545
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
010B50I         GAB=(DU(I,J,KK)+DU(II,J,KK)+DU(I,JJ,KK)+DU(II,JJ,KK))*0.25      546
010BF4I         TXXEQ=(TXXE(I,J,K)+TXXE(II,JJ,KK))*0.5                          547
010C4CI         TXXWQ=(TXXW(I,J,K)+TXXW(II,JJ,KK))*0.5                          548
010CA4I         TYYNQ=(TYYN(I,J,K)+TYYN(II,JJ,KK))*0.5                          549
010CFCI         TYYSQ=(TYYS(I,J,K)+TYYS(II,JJ,KK))*0.5                          550
010D54I         TZZTQ=(TZZT(I,J,K)+TZZT(II,JJ,KK))*C.5                          551
010DACI         TZZBQ=(TZZB(I,J,K)+TZZB(II,JJ,KK))*0.5                          552
010E04I         TQ=(TJO(I,J,K)+TJO(II,JJ,KK))*C.5                               553
010E5CI         AE(I,J,K)=TQ*(GAE*TXXEQ)                                        554
010E94I         AW(I,J,K)=TQ*(GAW*TXXWQ)                                        555
010ECCI         AN(I,J,K)=TQ*(GAN*TYYNQ)                                        556
010F04I         AS(I,J,K)=TQ*(GAS*TYYSQ)                                        557
010F3CI         AT(I,J,K)=TQ*(GAT*TZZTQ)                                        558
010F74I         AB(I,J,K)=TQ*(GAB*TZZBQ)                                        559
          C-----P' SOURCE
010FACI         P1=1.0                                                          560
010FB8I         P2=1.0                                                          561
010FC4I         P3=1.0                                                          562
010FD0I         P4=1.0                                                          563
010FDCI         P5=1.0                                                          564
010FE8I         P6=1.0                                                          565
010FF4I         IF(GAN .EQ. 0.0) P1=0.0                                         566
011012I         IF(GAS .EQ. 0.0) P2=0.0                                         568
011030I         IF(GAE .EQ. 0.0) P3=0.0                                         570
01104EI         IF(GAW .EQ. 0.0) P4=0.0                                         572
01106CI         IF(GAT .EQ. 0.0) P5=0.0                                         574
01108AI         IF(GAB .EQ. 0.0) P6=0.0                                         576
0110A8I         PQ=C.5*((P1+P2)*(P3+P4+P5+P6)+(P3+P4)*(P1+P2+P5+P6)+            578
               1          (P5+P6)*(P1+P2+P3+P4))
01112AI         IF(PQ .EQ. 0.0) PQ=1.0                                          579
011148I         SU(I,J,K)=(P1*P5*SUK(I,J,K)+P5*P2*SUK(I,J-1,K)+                 581
               1            P2*P6*SUK(I,J-1,K-1)+P6*P1*SUK(I,J,K-1)+
               2            P3*P5*SPK(I,J,K)+P5*P4*SPK(I-1,J,K)+
               3            P4*P6*SPK(I-1,J,K-1)+P6*P3*SPK(I,J,K-1)+
               4            P1*P3*SP(I,J,K)+P3*P2*SP(I,J-1,K)+
               5            P2*P4*SP(I-1,J-1,K)+P4*P1*SP(I-1,J,K))/PQ
01140AI     100 CONTINUE                                                        582
          C-----LINK COEFF. ASSEMBLY AND BLOCKAGES
011452I         DO 870 I=IS,IT                                                  583
01146AI         DO 870 J=JS,JT                                                  584
011482I         DO 870 K=KS,KT                                                  585
01149AI         ANAB=AE(I,J,K)+AW(I,J,K)+AN(I,J,K)+AS(I,J,K)+AT(I,J,K)+AB(I,J,K)  586
011584I         AP(I,J,K)=ANAB                                                  587
0115B0I         IF(MC(I,J,K) .LT. 2) GO TO 870                                  588
0115E6I         AP(I,J,K)=1.0                                                   590
011612I         AE(I,J,K)=0.0                                                   591
01163EI         AW(I,J,K)=0.0                                                   592
01166AI         AN(I,J,K)=0.0                                                   593
011696I         AS(I,J,K)=0.0                                                   594
0116C2I         AT(I,J,K)=0.0                                                   595
0116EEI         AB(I,J,K)=0.0                                                   596
01171AI         SU(I,J,K)=0.0                                                   597
011746I     870 CONTINUE                                                        598
          C-----LINEAR EQUATIONS SOLVER
01178EI         CALL LINERX(2,IS,JS,KS,IT,JT,KT,F)                              599
0117DCI         DO 850 I=1,ISWF                                                 600
0117F0I     850 CALL LINERX(1,IS,JS,KS,IT,JT,KT,F)                              601
```

