

NASA 30. 37 72

FEB 21 1982

NASA Contractor Report 3442

COMPLETED

139

A Computer Code for Swirling
Turbulent Axisymmetric
Recirculating Flows in Practical
Isothermal Combustor Geometries

D. G. Lilley and D. L. Rhode

GRANT NAG 3-74
FEBRUARY 1982

NASA

NASA Contractor Report 3442

**A Computer Code for Swirling
Turbulent Axisymmetric
Recirculating Flows in Practical
Isothermal Combustor Geometries**

D. G. Lilley and D. L. Rhode
Oklahoma State University
Stillwater, Oklahoma

Prepared for
Lewis Research Center
under Grant NAG 3-74

NASA
National Aeronautics
and Space Administration

**Scientific and Technical
Information Branch**

1982

<u>CONTENTS</u>	<u>Page</u>
1. INTRODUCTION	1
1.1 The Problem	1
1.2 Previous Work	3
1.3 Outline of the Present Contribution	4
2. THE MATHEMATICAL PROBLEM	6
2.1 The Governing Equations	6
2.2 The Solution Technique and Finite Difference Formulation	9
2.3 Boundary Conditions	17
2.4 Wall Functions and the Effect of Swirl	23
2.5 Boundary Conditions Summary	28
2.6 Solution Procedure	30
3. THE STARPIC COMPUTER PROGRAM	34
3.1 Outstanding Features	34
3.2 The MAIN Subprogram	40
3.3 Subroutine INIT	43
3.4 Subroutine PROPS	43
3.5 Subroutines CALCU and CALCV	47
3.6 Subroutine CALCP	48
3.7 The Other CALC Subroutines	49
3.8 Subroutine LISOLV	50
3.9 Subroutine PRINT	51
3.10 Subroutine PROMOD	51
3.11 Subroutine STRMFN	55
3.12 Subroutine PLOT	56

4.	USER'S GUIDE	58
4.1	Problem Specification	58
4.2	Iteration and Accuracy	61
4.3	The Sample Computation	64
5.	CLOSURE	67
6.	REFERENCES	68
7.	NOMENCLATURE	72
	APPENDICES	74
A.	Derivation of the Hybrid Differencing Scheme	74
B.	Derivation of Eq. (A7) from Eq. (A6)	80
C.	The Superiority of the Hybrid Differencing Scheme	81
D.	Fortran Symbol List	86
E.	Computer Program Listing	97

1. INTRODUCTION

1.1 The Problem

In combustion chamber development, designers are aided by experiments but, as a supplement to them, economical design and operation can be greatly facilitated by the availability of prior predictions of the flowfield. These may be obtained by use of a mathematical model incorporating a numerical finite difference prediction procedure. This work combines the rapidly developing fields of theoretical combustion aerodynamics and computational fluid dynamics, and its improvement and use will significantly increase understanding and reduce the time and cost of development.¹⁻⁶

The present research work is concerned with complementary experimental and theoretical studies and is described in a recent paper.⁷ The problem being investigated is restricted to steady turbulent flow in axisymmetric geometries, under low speed and nonreacting conditions - a study area highlighted recently⁸⁻¹⁰ as a fundamental research requirement in combustion modeling. The particular problem is concerned with turbulent flow of a given turbulence distribution in a round pipe entering an expansion into another round pipe, as illustrated in Fig. 1. The in-coming flow may possess a swirl component of velocity via passage through swirl vanes at angle ϕ [equal approximately to $\tan^{-1}(w_{in}/u_{in})$], and the side-wall may slope at an angle α , to the main flow direction. The resulting flowfield domain may possess a central toroidal recirculation zone CTRZ in the middle of the region on the axis, in addition to the possibility of a corner recirculation zone CRZ near the upper corner provoked by the rather sudden enlargement of the cross-sectional area. Of vital concern is the characterization of flows of this type in terms of the effects of side-wall angle α , degree of swirl ϕ , turbulence intensity k_{in} of the inlet stream and expansion ratio D/d on the resulting flowfield in terms of its time-mean and turbulence quantities.

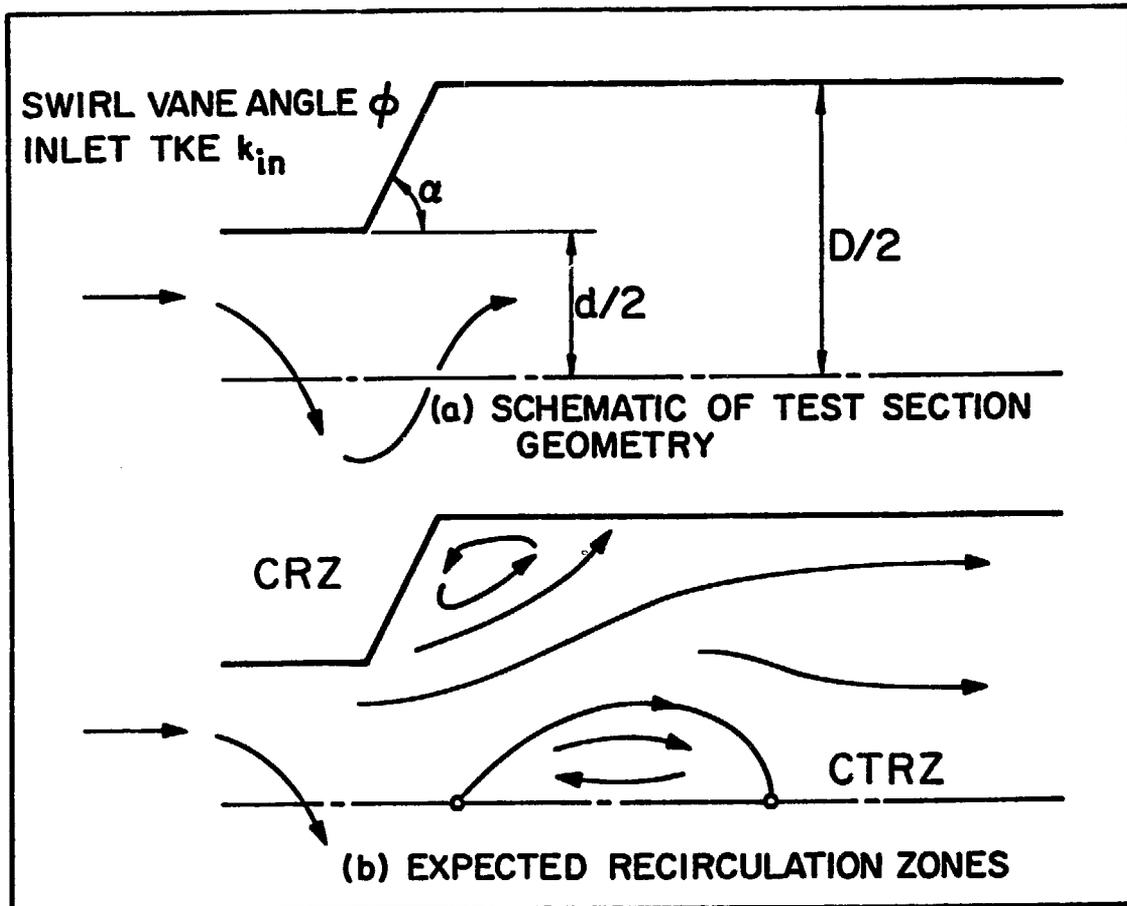


Fig. 1. The flowfield being investigated.

Such problems have received little attention, yet there is a definite need for work in this area even under nonreacting flow conditions.

The objective of the present document is to present details of the computational solution procedure which has been developed to perform the prediction function in the configuration just described. The finally developed computer program (written in Fortran 4) is code-named STARPIC (mnemonic for swirling turbulent axisymmetric recirculating flows in practical isothermal combustor geometries) and it is the purpose of this document to serve as a user's manual and describe it in sufficient detail for it to be used and amended by persons whose interest is confined more to the basis and results of the computations.

1.2 Previous Work

A mathematical solution of the flowfield of interest should provide results, if possible more cheaply, quickly, and correctly than possible by other means (for example, experiments on real-life systems or models). In order to achieve this, the model should simulate the flow in all its important respects (geometry, boundary conditions, physical properties of gases, turbulence, etc.) and provide a means whereby the governing equations may be solved. Mathematical models of steadily increasing realism and refinement are now being developed, both in the dimensionality of the model (together with the computational procedures) and in problems associated with the simulation of the physical processes occurring. Clearly there are two areas of difficulty: the simulation and solution.

Several previous publications discuss these problems at length, in terms of practical application,^{5,6} turbulence modeling^{11,12} and numerical solution of 2-D axisymmetric problems via the stream function ~ vorticity or primitive pressure ~ velocity approach.¹³⁻¹⁵ Whereas the former approach, used in the 1968 computer program from Imperial College for example¹⁶, reduces by one

the number of equations to be solved and eliminates the troublesome pressure (at the expense of trouble with the vorticity equation), the preferred approach now is SIMPLE (mnemonic for semi-implicit method for pressure linked equations) which focuses attention directly on the latter variables. Because it possesses many advantages, the present work has been developed immediately on this new technique, the basic ideas of which have been embodied into the 1974 Imperial College TEACH (teaching elliptic axisymmetric characteristics heuristically) computer program.¹⁷ Advances to this code have been made elsewhere and applied to other 2-D¹⁸⁻²⁵ and 3-D²⁶⁻²⁹ problems of interest to the combustor designer.

1.3 Outline of the Present Contribution

The present document serves as a user's manual for the STARPIC computer program, with three subsequent major sections dealing with the basic ideas, the computer program and the user's guide. Consideration is given to recent work in the finite difference solution, via a primitive variable code, of axisymmetric swirl flow in the combustor geometry of Fig. 1, where the inlet expansion sidewall may slope obliquely to the central axis. Section 2 describes the mathematical problem, by providing the basic equations and assumptions for the flow, and discussing the primitive variable solution technique to be used. Section 3 concerns itself with the organization and structure of the computer program. Sub-sections of the program are here described and linked with the equations previously noted in the text; and explanations are given of the special features of the program. Readers primarily interested in using the program, without concerning themselves greatly with the computational details, can give their main attention to Section 4. This is a user's guide to the program and the emphasis is on those parts of the program which will require alteration to suit particular flow configura-

tions, to obtain the desired output, to test for accuracy and to improve it when necessary. Appendices are provided dealing with the hybrid differencing scheme and reasons for its superiority, a Fortran symbol list, a listing of the STARPIC computer program, The program output is included in a microfiche supplement at the back of the report.

2. THE MATHEMATICAL PROBLEM

2.1 The Governing Equations

The turbulent Reynolds equations for conservation of mass, momentum (in x , y and θ directions), turbulence energy k and turbulence dissipation rate ϵ , which govern the 2-D axisymmetric, swirling, steady flow may be taken as previously.¹⁹⁻¹⁵ The transport equations are all similar and contain terms for convection and diffusion (via turbulent flux terms) and source S_ϕ of a general variable (which contains terms describing the generation (creation) and consumption (dissipation) of ϕ). Introducing turbulent exchange coefficients and the usual turbulent diffusion-flux (stress ~ rate of strain type) laws, it can be shown that the similarity between the differential equations and their diffusion relations allows them all to be put in the common form:

$$\frac{1}{r} \left[\frac{\partial}{\partial x} (\rho u r \phi) + \frac{\partial}{\partial r} (\rho v r \phi) - \frac{\partial}{\partial x} (r \Gamma_\phi \frac{\partial \phi}{\partial x}) - \frac{\partial}{\partial r} (r \Gamma_\phi \frac{\partial \phi}{\partial r}) \right] = S_\phi \quad (1)$$

for $\phi = 1$ (continuity equation), u , v , w (three velocity components), k and ϵ (two turbulence quantities). The forms for the source term S_ϕ are given in Table 1, where certain quantities are defined as follows:

$$S^u = \frac{\partial}{\partial x} (\mu \frac{\partial u}{\partial x}) + \frac{1}{r} \frac{\partial}{\partial r} (r \mu \frac{\partial v}{\partial x}) \quad (2)$$

$$S^v = \frac{\partial}{\partial x} (\mu \frac{\partial u}{\partial r}) + \frac{1}{r} \frac{\partial}{\partial r} (r \mu \frac{\partial v}{\partial r}) \quad (3)$$

$$G = \mu \left[2 \left\{ \left(\frac{\partial u}{\partial x} \right)^2 + \left(\frac{\partial v}{\partial r} \right)^2 + \left(\frac{v}{r} \right)^2 \right\} + \left(\frac{\partial u}{\partial r} + \frac{\partial v}{\partial x} \right)^2 + \left\{ r \frac{\partial}{\partial r} (w/r) \right\}^2 + \left(\frac{\partial w}{\partial x} \right)^2 \right] \quad (4)$$

Implicit here is the use of the two-equations k - ϵ turbulence model^{11,12}

Table 1. The Form of the Source Term in the General Equation for ϕ (Eq. (i))

ϕ	S_ϕ
u	$-\frac{\partial p}{\partial x} + S^u$
v	$-\frac{\partial p}{\partial r} + \frac{\rho w^2}{r} - \frac{2\mu v}{r^2} + S^v$
w	$-\frac{\rho v w}{r} - \frac{w}{r^2} \frac{\partial}{\partial r} (r\mu)$
k	$G - C_D \rho \epsilon$
ϵ	$(C_1 \epsilon G - C_2 \rho \epsilon^2) / k$

$$\mu = C_{\mu} \rho k^2 / \epsilon + \mu_0 \quad (5)$$

$$\Gamma_{\phi} = \mu / \sigma_{\phi} \quad (6)$$

which is commonly used in computer codes for turbulent flow prediction.

These equations have to be solved for the time-mean pressure p and velocity components u , v , and w . Then other useful designer information like streamline plots showing breakaway and reattachment points, recirculation zones and stagnation points are easily produced. Streamline plots are obtained from the dimensionless stream function which is given by

$$\psi^* = \int_0^r u r dr / \int_0^{d/2} u r dr \quad (7)$$

This quantity is calculated for all u -cells, and the coordinates of points constituting each of eleven streamlines ($\psi^* = 0.0$ through $\psi^* = 1.0$) is stored for plotting. Earlier publications¹⁷⁻²¹ provide details of the present simulation and solution technique, and only highlights of the primitive-variable approach need be given here in the context of the specific problem being investigated.

All the linkages provide a high degree of non-linearity in the total problem, and given the numerical analysis of fluid flow its peculiar difficulty and flavor. The above equations do not alone specify the problem; additional information of two kinds is needed: initial and boundary conditions for all the dependent variables. Details of this are given in Section 2.3. The difficulties in solution spring mainly from the interlinkages between the ϕ 's. Those between the axial and radial velocity components are of a peculiar kind, each containing an unknown pressure

gradient and the components are linked additionally by another equation, that of mass conservation, in which pressure does not appear. A successful solution procedure is one which takes account of these interactions and ensures that successive adjustments, which must be made to one variable after another, form an involuntary convergent sequence.

2.2 The Solution Technique and Finite Difference Formulation

Solution may be via the stream function-vorticity or primitive pressure-velocity approach, and, as discussed in Section 1.2, the present solution technique is based on the latter. The basic TEACH computer program using SIMPLE with TDMA provides the starting point upon which the present work is based.¹⁷ The following features are incorporated into the program:

- (i) a finite difference procedure is used in which the dependent variables are the velocity components and pressure;
- (ii) the pressure is deduced from an equation which is obtained by the combination of the continuity equation and the momenta equations (yielding a new form of what is known in the literature as the Poisson equation for pressure);
- (iii) the idea is present at each iteration of a first approximation of u , v , and p followed by a succeeding correction;
- (iv) the procedure incorporates displaced grids for the axial and radial velocities u and v , which are placed between the nodes where pressure p and other variables are stored; and
- (v) an implicit line-by-line relaxation technique is employed in the solution procedure (requiring a tridiagonal matrix to be inverted in order to update a variable at all points along a column), using the TDMA (tri-diagonal matrix algorithm).

The incorporation of these enhances the accuracy and rapidity of convergence

of the finally developed computer program.

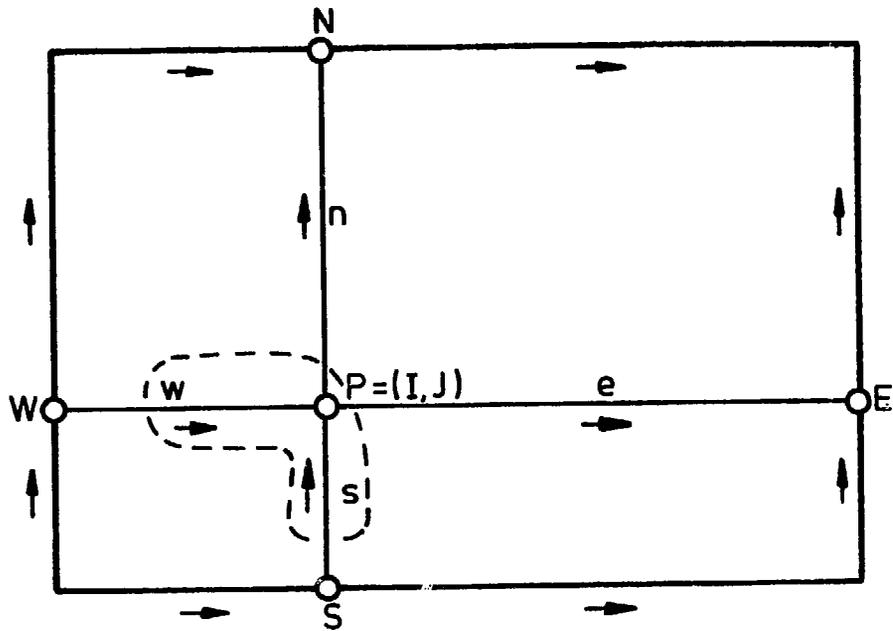
The finite difference equations are solved on a complex mesh illustrated in Fig. 2. The intersections, the point P for example, of the solid lines mark the grid nodes where all variables except the u and v velocity components are stored. The latter are stored at points which are denoted by arrows (and labeled w and s respectively) located midway between the grid intersections, and the boomerang-shaped envelope encloses a triad of points with reference location P at (I,J). Details of the special merits of this staggered grid system have been reported previously!^{7,18} The different control volumes C, U and V which are appropriate for the P, w and s locations respectively are given in Fig. 3.

The finite difference equations for each ϕ are obtained by integrating Eq. (1) over the appropriate control volume (centered about the location of ϕ) and expressing the result in terms of neighboring grid point values. The convection and diffusion terms become surface integrals of the convection and diffusive fluxes while the source term is linearized, resulting in

$$\begin{aligned} & [\rho u \phi - \Gamma_{\phi} \frac{\partial \phi}{\partial x}]_e A_e - [\rho u \phi - \Gamma_{\phi} \frac{\partial \phi}{\partial x}]_w A_w \\ & + [\rho v \phi - \Gamma_{\phi} \frac{\partial \phi}{\partial r}]_n A_n - [\rho v \phi - \Gamma_{\phi} \frac{\partial \phi}{\partial r}]_s A_s = [S_p \phi_p + S_U] Vol \end{aligned} \quad (8)$$

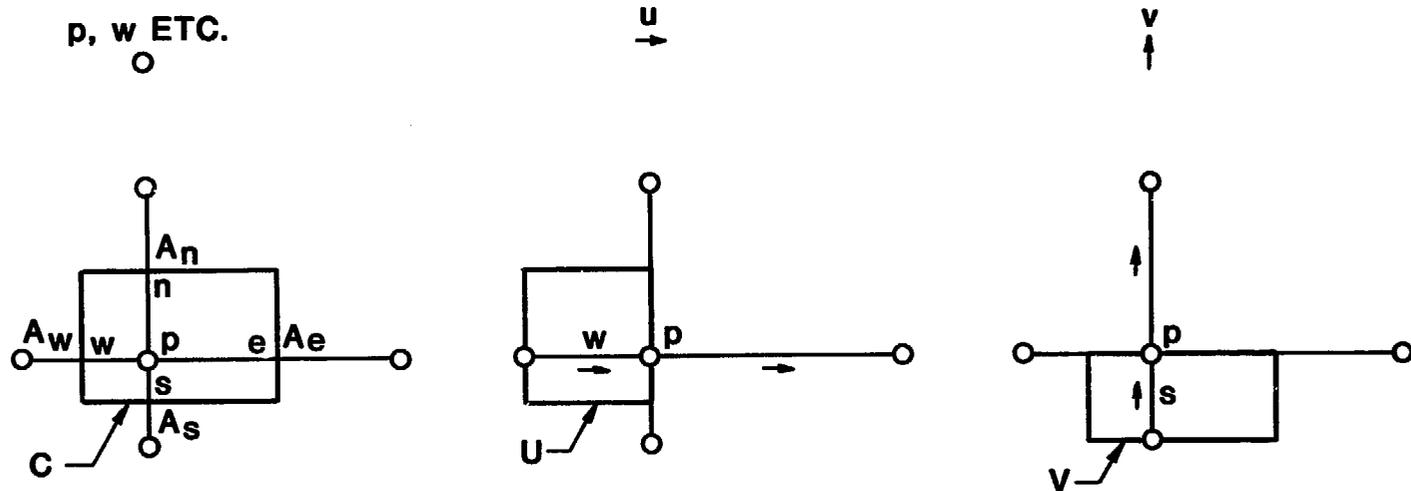
where S_p^{ϕ} and S_U^{ϕ} are tabulated in Table 2 and subscripts n, s, e and w refer to north, south, east and west cell faces.

The additional source terms in the equations for swirling flows may induce instability if one is not careful. Only terms which are always negative are permitted in the SP(I,J) expression in order to aid the convergence of the iterative solution procedure by increasing the diagonal dominance of the equation set to be solved at each iteration. Inspection of the arrangement given in Table 2 reveals that the forms given enjoy this desirable stabilizing



THREE GRIDS : FOR p, w ETC. - AT POSITION MARKED (O)
 FOR u VELOCITY - AT POSITION MARKED (\rightarrow)
 FOR v VELOCITY - AT POSITION MARKED (\uparrow)

Fig. 2. Staggered grid and notation for the rectangular computational mesh.



**CONTROL VOLUMES C, U, V FACE
AREAS A_n , A_s , A_e AND A_w FOR C, SIMILAR FOR U AND V**

Fig. 3. The three control volumes associated with points of the three grids.

Table 2. The Form of the Components of the Linearized Source Term*, The Cell Volume Integral $\int_V S_\phi dV = S_p^\phi \phi_p + S_U^\phi$ of Eq. (1).

ϕ	Γ_ϕ	S_p^ϕ/V	S_U^ϕ/V
1	0	0	0
u	μ	0	$S^u - \frac{\partial P}{\partial x}$
v	μ	$-2\frac{\mu}{r^2}$	$S^v + \frac{\rho w^2}{r} - \frac{\partial P}{\partial r}$
w	μ	0	$-\frac{\rho v w}{r} - \frac{w}{r^2} \frac{\partial}{\partial r} (r\mu)$
k	μ/σ_k	$-C_{\mu} C_D \rho^2 k/\mu$	G
ϵ	μ/σ_ϵ	$-C_2 \rho \epsilon/k$	$C_1 C_{\mu} G \rho k/\mu$

In this table certain quantities are defined as follows:

$$S^u = \frac{\partial}{\partial x} \left(\mu \frac{\partial u}{\partial x} \right) + \frac{1}{r} \frac{\partial}{\partial r} \left(r\mu \frac{\partial v}{\partial x} \right)$$

$$S^v = \frac{\partial}{\partial x} \left(\mu \frac{\partial u}{\partial r} \right) + \frac{1}{r} \frac{\partial}{\partial r} \left(r\mu \frac{\partial v}{\partial r} \right)$$

$$G = \mu \left[2 \left\{ \left(\frac{\partial u}{\partial x} \right)^2 + \left(\frac{\partial v}{\partial r} \right)^2 + \left(\frac{v}{r} \right)^2 \right\} + \left(\frac{\partial u}{\partial r} + \frac{\partial v}{\partial x} \right)^2 \right. \\ \left. + \left\{ r \frac{\partial}{\partial r} \left(\frac{w}{r} \right) \right\}^2 + \left(\frac{\partial w}{\partial x} \right)^2 \right]$$

*In this table, V stands for the cell control volume and $\mu \equiv \mu_{eff}$.

characteristics. A false source stabilizing trick, which has no effect on the final solution, is used in the form of a false source of magnitude:

$$S_{\text{false}} = |\dot{m}_{\text{net}}| (\phi_P^{\text{old}} - \phi_P) \quad (9)$$

That is used in the program although not detailed in Table 2.

For the representation of the convective and diffusive terms over the cell control volume surfaces a hybrid scheme is used,³⁰ which is a combination of the so-called central and upwind differences. Consider the transport across one face of a control volume. Figure 4 shows the western face of area A_W normal to the x-direction, which lies midway between the grid-points W and P distant δx apart. The contribution c_W to the surface integral by the western face area A_W , for example, is given by

$$c = \begin{cases} (\rho u)_W A_W (\phi_W + \phi_P)/2 - (\Gamma_\phi)_W A_W (\phi_P - \phi_W)/\delta x & \text{for } |Pe| < 2 \\ (\rho u)_W A_W \phi_W & \text{for } Pe \geq 2 \\ (\rho u)_W A_W \phi_P & \text{for } Pe \leq -2 \end{cases} \quad (10)$$

where $Pe = (\rho u)_W \delta x / (\Gamma_\phi)_W$ is a cell Peclet number calculated at the western face of the cell. Similar expressions are used at the other three faces of the cell. Details regarding the derivation and superiority of this differencing technique may be found in Appendixes A, B and C.

When the various terms are handled in this manner the following general equation is obtained:

$$a_P^\phi \phi_P = \sum_j a_j^\phi \phi_j + S_U^\phi \quad (11)$$

$$\text{where } a_P^\phi = \sum_j a_j^\phi - S_P^\phi$$

\sum_j = sum over N, S, E and W neighbors

thus linking each ϕ -value at a point P with its four neighboring values.

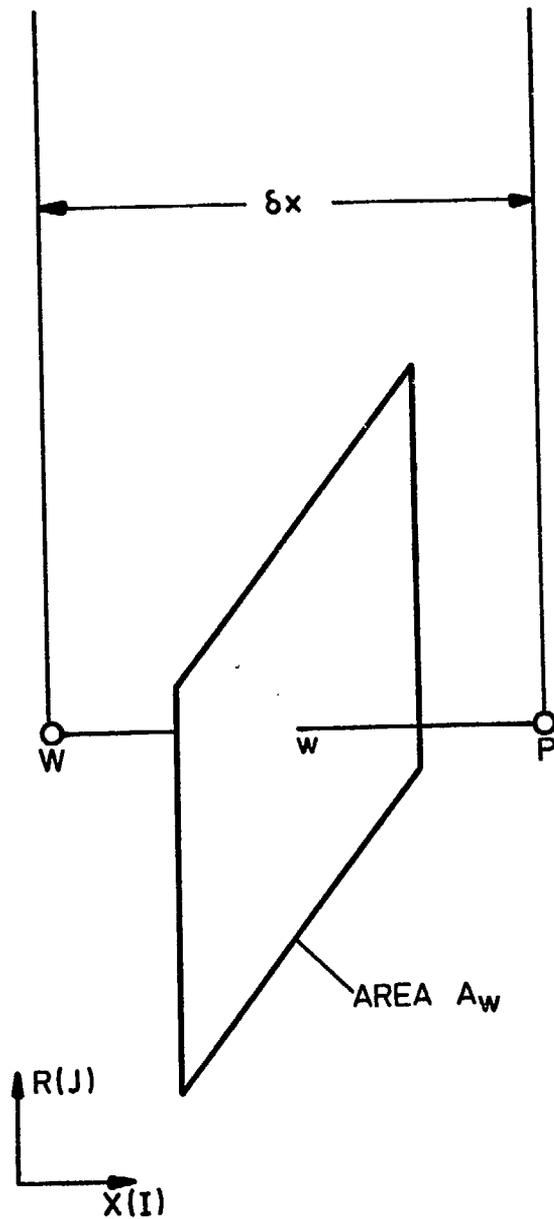


Fig. 4. One face of a control volume across which convective and diffusive fluxes are considered.

When $\phi = W, k$ and ϵ , equations of this type hold for values located at intersections of the grid system, but for $\phi = u$ and v there are two points of difference: (i) these variables are located to the west and south respectively of the point P and (ii) pressure gradient terms $\partial p/\partial x$ and $\partial p/\partial r$ respectively occur on the right hand side of their equations in addition to the usual source expression. Since the pressure field is not known when the u and v equations are required to be used at any stage of the iteration process, the best estimate so far of the pressure field p^* is used. Solving the u and v equations with this p^* field gives estimates u^* and v^* in the flowfield. These p^*, u^* and v^* fields constitute any any iteration stage a first approximation to the solution and they are immediately followed by a succeeding correction at each point P :

$$p_p = p_p^* + p_p' \quad (12)$$

$$u_p = u_p^* + D^u(p_W' - p_p') \quad (13)$$

$$v_p = v_p^* + D^v(p_S' - p_p') \quad (14)$$

where $D^u = A_w/a_p^u$

$$D^v = \frac{1}{2}(A_n + A_s)/a_p^v$$

These corrections can be applied only after the pressure correction p' field has been found and this is obtained from a Poisson equation for pressure which may be deduced from a combination of the continuity and momentum equations, resulting in finite difference equations like

$$a_p^p p_p' = \sum_j a_j^p p_j' + S_U^p \quad (15)$$

where $a_p^p = \sum_j a_j^p$

\sum_j = sum over N, S, E and W neighbors

S_U^p = term which results from mass sources of u^* and v^* fields.

Thus the application of Eqs. (13) and (14) at any iteration stage brings the first approximation obtained from solution of momentum equations into line with the requirement of continuity - it is here where the continuity equation is used, incorporated in fact in the derivation of the equation for p' , Eq. (15).

2.3 Boundary Conditions

There are several methods of numerically positioning an irregular boundary, such as a sloping boundary segment, in finite difference computer programs. In order to retain conceptual simplicity, the stairstep simulation shown in Fig. 5 is utilized. The figure shows an example of grid specification for the geometry under consideration. The flowfield is covered with a nonuniform rectangular grid system. Typically the boundary of the solution domain falls halfway between its immediate nearby parallel gridlines. Clearly, specification of the x and r coordinates of the gridlines, together with information concerned with the position of the sloping sidewall boundary (via specification of $JMAX(I)$ for each I) is sufficient to determine the flowfield of interest.

Up to this point the application of boundary conditions has not yet been considered, and the formulation has been concerned with regular internal mesh points. Insertion of correct boundary conditions requires amendment of the finite difference formulation for the near boundary points.¹⁶ For cells

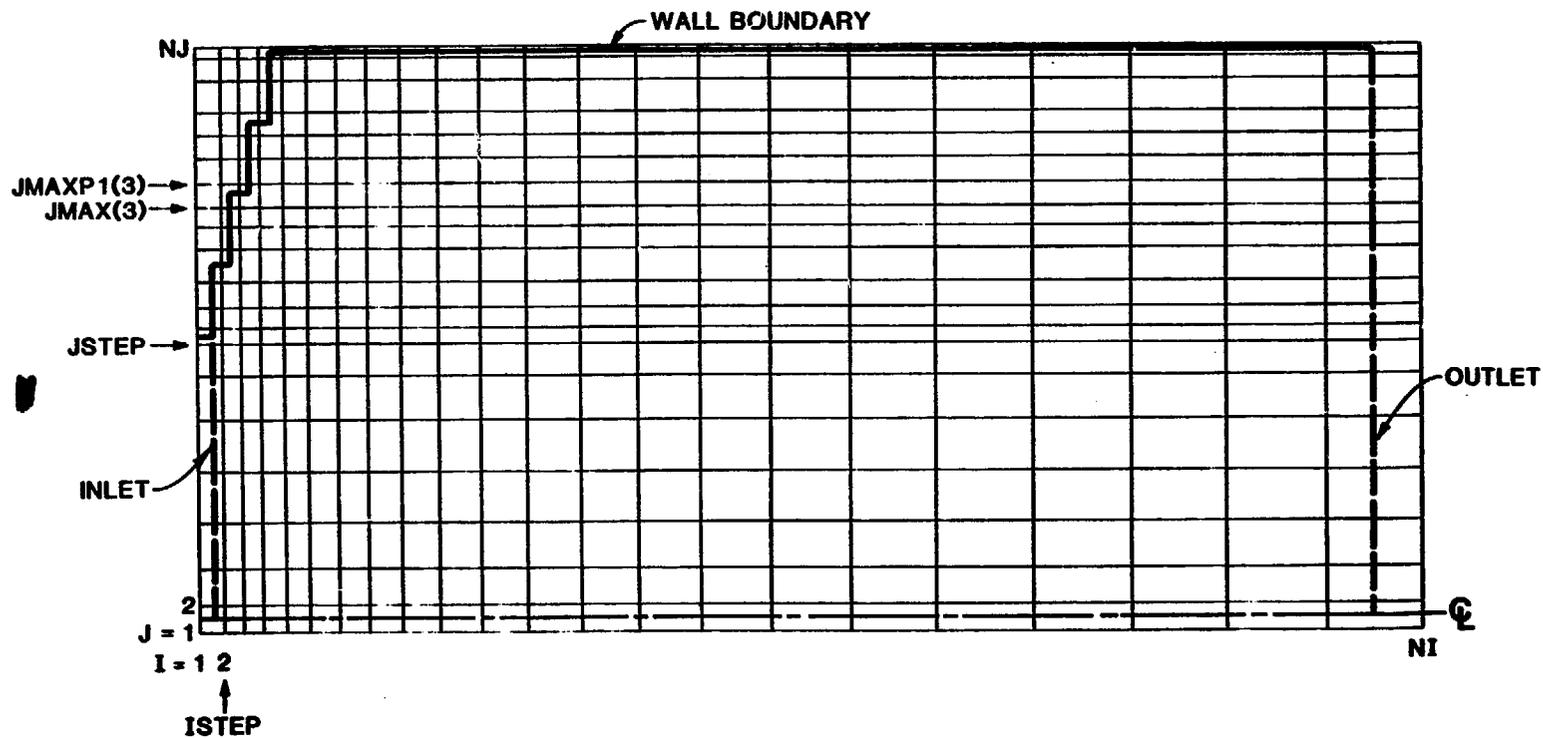


Fig. 5. An example of a grid system being employed to fit the flow domain.

adjacent to the west wall for example, as shown in Fig. 6, there is no convective flux through the cell west boundary which is coincident with an impervious wall. Hence the term for total ϕ -flux through this cell boundary in Eq. (8) is set to zero, and the diffusive flux term $\Gamma_\phi \partial\phi/\partial x$ is transferred to the right-hand side of the equation as a false source. In the case of a western wall boundary the normal P-W link is broken by setting:

$$a_W^\phi = 0 \quad (16)$$

and the correct expression is inserted by way of the S_U^ϕ and S_P^ϕ values:

- (i) If a value M for the diffusion terms $\Gamma_\phi \partial\phi/\partial x$ is specified for the western face of a cell adjacent to a western wall, then as seen in Eq. (8):

$$S_U^\phi = -M/\text{Vol} \quad (17)$$

$$S_P^\phi = 0 \quad (18)$$

It is clear that this specification results in correct ϕ balance for the boundary cell, and that if zero normal gradient is the required condition ($M = 0$) then Eq. (16) is sufficient in itself.

- (ii) If the value of ϕ on the boundary ϕ_B is specified, for a western boundary for example, then:

$$S_U^\phi = a_W' \phi_B \quad (19)$$

$$S_P^\phi = -a_W' \quad (20)$$

where

$$a_W' = (\Gamma_\phi)_W A_W / \delta x$$

It is clear that this specification gives the correct gradient-diffusion flux out of the western boundary of the cell.

(iii) If ϕ_p is required to be fixed at a value of ϕ_F then:

$$S_U^\phi = \phi_F \cdot 10^{30} \quad (21)$$

$$S_p^\phi = -10^{30} \quad (22)$$

to that these terms dominate in the equation for ϕ_p with solution $\phi_p = \phi_F$.

(iv) Velocities tangential to a wall boundary usually have boundary conditions of the type v_B given, τ_B (= tangential shear stress) given, or $\tau_B = \alpha v_p + \beta$ (from drag law for example). Any of these may be inserted by the usual treatment (break link and insert via linearized source).

(v) Velocities normal to a wall boundary are given zero values on the wall, but because this leads to incorrect gradient calculations for the first interior point away from the wall, zero normal gradient conditions are applied via breaking the appropriate coupling coefficient in the finite difference equation for this first interior point.

The above discussion about boundary condition types (i) - (v) has been concerned with a western wall boundary shown in Fig. 6. Analogous discussion is appropriate with a northern wall boundary shown in Fig. 7.

Conditions must be specified on the entire boundary around the solution domain, and special novelties concern the pressure correction p' , velocity components and the two turbulence quantities k and ϵ . At the inlets all variables are given definite fixed values (Dirichlet conditions) and at the outlet all variables are given zero normal gradient $\partial\phi/\partial x = 0$ (Neumann conditions) via

$$a_E^\phi = 0 \quad (23)$$

except radial velocity v which is set to zero.

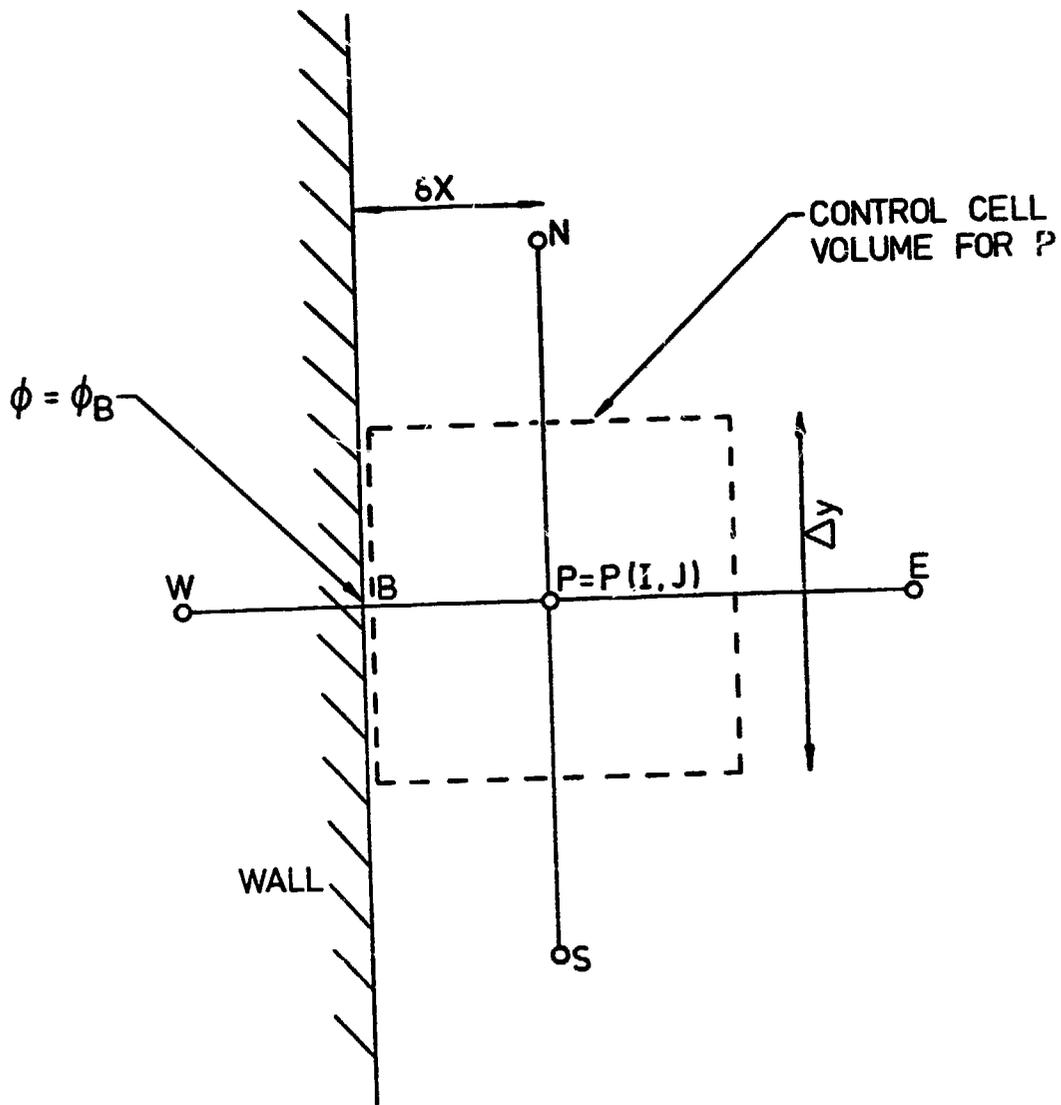


Fig. 6. Wall ϕ -value prescription for western wall.

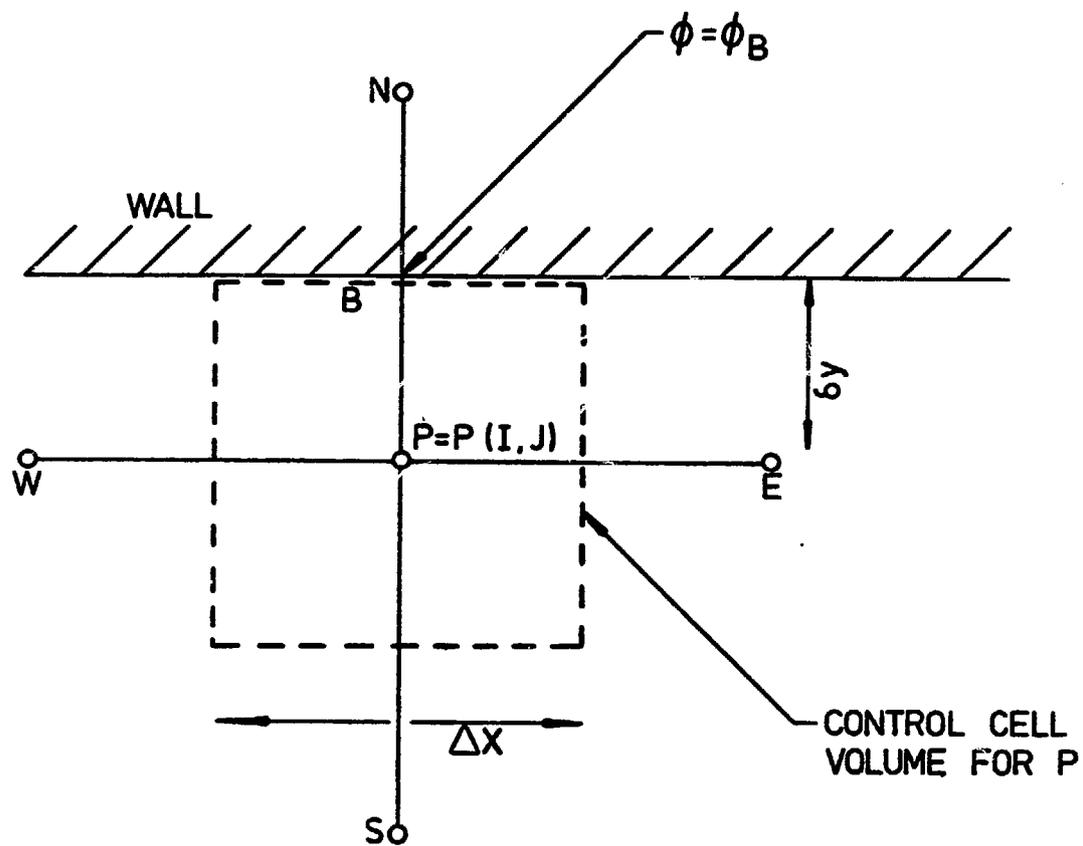


Fig. 7. Wall ϕ -value prescription for northern wall.

The u velocities are then readjusted slightly at each iteration so as to ensure overall mass conservation as compared with the inlet flow. At the symmetry axis $\partial\phi/\partial r = 0$ is assumed for all variables except v (which is given a definite zero value) and w (which is given an implied zero value as follows. Based on measurements³¹ of swirl velocity, the BC for w along the centerline specifies solid body rotation through linear interpolation of $w(I,2)$ between the latest value of $w(I,3)$ and zero for w on the centerline. Also p' is given zero normal gradient specification everywhere except at the inlet, where it has zero value: as far as the aerodynamics is concerned it is immaterial what the absolute pressure is, only pressure gradients entering the equations.

2.4 Wall Functions and the Effect of Swirl

To avoid the need for detailed calculations in the nearwall regions, equations are introduced to link velocities, k and ϵ on the wall to those in the logarithmic region. This one-dimensional Couette flow characterization of the flow (diffusion perpendicular to the wall is dominant) is extremely useful. Also it provides a way around this region of steep nonlinear variation of the variables and the fact that laminar and turbulent effects become of the same order of magnitude. The new equations introduced, called 'wall functions', are introduced and used in the finite difference calculations at near-wall points. They occur in the momenta equations and k -generation terms, and their implementation is discussed elsewhere^{11,12} together with appropriate near-wall ϵ specification.

The effect of swirl on wall function specification is handled as follows: The previous ideas are extended to find total tangential wall shear stress near boundaries [involving x and θ directions for an $r = \text{constant}$ wall, and r and θ directions for an $x = \text{constant}$ wall]. Then appropriate

components are deduced directly [for the u and w velocities which are tangential to an $r = \text{constant}$ wall, and v and w velocities which are tangential to an $x = \text{constant}$ wall]. The effects on u , v and w momentum equations are incorporated via the usual linearized source technique.

Northern Wall. The applicability of wall function formulations for swirl flows, especially for swirl momentum, appears uncertain a priori. However, from boundary layer velocity measurements in turbulent, swirling, pipe flows Backshall and Landis³¹ showed that total tangential velocity near the northern top wall $V = (u^2 + w^2)^{1/2}$ is correlated by the universal velocity profile

$$V^+ = (1/\kappa) \ln (E Y^+) \quad (24)$$

where κ and E are constants. The dimensionless total velocity V^+ and total distance Y^+ are obtained by nondimensionalizing with respect to the total shear velocity $(\tau_t/\rho)^{1/2}$. Hence the total tangential north wall shear stress $\tau_t = (\tau_{rx}^2 + \tau_{r\theta}^2)^{1/2}$ is obtained from

$$V(\tau_k \rho)^{1/2} / \tau_t = (1/\kappa) \ln [E y (\tau_k / \rho)^{1/2} / \nu] \quad (25)$$

where τ_k is an approximation for τ_t very near the wall. The quantity τ_k is formulated by observing that convection and diffusion of turbulence kinetic energy are nearly always negligible in this region.¹¹ Deleting these terms from the k -transport equation and invoking isotropic viscosity μ_{eff} leads to the result

$$\tau_k = (C_D C_\mu)^{1/2} \rho k \quad (26)$$

Thus one obtains

$$\tau_t = - V_p \kappa \rho C_\mu^{\frac{1}{4}} C_D^{\frac{1}{4}} k_p^{\frac{1}{2}} / \ln(E Y_p^+) \quad (27)$$

where the negative sign is inserted since τ_t and V_p must have opposite directions. The quantities with subscript P are evaluated at the appropriate near-wall points as shown in Fig. 7 for this case of a northern wall.

The north wall diffusion flux for the x-momentum equation is $\mu_{\text{eff}} \partial u / \partial r$. The τ_{rx} component of τ_t is given by

$$\tau_{rx} = \mu_{\text{eff}} \left(\frac{\partial u}{\partial r} + \frac{\partial v}{\partial x} \right) \quad (28)$$

however $\partial v / \partial x$ approaches zero near the north wall.¹⁷ Hence τ_{rx} is the required diffusion flux term to be included as a false source for north wall u-cells; its wall function expression asserts the required boundary influence by indirectly incorporating the boundary condition $u_B = 0$. The above wall function formulation, Eq. (27), is multiplied by the factor $\cos \theta$ ($= u/V$) to obtain τ_{rx} , where θ is the angle between the total tangential velocity vector near the wall and the axial velocity vector ($\theta = \arctan w/u$). This is valid since θ is essentially constant near the wall.³² Thus this wall function for u is

$$\mu_{\text{eff}} \frac{\partial u}{\partial r} = [-\kappa \rho C_\mu^{\frac{1}{4}} C_D^{\frac{1}{4}} k_p^{\frac{1}{2}} / \ln(E Y_p^+)] u_p \quad (29)$$

which is precisely the same expression found for nonswirling flows in TEACH, although different values result due to swirl effects on k_p . Also, this result is consistent with an expression obtained by Date.³³

For w-cells along the north wall, the diffusion is $\mu_{\text{eff}} \partial w / \partial r$. This quantity appears in the expression for $\tau_{r\theta}$ which is

$$\tau_{r\theta} = \mu_{\text{eff}} \left(\frac{\partial w}{\partial r} - \frac{w}{r} \right) \quad (30)$$

Applying the factor $\sin \theta (=w/V)$ to the τ_t wall function yields the following expression for $\tau_{r\theta}$:

$$\tau_{r\theta} = [-\kappa \rho C_\mu^{1/4} C_D^{1/4} k_p^{1/2} / \ln(E Y_p^+)] w_p \quad (31)$$

Hence the north wall function for w is given by

$$\mu_{\text{eff}} \frac{\partial w}{\partial r} = [-\kappa \rho C_\mu^{1/4} C_D^{1/4} k_p^{1/2} / \ln(E Y_p^+)] + \mu_{\text{eff}} / r] w_p \quad (32)$$

Western wall. Momentum wall functions along the west wall are similarly formulated. The total tangential velocity is now $V = (v^2 + w^2)^{1/2}$, and τ_{xr} reduces to $\mu_{\text{eff}} \partial v / \partial x$. Equation (27) for τ_t is again obtained and the τ_{xr} wall function is formulated by employing the factor $\cos \phi$; ϕ is the angle between the total tangential velocity vector and the radial velocity vector ($\phi = \arctan w/v$). Thus one obtains the v-momentum expression:

$$\mu_{\text{eff}} \frac{\partial v}{\partial x} = [-\kappa \rho C_\mu^{1/4} C_D^{1/4} k_p^{1/2} / \ln(E Y_p^+)] v_p \quad (33)$$

The w-equation wall function on the western wall approximates $\mu_{\text{eff}} \partial w / \partial x$, which is $\tau_{x\theta}$. Here the factor $\sin \phi$ is used to obtain the necessary wall function as

$$\mu_{\text{eff}} \frac{\partial w}{\partial x} = [-\kappa \rho C_\mu^{1/4} C_D^{1/4} k_p^{1/2} / \ln(E Y_p^+)] w_p \quad (34)$$

Other details. In accordance with Gosman and Pun,¹⁷ if Y^+ for a cell adjacent to any solid boundary is less than 11.63, it lies within the laminar sublayer where the laminar viscosity dominates. Hence, the wall function employed for north wall u-cells, for example, is

$$\mu_{\text{eff}} \frac{\partial u}{\partial r} = -\mu_l u_p / \delta_p \quad (35)$$

where $u_B = 0$ has been employed. Similar expressions are analogously obtained for other momentum diffusion terms of the problem.

For turbulence energy, from Eq. (30)

$$k_p / |\tau_B| = (C_\mu C_D)^{-\frac{1}{2}} = \text{constant} \quad (36)$$

near a solid boundary which implies that the wall flux is zero. Accordingly, zero normal gradient prescription for k is appropriate for wall cells via Eq. (16). Also the shear stress wall functions are employed for the k -generation source G for k -cells along a wall. The general expression for G , given in Table 2, can alternately be expressed as

$$G = 2\mu_{\text{eff}} \left[\left(\frac{\partial u}{\partial x} \right)^2 + \left(\frac{\partial v}{\partial r} \right)^2 + \left(\frac{v}{r} \right)^2 \right] + [\tau_{rx}^2 + \tau_{r\theta}^2 + \tau_{x\theta}^2] / \mu_{\text{eff}} \quad (37)$$

For north wall cells $\tau_t^2 = \tau_{rx}^2 + \tau_{r\theta}^2$ which yields

$$G = 2\mu_{\text{eff}} \left[\left(\frac{\partial u}{\partial x} \right)^2 + \left(\frac{\partial v}{\partial r} \right)^2 + \left(\frac{v}{r} \right)^2 \right] + \tau_t^2 / \mu_{\text{eff}} + \mu_{\text{eff}} \left(\frac{\partial w}{\partial x} \right)^2 \quad (38)$$

where the previous wall function expression for τ_t is employed. Similarly, the result for west wall cells is

$$G = 2\mu_{\text{eff}} \left[\left(\frac{\partial u}{\partial x} \right)^2 + \left(\frac{\partial v}{\partial r} \right)^2 + \left(\frac{v}{r} \right)^2 \right] + \tau_t^2 / \mu_{\text{eff}} + \mu_{\text{eff}} \left(\frac{\partial w}{\partial r} - \frac{w}{r} \right)^2 \quad (39)$$

where $\tau_t^2 = \tau_{xr}^2 + \tau_{x\theta}^2$.

As for energy dissipation, near a wall the length scale is assumed to be proportional to the normal distance from the wall which leads to

$$\epsilon = \frac{(C_\mu C_D)^{3/4}}{C_D \kappa} \frac{k_p^{3/2}}{\delta_p} \quad (40)$$

This is the effective wall boundary condition on ϵ and in the code this value is fixed for near wall points via the linearized source procedure, using Eqs. (21) and (22).

2.5 Boundary Conditions Summary

Table 3 summarizes the boundary values used in the present study, which are incorporated into the computer program. The symbol N in the table signifies that zero normal gradient prescription is appropriate. Footnotes (a) to (g) are dealt with as follows:

- (a) Inlet values UIN etc. are directly specified and easily altered.
- (b) The axial velocities at the outlet are deduced from their immediate upstream values by adding a fixed amount of each of U(NIMI,J) (J = 1 to NJMI); this amount is calculated to ensure overall mass conservation.
- (c) Near wall tangential velocities are connected with their zero wall values by way of the tangential shear stress wall functions.
- (d) Near-axis swirl velocity at J = 2 is fixed for solid body rotation in terms of the previous w at J = 3 and its zero value on the axis.
- (e) Outlet values of k and ϵ are unimportant because of the limited upstream influence at high Reynolds numbers and zero gradient is

Table 3. Boundary Conditions (N stands for zero normal gradient specified)
 (Footnotes are detailed in section 2.3 of the text)

Algebraic Variable	Inlets ^(a)	Outlet	Top Wall	Side wall	Symmetry Axis
u	UIN	adjusted ^(b)	(c)	N	N
v	0	0	N	(c)	0
w	WIN	N	(c)	(c)	0 ^(d)
κ	TEIN	N ^(e)	N	N	N
ϵ	EDIN	N ^(e)	(f)	(f)	N
p'	0 ^(g)	N	N	N	N

specified. If the local Peclet number is greater than +2, there is no upstream influence because of the implications of the hybrid formulation of convection and diffusion, and the downstream convection dominates.

- (f) Near wall ϵ values are fixed using the length scale near the wall and the current value of k .
- (g) The pressure correction p' possesses zero normal gradient specification everywhere except the inlet. Just one internal pressure p specification at (2,2) is then required to allow all the pressures to be calculated.

This concludes the discussion of boundary condition specification; its implementation is returned to in Section 4 of the present text.

2.6 Solution Procedure

The finite difference equations and boundary conditions constitute a system of strongly-coupled simultaneous algebraic equations. Though they appear linear they are not since the coefficients and source terms are themselves functions of some of the variables, and the velocity equations are strongly linked through the pressure. The solution proceeds by the cyclic repetition of the following steps:

- (i) Guess the values of all variables including p^* . Hence calculate auxiliary variables like turbulent viscosity etc.
- (ii) Solve the axial and radial momentum equations to obtain u^* and v^* from equations like Eq. (11).
- (iii) Solve the pressure correction equation, Eq. (15), to obtain p' .
- (iv) Calculate the pressure p and the corrected velocities u and v from Eqs. (12) through (14).
- (v) Solve the equations like Eq. (11) for the other ϕ 's successively.

- (vi) Treat the new values of the variables as improved guesses and return to step (i); repeat the process until convergence.

In the solution procedure algebraic equations like Eq. (11) are solved many times, coefficient and source updating being carried out prior to each occasion. The practice used here is to make use of the well-known tridiagonal matrix algorithm (TDMA), whereby a set of equations, each with exactly three unknowns in a particular order except the first and last which have exactly two unknowns, may be solved sequentially. One considers the values at grid-points along a vertical gridline to be unknown (values at P, N and S for each point P), but take as known, most recent values being used, the values at each E and W neighbor. The TDMA is then applied to this vertical gridline. In this manner one can traverse along all the lines in the vertical direction sequentially from left to right of the integration domain.

Two points clarify the solution technique regarding the staircase sloping boundary. Firstly, interior points adjacent to the boundary must 'feel' the boundary in the usual way. Thus values in the external field must not be inadvertently picked up at these points. Usually the standard coupling coefficient of interest is set to zero and the wall effect given by way of a false source, according to the previous description. Secondly, if the TDMA is applied in columns over the entire 2-D array of points, the solution at external points can be automatically forced to zero by giving S_p^ϕ a very large negative value at these locations. This term then dominates in its finite difference equation at P with solution $\phi = 0$. Alternatively, the domain of TDMA execution can be restricted to the physically meaningful portion of the 2-D array of points. The latter technique is incorporated into the present computer code.