```
685                C-----PRESSURE AND VELOCITIES CORRECTIONS                                1476
686   011854I            PPREF=F(2,2,2)                                                     1477   602
687   011864I            DO 880 I=IS,IT                                                     1478   603
688   01187CI            DO 880 J=JS,JT                                                     1479   604
689   011894I            DO 880 K=KS,KT                                                     1480   605
690   0118ACI            IF(MC(I,J,K) .GE. 2) GO TO 880                                     1481   606
691   0118E2I            P1=F(I,J,K)-PPREF                                                  1482   608
692   01191AI            P(I,J,K)=P(I,J,K)+ALP*P1                                           1483   609
693   011972I            ERRF=AMAX1(ERRF,ABS(P1))                                           1484   610
694   0119AEI      880   CONTINUE                                                           1485   611
695   0119F6I            DO 600 I=2,LT                                                      1486   612
696   011A0AI            DO 600 J=2,MT                                                      1487   613
697   011A1EI            DO 600 K=2,NT                                                      1488   614
698   011A32I            IF(MC(I,J,K) .GE. 1) GO TO 600                                     1489   615
699   011A68I            II=I+1                                                             1490   617
700   011A76I            JJ=J+1                                                             1491   618
701   011A84I            KK=K+1                                                             1492   619
702   011A92I            PE=(F(II,J,K)+F(II,JJ,K)+F(II,J,KK)+F(II,JJ,KK))*0.25              1493   620
703   011B4EI            PW=(F(I,J,K)+F(I,JJ,K)+F(I,J,KK)+F(I,JJ,KK))*0.25                  1494   621
704   011C0AI            PN=(F(I,JJ,K)+F(II,JJ,K)+F(I,JJ,KK)+F(II,JJ,KK))*0.25              1495   622
705   011CC6I            PS=(F(I,J,K)+F(II,J,K)+F(I,J,KK)+F(II,J,KK))*0.25                  1496   623
706   011D82I            PT=(F(I,J,KK)+F(II,J,KK)+F(I,JJ,KK)+F(II,JJ,KK))*0.25              1497   624
707   011E3EI            PB=(F(I,J,K)+F(II,J,K)+F(I,JJ,K)+F(II,JJ,K))*0.25                  1498   625
708   011EFAI            PCXI=PE-PW                                                         1499   626
709   011F0CI            PEDA=PN-PS                                                         1500   627
710   011F1EI            PSCI=PT-PB                                                         1501   628
711   011F30I            CXQ=CX(I,J,K)                                                      1502   629
712   011F5CI            EXQ=EX(I,J,K)                                                      1503   630
713   011F88I            SXQ=SX(I,J,K)                                                      1504   631
714   011FB4I            CYQ=CY(I,J,K)                                                      1505   632
715   011FE0I            EYQ=EY(I,J,K)                                                      1506   633
716   01200CI            SYQ=SY(I,J,K)                                                      1507   634
717   012038I            CZQ=CZ(I,J,K)                                                      1508   635
718   012064I            EZQ=EZ(I,J,K)                                                      1509   636
719   012090I            SZQ=SZ(I,J,K)                                                      1510   637
720   0120BCI            PXX=PCXI*CXQ+PEDA*EXQ+PSCI*SXQ                                     1511   638
721   0120EAI            PYY=PCXI*CYQ+PEDA*EYQ+PSCI*SYQ                                     1512   639
722   012118I            PZZ=PCXI*CZQ+PEDA*EZQ+PSCI*SZQ                                     1513   640
723   012146I            U(I,J,K)=U(I,J,K)-DU(I,J,K)*PXX                                    1514   641
724   0121C0I            V(I,J,K)=V(I,J,K)-DU(I,J,K)*PYY                                    1515   642
725   01223AI            W(I,J,K)=W(I,J,K)-DU(I,J,K)*PZZ                                    1516   643
726   0122B4I      600   CONTINUE                                                           1517   644
727   0122FCI            RETURN                                                             1518   645
728   012304I            END                                                                1519   646

NO  ERRORS:F70  R05-01.0C  SUBROUTINE  SOLVEQ    02/21/86  09:58:31  TABLE SPACE:  18 KB
STATEMENT BUFFER:   20 LINES/1321 BYTES     STACK SPACE: 203 WORDS
SINGLE PRECISION FLOATING PT SUPPORT REQUIRED FOR EXECUTION
```