At each iteration it is necessary to employ some degree of under-relaxation when solving equations like Eq. (11). A weighted average of the

newly calculated value and the previous value is taken at each point. If ϕ_p^{n+1} represents the newly calculated value of ϕ at the $n+1$ iteration, it is obtained from

$$a_p^\phi \tilde{\phi}_p^{n+1} = \sum_j a_j^\phi \phi_j^n + S_U^\phi \quad (41)$$

The notation and similarity to Eq. (11) is obvious and

$$\phi_p^{n+1} = f \tilde{\phi}_p^{n+1} + (1-f) \phi_p^n \quad (42)$$

where $\tilde{\phi}_p^{n+1}$ represents the underrelaxed value of ϕ_p , employing the no-relaxation value ϕ_p^{n+1} and the underrelaxation parameter f ($0 < f \leq 1$). Rather than calculate and store the $\tilde{\phi}_p^{n+1}$ values and then apply Eq. (42), it is more convenient to modify and use Eq. (41) directly in the form

$$(a_p^\phi/f) \phi_p^{n+1} = \sum_j a_j^\phi \phi_j^n + [S_U^\phi + (1-f) \frac{a_p^\phi}{f} \phi_p^n] \quad (43)$$

so that the underrelaxed value is calculated immediately. This is the version used in the present code.

The effect of underrelaxation factor values on convergence rate is considerable. Unacceptably slow convergence or divergence of the solution is obtained if the factors are too low or too high, respectively. Large pressure corrections arise which produce large u - and v -velocity corrections. If these corrections are too large per iteration, the nonlinearity of the FDEs causes divergence.

Velocity and pressure corrections per iteration become smaller as the solution proceeds toward convergence. Thus the underrelaxation factors (especially for u - and v -velocities) have small initial values, to prevent

divergence, and increase for faster convergence as the corrections become smaller. Hence the underrelaxation factors for u, v and w are increased linearly during the first 40 iterations.

Increased accuracy of the initial estimate of field variables clearly reduces the amount of computational work required. Also it yields a slower rate-of-variable change and smaller corrections for continuity, which promotes convergence.

Since the user generally desires to utilize such a computer code to solve a series of problems, as in a parametric study for example, he almost always has converged (or partially-converged) solutions from previous computer runs of a similar problem. By storing field values of all variables on disk upon convergence for each swirl loop, a much better initial estimate conveniently becomes available for corresponding swirl cases of future problems, or for more stringent convergence of the same problem.

Final convergence is decided by way of a residual-source criterion, which measures the departure from exactness for the variable ϕ at the point P. The residual sources are defined for each variable at each point by equations like

$$R_p^\phi = a_p^\phi \phi_p - \sum_j a_j^\phi \phi_j - S_U^\phi \quad (44)$$

When each of these quantities becomes smaller than a certain fraction of a reference value, the finite difference equations are declared to be solved. Typically, the solution is considered to be converged if the cumulative sum of the absolute residuals throughout the field for all variables is less than 0.5 per cent of the inlet flow rate of the corresponding variable.

3. THE STARPIC COMPUTER PROGRAM

3.1 Outstanding Features

The case described. The computer program code described and listed performs the function outlined in Section 2 for the particular 2-D axisymmetric geometry shown schematically in Fig. 5. The flowfield is covered with a gradually expanding rectangular grid system. The quantity EPSX specifies the degree of grid expansion in the axial direction, and in the radial direction it is determined through specification of the radial position of each individual horizontal grid line. The boundary of the solution domain falls halfway between its immediate nearby parallel gridlines; and clearly specification of JMAX(I) for each I (see Fig. 5) is sufficient to determine the flowfield of interest. This provides a stairstep approximation to any monotonically increasing top wall radius as required so as to simulate any sudden expansion, sloping straight or curved sidewall. The code is equipped to handle any of these with ease. The program is an advanced version of a framework of ideas embodied into the 1974 Imperial College TEACH computer program,¹⁷ from which the present work has been developed. The following discussion is a commentary on the final developed program. A listing is supplied as Appendix E and should be read simultaneously with the discussion.

General arrangement. The STARPIC program (see the flow chart of Fig. 8) has been written (in Fortran 4) primarily with a view to maximizing simplicity and ease of amendment at the expense of computer time. This is achieved by limiting to the MAIN subprogram those features which characterize the particular flow conditions being investigated, while different boundary conditions may be inserted by way of the PROMOD (problem modifications) subroutine. The various functions of these and other subroutines are given briefly in Table 4 and in detail in sections 3.2 to 3.12. MAIN is the section of the

STARPIC

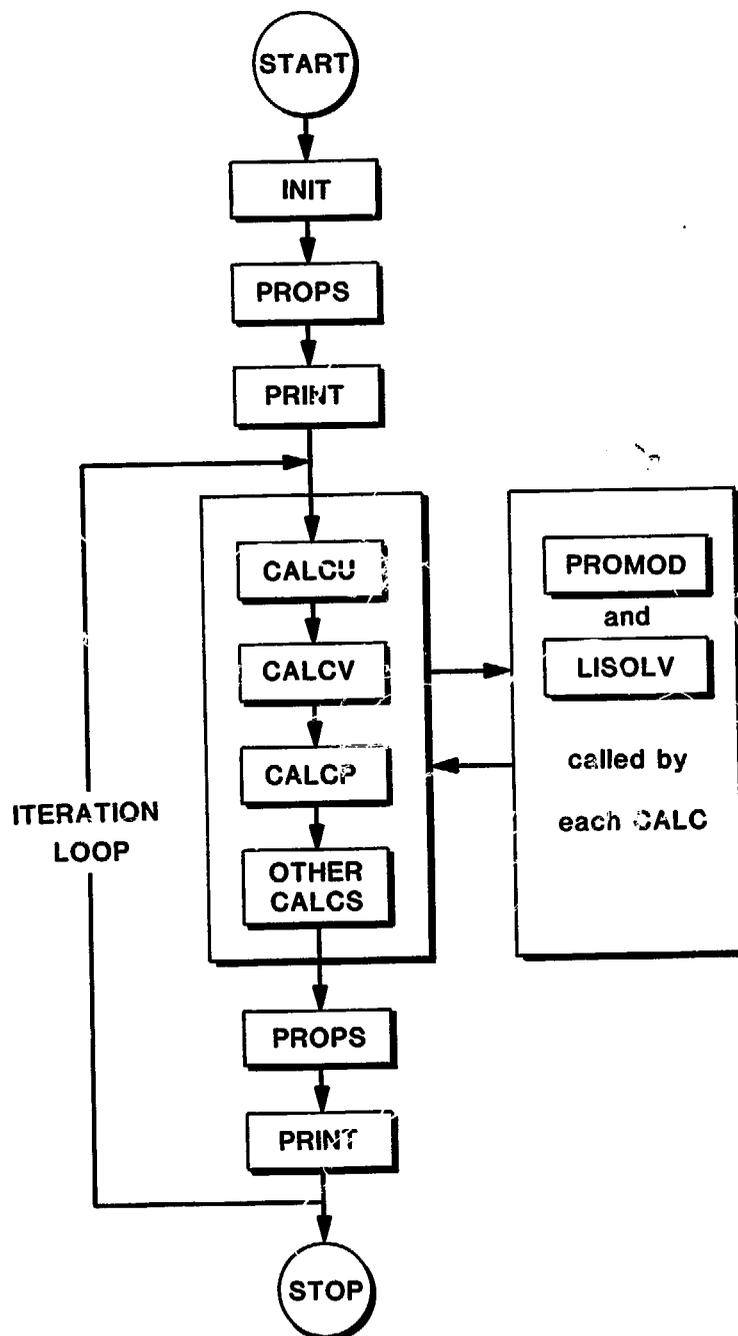


Fig. 8. Flow chart of computer program showing the subroutine arrangement.

Table 4. Subroutine Tasks

<u>Subroutine</u>	<u>Task</u>
MAIN	Controls and monitors the entire sequence of calculations: initialization, properties and initial output; the iteration loop with calls to update main variables, other mixture properties and intermediate output; and, after termination of the iteration loop, final output, an increment in inlet degree of swirl and a return to the beginning again.
INIT	Sets values to the numerous geometric quantities concerned with grid structure, see Figs. 9-11, and initializes most variables to zero or other reference value.
PROPS	Updates the fluid properties via calculation of turbulent viscosity, under-relaxed using its previous value. In nonisothermal flows, perhaps with chemical reaction, additional species' mass fractions, temperature and density are also calculated here, with an appeal to PROMOD (1) for any other modifications.
PRINT	Prints out an entire variable field according to a standard format.
CLACU and CALCV	Calculates coupling coefficients of finite difference equation for axial velocity u^* and radial velocity v^* , calls PROMOD (2) and PROMOD (3) for boundary modifications and LISOLV for entire field of variables to be updated to get u^* and v^* fields.

Table 4 (continued)

CALCP	Calculates coupling coefficients of finite difference equation for pressure correction p' ; calls PROMOD (4) for boundary modifications and LISOLV to obtain p' field. The subroutine closes with p^* , u^* and v^* being 'corrected' with p' , u' and v' .
Other CALC Subroutines	Calculates coupling coefficients of appropriate finite difference equation, calls appropriate part of PROMOD and then LISOLV for complete update of the variable in question.
PROMOD	Modifies the values of the finite difference equation coefficients, or the variables, near walls or other boundaries where particular conditions apply. The subroutine is divided into chapters, each handling a particular variable and being called from a CALC subroutine, and each chapter considers all the boundaries around the solution domain.
LISOLV	Updates entire field of a particular variable, by applying TDMA (tridiagonal matrix algorithm) to all the lines in the r-direction sequentially from left to right of the integration domain.

program to which a user will devote most of his attention, provided that he wishes to investigate configurations easily derivable from that shown in Fig. 5. He may also be concerned with PROMOD to adjust boundary condition types. Moreover all the subprograms are divided into chapters, each with a prescribed task to accomplish, thus decreasing the possibility of error on problem specification and modification.

MAIN controls the entire iterative solution procedure, with initial calls to INIT (initialization), PROPS (properties) and PRINT (for output of all variable fields). Repeated intermediate calls occur during the iteration loop to the CALC subroutines (CALCU for updating calculation of the u-velocity field, etc.), PROPS and PRINT (when required). Finally, after convergence or MAXIT (maximum iterations) number of iterations, PRINT is called again, together with the calculation and output of any special quantities required which are deducible from the flowfield values. Each of the CALC subroutines calls PROMOD (problem modifications) for modifications to the usual formulation because of boundary conditions. Subroutine LISOLV (line solve) is also called by each CALC subroutine to make several update sweeps of the relevant field of variables by applying the TDMA (tridiagonal matrix algorithm) to all the lines in the r-direction sequentially from left to right of the integration domain (NSWPU times for the u-field, etc. - number of sweeps for u).

Major variables. The discussion of Fortran variables is restricted to those symbols enjoying major significance in the accompanying listing of the program in Appendix E. Other symbols will readily yield their meanings on inspection of the context and their memonics; or they appear in parts of the program which the user is enjoined not to disarrange, the user restricting most of his attention to MAIN and PROMOD. A glossary of Fortran symbols is given in Appendix D. Table 5 lists principal dependent variables and controlling parameters.

Table 5. Principal Dependent Variables and Controlling Parameters

Algebraic Variable	Fortran Variable	Alphanumeric Heading	Logical Variable	Inlet Value	CALC Subroutine	Underrelaxation factor	Number of sweeps of LISOLV per iteration	Prandtl/Schmidt Number	Residual Source Term
u	U	HEDU	INCALU	UIN	CALCU	URFU	NSWPU	-	RESORU
v	V	HEDV	INCALV	VIN	CALCV	URFV	NSWPV	-	RESORV
w	W	HEDW	INCALW	WIN	CALCW	URFW	NSWPW	PRW	RESORW
k	TE	HEDK	INCALK	TEIN	CALCTE	URFK	NSWPK	PRTE	RESORK
ϵ	ED	HEDD	INCALD	EDIN	CALCED	URFE	NSWPD	PRED	RESORE
p'	PP	HEDP	INCALP	-	CALCP	URFP	NSWPP	-	(RESORM for mass calculated in CALCP)
μ	VIS	HEDVIS	INPRO	-	PROPS	URFVIS	-	-	-

3.2 The MAIN Subprogram

General arrangement. MAIN is that section of the program to which a user will devote most of his attention. The function of MAIN has been described in section 3.1. It is divided into chapters, each with a specific task, and a description of the individual chapters of MAIN now follows.

CO Preliminaries. Dimension, common and data blocks are followed by user input logical variables, which activate (when specified as true) certain special features of the program. As discussed in detail in Chapter 4, IFINE specifies a fine mesh in the x-direction with NI=35 as opposed to a coarser mesh with NI=23; IWRITE writes all dependent variables onto allocated disk storage; NONDIM calculates and prints the nondimensional solution in addition to the dimensional one; IREAD reads from allocated disk storage the initial solution guess, which is a previous solution of a similar problem; INITAL prints the initial guess of the solution; and INPLOT produces a line-printer plot of streamlines. Specification of NSWPU (number of sweeps for U), etc. and input read statements for alphanumeric headings HEDU (heading for U velocity), etc. are also located here. These alphanumeric headings are the only input data which appear after the program and are listed here in order:

U VELOCITY
V VELOCITY
W VELOCITY
PRESSURE
TEMPERATURE
TURBULENCE ENERGY
TURBULENCE DISSIPATION
VISCOSITY

DIMENSIONLESS LENGTH SCALE
DIMENSIONLESS STREAM FUNCTION
RADIAL COORDINATE OF STREAMLINES
DIMENSIONLESS U VELOCITY
DIMENSIONLESS V VELOCITY
DIMENSIONLESS W VELOCITY
DIMENSIONLESS PRESSURE
DIMENSIONLESS TURBULENCE ENERGY
DIMENSIONLESS STREAMLINE COORDS
DIMENSIONLESS EFF. VISCOSITY

C1 Parameters and control indices. This chapter really defines the problem to be solved. It is implicit that the problem is axisymmetric in cylindrical coordinates with the setting of INDCOS = 2. The problem domain is specified by dealing with the grid exemplified in Fig. 5, and giving values to the key parameters there-in. Other matters include dependent variable selection (setting .TRUE. or .FALSE. to INCALU etc.), fluid properties (see Prandtl/Schmidt numbers of Table 4), turbulence constants, boundary values (see inlet values of Table 4), pressure calculation reference point (IPREF, JPREF) and program control and monitor.

C2 Initial operations. Based on the problem specification already accomplished in C0 and C1, certain other initial operations are needed prior to the iteration procedure. In C2 the geometric quantities for the grids (see Figs. 9-11) are calculated and main 2-D array variables are set to zero or obvious initial values by way of the call to subroutine INIT. Returning to MAIN, the dependent variable fields are first specified at the inlet boundary including swirl velocity which uses VANB or SWNB for flat or solid-body rotation profile according to whether NSBR (number for solid body rotation) equals 0 or 1. Then estimates of dependent variables in the

field are calculated, as many of them depend on inlet values. Following the call to PROPS for fluid properties, an initial output is made of certain flow and geometric quantities. Finally, if IREAD = .TRUE. each dependent variable array is read sequentially from the disk storage allocated to unit 12 to re-specify the initial field estimates with improved values. Thus the user should specify IREAD = .TRUE. when a previous solution to a similar problem is available on disk, as this reduces the number of iterations required for convergence.

C3 Iteration loop. This is the section of MAIN that is repeated at each iteration where NITER counts the number of iterations. The task of C3 is several-fold: to adjust the underrelaxation factors for enhanced convergence based on numerical experience; to update dependent variables (via calls to subroutine CALCU if INCALU is true, etc.); to update fluid properties (if INPR0 is true the call to PROPS updates secondary dependent variables); to give intermediate output of residual source sums and field values at monitored location (IMON, JMON) for all values of NITER; to include appropriate field prints when NITER=JPRINT; and finally to make appropriate iteration termination tests.

C4 Final operations and output. Having decided to terminate the iteration process, control transfers to C4 to obtain final variable field print-outs and to calculate and print nondimensional variables if NONDIM = .TRUE. Also, a call to subroutine STRMFN calculates dimensionless stream function and determines the coordinates of points along NSTLN streamlines for plotting. Further, if IWRITE = .TRUE. each dependent variable array is stored sequentially on the disk space allocated to unit 11.

Control has now finished with the current problem. However, if LFS (loop for swirl) is less than LFSMAX (loop for swirl maximum) then LFS is incremented by 1, and the inlet swirl velocity profile is recalculated

for the next swirl problem based on SWNB (swirl number block) or VANB (vane angle block) according to whether NSBR (number for solid body rotation) equals 0 or 1. Before control begins C3 for the new swirl problem, the previous solution of a similar problem will be read from disk storage allocated to unit 12 if IREAD = .TRUE. This improved initial field estimate reduces the number of iterations required for convergence.

3.3 Subroutine INIT

INIT consists of two main chapters:

C1 Calculate geometric coefficients. Geometric quantities for the three grids for C, U and V cell control volumes (see Figs. 3 and 9-11) are here calculated once and for all for the particular geometric system previously specified in MAIN. Notice that setting INDCOS = 1 suppresses the variation of R with Y and thus obtains cartesian coordinates. In the current program, however, the problem is correctly coded only for cylindrical coordinates.

C2 Set variables to zero. In C2 most dependent and other 2-D array variables are set to zero throughout the flowfield although some are set to obvious nonzero initial values.

3.4 Subroutine PROPS

There is only one chapter:

C1 Viscosity. The turbulent viscosity μ is calculated from the two-equation k- ϵ turbulence model, Eqs. (5) and (6), making use of the URFVIS underrelaxation parameter.

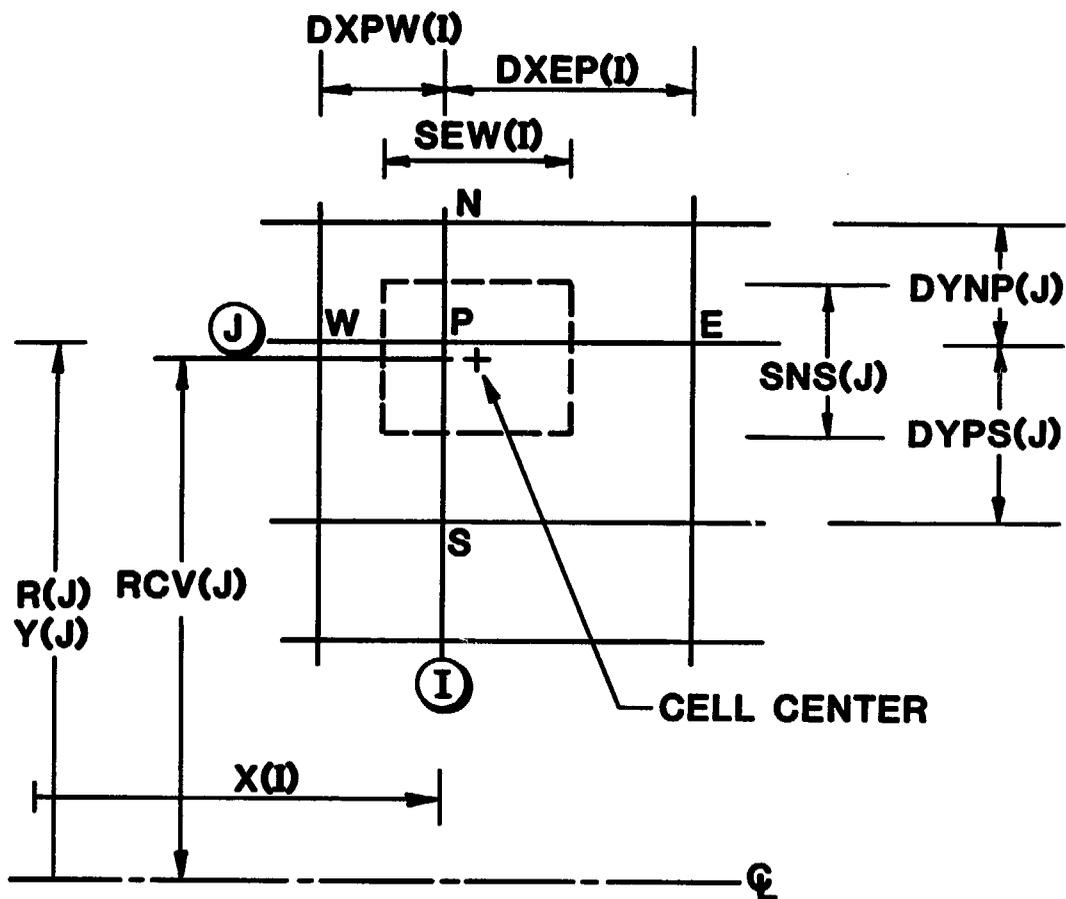


Fig. 9. Fortran variables related to the grid for C cells.

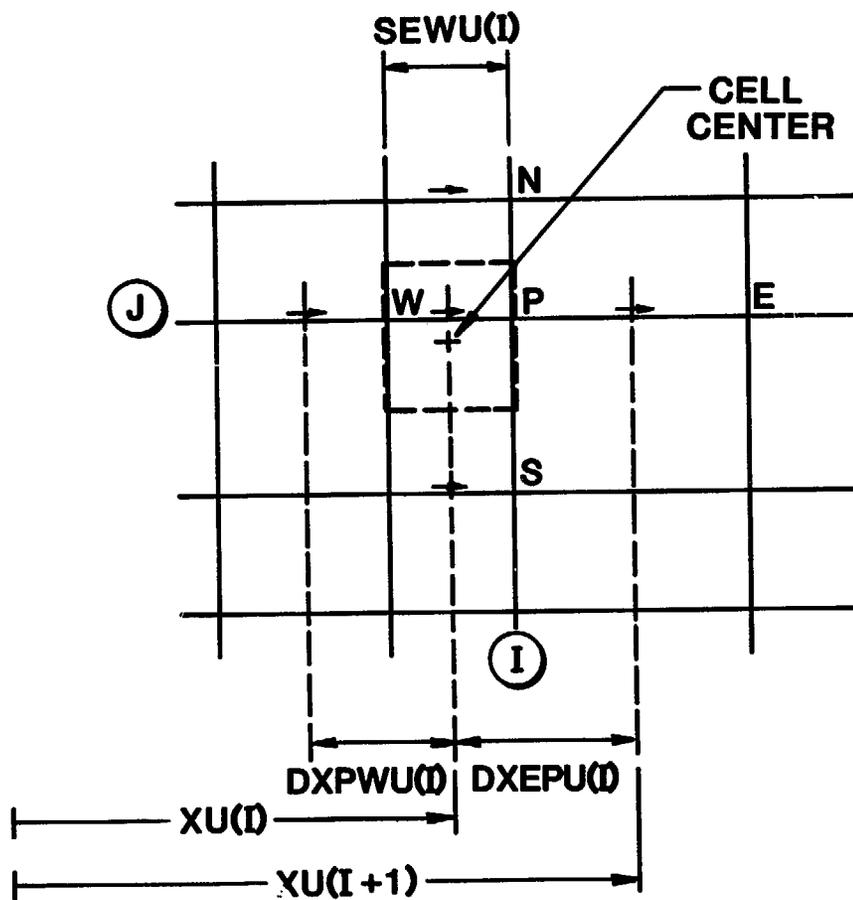


Fig. 10. Fortran variable_ related to the grid for U cells.

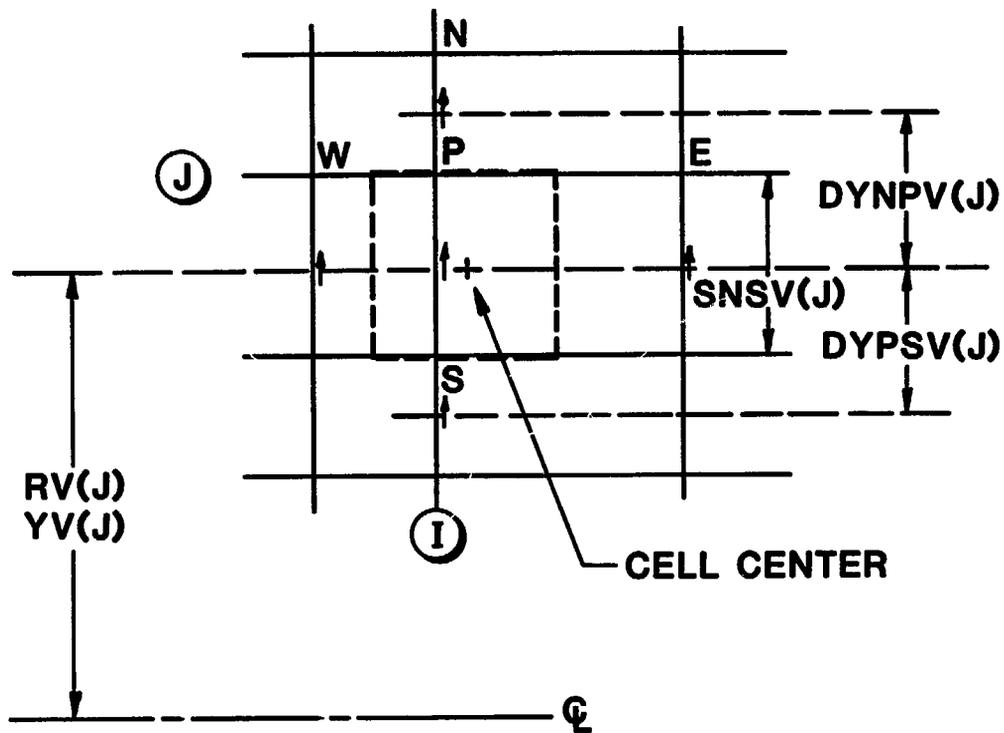


Fig. 11. Fortran variables related to the grid for V cells.

3.5 Subroutines CALCU and CALCV

CALCU has the task of calculating the appropriate coupling coefficients linking the value of U^* at each 'w' point of the mesh system with its four neighbors via Eq. (11). The formulation is completely general and correct only for internal points but the call to PROMOD (2) (problem modifications, chapter 2) corrects for near boundary points by applying the correct formulation via a linearized source technique (see section 2.3). Underrelaxation via Eq. (43) precedes the updating of the U^* field via a call to LISOLV, this being called NSWPU (number of sweeps for U) times for multiple inner updates of the U^* field. The task of CALCV is similar as it renews the V^* field and utilizes PROMOD (3). The chapters are:

C1 Assembly of coefficients. Two nested loops for I and J ensure that all internal points of the mesh are considered in C1. For each (I, J) - point the following sequence of calculations is performed:

- (i) the cell face areas and volume (utilizing the geometric variables of Figs. 9-11.
- (ii) the convection coefficients via the mass-velocities GN etc. and mass flow fluxes through the faces CN etc.
- (iii) the diffusion coefficients utilizing viscosities, areas and distances to obtain DN etc.
- (iv) the coefficients of source terms - here only the 'false' source stabilizing trick components of Eq. (9) are used; other parts of the source term are added in later.
- (v) the main coefficients AN etc. are assembled, using the 'hybrid' formulation of central and upwind differences, Eq. (10). The SU and SP components of the source expression are now determined with the pressure term going into SU, see Tables 1 and 2.

C2 Problem modifications. The general formulation of C1 is modified for near boundary points via the call to PROMOD. The call is to Chapter 2 of PROMOD in the case of CALCU and to Chapter 3 in the case of CALCV.

C3 Final coefficient assembly and residual source calculation. Two nested loops for I and J allow all points of the grid to be dealt with in turn. Firstly, $AP (= a_p)$ is calculated and RESORU (residual source for U) is incremented, and secondly, the underrelaxation demands of Eq. (43) are applied.

C4 Solution of difference equation. LISOLV is called NSWPU times in order to update the U^* field several times (NSWPV times in the case of V^*). The LISOLV parameter list begins 3, 2, ... in the case of CALCU, but 2, 3, ... in the case of CALCV. Careful observation of the relationship between the physical and reference storage locations of the displaced grid systems for U and V confirm this requirement.

3.6 Subroutine CALCP

CALCP deals with the variable PP (= p' the pressure correction) in a similar fashion to the CALCU and CALCV subroutines just described. However, there are several points of difference:

- (i) the subroutine is concerned with Eq. (15) and the calculation of p' ,
- (ii) the assembly of coefficients is simpler than the previous two subroutines,
- (iii) the subroutine closes with the application of the p' field to correct the latest estimates so far of the u , v and p fields, that is u^* , v^* and p^* , by appropriate use of Eqs. (12) through (14).

C1 Assembly of coefficients. Two nested loops sweep through the entire grid system. At each point the following sequence of calculations is performed: areas and volume of cells, coupling coefficients, source terms and absolute mass sources.

C2 Problem modification. A call statement to PROMOD (4) is included.

C3 Final coefficient assembly. Coefficient AP is calculated for all points of the mesh.