```
000000I          SUBROUTINE LINERX(ISOL,IS,JS,KS,LT,MT,NT,F)                                   1520
000004I          DIMENSICN A(20),B(20),C(20),D(20),F(21,18,10),BB(21,18,10)                    1521
000004I          COMMON                                                                         1522
                1/COEF/ AP(21,18,10),SU(21,18,10),SP(21,18,10),SUK(21,18,10),                  1523
                2   SPK(21,18,10),AE(21,18,10),AW(21,18,10),AN(21,18,10),                       1524
                3   AS(21,18,10),AT(21,18,10),AB(21,18,10),APO(21,18,10)                        1525
           C-----LINEAR EQUATIONS SOLVERS                                                       1526
000004I          GO TO (1,2),ISOL                                                               1527     1
           C-----LINE-RELAXATION USING TDMA                                                     1528
003C86I      1   CONTINUE                                                                       1529     2
003C86I          DO 10 I=IS,LT                                                                  1530     3
003C9EI          DO 10 J=JS,MT                                                                  1531     4
003CB6I          DO 10 K=KS,NT                                                                  1532     5
003CCEI          BB(I,J,K)=AE(I,J,K)*F(I+1,J,K)+AN(I,J,K)*F(I,J+1,K)+                           1533     6
                1          SU(I,J,K)                                                            1534
003DC8I     10   CONTINUE                                                                       1535     7
003E10I          A(KS-1)=0.0                                                                    1536     8
003E24I          DO 100 I=IS,LT                                                                 1537     9
003E3CI          DO 100 J=JS,MT                                                                 1538    10
003E54I          C(KS-1)=F(I,J,KS-1)                                                            1539    11
003E90I          DO 101 K=KS,NT                                                                 1540    12
003EA8I          A(K)=AT(I,J,K)                                                                 1541    13
003EDAI          B(K)=AB(I,J,K)                                                                 1542    14
003F0CI          C(K)=BB(I,J,K)+AW(I,J,K)*F(I-1,J,K)+                                           1543    15
                1     AS(I,J,K)*F(I,J-1,K)                                                      1544
003FECI          D(K)=AP(I,J,K)                                                                 1545    16
00401EI          TERM1=D(K)-B(K)*A(K-1)+1.E-30                                                  1546    17
004054I          TERM=1.0/TERM1                                                                 1547    18
004066I          A(K)=A(K)*TERM                                                                 1548    19
004088I    101   C(K)=(C(K)+B(K)*C(K-1))*TERM                                                   1549    20
0040E0I          DO 102 KK=KS,NT                                                                1550    21
0040F8I          K=KS+NT-KK                                                                     1551    22
004110I    102   F(I,J,K)=A(K)*F(I,J,K+1)+C(K)                                                  1552    23
00419EI    100   CONTINUE                                                                       1553    24
0041CEI          RETURN                                                                         1554    25
           C-----BLOCK CORRECTION FOR PRESSURE FIELD                                            1555
0041D6I      2   CONTINUE                                                                       1556    26
0041D6I          DO 201 I=IS,LT                                                                 1557    27
0041EEI          A(I)=0.0                                                                       1558    28
004202I          B(I)=0.0                                                                       1559    29
004216I          C(I)=0.0                                                                       1560    30
00422AI          D(I)=0.0                                                                       1561    31
00423EI          DO 201 J=JS,MT                                                                 1562    32
004256I          DO 201 K=KS,NT                                                                 1563    33
00426EI          A(I)=A(I)-AE(I,J,K)                                                            1564    34
0042B0I          B(I)=B(I)-AW(I,J,K)                                                            1565    35
0042F2I          C(I)=C(I)+SU(I,J,K)                                                            1566    36
004334I    201   CONTINUE                                                                       1567    37
00437CI          DO 202 I=IS,LT                                                                 1568    38
004394I          ANAB=-A(I)-B(I)                                                                1569    39
0043B3CI   202   D(I)=ANAB*1.5                                                                  1570    40
0043EEI          DO 203 I=IS+1,LT                                                               1571    41
00440AI          TERM=B(I)/D(I-1)                                                               1572    42
00442EI          D(I)=D(I)-TERM*A(I-1)                                                          1573    43
004462I    203   C(I)=C(I)-TERM*C(I-1)                                                          1574    44
0044AEI          F(LT,JS,KS)=C(LT)/D(LT)                                                        1575    45
0044F6I          DO 204 II=IS+1,LT                                                              1576    46
```

```
58  004512I          I=IS+LT-II                                                          1577      47
59  00452AI    204   F(I,JS,KS)=(C(I)-A(I)*F(I+1,JS,KS))/D(I)                            1578      48
60  0045CAI          DO 205 I=IS,LT                                                      1579      49
61  0045E2I          PPBLK=F(I,JS,KS)                                                    1580      50
62  004614I          DO 205 J=JS+1,MT                                                    1581      51
63  004630I          DO 205 K=KS+1,NT                                                    1582      52
64  00464CI          F(I,J,K)=PPBLK                                                      1583      53
65  00467EI    205   CONTINUE                                                            1584      54
66  0046C6I          RETURN                                                              1585      55
67  0046CEI          END                                                                 1586      56

NO  ERRORS:F7D  R05-01.0C  SUBROUTINE  LINERX     02/21/86  09:59:43  TABLE SPACE:   3 KB
STATEMENT BUFFER:   20 LINES/1321 BYTES     STACK SPACE: 199 WORDS
SINGLE PRECISION FLOATING PT SUPPORT REQUIRED FOR EXECUTION
```

```
000000I         SUBROUTINE BOUNC(IE,F)
000004I         DIMENSION F(21,18,10)
000004I         COMMON
               1/VAR/U(21,18,10),V(21,18,10),P(21,18,10),DK(21,18,10),
               2  DE(21,18,10),ERRU,ERRV,ERRM,ERRK,ERRE,ERRW,
               3  PF(21,18,10),W(21,18,10),TM(21,18,10)
               1/PRCP/ VISE(21,18,10),DEN(21,18,10),VISC,DENIN,FLOWIN
               1/PCOR/ DU(21,18,10),DV(21,18,10),DW(21,18,10)
               1/TUR/ SIGK,SIGE,CMU,C1,C2,CMU1,CMU2,E,CK,HINUM,SMNUM,ANV1(800),
               2 YN(800),YN1(800),SINX(800),SINY(800),SINZ(800),ANW1(800),
               3 YPLN(800),TAUN(800),IBC(800),JBC(800),KBC(800),IITY(800),
               4 TALW(800),GEN(21,18,10),MC(21,18,10),IJLO(21,18,10),IITO
               1/COEF/ AP(21,18,10),SU(21,18,10),SP(21,18,10),SUK(21,18,10),
               2    SPK(21,18,10),AE(21,18,10),AW(21,18,10),AN(21,18,10),
               3    AS(21,18,10),AT(21,18,10),AB(21,18,10),APO(21,18,10)
000004I         COMMON
               1/TRAN/ X(21,18,10),Y(21,18,10),Z(21,18,10),TJO(21,18,10),
               2    CX(21,18,10),CY(21,18,10),CZ(21,18,10),
               3    EX(21,18,10),EY(21,18,10),EZ(21,18,10),
               3    SX(21,18,10),SY(21,18,10),SZ(21,18,10)
               1/LIMT/ L,M,LT,MT,L1,L2,M1,M2,LO,MO,ISWU,ISWV,ISWP,ISWK,ISWE,
               2    ALU,ALV,ALP,ALK,ALE,ALVIS,ALW,N,N1,N2,NO,ISWW,IG,NT,ALC,DTT
          C-----EVALUATE WALL BOUNDARY CONDITIONS USING WALL FUNCTIONS
000004I         DO 150 III=1,IITO                                              1
000024I         I=IBC(III)                                                     2
000038I         J=JBC(III)                                                     3
00004CI         K=KBC(III)                                                     4
000060I         GO TO (1,2,3,4,5,6), IITY(III)                                 5
00009EI       1 CONTINUE                                                       6
          C-----NORTH
00009EI         YP=YN(III)                                                     7
000032I         YP1=YN1(III)                                                   8
0000C6I         CDK=DK(I,J,K)                                                  9
0000F2I         CDE=DE(I,J,K)                                                 10
00011EI         CDEN=DEN(I,J,K)                                               11
00014AI         CALL WALLFN(IE,YP,YP1,CDK,CDE,CDEN,SINX(III),SINY(III),       12
               1     SINZ(III),F(I,J+1,K),AN(I,J,K),SU(I,J,K),SP(I,J,K),
               2     TAUN(III),YPLN(III),GEN(I,J,K),VISE(I,J,K),
               3     U(I,J+1,K),V(I,J+1,K),W(I,J+1,K),U(I,J,K),V(I,J,K),
               4     W(I,J,K),U(I,J-1,K),V(I,J-1,K),W(I,J-1,K),
               5    -SPK(I,J,K),SPK(I,J,K),ANV1(III),ANW1(III),TJO(I,J,K))
000584I         GO TO 150                                                     13
00058AI       2 CONTINUE                                                      14
          C-----SOUTH
0005BAI         YP=YN(III)                                                    15
0005CEI         YP1=YN1(III)                                                  16
0005E2I         CDK=DK(I,J,K)                                                 17
000C60EI        CDE=DE(I,J,K)                                                 18
00063AI         CDEN=DEN(I,J,K)                                               19
000666I         CALL WALLFN(IE,YP,YP1,CDK,CDE,CDEN,SINX(III),SINY(III),       20
               1     SINZ(III),F(I,J-1,K),AN(I,J,K),SU(I,J,K),SP(I,J,K),
               2     TAUN(III),YPLN(III),GEN(I,J,K),VISE(I,J,K),
               3     U(I,J-1,K),V(I,J-1,K),W(I,J-1,K),U(I,J,K),V(I,J,K),
               4     W(I,J,K),U(I,J+1,K),V(I,J+1,K),W(I,J+1,K),
               5    -SPK(I,J,K),SPK(I,J,K),ANV1(III),ANW1(III),TJO(I,J,K))
000AD0I         GO TO 150                                                     21
000AD6I       3 CONTINUE                                                      22
```