C4 Solution of difference equations. LISOLV is called NSWPP times in order to obtain a good estimate of the p' field. The parameter list, beginning 2, 2, ..., is appropriate since pressures are stored at grid intersections.

C5 Correct velocities and pressure. When this stage is reached, first estimates of the u , v and p fields (U^* from CALCU, V^* from CALCV and p^* from previous iteration) are known. Equations (12) through (14) are applied to obtain better estimates and ensure that the U^* and V^* momentum equation first-estimate solutions are brought into conformity with the continuity equation. Notice that the DO loop ranges are a little complicated so as to operate only on internal field points. There is no underrelaxation here on the velocity components (this was done prior to obtaining the U^* and V^* fields). Also, the required p' correction is added to the latest estimate of the pressure field p^* .

3.7 The other CALC subroutines

The other main dependent variables which are to be solved from their governing partial differential equations are w , k , and ϵ as seen in Table 1. Each of these is dealt with in a separate subroutine denoted by the name CALC followed by W, TE and ED. They are each called sequentially in this order directly from the MAIN subprogram; and each of them is structured

similarly to the CALCU and CALCV subroutines described in section 3.5. Only highlights of the differences need be described.

C1 Assembly of coefficients. Since all these variables are physically located at the grid intersections, both the I and J loops begin with the value 2 and ensure a sweep over all internal mesh points. The third part of C1 is concerned with diffusion coefficients and it is here that the Prandtl/Schmidt numbers PRTE, etc. in Table 4 are brought into play: viscosities are divided by PRTE, etc. in order to obtain the appropriate exchange coefficients. The fourth part concerns source terms and rather longer expressions are formulated as required from Table 1.

C2, C3 and C4. These bear such a strong resemblance to their corresponding chapters already described in section 3.5 that no further elaboration need be given here, except to remark that the LISOLV call always begins with the parameters 2, 2.

3.8 Subroutine LISOLV

LISOLV (line solver) is called from each of the CALC subroutines. When called it provides one complete sweep of the relevant interior points at which the variable is located. Also it replaces the 2-D array PHI, the last parameter of the call list, with the solution of the equations previously built-up in the calling subroutine. This is effected by means of the well-known TDMA (tridiagonal matrix algorithm) being applied to each vertical gridline in turn, and traversing along all such lines in the vertical direction sequentially from left to right of the integration domain.

The subroutine begins with a DO loop for the W-E sweep. There is some complexity here involving ISTART and JSTART, the first two parameters of the call list (these are usually given the values 2, 2 but are 3, 2 and 2, 3 when the call is from CALCU and CALCV respectively). But careful perusal of

the steps involved will assure the reader that the calculations are correct. The subsequent parts of the subroutine concern implementation of the TDMA technique: with the S-N traverse, assembling the TDMA coefficients, the calculation of the coefficients of the recurrence formula and the back-substitution to obtain the solution values of the S-N gridline.

3.9 Subroutine PRINT

The task of PRINT is to print out the values of a 2-D array PHI, together with associated heading HEAD, the last parameter of the call list. The X and Y coordinates are written horizontally and vertically around the output matrix of values. Again, as discussed in Section 3.8, the first two parameters of the call list, ISTART and JSTART, are usually 1,1 so that all internal and external values are printed to help diagnostics. Also, for U the XU values are printed and for V the YV values so that correct physical position values accompany the output of these variables which are located on displaced grids.

3.10 Subroutine PROMOD

The finite difference equation coefficients are formulated in each of the CALC subroutines on the assumption that each point is a usual internal point. Of course some points lie near or on the rectangular boundary of the flow domain in which the solution is sought and the general formulation of a CALC routine is not correct at these points. So that all required corrections to the general formulation are simple to understand and apply, each CALC routine calls a particular chapter of PROMOD (problem modifications). This subroutine has the task of modifying the values of the finite difference equation coefficients, or the variables, near walls or other boundaries where particular conditions apply. Each chapter concerns itself solely with one

particular variable and each chapter considers in turn each of the boundaries around the solution domain. Boundary conditions are thus easy to formulate and simple to apply; and this novel feature makes PROMOD second only to MAIN as a subroutine to which a prospective user needs to apply his attention.

According to the value given to NCHAP (number of chapter required) of the call list, control proceeds to one and only one of the eight chapters of this subroutine. These chapters deal with:

- C1 Properties
- C2 U momentum
- C3 V momentum
- C4 Pressure correction
- C5 Thermal Energy
- C6 Turbulent kinetic energy
- C7 Dissipation
- C8 W momentum

where C5 is retained from the original TEACH arrangement, but is not used in the present version of the code.

Figure 5 reveals that amongst the boundaries of the solution domain only at the inlet is the usual formulation of the finite difference equations, as computed in the CALC subroutines, correct. At other boundaries of the solution domain appropriate sections of PROMOD have to supply the correct influence of the boundary conditions on the coupling coefficients a_j^ϕ ($j = N, S, E, W$ and P) and components S_U^ϕ and S_P^ϕ of the linearized source term. In PROMOD each of the chapters is further divided into sections dealing with

- (i) Northern top wall, $I = 2$ to $NIM1$
- (ii) Southern symmetry axis, $I = 2$ to $NIM1$
- (iii) Western side wall, $J = 2$ to $NJM1$
- (iv) Eastern outlet, $J = 2$ to $NJM1$

In what follows all four cases of a N, S, E and W near-boundary point are dealt with simultaneously. One merely reads the appropriate line (designated by (N), (S), (E) and (W)) of the equations which follow, according to the particular case in which the reader is temporarily interested. Notation follows that of the boundary cells of Figs. 6 and 7.

Boundary conditions are inserted for each boundary cell. The first step generally involves breaking the link to any adjoining external cell by setting to zero the appropriate a_j of Eq. (11). Then, for each boundary

$$\begin{aligned}
 a_N(I, J) &= 0 & (N) \\
 a_S(I, J) &= 0 & (S) \\
 a_E(I, J) &= 0 & (E) \\
 a_W(I, J) &= 0 & (W)
 \end{aligned}
 \tag{45}$$

This step is followed by insertion of the correct boundary flux (diffusion and/or convection) for the cell boundary in question into the linearized source terms SU and SP, as exemplified in the following sections.

Neumann conditions. If zero normal gradient condition is the one to be applied, breaking the appropriate link is all that is required. For then the terms

$$\begin{aligned}
 a_N(\phi_P - \phi_N) &= 0 & (N) \\
 a_S(\phi_P - \phi_S) &= 0 & (S) \\
 a_E(\phi_P - \phi_E) &= 0 & (E) \\
 a_W(\phi_P - \phi_W) &= 0 & (W)
 \end{aligned}
 \tag{46}$$

are all zero in Eq. (11) so there is no flux through the boundary in question.

Dirichlet conditions. If the value of ϕ is prescribed as ϕ_B on the boundary then the usual boundary link is broken via Eq. (16). For a solid boundary the convective flux through the boundary face of the cell is zero,

but the diffusive flux is typically nonzero. Hence the diffusion term of Eq. (8) is transferred to the right-hand side RHS of the equation where it is incorporated into SU(I,J) and SP(I,J); on the RHS it is expressed in terms of ϕ_p and ϕ_B as

$$\begin{aligned}
 \Gamma_B (\phi_B - \phi_p) r_B \Delta x / \delta y & \quad (N) \\
 \Gamma_B (\phi_B - \phi_p) r_B \Delta x / \delta y & \quad (S) \\
 \Gamma_B (\phi_B - \phi_p) r_B \Delta y / \delta x & \quad (E) \\
 \Gamma_B (\phi_B - \phi_p) r_B \Delta y / \delta x & \quad (W)
 \end{aligned}
 \tag{47}$$

where Γ_B is the appropriate exchange coefficient for ϕ evaluated at point B. Appropriate cell boundary face areas appear here. North and south face areas are given by $r_B \Delta \theta \Delta x$ whereas east and west areas are expressed as $r_B \Delta \theta \Delta r$; but $\Delta \theta$ cancels from the governing equations. These expressions may be split between SU(I,J) and SP(I,J) as

$$SU(I,J) = SU(I,J) + \begin{cases} \Gamma_B r_B \phi_B \Delta x / \delta y & (N) \\ \Gamma_B r_B \phi_B \Delta x / \delta y & (S) \\ \Gamma_B r_B \phi_B \Delta y / \delta x & (E) \\ \Gamma_B r_B \phi_B \Delta y / \delta x & (W) \end{cases}
 \tag{48}$$

and

$$SP(I,J) = SP(I,J) - \begin{cases} \Gamma_B r_B \Delta x / \delta y & (N) \\ \Gamma_B r_B \Delta x / \delta y & (S) \\ \Gamma_B r_B \Delta y / \delta x & (E) \\ \Gamma_B r_B \Delta y / \delta x & (W) \end{cases}
 \tag{49}$$

Flux specified. If the flux of ϕ through a boundary face of the cell is specified. When brought to the RHS of Eq. (8) it may be approximately expressed as $b\phi_p + c$ where b and c are known (if possible, retaining $(-b)$ and (c) both positive to aid the convergence of the finite difference iteration procedure). This is implemented by breaking the boundary link and augmenting the source terms appropriately:

$$SU(I,J) = SU(I,J) + c \quad (N,S,E, \text{ and } W) \quad (50)$$

$$SP(I,J) = SP(I,J) + b \quad (N,S,E, \text{ and } W) \quad (51)$$

Fixing a value at an internal point. The value of ϕ at a near-boundary point P may be fixed at a value of ϕ_F via setting

$$SU(I,J) = \phi_F 10^{30} \quad (N,S,E, \text{ and } W) \quad (52)$$

$$SP(I,J) = -10^{30} \quad (N,S,E, \text{ and } W) \quad (53)$$

so that these terms dominate in the equation for ϕ at P , with the solution $\phi_p = \phi_F$.

Convective and diffusive influences. If both convection and diffusion through a cell boundary occurs, the hybrid formulation should be used. In this situation the contribution C for the flux passing the surface in question (N,S,E , or W) is calculated according to Eq. (10) and used along with the other surface contributions and source terms to formulate the usual finite difference equation. This is ensured automatically in the program for a normal inflow/outflow boundary.

3.11 Subroutine STRMFN

Subroutine STRMFN has been developed to calculate the dimensionless stream function

$$STFN = \int_0^r urdr / \int_0^{d/2} urdr \quad (54)$$

for all internal field points of the u-cell grid. The quadrature formulation for calculating STFN is essentially the Trapezoidal Rule. The integration proceeds radially one step SNSV(J) at a time so that a finite difference mesh which is either uniform or nonuniform in the radial direction is permissible.

The coordinates of points constituting each of NSTLN streamlines with dimensionless stream function values STVAL ranging from 0.0 to 1.0 are subsequently calculated for plotting. The present code calculates the coordinates of eleven streamlines with $\psi^* = 0.0, 0.1, \dots, 1.0$. Each streamline is represented by NI-1 points whose dimensionless x- and r-coordinates are stored as XUND(I) and YSTLND(I,K) (K = 1,...,11) respectively, where the k-index indicates the particular streamline. These two arrays are employed in producing COMPLIT streamline plots via the usual CALCOMP subroutines. Immediate line printer plots of alternate streamlines are obtained through the use of corresponding arrays XUDPLT(I) and YSLPLT(N,I) (N = 1,...,6) for the six streamlines $\psi^* = 0.0, 0.2, \dots, 1.0$.

3.12 Subroutine PLOT

The PLOT subroutine is a variant of the ϕ - profile line printer plot routine supplied with some versions of the GENMIX computer program. It is now described at length in a recent text.³⁴ In the present version supplied, it is called twice after each converged solution has been obtained. Firstly, with the parameter LARGE equal to 0 (giving a small-sized plot) and secondly with LARGE equal to 1 (giving a larger-sized plot). Prominent in the call list are XUDPLT(I) and YSLPLT(N,I) (N = 1,...,6) as described in Section 3.1 from which the required six streamlines are plotted. A 50 x 100 array is filled with alphanumeric characters as appropriate so as to give labels

0,2,4,6,8 and 1 corresponding to the nondimensional streamlines $\psi^* = 0.0,$
0.2,...,1.0.

4. USER'S GUIDE

4.1 Problem Specification

General. The listing of the program, which is supplied in Appendix E, is used for precisely those calculations which have been described in Sections 2 and 3 of this report. The user of the program will wish to make other calculations; for this he must modify the program. Although the description of the program which was given in Section 3 will allow the discerning reader already to distinguish the parts of the program which must be altered, the following notes will aid him to make the modifications to suit his case, without going through the whole program.

The general principles which the user should adopt are:

- (i) Read through MAIN, paying special attention to the Chapter titles, and other subtitles, and considering whether any of the features of the problem which relate to those titles have suffered changes which should be incorporated.
- (ii) Make the corresponding changes to MAIN.
- (iii) If modifications are desired to the turbulence model, introduce these in C2 of MAIN and/or C2 of PROPS.
- (iv) If different wall boundary conditions are required, read carefully Sections 2.3 and 3.10 and then change PROMOD accordingly.
- (v) Check that the COMMON statements are still adequate.
- (vi) Leave the general parts of the program alone, including the CALC subroutines, INIT, PROPS, LISOLV and PRINT (except those minor nonstructural parts just mentioned).

The following discussion is organized under the headings: Input data specification, system specification, boundary conditions, and turbulence specification.

Input data specification. Much of the data is internally specified. However, if IREAD = .TRUE. has been declared in C0 of MAIN, the unformatted reading of all input data from disk storage is activated. This input requires the proper job control "card" to allocate the logical unit number of the READ statement to the disk storage data file from which the data is to be read. The dependent variable fields are read from disk storage as an improved (near-solution) initial estimate to reduce the number of iterations required for convergence.

As seen in Section 4.2, the X(I) and Y(J) [=R(J)] arrays precede the dependent variable arrays sequentially written on each data file stored. Hence in C0 of MAIN, these two independent variable arrays are read to enable the next unformatted READ in C2 of MAIN to read the U(I,J), etc. dependent variables. After the solution is printed in C4 of MAIN, the dependent variable arrays are re-initialized for the next swirl loop by another series of unformatted reads from disk storage. Also, as discussed in Section 3, alphanumeric headings are read in C0 of MAIN using an A format field.

System specification. Minor variants of the sample grid system of Fig. 4 can be derived simply by appropriate modifications being made to the grid section of C1 of MAIN. Appropriate choice of the integers NI, NJ, ISTEP, JSTEP and JMAX(I) together with RLARGE (=D/2) ALTOT (= total length), EPSX (the gradual expansion rate in the axial direction) and Y(J) defines completely the axisymmetric geometry with INDCOS = 2. Assigning true or false to INCALU etc. activates the calls to subroutine CALCU etc. during the subsequent iteration cycle to update the U-velocity field. They also determine whether printing is required. Clearly, true settings will generally be the ones to use for all the dependent variables. Specification is also required for

fluid properties, including Prandtl/Schmidt numbers and all material constants in SI units.

Boundary conditions. C1 of MAIN specifies boundary values of all the dependent variables. Hence specification of UIN etc. are required for the main inlet. Most inlet profiles of the turbulent flow are then taken to be uniform in C2 of MAIN and the user can easily amend this if desired. Employed here is the swirl number block SWNB or vane angle block VANB according to the value of NSBR (number for solid body rotation). The specification NSBR = 1 activates SWNB, which gives solid body rotation from a swirl generator, and NSBR = 0 activates VANB for a flat swirl velocity profile from swirl vanes. In either case, the desired inlet values of swirl velocity W are assigned appropriately.

Initial field values of all the dependent variables are specified in C2 of INIT and C2 of MAIN. The user will be content to leave these alone, for as soon as the iteration process is underway, these estimates are improved. Only occasionally in some cases will they be too inaccurate to permit a smooth iterative process to continue from their values - in such cases more realistic values closer to the true solution and/or underrelaxation must be specified.

Turbulence specification. Minor adjustments to the turbulence model already in the program can be simply achieved by altering the turbulence constants part of C1 of MAIN and C2 of PROPS for the viscosity calculation. If one desired to make a computation for laminar flow, VIS must be filled with appropriate values in the DO 100 loop of PROPS. Also, the logical variables INCALK and INCALD could be specified as false to suppress the calculation of k and ϵ and the complexity in PROMOD for velocities near wall boundaries would be removed. When a different turbulence model is required, of course, C2 of PROPS must reflect this. Should a nonisotropic

or direct turbulent stress specification model be required, more far-reaching changes would be required to the diffusion term calculations in all the CALC subroutines, possibly with other CALC routines for solving stress equations. It is here that current knowledge is uncertain and much current research activity is being directed toward more realistic turbulence models.

4.2 Iteration and Accuracy

Iteration control. The iteration process is monitored by comparing the sum of the absolute values of residual source of mass in the field, RESORM, with a preset value, SORMAX, representing maximum source. Similarly for other variables U, V, W, etc. iteration is terminated for a particular problem when the largest residual source is less than SORMAX, or number of iterations NITER is greater than the maximum number of iterations MAXIT to be allowed. By analogy a divergence criterion can also be included if required, so as to prematurely terminate a computation giving results which appear not to be converging.

Fortran variables which are used to influence the iteration behavior are NSWPU etc. (representing number of update sweeps for U etc.) and URFU etc. (representing underrelaxation for U etc.). If divergence is found, the remedy often lies in increasing the former (especially NSWPP for pressure) and reducing the latter. However, the accompanying listing includes reasonable values which have been found convenient to use. They are automatically re-assigned in the iteration process according to the inlet swirl strength and number of iterations made toward convergence, as inspection of the listing reveals.

Output. At every iteration, RESORM etc. are printed out along with monitored values of variables at the location specified by IMON and JMON. Full field prints presently after 100 iterations and increments of 50 thereafter. Full field prints are obtained for each dependent variable with a true logical variable (like INCALU for U-velocity) by way of a call to subroutine PRINT. The user, of course, is entirely free to make any output he wishes at each iteration stage in C3 of MAIN and to make a field print of any 2-D array via a call to PRINT. This structure is a flexible one which can be utilized to advantage.

If IWRITE = .TRUE. has been specified in C0 of MAIN, the unformatted writing of all output data onto disk storage is activated. This output requires the proper job control "card" to allocate the logical unit number of the WRITE statement to the disk storage data file which is to receive the data. First, the X(I) and then the Y(J) [=R(J)] arrays are written to disk storage allocated to logical unit 11, as seen in the initial output portion of MAIN, C2. Also, located here is the writing of XUND(I), nondimensional XU(I), to disk storage allocated to logical unit 14.

Upon completion of the solution process for each swirl loop each field variable is nondimensionalized and printed (if NONDIM = .TRUE.) subsequent to the printing of dimensional variables. Then, the following dependent variable arrays are written onto disk (logical unit 11): U(I,J), V(I,J), W(I,J), P(I,J), TE(I,J), ED(I,J), VIS(I,J) and STFNI(I,J). Immediately following is the WRITE of YSTLND(I,K) to disk (logical unit 14), where YSTLND(I,K) contains the dimensionless radial coordinate corresponding to each dimensionless axial coordinate XUND(I) along each K streamline. These arrays are subsequently used in obtaining CALCOMP streamline plots.

Independence of grid size. A computation may be considered to be accurate if alterations in the grid size produce no significant changes in the values of dependent variables, or the fluxes, at the points in the flow or its boundary which are interesting to the user of the program. Such alterations are obtained by changing the values of NI and NJ (number of grid intervals in I and J (i.e. X and Y) directions) along with EPSX and Y(J). Specification of these quantities is made in C1 of MAIN. Of course, large values of NI and NJ increase the computer time significantly, so smallest values should be used that are consistent with acceptable accuracy. There is no way but trial to establish what these values are, but a 23 x 21 or 35 x 21 grid is probably sufficient, as available by simple choice in the delivered code.

Comparison with exact or experimental values. Further tests of accuracy can be made by setting the geometry and boundary conditions of the program to correspond to a problem for which exact solutions or experimental results are available. For example, the inlet step could be dispensed with and standard pipe flow results. In either case the presence of swirl and turbulence is optional. Hence comparison of the predictions of the program for this degenerate case with exact or experimental results can provide valuable insight into the accuracy with which the differential equations are being solved.

'Trouble-shooting' - It is possible to mention only a few of the remedies which can be employed when the program, in the course of being adapted to a new problem, generates computations that are evidently faulty.

To prevent divergence, the source terms should be formulated so that -SP is always positive for each ϕ variable. This keeps the point P dominant in its finite difference form with its N, S, E and W neighbors and aids the

stability of the iteration process. The forms shown in Table 2 enjoy this characteristic.

Lack of conservation of properties which ought to be conserved is often the result of incompatible specification of boundary conditions and fluid properties, for example. One has to take extreme care. It is not possible to say much more in general than that inadvertently introduced incompatibilities are among the most common causes of error.

4.3 The Sample Computation

The sample computation code listing (Appendix E) and the results given in the microfiche supplement are concerned with the inert turbulent flow in an idealized combustion chamber. An air flow (constant density $\rho = 1.211 \text{ kg/m}^3$ and molecular viscosity $\mu = 1.8 \times 10^{-5} \text{ kg/ms}$) enters a 45 deg. expansion ($\alpha = 45 \text{ deg.}$) from an inlet pipe (radius $d/2 = 0.03125 \text{ m}$) to a large pipe (radius $D/2 = 0.0625 \text{ m}$). The total length of the flow domain is 0.5 m so as to be sufficient to enclose any expected recirculation zones. Computer runs through a range of seven inlet swirl vane angles $\phi (= \arctan w_{in}/u_{in})$ equal to 0, 45, 55, 60, 65, 68 and 70 deg. are given, although serious production runs with the larger swirl strengths are made with the total length further extended to about 0.8 m in order to cover a larger domain associated with a longer central recirculation zone. A maximum of 200 iterations for each swirl strength is allowed, with the solutions for each value of ϕ being used as the initial starting values for the next higher value of ϕ .

The flow domain is covered by a 23 x 21 mesh system (NI = 23 for the x-direction, NJ = 21 for the r-direction). The sidewall angle of $\alpha = 45 \text{ deg.}$ is accommodated by way of four steps each of three cells high in the radial direction. The specification is

```
ISTEP = 2
JSTEP = 8
JMAX(1) = 8
JMAX(2) = 11
JMAX(3) = 14
JMAX(4) = 17
JMAX(I) = 20 (I = 5,6,...,21)
```

In the x-direction a uniformly expanding grid up to $x = 0.375$ m at $I = 20$ is specified via $EPSX = 1.11$, then equidistant addition grid lines are specified up to $x = 0.5$ m at $NI = 23$. This grid is appropriate for sidewall angles $\alpha = 45$ and 90 deg., though with $\alpha = 70$ deg. a uniformly expanding grid up to $x = 0.375$ m at $I = 30$ is specified via $EPSX = 1.102$, then equidistant additional grid lines are specified up to $x = 0.55$ m at $NI = 35$. In the r-direction the 21 grid lines are clustered near the expansion corner, centerline and north wall since additional refinement is required in regions with expected large gradients. The choice of x-grid is determined by setting IFINE to be true or false, giving fine or coarse grid system, respectively. Both grids are arranged

- (i) to give good resolution in strategic locations near the inlet, exit, projecting corners and front and back stagnation points,
- (ii) to give corresponding axial locations at which to compare predictions with present and previous measurements.

Finally, the choice of logical variables in Chapter 0 of MAIN is self-explanatory giving rather detailed output but not reading from or writing to any disk storage locations.

When the user wishes to compute a problem with a sidewall angle α different from 45 deg. [as in the listing supplied in Appendix E] he has

to specify the x and r [\equiv y] grid line locations and JMAX(I) for each I. Many variants of the flow domain of Fig. 5 discussed in Section 2.3 can thus be investigated. Auxiliary computations lead to the required specification. For example, other values of α are obtained by modifying the x-grid spacing near the inlet, which depends on NI, EPSX and ALTOT, and comparing it with the corresponding r-grid spacing near the expansion corner. That is, $XU(3) - XU(2)$ is compared with $YV(JSTEP+2) - YV(JSTEP+1)$ to get the aspect ratio by the expansion corner. When a multi-cell step is envisioned a greater range of YV values are considered. Corresponding values of α can be tabulated with various combinations of NI and EPSX, and the user can then judiciously choose an appropriate combination with JMAX(I) values for his needs.

Another parameter frequently requiring different values is swirl strength. The program is set up via LFS and LFSMAX to sequentially calculate through a range of seven swirl vane angles. These angles are specified in a DATA statement for the array VANB which is activated if NSBR = 0. If NSBR = 1 the swirl number for consecutive swirl numbers to be investigated is specified via a DATA statement for the array SWNB.

5. CLOSURE

The STARPIC computer program has been developed to predict swirling recirculating inert turbulent flows in axisymmetric combustion chambers. The technique involves a staggered grid system for axial and radial velocities, a line relaxation procedure for efficient solution of the equations, a two-equation $k - \epsilon$ turbulence model, a stairstep boundary representation of the expansion flow, and realistic accommodation of swirl effects. Predictions of this type allow some results to be obtained more cheaply, quickly, and correctly than currently possible by the almost exclusive use of experimental means. Further development and application is providing a valuable supplementary technique for designers of practical combustion equipment.

This report has dealt with the computational problem and shown how the mathematical basis and computational scheme may be translated into a computer program. A flow chart, Fortran 4 listing, notes about various subroutines and a user's guide are supplied as an aid to prospective users of the code.

6. REFERENCES

1. Lilley, D. G., Flowfield Modeling in Practical Combustors: A Review, *Journal of Energy*, Vol. 3, July-Aug. 1979, pp. 193-210.
2. Lilley, D. G., Prospects for Computer Modeling in Ramjet Combustors, AIAA Paper No. 80-1189, Hartford, CT, June 30-July 2, 1980.
3. Syred, N. and Beer, J. M., Combustion in Swirling Flows: A Review, *Combustion and Flame*, Vol. 23, 1974, pp. 143-201.
4. Lilley, D. G., Swirl Flows in Combustion: A Review, *AIAA Journal*, Vol. 15, No. 8, August 1977, pp. 1063-1078.
5. Beer, J. M. and Chigier, N. A., *Combustion Aerodynamics*, Applied Science, London and Halstead-Wiley, New York, 1972.
6. Gupta, A. K., Lilley, D. G. and Syred, N., *Swirl Flows*, Abacus Press, Tunbridge Wells, England, 1981.
7. Rhode, D. L., Lilley, D. G. and McLaughlin, D. K., On the Prediction of Swirling Flowfields Found in Axisymmetric Combustor Geometries. ASME Symposium on Fluid Mechanics of Combustion Systems, held in Boulder, Colorado, June 22-24, 1981.
8. Gerstein, M. (ed), *Fundamentals of Gas Turbine Combustion*, NASA-CP-2087, 1979. Workshop held at NASA Lewis Research Center, Cleveland, OH, Feb. 6-7, 1979.
9. Murthy, S. N. B., Summary Report DOD Colloquium on Gas Turbine Combustor Modeling. Project SQUID Technical Report PU-R1-79, School of Mech. Engr., Purdue University [for ONR], Nov. 1979.
10. Lefebvre, A. H. (ed), *Gas Turbine Combustor Design Problems*, Hemisphere-McGraw-Hill, New York, 1980.
11. Launder, B. E. and Spalding, D. B., *Mathematical Models of Turbulence*, Academic Press, London, England, 1972.

12. Launder, B. E. and Spalding, D. B., The Numerical Computation of Turbulent Flows, *Comp. Methods in Appl. Mech. and Engrg.*, Vol. 3, March 1974, pp. 269-289.
13. Roache, P. J., *Computational Fluid Dynamics*, Hermosa, Albuquerque, NM, 1972.
14. Chow, C.-Y., *An Introduction to Computational Fluid Mechanics*, Wiley, New York, NY, 1979.
15. Patankar, S. V., *Numerical Heat Transfer and Fluid Flow*, Hemisphere-McGraw-Hill, New York, 1980.
16. Gosman, A. J., Pun, W. M., Runchal, A. K., Spalding, D. B., and Wolfshtein, M. W., *Heat and Mass Transfer in Recirculating Flows*, Academic Press, London, England, 1969.
17. Gosman, A. D., and Pun, W. M., *Calculation of Recirculating Flows*, Rept. No. HTS/74/12, 1974, Dept. of Mechanical Engineering, Imperial College, London, England.
18. Gosman, A. D. and Ideriah, F. J. K., *TEACH-2E: A General Computer Program for Two-Dimensional, Turbulent, Recirculating Flows*. Report, Dept. of Mech. Eng., Imperial College, London, June 1976.
19. Lilley, D. G., *Primitive Pressure-Velocity Code for the Computation of Strongly Swirling Flows*, *AIAA Journal*, Vol. 14, June 1976, pp. 749-756.
20. Khalil, E. E., Spalding, D. B. and Whitelaw, J. H., *The Calculation of Local Flow Properties in Two-Dimensional Furnaces*, *Int. J. Heat Mass Trans.*, Vol. 18, 1975, pp. 775-791.
21. Novick, A. S., Miles, G. A., and Lilley, D. G., *Numerical Simulation of Combustor Flowfields: A Primitive Variable Design Capability*, *J. of Energy*, Vol. 3, No. 2, March-April, 1979, pp. 95-105.