```
                 C-----EAST
000AD6I                YP=YN(III)                                                           23
000AEAI                YP1=YN1(III)                                                         24
000AFEI                CDK=DK(I,J,K)                                                        25
000B2AI                CDE=DE(I,J,K)                                                        26
000B56I                CDEN=DEN(I,J,K)                                                      27
000B82I                CALL WALLFN(IE,YP,YP1,CDK,CDE,CDEN,SINX(III),SINY(III),              28
                      1     SINZ(III),F(I+1,J,K),AN(I,J,K),SU(I,J,K),SP(I,J,K),
                      2     TAUN(III),YPLN(III),GEN(I,J,K),VISE(I,J,K),
                      3     U(I+1,J,K),V(I+1,J,K),W(I+1,J,K),U(I,J,K),V(I,J,K),
                      4     W(I,J,K),U(I-1,J,K),V(I-1,J,K),W(I-1,J,K),
                      5    -SPK(I,J,K),SPK(I,J,K),ANV1(III),ANW1(III),TJC(I,J,K))
000FECI                GO TO 150                                                            29
000FF2I           4    CONTINUE                                                             30
                 C-----WEST
000FF2I                YP=YN(III)                                                           31
001006I                YP1=YN1(III)                                                         32
00101AI                CDK=DK(I,J,K)                                                        33
001046I                CDE=DE(I,J,K)                                                        34
001072I                CDEN=DEN(I,J,K)                                                      35
00109EI                CALL WALLFN(IE,YP,YP1,CDK,CDE,CDEN,SINX(III),SINY(III),              36
                      1     SINZ(III),F(I-1,J,K),AN(I,J,K),SU(I,J,K),SP(I,J,K),
                      2     TAUN(III),YPLN(III),GEN(I,J,K),VISE(I,J,K),
                      3     U(I-1,J,K),V(I-1,J,K),W(I-1,J,K),U(I,J,K),V(I,J,K),
                      4     W(I,J,K),U(I+1,J,K),V(I+1,J,K),W(I+1,J,K),
                      5    -SPK(I,J,K),SPK(I,J,K),ANV1(III),ANW1(III),TJC(I,J,K))
001508I                GO TO 150                                                            37
00150EI           5    CONTINUE                                                             38
                 C-----TOP
00150EI                YP=YN(III)                                                           39
001522I                YP1=YN1(III)                                                         40
001536I                CDK=DK(I,J,K)                                                        41
001562I                CDE=DE(I,J,K)                                                        42
00158EI                CDEN=DEN(I,J,K)                                                      43
0015BAI                CALL WALLFN(IE,YP,YP1,CDK,CDE,CDEN,SINX(III),SINY(III),              44
                      1     SINZ(III),F(I,J,K+1),AN(I,J,K),SU(I,J,K),SP(I,J,K),
                      2     TAUN(III),YPLN(III),GEN(I,J,K),VISE(I,J,K),
                      3     U(I,J,K+1),V(I,J,K+1),W(I,J,K+1),U(I,J,K),V(I,J,K),
                      4     W(I,J,K),U(I,J,K-1),V(I,J,K-1),W(I,J,K-1),
                      5    -SPK(I,J,K),SPK(I,J,K),ANV1(III),ANW1(III),TJC(I,J,K))
001A18I                GO TO 150                                                            45
001A1EI           6    CONTINUE                                                             46
                 C-----BOTTOM
001A1EI                YP=YN(III)                                                           47
001A32I                YP1=YN1(III)                                                         48
001A46I                CDK=DK(I,J,K)                                                        49
001A72I                CDE=DE(I,J,K)                                                        50
001A9EI                CDEN=DEN(I,J,K)                                                      51
001ACAI                CALL WALLFN(IE,YP,YP1,CDK,CDE,CDEN,SINX(III),SINY(III),              52
                      1     SINZ(III),F(I,J,K-1),AN(I,J,K),SU(I,J,K),SP(I,J,K),
                      2     TAUN(III),YPLN(III),GEN(I,J,K),VISE(I,J,K),
                      3     U(I,J,K-1),V(I,J,K-1),W(I,J,K-1),U(I,J,K),V(I,J,K),
                      4     W(I,J,K),U(I,J,K+1),V(I,J,K+1),W(I,J,K+1),
                      5    -SPK(I,J,K),SPK(I,J,K),ANV1(III),ANW1(III),TJC(I,J,K))
001F28I         150    CONTINUE                                                             53
001F40I                RETURN                                                               54
001F46I                END                                                                  55
```

```
NO  ERRORS:F7D  RDS-01.DC  SUBROUTINE  BOUND     02/21/86  10:00:46  TABLE SPACE:   7 KB
STATEMENT BUFFER:   20 LINES/1321 BYTES    STACK SPACE: 136 WORDS
SINGLE PRECISION FLOATING PT SUPPORT REQUIRED FOR EXECUTION
```

```
000000I        SUBROUTINE WALLFN(IE,YP,YP1,CK1,DE1,DEN1,SINX1,SINY1,                1701
              1     SINZ1,F1,AN1,SU1,SP1,                                           1702
              2     TAUN1,YPLN1,GEN1,VISE1,                                         1703
              3     U1,V1,W1,UC,VO,                                                 1704
              4     WO,U2,V2,W2,                                                    1705
              5     SUK1,SPK1,ANV2,ANW2,TJO1)                                       1706
000004I        COMMON                                                               1707
              1/PRCP/ VISE(21,18,10),CEN(21,18,10),VISC,DENIN,FLOWIN                1708
              1/TUR/ SIGK,SIGE,CMU,C1,C2,CMU1,CMU2,E,CK,HINUM,SMNUM,ANV1(800),      1709
              2 YN(800),YN1(800),SINX(800),SINY(800),SINZ(800),ANW1(800),           1710
              3 YPLN(800),TAUN(800),ISC(800),JBC(800),KSC(800),IITY(300),           1711
              4 TAUW(800),GEN(21,18,10),MC(21,18,10),IJLO(21,18,10),IITO            1712
         C-----WALL FUNCTIONS USING LOGARITHMIC WALL LAW                            1713
000004I        GO TO (1,2,3,4,5,6,7), IE                                            1714      1
00003EI      1 CONTINUE                                                             1715      2
         C-----U                                                                    1716
00003EI        SQRTK=SQRT(CK1)                                                      1717      3
00005AI        YPLN1=DEN1*SQRTK*CMU1*YP/VISC                                        1718      4
00007EI        IF(YPLN1 .LE. 11.63) GO TO 111                                       1719      5
000096I        TMULT=DEN1*CMU1*SQRTK*CK/ALOG(E*YPLN1)                               1720      7
0000E0I        GO TO 112                                                            1721      8
0000E6I    111 TMULT=VISC/YP                                                        1722      9
0000F8I    112 TAUN1=-TMULT                                                         1723     10
00010AI        PTA=TJC1*SINX1*TMULT/YP1                                             1724     11
000123I        SP1=SP1-PTA                                                          1725     12
00013AI        SU1=SU1+PTA*F1                                                       1726     13
000152I        P1=SQRT(1.-SINX1**2)                                                 1727     14
00018AI        ANV2=AN1*SQRT(1.-SINY1**2)                                           1728     15
0001C8I        ANW2=AN1*SQRT(1.-SINZ1**2)                                           1729     16
000208I        AN1=P1*AN1                                                           1730     17
00021AI        RETURN                                                               1731     18
000220I      2 CONTINUE                                                             1732     19
         C-----V                                                                    1733
000220I        TMULT=-TAUN1                                                         1734     20
000232I        PTA=TJC1*SINY1*TMULT/YP1                                             1735     21
000250I        SP1=SP1-PTA                                                          1736     22
000262I        SU1=SU1+PTA*F1                                                       1737     23
00027AI        AN1=ANV2                                                             1738     24
000286I        RETURN                                                               1739     25
00028CI      3 CONTINUE                                                             1740     26
         C-----W                                                                    1741
00028CI        TMULT=-TAUN1                                                         1742     27
00029EI        PTA=TJC1*SINZ1*TMULT/YP1                                             1743     28
0002BCI        SP1=SP1-PTA                                                          1744     29
0002CEI        SU1=SU1+PTA*F1                                                       1745     30
0002E6I        AN1=ANW2                                                             1746     31
0002F2I        RETURN                                                               1747     32
0002F8I      4 CONTINUE                                                             1748     33
         C-----TM                                                                   1749
0002F8I        RETURN                                                               1750     34
0002FEI      5 CONTINUE                                                             1751     35
         C-----K                                                                    1752
0002FEI        SQRTK=SQRT(CK1)                                                      1753     36
00031AI        IF(YPLN1 .LE. 11.63) GO TO 511                                       1754     37
000332I        DITM=DEN1*CMU2*SQRTK*ALOG(E*YPLN1)/(CK*YP)                           1755     39
000382I        GO TO 512                                                            1756     40
000388I    511 DITM=DEN1*CMU2*SQRTK*YPLN1/YP                                        1757     41
```

```
58  0003ACI    512  CONTINUE                                                  1758    42
59  0003AC2         DDU=U0-U1                                                 1759    43
60  00038EI         DDV=V0-V1                                                 1760    44
61  0003D0I         DDW=W0-W1                                                 1761    45
62  0003E2I         P1=DDU**2+DDV**2+DDW**2                                   1762    46
63  00042AI         DDU=U2-U1                                                 1763    47
64  00043CI         DDV=V2-V1                                                 1764    48
65  00044EI         DDW=W2-W1                                                 1765    49
66  000460I         P2=DDU**2+DDV**2+DDW**2                                   1766    50
67  0004A8I         YP2=2*YP1-YP                                              1767    51
68  0004C0I         GENR=GEN1-VISE1*P2/YP2/YP2                                1768    52
69  0004E6I         IF(GENR .LE. 0.0) GENR=0.0                                1769    53
70  000504I         SU1=TJC1*(SUK1*CK1+(P1*TAUN1**2/VISE1+GENR))              1770    55
71  00053EI         SP1=TJC1*(SPK1-CITM)                                      1771    56
72  000556I         AN1=0.0                                                   1772    57
73  000562I         RETURN                                                    1773    58
74  000568I      6  CONTINUE                                                  1774    59
75             C-----E                                                        1775
76  000568I         TERM=CMU2/(CK*YP)                                         1776    60
77  000582I         SU1=HINUM*TERM*CK1**1.5                                   1777    61
78  000582I         SP1=-HINUM                                                1778    62
79  0005C4I         RETURN                                                    1779    63
80  0005CAI      7  CONTINUE                                                  1780    64
81             C-----PF                                                       1781
82  0005CAI         RETURN                                                    1782    65
83  0005D0I         END                                                       1783    66

NO  ERRORS:F7D  R05-01.0C  SUBROUTINE  WALLFN    02/21/86  10:01:12  TABLE SPACE:   5 KB
STATEMENT BUFFER:   20 LINES/1321 BYTES    STACK SPACE: 122 WORDS
SINGLE PRECISION FLOATING PT SUPPORT REQUIRED FOR EXECUTION
```