22. Wuerer, J. E. and Samuelson, G. S., Predictive Modeling of Backmixed Combustor Flows: Mass and Momentum Transport, AIAA Paper 79-0215, New Orleans, LA., Jan. 15-17, 1979.
23. Habib, M. A. and Whitelaw, J. H., Velocity Characteristics of Confined Coaxial Jets With and Without Swirl, ASME Paper No. 79-WA/FE-21, New York, Dec. 2-7, 1979.
24. El Banhawy, Y. and Whitelaw, J. H., Calculation of the Flow Properties of a Confined Kerosene-Spray Flame, AIAA J., Vol. 18, No. 2, Dec. 1980, pp. 1503-1510.
25. Srinivasan, R. and Mongia, H. C., Numerical Computations of Swirling Recirculating Flows, Final Report, NASA-CR-165196, Sept. 1980.
26. Patankar, S. V. and Spalding, D. B., A Computer Model for Three-Dimensional Flow in Furnaces, 14th Symp. (Int.) on Comb., The Comb. Inst., Pittsburgh, Pa., 1973, pp. 605-614.
27. Serag-Eldin, M. A. and Spalding, D. B., Computations of Three-Dimensional Gas-Turbine Combustion Chamber Flows, Trans. ASME J. of Eng. for Power, Vol. 101, July 1979, pp. 326-336.
28. Mongia, H. C. and Reynolds, R. S., Combustor Design Criteria Validation Vol. III - User's Manual, Report USARTL-TR-78-55C, U.S. Army Res and Tech. Lab., Ft. Eustis, VA, Feb. 1979. [See also Vols. I and II].
29. Swithenbank, J., Turan, A. and Felton, P. G., Three-Dimensional Two-Phase Mathematical Modeling of Gas Turbine Combustors, in Gas Turbine Combustor Design Problems (Lefebvre, A. H. ed.), Hemisphere-McGraw-Hill, New York, 1980, pp. 249-314.
30. Spalding, D. B., A Novel Finite Difference Formulation of Differential Expressions Involving Both First and Second Derivatives, Int. J. Num. Meth. in Eng., Vol. 4, 1972, pp. 551-559.

31. Weske, D. R. and Sturov, G. Ye, Experimental Study of Turbulent Swirled Flows in a Cylindrical Tube, Fluid Mechanics - Soviets Research, Vol. 3, No. 1, Jan.-Feb., 1974, pp. 77-82.
32. Blackshall, R. G. and Landis, F., The Boundary-Layer Velocity Distribution in Turbulent Swirling Pipe Flow, Journal Basic Engr., ASME Trans., Dec. 1969, pp. 728-733.
33. Date, A. W., Prediction of Fully-Developed Flow in a Tube Containing A Twisted Tape, Int. Jl. Heat Mass Transfer, Vol. 17, pp. 845-859.
34. Spalding, D. B., GENMIX - A General Computer Program for Two-Dimensional Parabolic Phenomena. Pergamon Press, Oxford, England, 1977.

7. NOMENCLATURE

A	Cell face area, Eq. (8)
a	Coupling coefficient, Eq. (11)
C	Contribution to cell surface integral, Eq. (10)
C,U,V	Control cell volumes for ϕ , u, v, Fig. 3
D	Chamber diameter, Fig. 1
d	Nozzle diameter, Fig. 1
E	Constant in law of wall, Eq. (24)
f	Underrelaxation parameter, Eq. (42)
G	k - generation term, Eq. (4)
I,J	Mesh point, Fig. 2
k	Kinetic energy of turbulence, Eq. (1)
\dot{m}_{net}	Net outflow of mass from cell, Eq. (9)
Pe	Cell Peclet number = $\rho u \delta / \Gamma$, Eq. (10)
p	Time-mean pressure, Eq. (1)
R	Residual source, Eq. (44)
S	Source term (with subscript), Eq. (1)
S_p, S_U	Components of linearized source term, Eq. (8)
V	Magnitude of total velocity vector, Eq. (24)
$\underline{v} = (u,v,w)$	Time-mean velocity (in x,r, θ directions), Eq. (1)
x,r, θ	Axial, radial, circumferential polar co-ordinates, Eq. (1)
y	Distance normal to a wall, Eq. (25)
Γ	Turbulent exchange coefficient, Eq. (1)
α	Side-wall angle, Fig. 1
Γ	Exchange coefficient, Eq. (7)

δx	Axial distance between two neighboring mesh points, Fig. 4
ϵ	Turbulence energy dissipation rate, Eq. (1)
κ	Constant in log law, Eq. (24)
μ	Effective viscosity, Eq. (5)
ρ	Time-mean density, Eq. (1)
σ	Prandtl-Schmidt number, Eq. (6)
τ	Wall shear stress, Eq. (25)
ϕ	Swirl vane angle, Fig. 1 dependent variable, Eq. (1)
ψ	Stream function, Eq. (7)

Superscripts

old	Last iterate value, Eq. (9)
*	Preliminary u, v and p field based on estimated pressure field p^* , Eqs. (12)-(15)
'	Correction value to u^*, v^*, p^* to get u, v, p, Eqs. (12)-(15)

Subscripts

in	Inlet, Fig. 1
l	Laminar value, Eq. (5)
n,s,e,w	North, south, east, west faces of cell, Eq. (8)
P,N,S,E,W	Point, north, south, east, west neighbors, Fig. 2
t	Total, Eq. (25)

APPENDIX A DERIVATION OF THE HYBRID DIFFERENCING SCHEME

In the discussion of Section 2.2, it was remarked that a hybrid differencing scheme is used in representing the combined effects of convection and diffusion at cell boundaries. It possesses certain advantages over the well-known central and upstream differencing schemes, and it is convenient to provide here sufficient derivation, in the context of the present 2-D axisymmetric flow problem. The reasons and choices are aided by comparison of possible finite difference representations with the exact solution of a simplified 1-D convection and diffusion problem, whose solution values and flux calculations therefrom provides useful guidance.

Consider the transport across one face of a control volume, for example the western face of area A_w normal to the x-direction which lies midway between the gridpoints W and P distant δx apart, as shown earlier in Fig. 4. The contribution C_w of the convection and diffusion terms of Eq. (1) to the surface integral for the western face is

$$C_w = [\rho u \phi - \Gamma_\phi \frac{\partial \phi}{\partial x}]_w A_w \quad (A1)$$

and the task is to derive a finite difference expression to represent the right hand side realistically in terms of ϕ and Γ_ϕ values at W and P using appropriate differencing. The choice will be aided by comparison with the exact solution of

$$\rho u \phi - \Gamma_\phi \frac{\partial \phi}{\partial x} = 0 \quad (A2)$$

in the interval $W \leq x \leq P$, taken without loss of generality to be $0 \leq x \leq \delta x$, where $\Gamma_\phi = (\Gamma_\phi)_w = \text{constant}$ and ρ and u are presumed to be constant known values evaluated at $x = \delta x/2$. Solving this one-dimensional

Eq. (A2) subject to the two grid values

$$\begin{aligned} x = 0 & \quad \phi = \phi_W \\ x = \delta x & \quad \phi = \phi_P \end{aligned} \tag{A3}$$

yields the solution

$$\phi = \phi_W + (\phi_P - \phi_W) \left[\frac{e^{Pe \cdot x/\delta x} - 1}{e^{Pe} - 1} \right] \tag{A4}$$

where the cell Peclet number has been defined by

$$Pe = \rho u \delta x / \Gamma_\phi \tag{A5}$$

which is evaluated at $x = \delta x/2$, i.e. at the western face of the cell around P. The exact solution, Eq. (A4), is portrayed in Fig. 12 where inspection shows that the value of ϕ always lies within the extreme values of ϕ_W and ϕ_P . When Pe becomes very large and positive, because of axial velocity u being very large and from left to right in Fig. 3, the ϕ - x curve is nearly horizontal and nearly equal to ϕ_W at all points, with a steep rise to the value ϕ_P as x approaches δx (i.e. near P). Conversely when Pe is very large and negative, because of axial velocity u being very large and from right to left for example, the ϕ - x curve is nearly equal to ϕ_P at all points with the steep part near $x = 0$ (i.e. near W). When Pe is near to zero in value, the linear interpolation straight line is close to the exact solution.

The present task is to look at the net transfer of ϕ across the mid-plane at $x = \delta x/2$ and to deduce a satisfactory finite difference analogue of the differential equation Eq. (A2) in terms of ϕ_W and ϕ_P . The flux of ϕ across the mid-plane at $x = \delta x/2$ is

$$Q_{\delta x/2} = \left[\rho u \phi - \Gamma_\phi \frac{\partial \phi}{\partial x} \right]_{x = \delta x/2} \tag{A6}$$

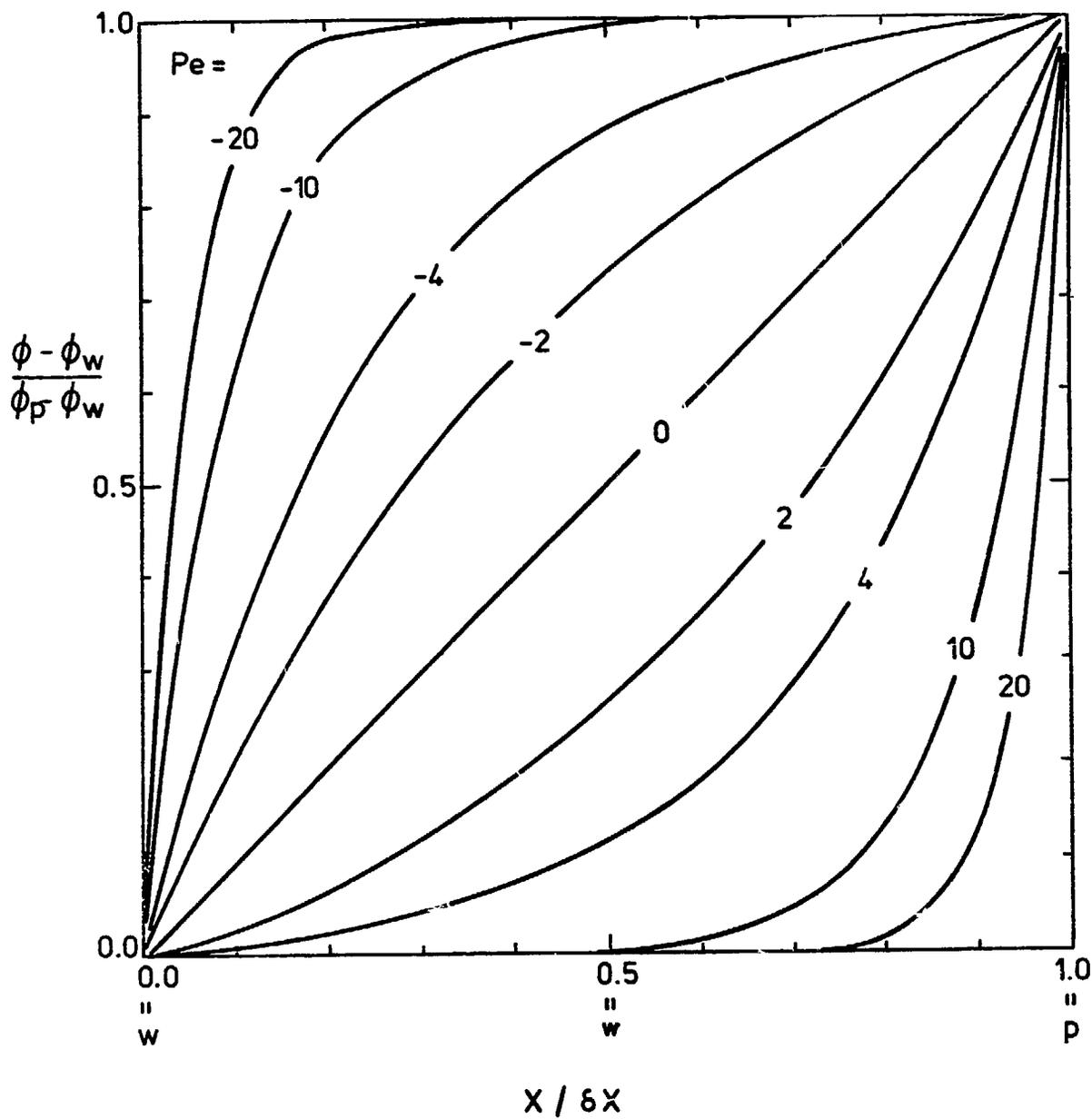


Fig. 12. ϕ versus x curves for different values of the cell Peclet number Pe (assuming $\phi_w < \phi_p$).

and upon substituting the exact solution for ϕ and $\frac{\partial\phi}{\partial x}$ at $x = \delta x/2$ using Eq. (A4) yields (see Appendix B)

$$\begin{aligned}
 Q_{\delta x/2} &= (\rho u)_{\delta x/2} \left\{ \frac{\phi_W e^{Pe} - \phi_P}{e^{Pe} - 1} \right\} \\
 &= (\rho u)_{\delta x/2} \left\{ \frac{\phi_W e^{Pe}}{e^{Pe} - 1} - \frac{\phi_P e^{-Pe}}{1 - e^{-Pe}} \right\}
 \end{aligned}
 \tag{A7}$$

Three limits of this expression can be examined:

$$Pe \rightarrow +\infty \quad Q_{\delta x/2} \rightarrow (\rho u)_{\delta x/2} \cdot \phi_W
 \tag{A8}$$

$$Pe \rightarrow -\infty \quad Q_{\delta x/2} \rightarrow (\rho u)_{\delta x/2} \cdot \phi_P
 \tag{A9}$$

$$|Pe| \rightarrow 0 \quad Q_{\delta x/2} \rightarrow (\rho u)_{\delta x/2} \left\{ \phi_W \frac{1 + Pe + \dots}{Pe (1 + Pe/2 + \dots)} - \phi_P \frac{1 - Pe + \dots}{Pe (1 - Pe/2 + \dots)} \right\}
 \tag{A10}$$

$$= (\rho u)_{\delta x/2} (\phi_W + \phi_P)/2 - \Gamma_{\delta x/2} \frac{\phi_P - \phi_W}{\delta x}
 \tag{A11}$$

where terms of order Pe^2 and higher have been omitted from the last expression. Rearrangement and substitution of $\Gamma_{\phi, \delta x/2} / \delta x$ for $(\rho u)_{\delta x/2} / Pe$ has been made to obtain the final form of Eq. (A11). This last equation is equivalent to central differencing of the diffusion part (in square brackets) of Eq. (A1). The first two limits of Eqs. (A8) and (A9) are the upstream convection dominated limits where diffusion terms are negligible. In computing it is necessary to use a finite difference expression for the flux passing the surface for the western face contribution

$$C_W = Q_{\delta x/2} \cdot A_W
 \tag{A12}$$

which exhibits these three limiting cases. The hybrid scheme recommended and used in Refs. 17-23 exhibits these properties via the use of

$$C_w = \begin{cases} (\rho u)_w A_w (\phi_w + \phi_p) / 2 - (\Gamma_\phi)_w A_w (\phi_p - \phi_w) / \delta x & \text{for } |Pe| < 2 \\ (\rho u)_w A_w \phi_w & \text{for } Pe \geq 2 \\ (\rho u)_w A_w \phi_p & \text{for } Pe \leq -2 \end{cases} \quad (A13)$$

where $Pe = (\rho u)_w \delta x / (\Gamma_\phi)_w$ is the cell Peclet number calculated at w . This hybrid scheme is denoted by HS.

Figure 13 shows the nondimensional flux passing the mid-plane at $x - \delta x/2$ versus Peclet number Pe . Four lines are shown representing the exact solution ES, the hybrid scheme HS, the central difference scheme CDS and the upwind difference scheme UDS. These lines show the overall superiority of the HS over the CDS and UDS approaches, for the particular case considered in Appendix C. However, the result is quite general.³⁰ These different schemes and the superiority of the hybrid scheme recommended and used in this work⁷ are fully discussed in Appendix C.

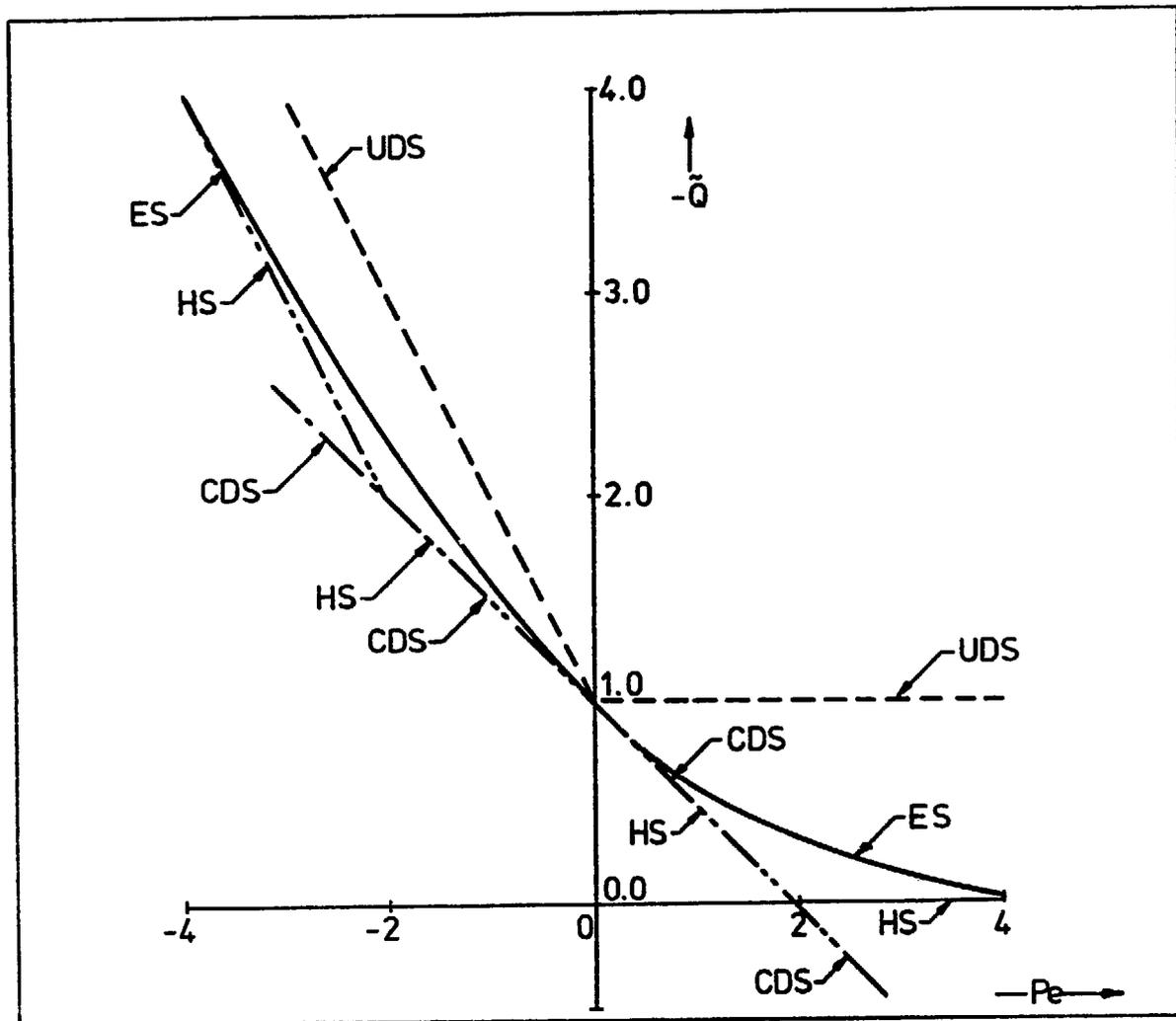


Fig. 13. Comparison of exact flux passing through western face with other approaches, as a function of Peclet number (ES, CDS, UDS, AND HS are described in Appendix C).

APPENDIX B DERIVATION OF EQ. (A7) FROM EQ. (A6)

Equation (A4) gives ϕ and $\frac{\partial\phi}{\partial x}$ evaluated at $x = \delta x/2$ as

$$\phi_{\delta x/2} = \phi_W + (\phi_P - \phi_W) \frac{e^{Pe/2} - 1}{e^{Pe} - 1} \quad (B1)$$

and

$$\left. \frac{\partial\phi}{\partial x} \right|_{\delta x/2} = (\phi_P - \phi_W) \cdot \frac{1}{e^{Pe} - 1} \cdot \frac{Pe}{\delta x} \cdot e^{Pe/2} \quad (B2)$$

Substitution into Eq. (A6) gives, where $\rho u = (\rho u)_{\delta x/2}$:

$$\begin{aligned} Q_{\delta x/2} &= \rho u \left\{ \phi_W + (\phi_P - \phi_W) \frac{e^{Pe/2} - 1}{e^{Pe} - 1} \right\} \\ &\quad - \Gamma_\phi \left\{ (\phi_P - \phi_W) \frac{1}{e^{Pe} - 1} \cdot \frac{Pe}{\delta x} \cdot e^{Pe/2} \right\} \\ &= \rho u \phi_W \left\{ 1 - \frac{e^{Pe/2} - 1}{e^{Pe} - 1} + \frac{e^{Pe/2}}{e^{Pe} - 1} \right\} \\ &\quad + \rho u \phi_P \left\{ \frac{e^{Pe/2} - 1}{e^{Pe} - 1} - \frac{e^{Pe/2}}{e^{Pe} - 1} \right\} \end{aligned} \quad (B3)$$

on substituting for Γ_ϕ in terms of the Peclet number $Pe = \rho u \delta x / \Gamma_\phi$.

continuing

$$\begin{aligned} Q_{\delta x/2} &= \rho u \phi_W \left\{ \frac{e^{Pe} - 1 + 1}{e^{Pe} - 1} \right\} + \rho u \phi_P \left\{ \frac{-1}{e^{Pe} - 1} \right\} \\ &= (\rho u)_{\delta x/2} \left\{ \phi_W \frac{e^{Pe}}{e^{Pe} - 1} - \phi_P \frac{1}{e^{Pe} - 1} \right\} \end{aligned} \quad (B4)$$

which is equivalent to Eq. (A7).

APPENDIX C THE SUPERIORITY OF THE HYBRID DIFFERENCING SCHEME

Here the concern is with exhibiting possible finite difference analogues of Eq. (A7) which expresses the exact solution for the flux $Q = Q_{\delta x/2}$ passing the mid-plane $x = \delta x/2$, as a function of local density and axial velocity evaluated at this point $\rho u = (\rho u)_{\delta x/2}$, the local Peclet number Pe and the ϕ -values at W and P. Three limiting cases for $Pe \rightarrow +\infty$, $Pe \rightarrow -\infty$ and $|Pe| \rightarrow 0$ were given in Eqs. (A8) - (A11), the first two being upstream convection dominated and the last one being equivalent to simple averaging for the convection term and central differencing for the diffusion term.

In computing it is advantageous to use a finite difference analogue of Eq. (A2) which gives the best fit for this flux Q over a range of ρ , u , Γ and δx likely to be encountered, as expressed via the Peclet number Pe . The suffix ϕ on Γ is omitted from here onward to avoid confusion. Nonsubscripted variables are evaluated at $x = \delta x/2$. The three popular practices, and the exact solution, are:

- (i) CDS - the central difference scheme. For all Pe numbers, use

$$Q = \rho u (\phi_P + \phi_W)/2 - \Gamma(\phi_P - \phi_W)/\delta x \quad (C1)$$

- (ii) UDS - the upstream difference scheme. For all Pe numbers, use

$$Q = \begin{cases} \rho u \phi_W - \Gamma(\phi_P - \phi_W)/\delta x & (u > 0) \\ \rho u \phi_P - \Gamma(\phi_P - \phi_W)/\delta x & (u < 0) \end{cases} \quad (C2)$$

- (iii) HS - the hybrid scheme. According to Pe and u , use (see Eq. (A13)):

$$Q = \begin{cases} \rho u (\phi_P + \phi_W)/2 - \Gamma(\phi_P - \phi_W)/\delta x & (|Pe| < 2) \\ \rho u \phi_W & (Pe \geq 2, u > 0) \\ \rho u \phi_P & (Pe \leq -2, u < 0) \end{cases} \quad (C3)$$

(iv) ES - the exact solution. According to Pe number, use (see Eq. (A7)):

$$Q = \rho u \left\{ \frac{\phi_W e^{Pe}}{e^{Pe} - 1} - \phi_P \frac{e^{-Pe}}{1 - e^{-Pe}} \right\} \quad (C4)$$

We wish to determine which of the CDS, UDS or HS is generally the best to use as an approximation to ES, over a range of Pe numbers. Nondimensionalizing the fluxes via

$$\hat{Q} = Q \cdot \frac{\delta x}{\Gamma} \cdot \frac{1}{\phi_P - \phi_W} \quad (5)$$

gives the following forms (using $Pe = \rho u \delta x / \Gamma$ when required):

$$(i) \text{ CDS } \hat{Q} = \frac{1}{2} \cdot Pe \cdot \frac{\phi_P + \phi_W}{\phi_P - \phi_W} - 1 \quad (C6)$$

$$(ii) \text{ UDS } \hat{Q} = \begin{cases} Pe \cdot \frac{\phi_W}{\phi_P - \phi_W} - 1 & (u > 0) \\ Pe \cdot \frac{\phi_P}{\phi_P - \phi_W} - 1 & (u < 0) \end{cases} \quad (C7)$$

$$(iii) \text{ HS } \hat{Q} = \begin{cases} \frac{1}{2} \cdot Pe \cdot \frac{\phi_P + \phi_W}{\phi_P - \phi_W} - 1 & (|Pe| < 2) \\ Pe \cdot \frac{\phi_W}{\phi_P - \phi_W} & (Pe \geq 2, u > 0) \\ Pe \cdot \frac{\phi_P}{\phi_P - \phi_W} & (Pe \leq 2, u < 0) \end{cases} \quad (C8)$$

$$(iv) \text{ ES } \hat{Q} = Pe \left\{ \frac{\phi_W e^{Pe}}{e^{Pe} - 1} - \frac{\phi_P e^{-Pe}}{1 - e^{-Pe}} \right\} \quad (C9)$$

It is useful to examine limiting cases of the ES, in a manner similar to that used in obtaining Eqs. (A8) - (A11). Now they are

$$\hat{Q} \rightarrow \begin{cases} -1 & (|PE| \rightarrow 0) \\ Pe \cdot \frac{\phi_W}{\phi_P - \phi_W} & (Pe \rightarrow +\infty) \\ Pe \cdot \frac{\phi_P}{\phi_P - \phi_W} & (Pe \rightarrow -\infty) \end{cases} \quad (C10)$$

In order to discriminate between the CDS, UDS and HS approaches to approximating the ES, we plot \hat{Q} versus Pe . The plot depends, of course, on the values of ϕ_P and ϕ_W . To illustrate the point, and without loss of generality of the result, let us take $\phi_W = 0$ and $\phi_P = 1$. For this case the \hat{Q} versus Pe relationships to plot are:

(i) CDS

$$\hat{Q} = \frac{1}{2} Pe - 1 \quad (C11)$$

(ii) UDS

$$\hat{Q} = \begin{cases} -1 & (u > 0) \\ Pe - 1 & (u < 0) \end{cases} \quad (C12)$$

(iii) HS

$$\hat{Q} = \begin{cases} \frac{1}{2} Pe - 1 & (|Pe| < 0) \\ 0 & (Pe \geq 2, u > 0) \\ Pe & (Pe \leq -2, u < 0) \end{cases} \quad (C13)$$

(iv) ES

$$\hat{Q} = -Pe \cdot \frac{e^{-Pe}}{1 - e^{-Pe}} \quad (C14)$$

The three limiting cases of the ES are also

$$\hat{Q} \rightarrow \begin{cases} -1 & (|Pe| \rightarrow 0) \\ 0 & (Pe \rightarrow +\infty) \\ -Pe & (Pe \rightarrow -\infty) \end{cases} \quad (C15)$$

In Fig. 13 are plotted four lines labelled CDS, UDS, HS and ES, each of which represents \hat{Q} as a function of Pe . That marked ES represents the exact solution and its three limiting cases are clearly seen. When $|Pe|$ is small, ES tends to -1 ; this is the value appropriate to \hat{Q} because of diffusion alone. When Pe is very large and positive, \hat{Q} tends to 0 ; when Pe is very large in magnitude, but negative, \hat{Q} tends to be equal to $-Pe$. With the CDS approach, \hat{Q} tends to $\frac{1}{2}Pe$ instead of zero when Pe is large and positive, and at the other extreme it tends to $\frac{1}{2}Pe$ instead of Pe . These disagreements are quite severe. And, of course, the finite difference equation coefficients obtained with the CDS approach when $|Pe| > 2$ are such as to encourage divergence during an iterative solution procedure. The UDS approach of the other hand, has errors which are larger than the CDS approach for moderate values of Pe , but when $|Pe|$ is large the UDS solution is more exact than the CDS one.