```
000000I          SUBROUTINE SYMOUT(IC,IE,IS,IT,JS,JT,KS,KT)
000004I          COMMON
                1/VAR/U(21,18,10),V(21,18,10),P(21,18,10),DK(21,18,10),
                2  DE(21,18,10),ERRU,ERRV,ERRM,ERRK,ERRE,ERRW,
                3  PP(21,18,10),W(21,18,10),TM(21,18,10)
                1/PRCP/ VISE(21,18,10),CEN(21,18,10),VISC,DENIN,FLOWIN
                1/PCOR/ DU(21,18,10),DV(21,18,10),DW(21,18,10)
                1/COEF/ AP(21,18,10),SU(21,18,10),SP(21,18,10),SUK(21,18,10),
                2    SPK(21,18,10),AE(21,18,10),AW(21,18,10),AN(21,18,10),
                3    AS(21,18,10),AT(21,18,10),AB(21,18,10),APO(21,18,10)
000004I          COMMON
                1/TRAN/ X(21,18,10),Y(21,18,10),Z(21,18,10),TJO(21,18,10),
                2    CX(21,18,10),CY(21,18,10),CZ(21,18,10),
                3    EX(21,18,10),EY(21,18,10),EZ(21,18,10),
                3    SX(21,18,10),SY(21,18,10),SZ(21,18,10)
                1/UNSTDY/UO(21,18,10),VO(21,18,10),WO(21,18,10),DKO(21,18,10),
                2    CEO(21,18,10),DENO(21,18,10),TMO(21,18,10)
                1/LIMT/ L,M,LT,MT,L1,L2,M1,M2,LO,MO,ISWU,ISWV,ISWP,ISWK,ISWE,
                2    ALU,ALV,ALP,ALK,ALE,ALVIS,ALW,N,N1,N2,NO,ISWW,IG,NT,ALC,CTT
          C-----SYMMETRIC, CYCLIC AND EXIT CONDITIONS AND LINK MODIFICATIONS
000004I          GO TO (1,2,3), IC                                            1
00002EI      1   CONTINUE                                                     2
          C-----BOTTOM
00002EI          K=1                                                          3
000036I          DO 10 I=1,L                                                  4
00004AI          DO 10 J=2,MT                                                 5
00005EI          U(I,J,K)=U(I,J,K+1)                                          6
0000ACI          V(I,J,K)=V(I,J,K+1)                                          7
0000FAI          W(I,J,K)=0.0                                                 8
000126I          TM(I,J,K)=TM(I,J,K+1)                                        9
000174I          DK(I,J,K)=DK(I,J,K+1)                                       10
0001C2I          DE(I,J,K)=DE(I,J,K+1)                                       11
000210I     10   CONTINUE                                                    12
          C-----EAST OUT (BASED ON INFLOW MASS FLOW RATE)
000240I          I=IT                                                        13
00024CI          FLOW=0.0                                                    14
000258I          ARDEN=0.0                                                   15
000264I          DO 50 J=2,JT                                                16
000278I          DO 50 K=2,KT                                                17
00028CI          UC=(V(I,J,K)+V(I,J-1,K)+V(I,J,K-1)+V(I,J-1,K-1))*0.25       18
00033CI          DENC=(DEN(I,J,K)+DEN(I,J-1,K)+DEN(I,J,K-1)+DEN(I,J-1,K-1))*0.25   19
0003ECI          P1=(X(I,J,K)+X(I,J,K-1)-X(I,J-1,K)-X(I,J-1,K-1))*0.5       20
00049CI          P2=(Y(I,J,K)+Y(I,J,K-1)-Y(I,J-1,K)-Y(I,J-1,K-1))*0.5       21
00054CI          P3=(Z(I,J,K)+Z(I,J,K-1)-Z(I,J-1,K)-Z(I,J-1,K-1))*0.5       22
0005FCI          Q1=(X(I,J,K)+X(I,J-1,K)-X(I,J,K-1)-X(I,J-1,K-1))*0.5       23
0006ACI          Q2=(Y(I,J,K)+Y(I,J-1,K)-Y(I,J,K-1)-Y(I,J-1,K-1))*0.5       24
00075CI          Q3=(Z(I,J,K)+Z(I,J-1,K)-Z(I,J,K-1)-Z(I,J-1,K-1))*0.5       25
00080CI          AREA=SQRT(P1*P1+P2*P2+P3*P3)*SQRT(Q1*Q1+Q2*Q2+Q3*Q3)        26
0008A8I          FLOW=FLOW+DENC*AREA*UC                                      27
0008C6I          ARDEN=ARDEN+DENC*AREA                                       28
0008DEI     50   CONTINUE                                                    29
00090EI          UINC=(FLOW-FLOWIN)/ARDEN                                    30
000926I          DO 60 J=2,JT                                                31
00093AI          DO 60 K=2,KT                                                32
00094EI          U(I+1,J,K)=U(I,J,K)                                         33
00099EI          V(I+1,J,K)=V(I,J,K)-UINC                                    34
0009F4I          W(I+1,J,K)=W(I,J,K)                                         35
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
58 000A44I          DK(I+1,J,K)=DK(I,J,K)                    1841 36
59 000A94I          DE(I+1,J,K)=DE(I,J,K)                    1842 37
60 000AE4I     60   CONTINUE                                 1843 38
61 000B14I          RETURN                                   1844 39
62            C-----LINK COEFF. MODIFICATIONS                1845
63 000B1AI      2   CONTINUE                                 1846 40
64            C-----EAST OUT                                 1847
65 000B1AI          I=IT                                     1848 41
66 000B26I          DO 200 J=2,JT                            1849 42
67 000B3AI          DO 200 K=2,KT                            1850 43
68 000B4EI          AE(I,J,K)=0.0                            1851 44
69 000B7AI    200   CONTINUE                                 1852 45
70 000BAAI          RETURN                                   1853 46
71            C-----UPDATE UNSTEADY COEFF.                   1854
72 000BB0I      3   CONTINUE                                 1855 47
73 000BB0I          IF(DTT .NE. 0.0) GO TO 301               1856 48
74 000BC8I          DO 300 I=IS,IT                           1857 50
75 000BE0I          DO 300 J=JS,JT                           1858 51
76 000BF8I          DO 300 K=KS,KT                           1859 52
77 000C10I    300   APO(I,J,K)=0.0                           1860 53
78 000C84I          RETURN                                   1861 54
79 000C8AI    301   CONTINUE                                 1862 55
80 000C8AI          DO 310 I=IS,IT                           1863 56
81 000CA2I          DO 310 J=JS,JT                           1864 57
82 000CBAI          DO 310 K=KS,KT                           1865 58
83 000CD2I          APO(I,J,K)=DENO(I,J,K)/DTT               1866 59
84 000D24I          UO(I,J,K)=U(I,J,K)                       1867 60
85 000D70I          VO(I,J,K)=V(I,J,K)                       1868 61
86 000DBCI          WO(I,J,K)=W(I,J,K)                       1869 62
87 000E08I          TMO(I,J,K)=TM(I,J,K)                     1870 63
88 000E54I          DKO(I,J,K)=DK(I,J,K)                     1871 64
89 000EA0I          DEO(I,J,K)=DE(I,J,K)                     1872 65
90 000EECI    310   CONTINUE                                 1873 66
91 000F34I          RETURN                                   1874 67
92 000F3AI          END                                      1875 68