The HS approach combines the advantages of the CDS and UDS ideas and its corresponding straight lines are marked on the figure. Evidently this prescription is somewhat more accurate, overall, than either of its associated predecessors. It clearly tends to the asymptotes of 0 and $-Pe$, as Pe tends to $+\infty$ and $-\infty$ respectively (making it superior to the UDS for large values of $|Pe|$), and is equivalent to the CDS approach where the latter may be usefully employed when $|Pe| < 2$. The HS approach also happens to keep the finite difference coefficients within bounds which

ensure convergence of the iterative scheme for solving the resulting set of algebraic equations. Hence the recommendation and current use of the hybrid differencing scheme as given by Eq. (A13), and quoted earlier as Eq. (10).

APPENDIX D FORTRAN SYMBOL LIST

A(J)	=	Coefficient of recurrence relation
AE(I,J)	=	Coefficient of combined convective/diffusive flux through east-wall of control volume
AL1	=	X-coordinate of inlet boundary of flow domain
AL2	=	X-coordinate of outlet boundary of flow domain
ALAMDA	=	Length scale factor at inlet of flow domain
ALPHA	=	Inlet sloping wall expansion angle
ALTOT	=	Total length of pipe of larger diameter
AN(I,J)	=	Coefficient of combined convective/diffusive flux through north-wall of control volume
AP(I,J)	=	Sum of coefficients of combined convective/diffusive fluxes through all four walls of control volume
AREAEW	=	Area of east/west wall of control volume
AREAN	=	Area of north-wall of control volume
AREAS	=	Area of south-wall of control volume
ARDEN	=	Area of east/west cell-wall times density of fluid
ARDENT	=	Sum of all east/wall ARDEN at a cross-section
AS(I,J)	=	Coefficient of combined convective/diffusive flux through south-wall of control volume
AW(I,J)	=	Coefficient of combined convective/diffusive flux through west-wall of control volume
B(J)	=	Coefficient of recurrence formulae
C(J)	=	Coefficient of recurrence relation
C1	=	Constant of turbulence model (=1.44)
C2	=	Constant of turbulence model (=1.92)
CAPPA	=	Von Karman constant (=0.4187)

CD = Constant of turbulence model (=1.0)
 CDTERM = $CMU * * 0.25$
 CE = Coefficient of convective flux through east-wall of control volume
 CMU = Constant of turbulence model (=0.09)
 CN = Coefficient of convective flux through north-wall of control volume
 CP = Maximum of zero and net outflow (SMP) from control volume
 CPO = CP
 CS = Coefficient of convective flux through south-wall of control volume
 CW = Coefficient of convective flux through west-wall of control volume
 D(J) = Coefficient of recurrence formulae
 DE = Coefficient of diffusive flux through east-wall of control volume
 DEN(I,J) = Density of fluid
 DENSIT = Density of fluid at inlet of the calculation domain
 DITERM = Coefficient of volume integral of energy dissipation rate in vicinity of walls
 DN = Coefficient of diffusive flux through north-wall of control volume
 DS = Coefficient of diffusive flux through south-wall of control volume
 DU(I,K) = Coefficient of velocity-correction term for U velocity
 DUDXE = $\partial u / \partial x$ at eastern face of U-cell
 DUDXW = $\partial u / \partial x$ at western face of U-cell
 DUDX = $\partial u / \partial x$ at main grid node (I,J)

DUDY = $\partial u/\partial y$ at main grid node (I,J)
DUDYE = $\partial u/\partial y$ at mid-point of east wall of V-cell
DUDYW = $\partial u/\partial y$ at mid-point of west wall of V-cell
DVDX = $\partial v/\partial x$ at main grid node (I,J)
DVDXN = $\partial v/\partial x$ at mid-point of north wall of U-cell
DVDXS = $\partial v/\partial x$ at mid-point of south wall of U-cell
DVDY = $\partial v/\partial y$ at main grid node (I,J)
DVDYN = $\partial v/\partial y$ at mid-point of north wall of V-cell
DVDYS = $\partial v/\partial y$ at mid-point of south wall of V-cell
DW = Coefficient of diffusive flux through west wall of control
volume
DWDX = $\partial W/\partial x$ at main grid node (I,J)
JWDY = $\partial W/\partial y$ at main grid node (I,J)
DXEP(I) = $X(I+1)-X(I)$
DXEPU(I) = $XU(I+1)-XU(I)$
DXPW(I) = $X(I)-X(I-1)$
DYNP(J) = $Y(J+1)-Y(J)$
DYNPV(J) = $YV(J+1)-YV(J)$
DYPS(J) = $Y(J)-Y(J-1)$
DYPSV(J) = $YV(J)-YV(J-1)$
ED(I,J) = Energy dissipation rate, ϵ
EDIN = Energy dissipation rate at inlet of flow domain (ϵ_{in})
ELOG = Constant of P-function for heat transfer at walls (=9.793)
EPSX = Grid expansion factor in axial direction
FACTOR = Area ratio for setting initial u-velocity field
FLOW = Mass flow rate at a cross-section based on calculated velocity
FLOWIN = Total mass flow rate entering pipes

GE = Mass flux through east-wall of cell
 GEN(I,J) = Generation of turbulence by shear from mean flow
 GENCOU = Part of generation term modified in terms of wall shear stress
 GENRES = Total unmodified generation of turbulence (GEN(I,J) less
 $\mu_t(\partial v/\partial x)^2$.
 GN = Mass flux through north-wall of cell
 GNW = Mass flux through north-wall of u-cell
 GP = Mass flux at location of velocity
 GREAT = A very large value (10^{30})
 GS = Mass flux through south-wall of cell
 GSW = Mass flux through south-wall of u-cell
 HEDA = Heading 'KPLUS = TE * RHO/TAUN'
 HEDB = Heading 'LENGTH SCALE/PIPE RADIUS'
 HEDD = Heading 'ENERGY DISSIPATION'
 HEDDK = Heading 'DIMENSIONLESS TURBULENCE ENERGY'
 HEDDP = Heading 'DIMENSIONLESS PRESSURE'
 HEDDSL = Heading 'DIMENSIONLESS STREAMLINE COORDS'
 HEDDU = Heading 'DIMENSIONLESS U VELOCITY'
 HEDDV = Heading 'DIMENSIONLESS V VELOCITY'
 HEDDVS = Heading 'DIMENSIONLESS EFF. VISCOSITY'
 HEDDW = Heading 'DIMENSIONLESS W VELOCITY'
 HEDK = Heading 'TURBULENCE ENERGY'
 HEDM = Heading 'VISCOSITY'
 HEDP = Heading 'PRESSURE'
 HEDSF = Heading 'DIMENSIONLESS STREAM FUNCTION'
 HEDSL = Heading 'RADIAL COORDINATE OF STREAMLINES'
 HEDT = Heading 'TEMPERATURE'

HEDU = Heading 'U VELOCITY'
 HEDV = Heading 'V VELOCITY'
 I = Index for dependent variables, and co-ordinates
 IFINE = Logical parameter for fine x-direction grid spacing
 IMON = I-index of monitoring location
 INCALA = Additional (unused) logical parameter for selection of dependent variables
 INCALB = Additional (unused) logical parameter for selection of dependent variables
 INCALD = Logical parameter for solution of ϵ -equation
 INCALK = Logical parameter for solution of k-equation
 INCALM = Additional (unused) logical parameter for selection of dependent variables
 INCALP = Logical parameter for solution of P'-equation
 INCALS = Logical parameter for calculation of stream function
 INCALU = Logical parameter for solution of U-equation
 INCALV = Logical parameter for solution of V-equation
 INCALW = Logical parameter for solution of W-equation
 INDCOS = Control index for definition of co-ordinate system
 (= 1 for plane flows; = 2 for axisymmetric flows)
 INITAL = Logical parameter to print initial field estimates
 INPLOT = Logical parameter to produce line printer streamline plots
 IREAD = Logical parameter to read initial field values from allocated disk storage
 IWRITE = Logical parameter to write solution field values to allocated disk storage
 INPRO = Logical parameter for updating of fluid properties
 IPREF = I-index of location where pressure is fixed

IPRINT = If equal to NITER, activates printing of residual sums and
 monitor values of field variables
 ISTEP = I-index of entrance plane, within calculation domain
 ISTM1 = ISTEP-1
 ISTP1 = ISTEP+1
 IT = I-index of maximum dimension of dependent variables
 J = Index for dependent variables, and co-ordinate
 JMAX(I) = Maximum value of j-index within flow domain
 JMON = J-index of monitoring location
 JPREF = J-index of location where pressure is fixed
 JPRINT = If equal to NITER, activates printing of field variable values
 JSTEP = J-index of horizontal plane next to wall of, and within,
 smaller pipe
 JSTM1 = JSTEP-1
 JSTP1 = JSTEP+1
 JT = J-index of maximum dimension of dependent variables
 LFS = Index for counting loops for swirl
 LFSMAX = Number of swirl loops to be run
 MAXIT = Maximum number of iterations to be completed if iteration
 sequence is not stopped by test on value of SORCE
 NI = Maximum value of I-index
 NIM1 = NI-1
 NITER = Number of iterations completed
 NJ = Maximum value of J-index
 NJM1 = NJ-1
 NJM2 = NJ-2

NONDIM - Activates the calculation and printing of dimensionless solution when specified as true

NSBR = Zero value specifies flat W profile; the value one specifies solid body rotation

NSTLN = Number of streamlines calculated

NSWPD = Number of application of line iteration for ϵ -equation

NSWPK = Number of application of line iteration for k-equation

NSWPP = Number of application of line iteration for P'-equation

NSWPU = Number of application of line iteration for U-equation

NSWPW = Number of application of line iteration for W-equation

P(I,J) = Pressure, P

PHI(I,J) = General representation for all dependent variables, ϕ

PP(I,J) = Pressure-correction, P'

PRANDT = Turbulent Prandtl number

PRED = Constant of turbulence model in ϵ -equation, σ_ϵ

PRTE = Constant of turbulence model in k-equation, σ_ϵ

PSTAR(I,J) = Dimensionless pressure

R(J) = Radius of main grid node (I,J) from symmetry axis

RCV(J) = Radius of C- and U-cell center

RESOR = Residual source for individual control volume

RESORE = Sum of residual sources within calculation domain for ϵ -equation

RSORK = Sum of residual sources within calculation domain for k-equation

RESORM = Sum of mass sources within calculation domain

RESORU = Sum of residual sources within calculation domain for U-equation

RESORV = Sum of residual sources within calculation domain for V-equation

RESORW = Sum of residual sources within calculation domain

RLARGE = Radius of large pipe

RSDRL = RSMALL/RLARGE
 RSMALL = Radius of small pipe
 RV(J) = Radius of location of V(I,J) from symmetry axis
 SEW(I) = $0.5*(DXEP(I) + DXPW(I))$
 SEWU(I) = $0.5*(DXEPU(I) + DXPWU(I))$
 SMP = Net outflow from control volume
 SNS(J) = $0.5*(DYNP(J) + DYPS(J))$
 SNSV(J) = $0.5*(DYNPV(J) + DYPV(J))$
 SORCE = Maximum of RESORM, RESORU, RESORV, RESORW, RESORK
 SORMAX = Maximum acceptable value of SORCE for converged solution
 SORVOL = GREAT * VOL
 SP(I,J) = Coefficient of linearized source treatment
 SPKD(I,J) = -CP, for k- and ϵ -equations
 SSC = Shear-stress coefficient
 STFN(I,J) = Dimensionless stream function
 STVAL(K) = Stream function value of streamlines
 SU(I,J) = Coefficient of linearized source treatment
 SUKD(I,J) = $CPO * TE(I,J)$, for k-equation
 = $CPO * ED(I,J)$, for ϵ -equation
 SWNB(LFS) = Inlet swirl number specification of WINST
 SWRLNO = Calculated inlet swirl number
 TAUN(I) = Shear stress at north wall-boundary of flow domain
 TAURX = North wall shear stress, x-component
 TAURW = North wall shear stress, θ -component
 TAUW(J) = Shear stress at west wall-boundary of flow domain
 TAUWR = West wall shear stress, r-component
 TAUW θ = West wall shear stress, θ -component
 TE(I,J) = Turbulence energy, k

TESTAR(I,J) = Dimensionless turbulence energy
 TEIN = Turbulence energy at inlet of flow domain (k_{in})
 TMULT = Coefficient of wall shear-stress expression
 TURBIN = Turbulence intensity factor at inlet of flow domain
 U(I,J) = Component of mean velocity in axial direction (u-velocity)
 UEFF = $\text{SQRT}[U(I,J)**2 + W(I,J)**2]$
 UIN = U-velocity at inlet of flow domain
 UINC = Uniform increment of u-velocity at outlet of flow domain
 ULARGE = $UIN * (RSMALL/RLARGE)**2$
 UMEAN = Mean u-velocity at inlet
 URFE = Under-relaxation factor for energy dissipation
 URFK = Under-relaxation factor for turbulence energy
 URFP = Under-relaxation factor for pressure-correction
 URFU = Under-relaxation factor for u-velocity
 URFV = Under-relaxation factor for v-velocity
 URFVIS = Under-relaxation factor for viscosity
 URFW = Under-relaxation factor for w-velocity
 USTAR(I,J) = Dimensionless u-velocity
 V(I,J) = Component of mean velocity in radial direction (v-velocity)
 VANB(LFS) = Swirl vane angle
 VAVG = Average v-velocity between nodes (I,J) and (I,J+1)
 VDR = $V(I,J)/RV(J)$
 VIS(I,J) = Effective viscosity
 VISCOS = Laminar viscosity
 VISE = Effective viscosity at mid-point of east-wall of cell

VISOLD = Value of effective viscosity before underrelaxation
 VISN = Effective viscosity at mid-point of north-wall of cell
 VISS = Effective viscosity at mid-point of south-wall of cell
 VISTAR(I,J)= Dimensionless effective viscosity
 VISW = Effective viscosity at mid-point of west-wall of cell
 VOL = Volume of cell or control-volume
 VSTAR(I,J)= Dimensionless v-velocity
 W(I,J) = w-velocity
 WIN = Inlet w-velocity from swirl vanes
 WINST = Inlet w-velocity at JSTEP from solid body rotation swirl
 generator
 WMONIN = Inlet swirl momentum
 WSTAR(I,J)= Dimensionless w-velocity
 X(I) = Distance from inlet plane in axial direction
 XMONIN = Momentum of fluid at inlet of flow domain
 XND(I) = Dimensionless X(I)
 XPLUSW(I) = Local Reynolds number based on friction velocity and
 distance from west wall-boundary of flow domain
 XU(I) = X-coordinate of storage location of U(I,J)
 XUDPLT(I) = Dimensionless XU(I) required for line printer streamline
 plots
 Y(J) = Distance from symmetry axis in radial direction
 YND(J) = Dimensionless Y(J)
 YPLUSN(J) = Local Reynolds number based on friction velocity and distance
 from north wall-boundary of flow domain
 YSLOPE = Height of first stairstep along sloping wall
 YSLPLT(N,I)= Dimensionless YSTLN(I,K) required for line printer streamline
 plots

YSTLN(I,K) = Radial coordinate of points representing streamlines
YSTLND(I,K) = Dimensionless YSTLN(I,K)
YV(J) = Y-coordinate of storage location of V(I,J)
YUND(J) = Dimensionless YV(J)

APPENDIX E COMPUTER PROGRAM LISTING

The computer program listing obtained for the sample computation described in Section 4.3 is now given.

```

C          SUBROUTINE CONTRO                                0000100
C          0000200
CA*****0000300
C          0000400
C          0000500
C          0000600
C          0000700
C          0000800
C          0000900
C          0001000
C          0001100
C          0001200
C          0001300
C          0001400
C          0001500
C          0001600
CA*****0001700
C          0001800
CHAPTER 0 0 0 0 0 0 0 0 0 PRELIMINARIES 0 0 0 0 0 0 0 0 0 0001900
C          0002000
          DIMENSION HEDU(9), HEDV(9), HEDW(9), HEDP(9), HEDT(9), HEDK(9),
          *HEDD(9), HEDM(9), HEDL(9), VANB(7), SWNB(7), HEDSF(9), HEDSL(9),
          #HEDDU(9), HEDDV(9), HEDDW(9),
          #HEDDP(9), HEDDK(9), HEDDSL(9), HEDDVS(9)
          DIMENSION YAXES(10), SYMB L(10)
          COMMON
          1/UVEL/RESORU, NSWPU, URFU, DXEPU(48), DXPWU(48), SEWU(48)
          1/VVEL/RESORV, NSWPV, URFV, DYNPV(24), DYPSV(24), SNSV(24)
          */WVEL/ RESORW, NSWPW, URFW
          1/PCOR/RESORM, NSWPP, URFP, DU(48,24), DV(48,24), IPREF, JPREF
          1/TEN/RESORK, NSWPK, URFK
          1/TDIS/RESORE, NSWPD, URFE
          */VAR/U(48,24), V(48,24), W(48,24), P(48,24), PP(48,24), TE(48,24),
          *ED(48,24), STFN(48,24), YSTLN(48,24), STVAL(24), USTAR(48,24),
          *VSTAR(48,24), WSTAR(48,24), PSTAR(48,24), TESTAR(48,24), YSTLND(48,24)
          #, VISTAR(48,24)
          1/ALL/IT, JT, NI, NJ, NIM1, NJM1, GREAT, JMAX(48), JMAXP1(48)
          1/GEOM/INDCOS, X(48), Y(24), DXEP(48), DXPW(48), DYNP(24), DYP(24),
          1 SNS(24), SEW(48), XU(48), YV(24), R(24), RV(24),
          # WFN(24), WFS(24), WFE(48), WFW(48), RCV(24), XND(48), XUND(48),
          #YND(24), YVND(24)
          COMMON
          1/FLUPR/URFVIS, VISCOS, DENSIT, PRANDT, DEN(48,24), VIS(48,24)
          1/KASE T1/ UIN, TEIN, EDIN, FLOWIN, ALAMDA,
          2 RSMALL, RLARGE, AL1, AL2, JSTEP, ISTEP, JSTP1, JSTM1, ISTP1, ISTM1
          1/TURB/GEN(48,24), CD, CMU, C1, C2, CAPP, ELOG, PRED, PRTE
          1/WALLF/YPLUSN(48), XPLUSW(24), TAUN(48), TAUM(24)
          1/COEF/AP(48,24), AN(48,24), AS(48,24), AE(48,24), AW(48,24), SU(48,24),
          1 SP(48,24)
          1/PLOTT/NSTLN, HPLTLN, NPTS, YSLPLT(10,48), XUDPLT(48), INPLOT
          LOGICAL INCALU, INCALV, INCALW, INCALP, INPRO, INCALK, INCALD, INCALM,
          *INCALA, INCALB, INCALS, INPLOT, IWRITE, NONDIM, IREAD, INLET,
          *INITAL, IFINE
C-----ALL PRIMARY USER INPUTS ARE LOCATED HERE
          DATA VANB /0.,.45.,.55.,.60.,.65.,.68.,.70./
          #, SWNB/0.,.5,1.0,1.25,1.50,1.75,2.0/
          DATA XAXIS /3HX1 /
          DATA YAXES /3H 00,3H 02,3H 04,3H 06,3H 08,3H 10,
          #3H 06,3H 07,3H 08,3H 09/
          DATA SYMB L /1H0,1H2,1H4,1H6,1H8,1H1,1H6,1H7,1H8,1H9/
C-----DELETE UNDERFLOW ERROR MESSAGES, CALL TRAPS WHEN USING WATFIV 0006000

```

```

C-----CALL TRAPS(1,1,4000)                                0006100
      CALL ERRSET(208,256,-1,1,0,0)                          0006200
C-----SET INPLOT=.TRUE. ONLY FOR STREAMLINE LINE-PRINTER PLOT 0006300
      INPLOT=.TRUE.                                           0006400
C-----SET IWRITE=.TRUE. ONLY FOR WRITING SOLN. ON DATA FILES 0006500
      IWRITE=.FALSE.                                          0006600
C-----SET NONDIM=.TRUE. ONLY FOR PRINTING DIMENSIONLESS SOLN. 0006700
      NONDIM=.TRUE.                                           0006800
C-----SET IREAD=.TRUE. ONLY FOR READING INITIAL GUESS OF SOLN. 0006900
C-----FROM DATA FILES                                     0007000
      IREAD=.FALSE.                                           0007100
C-----SET INITAL=.TRUE. ONLY FOR PRINTING INITIAL GUESS OF SOLN. 0007200
      INITAL=.TRUE.                                           0007300
C-----SET IFINE=.TRUE. ONLY FOR FINE GRID IN X-DIRECTION     0007400
      IFINE=.FALSE.                                           0007500
      IF(.NOT. IREAD) GO TO 150                                0007600
      READ(12) X                                               0007700
      READ(12) Y                                               0007800
150 CONTINUE                                                  0007900
      NSTLN=11                                                 0008000
      HPLTLN=6                                                 0008100
      MAXLN=10                                                 0008200
      NITER=0                                                  0008300
      JPRINT=NITER+300                                         0008400
      IPRINT=NITER+1                                           0008500
      LFS=1                                                    0008600
      LFSMAX=7                                                 0008700
      NSBR=0                                                   0008800
      MAXIT=NITER+200                                          0008900
C-----SEE STATEMENT 304 FOR MAXIT SPEC. FOR FURTHER SWIRL CASES 0009000
      DENSIT=1.211                                           0009100
      IT=48                                                    0009200
      JT=24                                                    0009300
      GREAT=1.E30                                             0009400
      NSWPU=4                                                  0009500
      NSWVP=3                                                  0009600
      NSWPM=3                                                  0009700
      NSWPP=5                                                  0009800
      NSWPK=3                                                  0009900
      NSWPD=3                                                  0010000
      READ(5,10) HEDU,HEDV,HEDW,HEDP,HEDT,HEDK,HEDD,HEDM,HEDL, 0010100
      $HEDSF,HEDSL,HEDDU,HEDDV,HEDDW,HEDDP,$EDDK,HEDDSL,HEDDVS 0010200
      010 FORMAT(9A4)                                         0010300
C                                                                0010400
CHAPTER 1 1 1 1 1 PARAMETERS AND CONTROL INDICES 1 1 1 1 1 1 0010500
C                                                                0010600
C-----GRID                                                 0010700
      ISTEP=2                                                  0010800
      JSTEP=8                                                  0010900
      INDCOS=2                                                 0011000
      NJ=21                                                    0011100
      NJM1=NJ-1                                               0011200
      ISTEP1=ISTEP+1                                          0011300
      ISTM1=ISTEP-1                                          0011400
      JSTEP1=JSTEP+1                                          0011500
      JSTM1=JSTEP-1                                          0011600
      RLARGE=.0625                                           0011700
      ALTOT=.375                                              0011800
      IF(IFINE) GO TO 120                                      0011900
C-----COARSE MESH GRID LINES IN X-DIRECTION                0012000

```

	NI=20	0012100
	NIM1=NI-1	0012200
	EPSX=1.11	0012300
	IF(EPSX-1.) 13,12,13	0012400
13	SUMX=0.5*EPSX**(NI-4)+(EPSX**(NI-3)-1.)/(EPSX-1.)+0.5	0012500
	GO TO 15	0012600
12	CONTINUE	0012700
	SUMX=NIM1-1	0012800
15	DX=ALOT/SUMX	0012900
	X(1)=-.5*DX	0013000
	X(2)=-X(1)	0013100
	DO 100 I=3,NIM1	0013200
	X(I)=X(I-1)+DX	0013300
100	DX=EPSX*DX	0013400
	X(NI)=X(NIM1)+(X(NIM1)-X(NI-2))	0013500
	X(21)=X(20)+(X(20)-X(19))	0013600
	X(22)=X(21)+(X(20)-X(19))	0013700
	X(23)=X(22)+(X(20)-X(19))	0013800
	NI=23	0013900
	NIM1=NI-1	0014000
	ALOT=(X(22)+X(23))/2.	0014100
	AL1=0.5*(X(ISTEP)+X(ISTM1))	0014200
	AL2=ALOT-AL1	0014300
C-----	-----FINE MESH GRID LINES IN X-DIRECTION	0014400
	IF(.NOT. IFINE) GO TO 130	0014500
120	NI=30	0014600
	NIM1=NI-1	0014700
	EPSX=1.102	0014800
	IF(EPSX-1.) 17,16,17	0014900
17	SUMX=0.5*EPSX**(NI-4)+(EPSX**(NI-3)-1.)/(EPSX-1.)+0.5	0015000
	GO TO 18	0015100
16	CONTINUE	0015200
	SUMX=NIM1-1	0015300
18	DX=ALOT/SUMX	0015400
	X(1)=-.5*DX	0015500
	X(2)=-X(1)	0015600
	DO 170 I=3,NIM1	0015700
	X(I)=X(I-1)+DX	0015800
170	DX=EPSX*DX	0015900
	X(NI)=X(NIM1)+(X(NIM1)-X(NI-2))	0016000
	DO 180 L=31,35	0016100
180	X(L)=X(L-1)+(X(NI)-X(NIM1))	0016200
	NI=35	0016300
	NIM1=NI-1	0016400
	ALOT=(X(NIM1)+X(NI))/2.	0016500
	AL1=0.5*(X(ISTEP)+X(ISTM1))	0016600
	AL2=ALOT-AL1	0016700
C-----	-----SPECIFY RADIAL HEIGHT(NO. OF J-CELLS) OF COMBUSTOR	0016800
C-----	-----WALL FOR EACH I GRID LINE	0016900
130	CONTINUE	0017000
	JMAX(I)=JSTEP	0017100
	JMAXPI(1)=JMAX(1)+1	0017200
	DO 160 I=2,NI	0017300
	JMAX(I)=JMAX(I-1)+3	0017400
	IF(JMAX(I-1) .EQ. NJM1) JMAX(I)=JMAX(I-1)	0017500
160	JMAXPI(I)=JMAX(I)+1	0017600
C-----	-----GRID LINES IN Y-DIRECTION	0017700
	Y(1)=-1.5625E-3	0017800
	Y(2)=1.5625E-3	0017900
	Y(3)=5.625E-3	0018000

Y(4)=1.0625E-2	0018100
Y(5)=1.625E-2	0018200
Y(6)=2.1875E-2	0018300
Y(7)=2.6875E-2	0018400
Y(8)=3.03125E-2	0018500
Y(9)=3.21875E-2	0018600
Y(10)=3.4375E-2	0018700
Y(11)=3.71875E-2	0018800
Y(12)=4.09375E-2	0018900
Y(13)=4.328125E-2	0019000
Y(14)=4.5625E-2	0019100
Y(15)=4.8125E-2	0019200
Y(16)=5.078125E-2	0019300
Y(17)=5.34375E-2	0019400
Y(18)=5.59375E-2	0019500
Y(19)=5.9375E-2	0019600
Y(20)=6.1875E-2	0019700
Y(21)=6.3125E-2	0019800
RSMALL=0.5*(Y(JSTEP)+Y(JSTP1))	0019900
C-----DEPENDENT VARIABLE SELECTION	0020000
INCALU=.TRUE.	0020100
INCALV=.TRUE.	0020200
INCALW=.TRUE.	0020300
INCALP=.TRUE.	0020400
INCALK=.TRUE.	0020500
INCALD=.TRUE.	0020600
INPRO=.TRUE.	0020700
INCALS=.TRUE.	0020800
C-----FLUID PROPERTIES	0020900
C-----TURBULENCE CONSTANTS	0021000
CMU=0.09	0021100
CD=1.00	0021200
C1=1.44	0021300
C2=1.92	0021400
CAPP=.4187	0021500
ELOG=9.793	0021600
PRED=CAPP*CAPP/(C2-C1)/(CMU*.5)	0021700
PRTE=1.0	0021800
C-----BOUNDARY VALUES	0021900
UIN=30.	0022000
ULARGE=UIN*(RSMALL/RLARGE)**2	0022100
TURBIN=.03	0022200
TEIN=TURBIN*UIN**2	0022300
ALAMDA=0.005	0022400
EDIN=TEIN**1.5/(ALAMDA*RLARGE)	0022500
VISCOS=1.8E-5	0022600
C-----PRESSURE CALCULATION	0022700
IPREF=2	0022800
JPREF=2	0022900
C-----PROGRAM CONTROL AND MONITOR	0023000
IMON=NIM1	0023100
JMON=8	0023200
SORMAX=.004	0023300
C	0023400
CHAPTER 2 2 2 2 2 2 INITIAL OPERATIONS 2 2 2 2 2 2 2 2 2	0023500
CALL INIT	0023600
C-----NONDIMENSIONALIZE X & Y VARIABLES FOR NONDIMENSIONAL OUTPUT	0023700
DO 50 I=1,NI	0023800
XND(I)=X(I)/(2.*RLARGE)	0023900
50 XUND(I)=XU(I)/(2.*RLARGE)	0024000

	DO 60 J=1,NJ	0024100
	YND(J)=Y(J)/(2.*RLARGE)	0024200
	60 YVND(J)=YV(J)/(2.*RLARGE)	0024300
C-----	INITIALISE VARIABLE FIELDS	0024400
	FLOWIN=0.0	0024500
	ARDEN=0.0	0024600
	ARDENT=0.	0024700
	XMONIN=0.	0024800
	WMONIN=0.0	0024900
	ANGMOM=0.	0025000
C		0025100
C-----	INLET SWIRL VELOCITY PROFILE	0025200
C		0025300
C-----	-W, USE SOLID BODY ROTATION MODEL	0025400
	WINST=2.*SWNB(LFS)/(1.+SWNB(LFS))*UIN	0025500
	DO 206 J=2,JSTEP	0025600
206	W(1,J)=WINST*R(J)/R(JSTEP)	0025700
C-----	-NSBR=0 - FLAT SWIRL VELOCITY PROFILE FROM SWIRL VANES	0025800
C-----	-NSBR=1 - SOLID BODY ROTATION FROM SWIRL GENERATOR	0025900
	IF(NSBR.EQ.1) GO TO 208	0026000
C-----	-W, FLAT PROFILE	0026100
	WIN=UIN*TAN(VANB(LFS)*3.14159/180.)	0026200
	DO 207 J=2,JSTEP	0026300
207	W(1,J)=WIN	0026400
208	CONTINUE	0026500
C-----	INITIALIZE U-,TE-,ED-, & W-FIELDS	0026600
	DO 200 J=2,JSTEP	0026700
	U(2,J)=UIN	0026800
	TE(1,J)=TEIN	0026900
	ED(1,J)=EDIN	0027000
	ARDEN=0.5*(DEN(1,J)+DEN(2,J))*RCV(J)*SNS(J)	0027100
	XMONIN=XMONIN+ARDEN*U(2,J)*U(2,J)	0027200
	WMONIN=WMONIN+ARDEN*U(2,J)*W(1,J)	0027300
	ANGMOM=ANGMOM+ARDEN*U(2,J)*W(1,J)*R(J)	0027400
	ARDENT=ARDENT+ARDEN	0027500
200	FLOWIN=FLOWIN+ARDEN*U(2,J)	0027600
	UMEAN=FLOWIN/ARDENT	0027700
	SWRLNO=ANGMOM/(XMONIN*RSMALL)	0027800
	IF(W(1,JSTEP).EQ.0.) WMONIN=1.	0027900
	DO 202 I=2,NI	0028000
	IJ=JMAXP1(I-1)	0028100
	FACTOR=(YV(JSTP1)*RV(JSTP1))/(YV(IJ)*RV(IJ))	0028200
	JJ=JMAX(I-1)	0028300
	DO 202 I=2,JJ	0028400
202	U(I,J)=UIN*FACTOR	0028500
	CONTINUE	0028600
C		0028700
	IF(NSBR.EQ.0) GO TO 219	0028800
	DO 209 I=2,NI	0028900
	JJ=JMAX(I)	0029000
	DO 209 J=2,JJ	0029100
	WINST=2.*SWNB(LFS)/(1.+SWNB(LFS))*U(I,2)	0029200
	W(I,J)=WINST*R(J)/R(JJ)	0029300
	TE(I,J)=TEIN	0029400
	ED(I,J)=EDIN	0029500
209	CONTINUE	0029600
	GO TO 221	0029700
C		0029800
219	CONTINUE	0029900
	DO 220 I=2,NI	0030000

	JJ=JMAX(I)	0030100
	DO 220 J=2, JJ	0030200
	TE(I, J)=TEIN	0030300
	ED(I, J)=EDIN	0030400
220	CONTINUE	0030500
C		0030600
221	CONTINUE	0030700
	DO 203 I=2, NIM1	0030800
203	YPLUSN(I)=11.0	0030900
	DO 204 J=JSTEP, NJ	0031000
	XPLUSW(J)=11.0	0031100
204	IF(J.EQ.JSTEP) XPLUSW(J)= 0.0	0031200
	URFVIS=.7	0031300
	CALL PROPS	0031400
C-----	INITIAL OUTPUT	0031500
	WRITE(6, 211)	0031600
	IK=JMAXP1(ISTEP)	0031700
	YSLOPE=YV(IK)-YV(JSTP.1)	0031800
	IF(JMAX(ISTEP) .LT. NJM1) ALPHA=ATAN(YSLOPE/SEWU(2))*180./3.14159	0031900
	IF((JSTEP .LT. NJM1) .AND. (JMAX(ISTEP) .EQ. NJM1)) ALPHA=90.	0032000
	WRITE(6, 225) ALPHA	0032100
	WRITE(6, 235) RSMALL	0032200
	WRITE(6, 240) RLARGE	0032300
	WRITE(6, 245) ALTOT	0032400
	RE=UIN*RSMALL*2.0*DENSIT/VISCOS	0032500
	WRITE(6, 250) RE	0032600
	WRITE(6, 255) VISCOS	0032700
	RSDRL=RSMALL/RLARGE	0032800
	WRITE(6, 260) DENSIT	0032900
295	CONTINUE	0033000
	IF(.NOT. IWRITE) GO TO 297	0033100
	WRITE(11) X	0033200
	WRITE(11) Y	0033300
	WRITE(14) XUND	0033400
297	CONTINUE	0033500
	IF(.NOT. IREAD) GO TO 298	0033600
	READ(12) U	0033700
	READ(12) V	0033800
	READ(12) W	0033900
	READ(12) P	0034000
	READ(12) TE	0034100
	READ(12) ED	0034200
	READ(12) VIS	0034300
	READ(12) STFJ	0034400
298	CONTINUE	0034500
	IF(.NOT. INITAL) GO TO 299	0034600
	IF(INCALU) CALL PRINT(1, 1, NI, NJ, IT, JT, XU, Y, U, HEDU)	0034700
	IF(INCALV) CALL PRINT(1, 1, NI, NJ, IT, JT, X, YV, V, HEDV)	0034800
	IF(INCALW) CALL PRINT(1, 1, NI, NJ, IT, JT, X, Y, W, HEDW)	0034900
	IF(INCALK) CALL PRINT(1, 1, NI, NJ, IT, JT, X, Y, TE, HEDK)	0035000
	IF(INCALD) CALL PRINT(1, 1, NI, NJ, IT, JT, X, Y, ED, HEDD)	0035100
299	CONTINUE	0035200
	RESORU=.005	0035300
	URFP=1.	0035400
	RESORV=.005	0035500
	RESORW=.005	0035600
C		0035700
CHAPTER	3 3 3 3 3 3 3 3 ITERATION LOOP 3 3 3 3 3 3 3 3	0035800
C		0035900
	WRITE(6, 310) IMON, JMON	0036000

300	NITER=NITER+1	0036100
	DO 330 I=2,NIM1	0036200
	DO 330 J=2,NJM1	0036300
330	PP(I,J)=0.	0036400
	IF(LFS .GE. 3) GO TO 425	0036500
	URFU=.5	0036600
	URFV=.5	0036700
	URFW=.6	0036800
	URFK=.7	0036900
	URFE=.7	0037000
	URFVIS=.7	0037100
	IF(LFS .LT. 3) GO TO 430	0037200
425	CONTINUE	0037300
C-----	INCREASE UNDERRELAXATION FACTORS AS CONVERGENCE NEARS	0037400
	URFU=.15+(FLOAT(NITER))*((.35-.15)/40.)	0037500
	IF(URFU .GT. .35) URFU=.35	0037600
	URFV=.20+(FLOAT(NITER))*((.25-.20)/40.)	0037700
	IF(URFV .GT. .25) URFV=.25	0037800
	URFW=.50+(FLOAT(NITER))*((.60-.50)/40.)	0037900
	IF(URFW .GT. .60) URFW=.60	0038000
	URFK=.70	0038100
	URFE=.70	0038200
	URFVIS=.70	0038300
	IF((RESORV .LT. .08) .AND. (RESORU .LT. .10)) URFU=.40	0038400
	IF((RESORV .LT. .06) .AND. (RESORU .LT. .08)) URFU=.45	0038500
	IF((RESORK .LT. .10E-1) .AND. (RESORE .LT. .10E+10)) URFE=.75	0038600
	IF((RESORK .LT. .4E-2) .AND. (RESORE .LT. .5E+9)) URFE=.80	0038700
	IF(RESORU .LT. SORMAX) URFU=.20	0038800
	IF(RESORV .LT. SORMAX) URFV=.20	0038900
	IF(RESORW .LT. SORMAX) URFW=.25	0039000
	IF(RESORK .LT. .10E-2) URFK=.60	0039100
	IF(RESORE .LT. .10E+8) URFE=.55	0039200
430	CONTINUE	0039300
C-----	UPDATE MAIN DEPENDENT VARIABLES	0039400
	IF(INCALU) CALL CALCU	0039500
	IF(INCALV) CALL CALCV	0039600
	IF(INCALP) CALL CALCP	0039700
	IF(INCALW) CALL CALCW	0039800
	IF(INCALK) CALL CALCTE	0039900
	IF(INCALD) CALL CALCED	0040000
C-----	UPDATE FLUID PROPERTIES	0040100
	IF(INPRO) CALL PROPS	0040200
C-----	INTERMEDIATE OUTPUT	0040300
	RESORM=RESORM/FLOWIN	0040400
	RESORU=RESORU/XMONIN	0040500
	RESORV=RESORV/XMONIN	0040600
	RESORW=RESORW/WMONIN	0040700
	RESORK=RESORK/(.5*FLOWIN*UMEAN*UMEAN)	0040800
	IF(NITER .NE. IPRINT) GO TO 301	0040900
	IPRINT=IPRINT+1	0041000
	WRITE(6,311) NITER,RESORU,RESORV,RESORW,RESORM,RESORK,	0041100
	*RESORE,U(IMON,JMON),V(IMON,JMON),W(IMON,JMON),P(IMON,NJM1),	0041200
	*ED(IMON,NJM1)	0041300
	IF(NITER .NE. JPRINT) GO TO 301	0041400
	IF(INCALU) CALL PRINT(1,1,NI,NJ,IT,JT,XU,Y,U,HEDU)	0041500
	IF(INCALV) CALL PRINT(1,1,NI,NJ,IT,JT,X,YV,V,HEDV)	0041600
	IF(INCALW) CALL PRINT(1,1,NI,NJ,IT,JT,X,Y,W,HEDW)	0041700
	IF(INCALP) CALL PRINT(1,1,NI,NJ,IT,JT,X,Y,P,HEDP)	0041800
	IF(INCALD) CALL PRINT(1,1,NI,NJ,IT,JT,X,Y,ED,HEDD)	0041900
	JPRINT=JPRINT+25	0042000

	WRITE(6,310) IMON,JMON	0042100
301	CONTINUE	0042200
C-----	TERMINATION TESTS	0042300
	SORCE=AMAX1(RESORM,RESORU,RESORV,RESORW,RESORK)	0042400
	IF(NITER.GE.MAXIT) GO TO 302	0042500
303	IF(NITER.GE.150.AND.SORCE.GE.3.0) GO TO 302	0042600
	IF(SORCE.GT.SORMAX.OR.NITER.LT.20) GO TO 300	0042700
302	CONTINUE	0042800
	IF(NITER.GE.150.AND.SORCE.GE.3.0) WRITE(6,960)	0042900
C		0043000
CHAPTER 4 4 4 4 4 4	FINAL OPERATIONS AND OUTPUT 4 4 4 4 4 4	0043100
C		0043200
440	CONTINUE	0043300
C-----	NONDIMENSIONALIZE PROBLEM SOLN.	0043400
	IF(.NOT.NONDIM) GO TO 700	0043500
	DO 600 I=1,NI	0043600
	DO 600 J=1,NJ	0043700
	USTAR(I,J)=U(I,J)/UIN	0043800
	VSTAR(I,J)=V(I,J)/UIN	0043900
	WSTAR(I,J)=W(I,J)/UIN	0044000
	PSTAR(I,J)=P(I,J)/(DENSIT*(UIN**2)/2.)	0044100
	TESTAR(I,J)=TE(I,J)/(UIN*UIN)	0044200
	SP(I,J)=0.0	0044300
	IF(ED(I,J).GT.1.E-15) SP(I,J)=TE(I,J)**1.5/ED(I,J)/RLARGE	0044400
	VISTAR(I,J)=VIS(I,J)/VISCOS	0044500
600	CONTINUE	0044600
700	CONTINUE	0044700
	IF(INCALS) CALL STRMFN	0044800
	WRITE(6,312)	0044900
	WRITE(6,410) LFS,NSBR,SWNB(LFS),VANB(LFS),SWRLNO,UMEAN,FLOWIN	0045000
	IF(INCALU) CALL PRINT(1,1,NI,NJ,IT,JT,XU,Y,U,HEDU)	0045100
	IF(INCALV) CALL PRINT(1,1,NI,NJ,IT,JT,X,YV,V,HEDV)	0045200
	IF(INCALW) CALL PRINT(1,1,NI,NJ,IT,JT,X,Y,W,HEDW)	0045300
	IF(INCALP) CALL PRINT(1,1,NI,NJ,IT,JT,X,Y,P,HEDP)	0045400
	IF(INCALP) CALL PRINT(1,1,NI,NJ,IT,JT,X,Y,PP,HEDPP)	0045500
	IF(INCALK) CALL PRINT(1,1,NI,NJ,IT,JT,X,Y,TE,HEDK)	0045600
	IF(INCALD) CALL PRINT(1,1,NI,NJ,IT,JT,X,Y,ED,HEDD)	0045700
	IF(INCALS) CALL PRINT(1,1,NI,NJ,IT,JT,XU,Y,STFN,HEDSF)	0045800
	IF(INCALS) CALL PRINT(1,1,NI,NSTLN,IT,JT,XU,STVAL,YSTLN,HEDSL)	0045900
	IF(.NOT.NONDIM) GO TO 750	0046000
	IF(INCALU) CALL PRINT(1,1,NI,NJ,IT,JT,XUND,YND,USTAR,HEDDU)	0046100
	IF(INCALV) CALL PRINT(1,1,NI,NJ,IT,JT,XND,YVND,VSTAR,HEDDV)	0046200
	IF(INCALW) CALL PRINT(1,1,NI,NJ,IT,JT,XND,YND,WSTAR,HEDDW)	0046300
	IF(INCALS) CALL PRINT(1,1,NI,NSTLN,IT,JT,XUND,STVAL,YSTLND,HEDDSL)	0046400
	IF(INCALP) CALL PRINT(1,1,NI,NJ,IT,JT,XND,YND,PSTAR,HEDDP)	0046500
	IF(INCALK) CALL PRINT(1,1,NI,NJ,IT,JT,XND,YND,TESTAR,HEDDK)	0046600
	IF(INCALK) CALL PRINT(2,2,NI,NJ,IT,JT,X,Y,SP,HEDL)	0046700
	IF(INCALK) CALL PRINT(1,1,NI,NJ,IT,JT,XND,YND,VISTAR,HEDDVS)	0046800
750	CONTINUE	0046900
	IF(.NOT.IWRITE) GO TO 702	0047000
	WRITE(11) U	0047100
	WRITE(11) V	0047200
	WRITE(11) W	0047300
	WRITE(11) P	0047400
	WRITE(11) TE	0047500
	WRITE(11) ED	0047600
	WRITE(11) VIS	0047700
	WRITE(11) STFN	0047800
	WRITE(14) YSTLND	0047900
702	CONTINUE	0048000

C-----CALCULATION OF SHEAR-STRESS COEFFICIENT ALONG LARGE DUCT WALL	0048100
WRITE(6,402)	0048200
DO 401 I=2,NIM1	0048300
SSC=ABS(TAUN(I))/(.5*DENSIT*UIN*UIN)	0048400
WRITE(6,403) I,XND(I),SSC	0048500
401 CONTINUE	0048600
WRITE(6,312)	0048700
C-----PLOT DIMENSIONLESS STREAMLINES	0048800
LARGE=0	0048900
IF(INPLOT .AND. INCALS) CALL PLOT (XUDPLT,IT,NPTS,XAXIS,YSLPLT,	0049000
#MAXLN,NPLTLN,YAXES,SYMB L,LARGE)	0049100
LARGE=1	0049200
IF(INPLOT .AND. INCALS) CALL PLOT (XUDPLT,IT,NPTS,XAXIS,YSLPLT,	0049300
#MAXLN,NPLTLN,YAXES,SYMB L,LARGE)	0049400
C-----RESET INITIAL CONDITIONS FOR ANOTHER SWIRL CASE	0049500
IF(LFS .GE. LFSMAX) GO TO 409	0049600
LFS=LFS+1	0049700
NITER=0	0049800
JPRINT=NITER+300	0049900
IPRINT=NITER+1	0050000
304 IF(LFS .GE. 3) MAXIT=NITER+200	0050100
IF(NSFR .EQ. 0) GO TO 405	0050200
WINST=2.*SWNB(LFS)/(1.+SWNB(LFS))*UIN	0050300
DO 406 J=2,JSTEP	0050400
406 W(1,J)=WINST*R(J)/R(JSTEP)	0050500
GO TO 408	0050600
405 WIN=UIN*TAN(VANB(LFS)*3.14159/180.)	0050700
DO 407 J=2,JSTEP	0050800
407 W(1,J)=WIN	0050900
408 FLOWIN=0.	0051000
ARDEN=0.	0051100
ARDENT=0.	0051200
XMONIN=0.	0051300
ANGMOM=0.	0051400
WMONIN=0.	0051500
C-----READ INITIAL GUESS OF NEXT SWIRL PROBLEM FROM	0051600
C-----PREVIOUS SOLN. OF SIMILAR PROBLEM	0051700
IF(.NOT. IREAD) GO TO 445	0051800
READ(12) U	0051900
READ(12) V	0052000
READ(12) W	0052100
READ(12) P	0052200
READ(12) TE	0052300
READ(12) ED	0052400
READ(12) VIS	0052500
READ(12) STFN	0052600
445 CONTINUE	0052700
DO 490 J=2,JSTEP	0052800
ARDEN=0.5*(DEN(1,J)+DEN(2,J))*RCV(J)*SNS(J)	0052900
XMONIN=XMONIN+ARDEN*U(2,J)*U(2,J)	0053000
WMONIN=WMONIN+ARDEN*U(2,J)*W(1,J)	0053100
ANGMOM=ANGMOM+ARDEN*U(2,J)*W(1,J)*R(J)	0053200
ARDENT=ARDENT+ARDEN	0053300
490 FLOWIN=FLOWIN+ARDEN*U(2,J)	0053400
UMEAN=FLOWIN/ARDENT	0053500
SHRLNO=ANGMOM/(XMONIN*RSMLL)	0053600
IF(W(1,JSTEP) .EQ. 0.) WMONIN=1.	0053700
WRITE(6,310) IMON,JMON	0053800
GO TO 300	0053900
409 CONTINUE	0054000

```

STOP
C-----FORMAT STATEMENTS
211 FORMAT(1H1,T37,'AXISYMMETRIC,ISOTHERMAL, GT COMBUSTOR FLOWFIELD SI
#MULATION',///,T35,'USING THE STAIRSTEP APPROXIMATION FOR THE SLOPIN
#G EXPANSION WALL',///,T53,'AND THE K-E TURBULENCE MODEL')
225 FORMAT(////,T40,'EXPANSION ANGLE(DEG.) =',T77,1PE13.3)
230 FORMAT(///,T40,'NUMBER OF STAIRSTEPS =',T81,I1)
235 FORMAT(///,T40,'INLET RADIUS(M) =',T77,1PE13.3)
240 FORMAT(///,T40,'COMBUSTOR RADIUS(M) =',T77,1PE13.3)
245 FORMAT(///,T40,'COMBUSTOR LENGTH(M) =',T77,1PE13.3)
250 FORMAT(///,T40,'INLET REYNOLDS NO.(USING DIAM.) =',T77,1PE13.3)
255 FORMAT(///,T40,'LAMINAR VISCOSITY(KG/M/SEC) =',T77,1PE13.3)
260 FORMAT(///,T40,'DENSITY(KG/CU. M) =',T77,1PE13.3,////)
310 FORMAT(13H0ITER I---, 9X,29HABSOLUTE RESIDUAL SOURCE SUMS,9X,
11H---I I---,37H FIELD VALUES AT MONITORING LOCATION(I,2,1H,,I2,
*6H) ---I/14H NO UMON,7X,'VMON',7X,'WMON',7X,'MASS',7X,
*'TKIN'
3,7X,4HDISP,9X,1HU,9X,1HV,10X,1HW,10X,1HP,10X,1HD/)
311 FORMAT(2X,I4,11E11.4)
312 FORMAT(1H0,59(2H- ))
402 FORMAT(///9X,1H1,5X,3HX/D,5X,10HS.S.COEFF.)
403 FORMAT(/5X,I5,2(1PE11.3))
410 FORMAT(/23H SWIRL CASE WITH LFS =,I3/
1 23H AND NSBR =,I3//
163H CORRESPONDS IF NSBR = 1 TO SWIRL GENERATOR WITH SWIRL NUMBER =
1,F10.3//37H OR IF NSBR = 0 TO SWIRL VANE ANGLE =,F10.3//,
11X,' COMPUTED INLET SWIRL NUMBER =',F10.4//,
11X,' COMPUTED INLET MEAN AXIAL VELOCITY =',F10.4//,
11X,' COMPUTED INLET MASS FLOW RATE =',F10.5/////
950 FORMAT(10X,6F10.5,/,10X,6F10.5,/,10X,4F10.5)
960 FORMAT(15X,' THE SOLN. IS NOT CONVERGING')
END
C-----
C
C
SUBROUTINE INIT
CA*****
C
CHAPTER 0 0 0 0 0 0 0 0 0 PRELIMINARIES 0 0 0 0 0 0 0
C
COMMON
1/UVEL/RESORU,NSWPU,URFU,DXEPU(48),DXPWU(48),SEWU(48)
1/VVEL/RESORV,NSWPV,URFV,DYNSV(24),DYPV(24),SNSV(24)
*/WVEL/ RESORW,NSWPW,URFW
1/PCOR/RESORM,NSWPP,URFP,DU(48,24),DV(48,24),IPREF,JPREF
*/VAR/U(48,24),V(48,24),W(48,24),P(48,24),PP(48,24),TE(48,24),
*ED(48,24),STFN(48,24),YSTLN(48,24),STVAL(24),USTAR(48,24),
*VSTAR(48,24),WSTAR(48,24),PSTAR(48,24),TESTAR(48,24),YSTLND(48,24)
#,VISTAR(48,24)
1/ALL/IT, JT, NI, NJ, NIM1, NJM1, GREAT, JMAX(48), JMAXP1(48)
1/GEOM/INDCOS,X(48),Y(24),DXEP(48),DXPW(48),DYNP(24),DYPS(24),
1 SNS(24),SEW(48),XU(48),YV(24),R(24),RV(24),
# WFN(24),WFS(24),WFE(48),WFW(48),RCV(24),XND(48),XUND(48),
#YND(24),YVND(24)
COMMON
1/FLUPR/URFVIS,VISCOS,DENSIT,PRANDT,DEN(48,24),VIS(48,24)
1/KASE T1/ UIN,TEIN,EDIN,FLOWIN,ALAMDA,
2 R:MALL,RLARGE,AL1,AL2,JSTEP,ISTEP,JSTP1,JSTM1,ISTP1,ISTM1
1/TURB/GEN(48,24),CD,CMU,C1,C2,CAPPA,ELOG,PRED,PRTE
1/WALLF/YP.USN(48),XPLUSW(24),TAUN(48),TAUW(24)

```

```

1/COEF/AP(48,24),AN(48,24),AS(48,24),AE(48,24),AW(48,24),SU(48,24), 0060100
1 SP(48,24) 0060200
1/PLOTT/NSTLN,NPLTLN,NPTS,YSPLT(10,48),XUDPLT(48),INPLOT 0060300
C 0060400
CHAPTER 1 1 1 1 1 CALCULATE GEOMETRICAL QUANTITIES 1 1 1 1 1 0060500
C 0060600
DO 100 J=1,NJ 0060700
R(J)=Y(J) 0060800
100 IF(INDCOS.EQ.1)R(J)=1.0 0060900
DXPW(1)=0.0 0061000
DXEP(NI)=0.0 0061100
DO 101 I=1,NIM1 0061200
DXEP(I)=X(I+1)-X(I) 0061300
101 DXPW(I+1)=DXEP(I) 0061400
DYPS(1)=0.0 0061500
DYNP(NJ)=0.0 0061600
DO 102 J=1,NJM1 0061700
DYNP(J)=Y(J+1)-Y(J) 0061800
102 DYPS(J+1)=DYNP(J) 0061900
SEW(1)=0.0 0062000
SEW(NI)=0.0 0062100
DO 103 I=2,NIM1 0062200
103 SEW(I)=0.5*(DXEP(I)+DXPW(I)) 0062300
SNS(1)=0.0 0062400
SNS(NJ)=0.0 0062500
DO 104 J=2,NJM1 0062600
104 SNS(J)=0.5*(DYNP(J)+DYPS(J)) 0062700
XU(1)=0.0 0062800
DO 105 I=2,NI 0062900
105 XU(I)=0.5*(X(I)+X(I-1)) 0063000
DXPWU(1)=0.0 0063100
DXPWU(2)=0.0 0063200
DXEPU(1)=0.0 0063300
DXEPU(NI)=0.0 0063400
DO 106 I=2,NIM1 0063500
106 DXPWU(I+1)=DXEPU(I) 0063600
SEWU(1)=0.0 0063700
DO 107 I=2,NI 0063800
107 SEWU(I)=X(I)-X(I-1) 0063900
C-----U-VELOCITIES WEIGHTING FACTORS 0064000
DO 111 I=2,NIM1 0064100
WFE(I)=SEWU(I+1)/(SEWU(I+1)+SEWU(I)) 0064200
IF(I.LE.2)GO TO 111 0064300
WFW(I)=SEWU(I-1)/(SEWU(I-1)+SEWU(I)) 0064400
111 CONTINUE 0064500
YV(1)=0.0 0064600
RV(1)=0.0 0064700
DO 108 J=2,NJ 0064800
108 RV(J)=0.5*(R(J)+R(J-1)) 0064900
YV(J)=0.5*(Y(J)+Y(J-1)) 0065000
RCV(1)=R(1) 0065100
RCV(NJ)=R(NJ) 0065200
DO 113 J=2,NJM1 0065300
113 RCV(J)=0.5*(RV(J+1)+RV(J)) 0065400
DYPSV(1)=0.0 0065500
DYPSV(2)=0.0 0065600
DYNPV(NJ)=0.0 0065700
DO 109 J=2,NJM1 0065800
109 DYNPV(J)=YV(J+1)-YV(J) 0065900
DYNPV(J)=YV(J+1)-YV(J) 0066000