NO ERRORS:F7D R05-01.0C SUBROUTINE SYMOUT 02/21/86 10:01:59 TABLE SPACE: 7 KB
STATEMENT BUFFER: 20 LINES/1321 BYTES STACK SPACE: 131 WORDS
SINGLE PRECISION FLOATING PT SUPPORT REQUIRED FOR EXECUTION
```

```
000000I          SUBROUTINE WALVAL(PW,IS,IT,JS,JT,KS,KT,F)
000004I          DIMENSION F(21,18,10)
000004I          COMMON
                1/TUR/ SIGK,SIGE,CMU,C1,C2,CMU1,CMU2,E,CK,HINUM,SMNUM,ANV1(800),
                2 YN(800),YN1(800),SINX(800),SINY(800),SINZ(800),ANW1(800),
                3 YPLN(800),TAUN(800),I3C(800),JBC(800),KBC(800),IITY(800),
                4 TAUW(800),GEN(21,18,10),MC(21,18,10),IJLO(21,18,10),IITO
              C-----ASSIGN WALL VALUES
000004I          DO 10 J=JS,JT
000028I          DO 10 K=KS,KT
000040I          F(IS-1,J,K)=PW*F(IS,J,K)
0000A2I       10 F(IT+1,J,K)=PW*F(IT,J,K)
000134I          DO 20 I=IS-1,IT+1
000152I          DO 20 K=KS,KT
00016AI          F(I,JS-1,K)=PW*F(I,JS,K)
0001CCI       20 F(I,JT+1,K)=PW*F(I,JT,K)
00025EI          DO 30 I=IS-1,IT+1
00027CI          DO 30 J=JS-1,JT+1
00029AI          F(I,J,KS-1)=PW*F(I,J,KS)
0002FAI       30 F(I,J,KT+1)=PW*F(I,J,KT)
00038AI          DO 40 III=1,IITO
00039EI          I=IBC(III)
0003B2I          J=JBC(III)
0003C6I          K=KBC(III)
0003DAI          GO TO (1,2,3,4,5,6), IITY(III)
000416I        1 F(I,J+1,K)=PW*F(I,J,K)
000478I          GO TO 40
00047EI        2 F(I,J-1,K)=PW*F(I,J,K)
0004E0I          GO TO 40
0004E6I        3 F(I+1,J,K)=PW*F(I,J,K)
000548I          GO TO 40
00054EI        4 F(I-1,J,K)=PW*F(I,J,K)
0005B0I          GO TO 40
0005B6I        5 F(I,J,K+1)=PW*F(I,J,K)
000616I          GO TO 40
00061CI        6 F(I,J,K-1)=PW*F(I,J,K)
00067CI       40 CONTINUE
000694I          RETURN
00069AI          END

NO  ERRORS:F7D  R05-01.0C  SUBROUTINE  WALVAL    02/21/86  10:02:24  TABLE SPACE:   3 KB
STATEMENT BUFFER:   20 LINES/1321 BYTES    STACK SPACE: 154 WORDS
SINGLE PRECISION FLOATING PT SUPPORT REQUIRED FOR EXECUTION
```

APPROVAL

A COMPUTER CODE FOR THREE-DIMENSIONAL INCOMPRESSIBLE FLOWS USING NONORTHOGONAL BODY-FITTED COORDINATE SYSTEMS

By Y. S. Chen

The information in this report has been reviewed for technical content. Review of any information concerning Department of Defense or nuclear energy activities or programs has been made by the MSFC Security Classification Officer. This report, in its entirety, has been determined to be unclassified.

G. F. McDONOUGH
Director, Systems Dynamics Laboratory

☆U.S. GOVERNMENT PRINTING OFFICE 1986—631-058/20122