```

109	DYPSV(J+1)=DYNPV(J)	0066100
	SNSV(1)=0.0	0066200
	DO 110 J=2,NJ	0066300
110	SNSV(J)=Y(J)-Y(J-1)	0066400
C-----	V-VELOCITIES WEIGHTING FACTORS	0066500
	DO 112 J=3,NJM1	0066600
	WFN(J)=SNSV(J+1)/(SNSV(J+1)+SNSV(J))	0066700
	WFS(J)=SNSV(J-1)/(SNSV(J-1)+SNSV(J))	0066800
112	CONTINUE	0066900
C		0067000
CHAPTER	2 2 2 2 2 2 SET VARIABLES TO ZERO 2 2 2 2 2 2	0067100
C		0067200
	DO 200 I=1,NI	0067300
	TAUN(I)=1.0	0067400
	DO 200 J=1,NJ	0067500
	TAUW(J)=1.0	0067600
	U(I,J)=0.0	0067700
	V(I,J)=0.0	0067800
	W(I,J)=0.	0067900
	P(I,J)=0.0	0068000
	PP(I,J)=0.0	0068100
	TE(I,J)=0.0	0068200
	ED(I,J)=0.0	0068300
	DEN(I,J)=DENSIT	0068400
	VIS(I,J)=VISCOS	0068500
	DU(I,J)=0.0	0068600
	DV(I,J)=0.0	0068700
	SU(I,J)=0.0	0068800
	SP(I,J)=0.0	0068900
	STFN(I,J)=0.	0069000
200	CONTINUE	0069100
	DO 300 I=1,NI	0069200
	DO 300 J=1,NSTLN	0069300
	YSTLN(I,J)=0.	0069400
	YSTLND(I,J)=0.	0069500
	STVAL(J)=0.	0069600
300	CONTINUE	0069700
	DO 400 N=1,NPLTLN	0069800
	DO 400 I=1,NI	0069900
	YSLPLT(N,I)=0.	0070000
400	CONTINUE	0070100
	RETURN	0070200
	END	0070300
C		0070400
C-----		0070500
C		0070600
C		0070700
	SUBROUTINE PROPS	0070800
CA*****		0070900
C		0071000
CHAPTER	0 0 0 0 0 0 0 0 PRELIMINARIES 0 0 0 0 0 0 0	0071100
C		0071200
	COMMON	0071300
	1/FLUPR/URFVIS,VISCOS,DENSIT,PRANDT,DEN(48,24),VIS(48,24)	0071400
	*/VAR/U(48,24), V(48,24), W(48,24), P(48,24), PP(48,24), TE(48,24),	0071500
	*ED(48,24),STFN(48,24),YSTLN(48,24),STVAL(24),USTAR(48,24),	0071600
	*VSTAR(48,24),WSTAR(48,24),PSTAR(48,24),TESTAR(48,24),YSTLND(48,24)	0071700
	*,VISTAR(48,24)	0071800
	1/ALL/IT,JT,NI,NJ,NIM1,NJM1,GREAT,JMAX(48),JMAXP1(48)	0071900
	1/TURB/GEN(48,24),CD,CMU,C1,C2,CAPPA,ELOG,PRED,PRTE	0072000
C		

```

CHAPTER 1 1 1 VISCOSITY 1 1 1 0072100
C DO 100 I=2,NIMI 0072200
DO 100 J=2,NJMI 0072300
VISOLD=VIS(I,J) 0072400
IF(ED(I,J).EQ.0.) GO TO 102 0072500
VIS(I,J)=DEN(I,J)*TE(I,J)**2*CMU/ED(I,J)-VISCOS 0072600
GO TO 101 0072700
102 VIS(I,J)=VISCOS 0072800
C-----UNDER-RELAX VISCOSITY 0072900
101 VIS(I,J)=URFVIS*VIS(I,J)+(1.-URFVIS)*VISOLD 0073000
100 CONTINUE 0073100
RETURN 0073200
END 0073300
C 0073400
C----- 0073500
C 0073600
C SUBROUTINE CALCU 0073700
CA***** 0073800
C 0073900
CHAPTER 0 0 0 0 0 0 0 0 PRELIMINARIES 0 0 0 0 0 0 0 0 0074000
C 0074100
COMMON 0074200
1/VEL/RESORU,NSWPU,URFU,DXEPU(48),DXPWU(48),SEWU(48) 0074300
1/VVEL/RESORV,NSWPV,URFV,DYNPV(24),DYPV(24),SNSV(24) 0074400
1/PCOR/RESORM,NSWPP,URFP,DU(48,24),DV(48,24),IPREF,JPREF 0074500
*/VAR/U(48,24),V(48,24),W(48,24),P(48,24),PP(48,24),TE(48,24), 0074600
*ED(48,24),STFN(48,24),YSTLN(48,24),STVAL(24),USTAR(48,24), 0074700
*VSTAR(48,24),WSTAR(48,24),PSTAR(48,24),TESTAR(48,24),YSTLND(48,24) 0074800
#,VSTAR(48,24) 0074900
1/ALL/IT, JT, NI, NIMI, NJMI, GREAT, JMAX(48), JMAXP1(48) 0075000
1/GEOM/INDCOS, X(8), Y(24), DXEP(48), DXPW(48), DYNP(24), DYP(24), 0075100
1 SNS(24), SEW(48), XU(48), YV(24), R(24), RV(24), 0075200
# WFN(24), WFS(24), WFE(48), WFW(48), RCV(24), XND(48), XUND(48), 0075300
#YND(24), YUND(24) 0075400
COMMON 0075500
1/FLUPR/URFVIS, VISCOS, DENSIT, PRANDT, DEN(48,24), VIS(48,24) 0075600
1/COEF/AP(48,24), AN(48,24), AS(48,24), AE(48,24), AW(48,24), SU(48,24), 0075700
1 SP(48,24) 0075800
1/KASE T1/UIIN, TEIN, EDIN, FLOWIN, ALAMDA, 0075900
2 RSMALL, RLARGE, AL1, AL2, JSTEP, ISTEP, JSTP1, JSTM1, ISTP1, ISTM1 0076000
C 0076100
CHAPTER 1 1 1 1 1 1 ASSEMBLY OF COEFFICIENTS 1 1 1 1 1 1 1 1 0076200
C 0076300
DO 100 I=3,NIMI 0076400
DO 101 J=2,NJMI 0076500
C-----COMPUTE AREAS AND VOLUME 0076600
AREAN=RV(J+1)*SEWU(I) 0076700
AREAS=RV(J)*SEWU(I) 0076800
AREAEW=RCV(J)*SNS(J) 0076900
VOL=RCV(J)*SEWU(I)*SNS(J) 0077000
C-----CALCULATE CONVECTION COEFFICIENTS 0077100
GN=0.5*(DEN(I,J+1)+DEN(I,J))*V(I,J+1) 0077200
GNW=0.5*(DEN(I-1,J)+DEN(I-1,J+1))*V(I-1,J+1) 0077300
GS=0.5*(DEN(I,J-1)+DEN(I,J))*V(I,J) 0077400
GSW=0.5*(DEN(I-1,J)+DEN(I-1,J-1))*V(I-1,J) 0077500
GE=DEN(I,J)*(U(I+1,J)*(1.0-WFE(I))+U(I,J)*WFE(I)) 0077600
GW=DEN(I-1,J)*(U(I-1,J)*(1.0-WFW(I))+U(I,J)*WFW(I)) 0077700
CN=0.5*(GN+GNW)*AREAN 0077800
CS=0.5*(GS+GSW)*AREAS 0077900
0078000

```

```

CE=GE*AREA EW                                0078100
CW=GW*AREA EW                                0078200
C-----CALCULATE DIFFUSION COEFFICIENTS      0078300
VISN=0.25*(VIS(I,J)+VIS(I,J+1)+VIS(I-1,J)+VIS(I-1,J+1)) 0078400
VISS=0.25*(VIS(I,J)+VIS(I,J-1)+VIS(I-1,J)+VIS(I-1,J-1)) 0078500
DN=VISN*AREAN/DYNP(J)                        0078600
DS=VISS*AREAS/DYPS(J)                       0078700
DE=VIS(I,J)*AREA EW/DXEPW(I)                 0078800
DM=VIS(I-1,J)*AREA EW/DXPWU(I)               0078900
C-----CALCULATE COEFFICIENTS OF SOURCE TERMS 0079000
SMP=CN-CS+CE-CW                              0079100
CP=AMAX1(0.0,SMP)                            0079200
CPO=CP                                        0079300
C-----ASSEMBLE MAIN COEFFICIENTS            0079400
AN(I,J)=AMAX1(ABS(0.5*CN),DN)-0.5*CN         0079500
AS(I,J)=AMAX1(ABS(0.5*CS),DS)+0.5*CS         0079600
DE=AMAX1(DE,-WFE(I)*CE,(1.0-WFE(I))*CE)     0079700
DW=AMAX1(DW,WFW(I)*CW,-(1.0-WFW(I))*CW)     0079800
AE(I,J)=DE-(1.0-WFE(I))*CE                   0079900
AW(I,J)=DW+(1.-WFW(I))*CW                    0080000
DU(I,J)=AREA EW                              0080100
DUDXE=(U(I+1,J)-U(I,J))/DXEPW(I)            0080200
DUDXW=(U(I,J)-U(I-1,J))/DXPWU(I)            0080300
SORCE1=(DUDXE*VIS(I,J)-DUDXW*VIS(I-1,J))/SEWU(I) 0080400
DVDXN=(V(I,J+1)-V(I-1,J+1))/SEWU(I)         0080500
DVDXS=(V(I,J)-V(I-1,J))/SEWU(I)            0080600
SORCE2=(RV(J+1)*VISN*DVDXN-RV(J)*VISS*DVDXS)/(RCV(J)*DYNPV(J)) 0080700
SU(I,J)=CPO*U(I,J)+DU(I,J)*(P(I-1,J)-P(I,J)) 0080800
SP(I,J)=SU(I,J)+(SORCE1+SORCE2)*VOL         0080900
101 CONTINUE                                 0081000
100 CONTINUE                                 0081100
C                                             0081200
CHAPTER 2 2 2 2 2 2 2 PROBLEM MODIFICATIONS 2 2 2 2 2 2 2 0081300
C                                             0081400
CALL PROMOD (2)                             0081500
C                                             0081600
CHAPTER 3 FINAL COEFF. ASSEMBLY AND RESIDUAL SOURCE CALCULATION 3 3 0081700
C                                             0081800
RESORU=0.0                                  0081900
DO 300 I=3,NIM1                             0082000
DO 301 J=2,NJMI                             0082100
AP(I,J)=AN(I,J)+AS(I,J)+AE(I,J)+AW(I,J)-SP(I,J) 0082200
DU(I,J)=DU(I,J)/AP(I,J)                     0082300
RESOR=AN(I,J)*U(I,J+1)+AS(I,J)*U(I,J-1)+AE(I,J)*U(I+1,J) 0082400
      +AW(I,J)*U(I-1,J)-AP(I,J)*U(I,J)+SU(I,J) 0082500
VOL=RCV(J)*SEWU(I)*SNS(J)                   0082600
SORVOL=GREAT*VOL                            0082700
IF(-SP(I,J).GT.0.5*SORVOL) RESOR=RESOR/SORVOL 0082800
RESORII=RESORU+ABS(RESOR)                   0082900
C-----UNDER-RELAXATION                     0083000
AP(I,J)=AP(I,J)/URFU                         0083100
SU(I,J)=SU(I,J)+(1.-URFU)*AP(I,J)*U(I,J)    0083200
DU(I,J)=DU(I,J)*URFU                         0083300
301 CONTINUE                                 0083400
300 CONTINUE                                 0083500
CHAPTER 4 4 4 SOLUTION OF DIFFERENCE EQUATION 4 4 4 4 4 4 4 0083600
C                                             0083700
DO 400 N=1,NSWPU                             0083800
                                             0083900
                                             0084000

```

```

400 CALL LISOLV(3,2,NI,JMAX,IT,JT,U,2)
RETURN
END
C
C-----
C
SUBROUTINE CALCV
CA*****
C
CHAPTER 0 0 0 0 0 0 0 0 0 PRELIMINARIES 0 0 0 0 0 0 0 0
C
COMMON
1/UVEL/RESORU,NSWPU,URFU,DXEPU(48),DXPWU(48),SEWU(48)
1/VVEL/RESORV,NSWPV,URFV,DYNPV(24),DYPSV(24),SNSV(24)
1/PCOR/RESORN,NSWPP,URFP,DU(48,24),DV(48,24),IPREF,JPREF
*/VAR/U(48,24),V(48,24),W(48,24),P(48,24),PP(48,24),TE(48,24),
*ED(48,24),STFN(48,24),YSTLN(48,24),STVAL(24),USTAR(48,24),
*VSTAR(48,24),WSTAR(48,24),PSTAR(48,24),TESTAR(48,24),YSTLND(48,24)
#VISTAR(48,24)
1/ALL/IT,JT,NI,NJ,NJM1,NJM1,GREAT,JMAX(48),JMAXP1(48)
1/GEOM/INDCOS,X(48),Y(24),DXEP(48),DXPW(48),DYNP(24),DYPS(24),
1 SNS(24),SEW(48),XU(48),YV(24),R(24),RV(24),
# WFN(24),WFS(24),WFE(48),WFW(48),RCV(24),XND(48),XUND(48),
#YND(24),YVND(24)
COMMON
1/FLUPR/URFVIS,VISCOS,DENSIT,PRANDT,DEN(48,24),VIS(48,24)
1/COEF/AP(48,24),AN(48,24),AS(48,24),AE(48,24),AW(48,24),SU(48,24),
1 SP(48,24)
1/KASE T1/UIN,TEIN,EDIN,FLOWIN,ALAMDA,
2 RSMALL,RLARGE,AL1,AL2,JSTEP,ISTEP,JSTP1,JSTM1,ISTP1,ISTM1
C
CHAPTER 1 1 1 1 1 1 ASSEMBLY OF COEFFICIENTS 1 1 1 1 1 1 1
C
DO 100 I=2,NIM1
DO 101 J=3,NJM1
C-----COMPUTE AREAS AND VOLUME
AREAN=R(J)*SEW(I)
AREAS=R(J-1)*SEW(I)
AREAEW=RV(J)*SNSV(J)
VOL=RV(J)*SEW(I)*SNSV(J)
C-----CALCULATE CONVECTION COEFFICIENTS
GN=DEN(I,J)*(V(I,J+1)*(1.0-WFN(J))+V(I,J)*WFN(J))
GS=DEN(I,J-1)*(V(I,J-1)*(1.0-WFS(J))+V(I,J)*WFS(J))
GE=0.5*(DEN(I+1,J)+DEN(I,J))*U(I+1,J)
GSE=0.5*(DEN(I,J-1)+DEN(I+1,J-1))*U(I+1,J-1)
GW=0.5*(DEN(I,J)+DEN(I-1,J))*U(I,J)
GSW=0.5*(DEN(I,J-1)+DEN(I-1,J-1))*U(I,J-1)
CN=GN*AREAN
CS=GS*AREAS
CE=0.5*(GE+GSE)*AREAEW
CW=0.5*(GW+GSW)*AREAEW
C-----CALCULATE DIFFUSION COEFFICIENTS
VISE=0.25*(VIS(I,J)+VIS(I+1,J)+VIS(I,J-1)+VIS(I+1,J-1))
VISW=0.25*(VIS(I,J)+VIS(I-1,J)+VIS(I,J-1)+VIS(I-1,J-1))
DN=VIS(I,J)*AREAN/DYNPV(J)
DS=VIS(I,J-1)*AREAS/DYPSV(J)
DE=VISE*AREAEW/DXEP(I)
DW=VISW*AREAEW/DXPW(I)
C-----CALCULATE COEFFICIENTS OF SOURCE TERMS
SMP=CN-CS+CE-CW

```

```

0084100
0084200
0084300
0084400
0084500
0084600
0084700
0084800
0084900
0085000
0085100
0085200
0085300
0085400
0085500
0085600
0085700
0085800
0085900
0086000
0086100
0086200
0086300
0086400
0086500
0086600
0086700
0086800
0086900
0087000
0087100
0087200
0087300
0087400
0087500
0087600
0087700
0087800
0087900
0088000
0088100
0088200
0088300
0088400
0088500
0088600
0088700
0088800
0088900
0089000
0089100
0089200
0089300
0089400
0089500
0089600
0089700
0089800
0089900
0090000

```



```

SUBROUTINE CALCX
CA*****
C
CHAPTER 0 0 0 0 0 0 0 0 PRELIMINARIES 0 0 0 0 0 0 0
C
COMMON
1/UVEL/RESORU,NSWPU,URFU,DXEPU(48),DXPWU(48),SEWU(48)
1/VVEL/RESORV,NSWPV,URFV,DYNPV(24),DYPSV(24),SNSV(24)
*/WVEL/RESORW,NSWPW,URFW
1/TEN/RESORK,NSWPK,URFK
1/TDIS/RESORE,NSWPD,URFE
*/VAR/U(48,24),V(48,24),W(48,24),P(48,24),PP(48,24),TE(48,24),
*ED(48,24),STFN(48,24),YSTLN(48,24),STVAL(24),USTAR(48,24),
*VSTAR(48,24),WSTAR(48,24),PSTAR(48,24),TESTAR(48,24),YSTLND(48,24)
#,VISTAR(48,24)
1/ALL/IT,JT,NI,NJ,NIM1,NJM1,GREAT,JMAX(48),JMAXP1(48)
1/GEOM/INDCOS,X(48),Y(24),DXEP(48),DXPW(48),DYNP(24),DYPS(24),
1 SNS(24),SEW(48),XU(48),YV(24),R(24),RV(24),
# WFN(24),WFS(24),WFE(48),WFW(48),RCV(24),XND(48),XUND(48),
#YND(24),YVND(24)
COMMON
1/FLUPR/URFVIS,VISCOS,DENSIT,PRANDT,DEN(48,24),VIS(48,24)
1/KASE T1/UIN,TEIN,EDIN,FLOWIN,ALAMDA,
2 RSMALL,RLARGE,AL1,AL2,JSTEP,ISTEP,JSTP1,JSTM1,ISTP1,ISTM1
1/TURB/GEN(48,24),CD,CMU,C1,C2,CAPPA,ELOG,PRED,PRTE
1/WALLF/YPLUSN(48),XPLUSW(24),TAUN(48),TAUW(24)
1/COEF/AP(48,24),AN(48,24),AS(48,24),AE(48,24),AW(48,24),SU(48,24),
1 SP(48,24)
C-----IF NO SWIRL, RETURN TO MAIN
IF(W(1,3) .LE. 0.) GO TO 500
C
CHAPTER 1 1 1 1 1 1 1 ASSEMBLY OF COEFFICIENTS 1 1 1 1 1 1
C
DO 100 I=2, NIM1
DO 101 J=2, NJM1
C
C COMPUTE AREAS AND VOLUME
C
AREAN=RV(J+1)*SEW(I)
AREAS=RV(J)*SEW(I)
AREAEW=RCV(J)*SNS(J)
VOL=RCV(J)*SNS(J)*SEW(I)
C
C-----CALCULATE CONVECTION COEFFICIENTS
C
GN=0.5*(DEN(I,J)+DEN(I,J+1))*V(I,J+1)
GS=0.5*(DEN(I,J)+DEN(I,J-1))*V(I,J)
GE=0.5*(DEN(I,J)+DEN(I+1,J))*U(I+1,J)
GW=0.5*(DEN(I,J)+DEN(I-1,J))*U(I,J)
CN=GN*AREAN
CS=GS*AREAS
CE=GE*AREAEW
CW=GW*AREAEW
C
C-----CALCULATE DIFFUSION COEFFICIENTS
C
VISN=0.5*(VIS(I,J)+VIS(I,J+1))
VISS=0.5*(VIS(I,J)+VIS(I,J-1))
VISE=0.5*(VIS(I,J)+VIS(I+1,J))
VISW=0.5*(VIS(I,J)+VIS(I-1,J))

```

```

0096100
0096200
0096300
0096400
0096500
0096600
0096700
0096800
0096900
0097000
0097100
0097200
0097300
0097400
0097500
0097600
0097700
0097800
0097900
0098000
0098100
0098200
0098300
0098400
0098500
0098600
0098700
0098800
0098900
0099000
0099100
0099200
0099300
0099400
0099500
0099600
0099700
0099800
0099900
0100000
0100100
0100200
0100300
0100400
0100500
0100600
0100700
0100800
0100900
0101000
0101100
0101200
0101300
0101400
0101500
0101600
0101700
0101800
0101900
0102000

```

DN=VISH*AREAN/DYNP(J)	0102100
DS=VISS*AREAS/DYPS(J)	0102200
DE=VISE*AREAEW/DXEP(I)	0102300
DW=VISW*AREAEW/DXPW(I)	0102400
C	0102500
C-----SOURCE TERMS	0102600
C	0102700
SMP=CN-CS+CE-CW	0102800
CP=AMAX1(0., SMP)	0102900
CPO=CP	0103000
C	0103100
C-----ASSEMBLE MAIN COEFFICIENTS	0103200
C	0103300
AN(I,J)=AMAX1(ABS(0.5*CN),DN)-0.5*CN	0103400
AS(I,J)=AMAX1(ABS(0.5*CS),DS)+0.5*CS	0103500
AE(I,J)=AMAX1(ABS(0.5*CE),DE)-0.5*CE	0103600
AW(I,J)=AMAX1(ABS(0.5*CW),DW)+0.5*CW	0103700
DV=0.5*(AREAN+AREAS)	0103800
VAVG=0.5*(V(I,J+1)+V(I,J))	0103900
SU(I,J)=CPO*W(I,J)	0104000
IF(INDCOS .EQ. 1) GO TO 101	0104100
SORCE1=-DEN(I,J)*VAVG*W(I,J) /RCV(J)	0104200
SORCE2=- (VISH*RV(J+1)-VISS*RV(J)) *W(I,J)/(DYNPV(J)*	0104300
*RCV(J)*RCV(J))	0104400
SU(I,J)=SU(I,J)+(SORCE1+SORCE2)*VOL	0104500
SP(I,J)=-CP	0104600
101 CONTINUE	0104700
100 CONTINUE	0104800
C	0104900
CHAPTER 2 2 2 2 2 2 PROBLEM MODIFICATIONS 2 2 2 2 2 2 2 2 2 2	0105000
C	0105100
CALL PROMOD (8)	0105200
C	0105300
CHAPTER 3 3 FINAL COEFFICIENT ASSEMBLY AND RESIDUAL SOURCE CALCULATION	0105400
C	0105500
RESORW=0.	0105600
DO 300 I=2, NIMI	0105700
DO 301 J=2, NJM1	0105800
AP(I,J)=AN(I,J)+AS(I,J)+AE(I,J)+AW(I,J)-SP(I,J)	0105900
RESOR=AN(I,J)*W(I,J+1)+AS(I,J)*W(I,J-1)+AE(I,J)*W(I+1,J)	0106000
* +AW(I,J)*W(I-1,J)-AP(I,J)*W(I,J)+SU(I,J)	0106100
VOL=RCV(J)*SNS(J)*SEW(I)	0106200
SORVOL=GREAT*VOL	0106300
IF(-SP(I,J) .GT. 0.5*SORVOL) RESOR=RESOR/SORVOL	0106400
IF(J .LE. 2) RESOR=0.	0106500
RESORW=RESORW+ABS(RESOR)	0106600
C	0106700
C-----UNDER RELAXATION	0106800
C	0106900
AP(I,J)=AP(I,J)/URFW	0107000
SU(I,J)=SU(I,J)+(1.-URFW)*AP(I,J)*W(I,J)	0107100
301 CONTINUE	0107200
300 CONTINUE	0107300
C	0107400
CHAPTER 4 4 4 SOLUTION OF DIFFERENCE EQUATIONS 4 4 4 4 4 4 4 4	0107500
C	0107600
DO 400 N=1, NSWPW	0107700
400 CALL LISOLV(2, 2, NI, JMAX, IT, JT, W,8)	0107800
500 CONTINUE	0107900
RETURN	0108000

```

C          END                                                    0108100
C-----
C          SUBROUTINE CALCP                                       0108200
CA*****                                                    0108300
C          CHAPTER 0 0 0 0 0 0 0 0 PRELIMINARIES 0 0 0 0 0 0 0 0 0 0108400
C          COMMON                                                0108500
          1/PCOR/RESORM,NSWPP,URFP,DU(48,24),DV(48,24),IPREF,JPREF 0108600
          */VAR/U(48,24), V(48,24), W(48,24), P(48,24), PP(48,24), TE(48,24), 0108700
          *ED(48,24),STFN(48,24),YSTLN(48,24),STVAL(24),USTAR(48,24), 0108800
          *VSTAR(48,24),WSTAR(48,24),PSTAR(48,24),TESTAR(48,24),YSTLND(48,24) 0108900
          #.VISTAR(48,24) 0109000
          1/ALL/IT,IT,NJ,NIM1,NJM1,GREAT,JMAX(48),JMAXPI(48) 0109100
          1/GEOM/INDCOS,X(48),Y(24),DXEP(48),DXPW(48),DYNP(24),DYP5(24), 0109200
          1 SNS(24),SEW(48),XU(48),YV(24),R(24),RV(24), 0109300
          # WFN(24),WFS(24),WFE(48),WFW(48),RCV(24),XND(48),XUND(48), 0109400
          #YND(24),YVND(24) 0109500
          COMMON 0109600
          1/FLUPR/URFVIS,VISCOS,DENSIT,PRANDT,DEN(48,24),VIS(48,24) 0109700
          1/COEF/AP(48,24),AN(48,24),AS(48,24),AE(48,24),AW(48,24),SU(48,24), 0109800
          1 SP(48,24) 0109900
          1/KASE T1/UIN,TEIN,EDIN,FLOWIN,ALAMDA, 0110000
          2 RSMALL,RLARGE,AL1,AL2,JSTEP,ISTEP,JSTP1,JSTM1,ISTP1,ISTM1 0110100
          RESORM=0.0 0110200
C          CHAPTER 1 1 1 1 1 1 ASSEMBLY OF COEFFICIENTS 1 1 1 1 1 1 1 0110300
C          DO 100 I=2,NIM1 0110400
          DO 101 J=2,NJM1 0110500
C-----COMPUTE AREAS AND VOLUME 0110600
          AREAN=RV(J+1)*SEW(I) 0110700
          AREAS=RV(J)*SEW(I) 0110800
          AREAEW=RCV(J)*SNS(J) 0110900
          VOL=RCV(J)*SNS(J)*SEW(I) 0111000
C-----CALCULATE COEFFICIENTS 0111100
          DENN=0.5*(DEN(I,J)+DEN(I,J+1)) 0111200
          DENS=0.5*(DEN(I,J)+DEN(I,J-1)) 0111300
          DENE=0.5*(DEN(I,J)+DEN(I+1,J)) 0111400
          DENW=0.5*(DEN(I,J)+DEN(I-1,J)) 0111500
          AN(I,J)=DENN*AREAN*DV(I,J+1) 0111600
          AS(I,J)=DENS*AREAS*DV(I,J) 0111700
          AE(I,J)=DENE*AREAEW*DU(I+1,J) 0111800
          AW(I,J)=DENW*AREAEW*DU(I,J) 0111900
C-----CALCULATE SOURCE TERMS 0112000
          CN=DENN*V(I,J+1)*AREAN 0112100
          CS=DENS*V(I,J)*AREAS 0112200
          CE=DENE*U(I+1,J)*AREAEW 0112300
          CW=DENN*U(I,J)*AREAEW 0112400
          SMP=CN-CS+CE-CW 0112500
          SP(I,J)=0.0 0112600
          SU(I,J)=-SMP 0112700
C-----COMPUTE SUM OF ABSOLUTE MASS SOURCES 0112800
          RESORM=RESORM+ABS(SMP) 0112900
          101 CONTINUE 0113000
          100 CONTINUE 0113100
C          CHAPTER 2 2 2 2 2 2 2 2 PROBLEM MODIFICATIONS 2 2 2 2 2 2 2 2 0113200
          0113300
          0113400
          0113500
          0113600
          0113700
          0113800
          0113900
          0114000

```

```

C          CALL PROMOD (4)                                0114100
C                                                    0114200
C          CHAPTER 3 3 3 3 3 FINAL COEFFICIENT ASSEMBLY 3 3 3 3 3 3 3 0114300
C                                                    0114400
C          DO 300 I=2,NIM1                                0114500
C          DO 301 J=2,NJM1                                0114600
C          301 AP(I,J)=AN(I,J)+AS(I,J)+AE(I,J)*AN(I,J)-SP(I,J) 0114700
C          300 CONTINUE                                    0114800
C                                                    0114900
C          CHAPTER 4 4 4 4 4 SOLUTION OF DIFFERENCE EQUATIONS 4 4 4 4 4 0115000
C                                                    0115100
C          DO 400 N=1,NSWPP                                0115200
C          400 CALL LISOLV(2,2,NI,JMAX,IT,JT,PP,4)        0115300
C                                                    0115400
C          CHAPTER 5 5 5 5 CORRECT VELOCITIES AND PRESSURE 5 5 5 5 5 5 0115500
C                                                    0115600
C          C-----VELOCITIES                            0115700
C          DO 503 I=2,NIM1                                0115800
C          JJ=JMAX(I)                                     0115900
C          DO 501 J=3,JJ                                   0116000
C          V(I,J)=V(I,J)+DV(I,J)*(PP(I,J-1)-PP(I,J))    0116100
C          501 CONTINUE                                    0116200
C          JJ=JMAX(I-1)                                   0116300
C          DO 502 J=2,JJ                                   0116400
C          IF(I.NE. 2) U(I,J)=U(I,J)+DU(I,J)*(PP(I-1,J)-PP(I,J)) 0116500
C          502 CONTINUE                                    0116600
C          503 CONTINUE                                    0116700
C          C-----PRESSURES (WITH PROVISION FOR UNDER-RELAXATION) 0116800
C          PPREF=PP(IPREF,JPREF)                          0116900
C          DO 506 I=2,NIM1                                0117000
C          JJ=JMAX(I)                                     0117100
C          DO 508 J=2,JJ                                   0117200
C          P(I,J)=P(I,J)+URFP*(PP(I,J)-PPREF)           0117300
C          C-----PP IS ZEROED AT TOP OF CHAPTER 3, MAIN 0117400
C          508 CONTINUE                                    0117500
C          506 CONTINUE                                    0117600
C          RETURN                                          0117700
C          END                                             0117800
C                                                    0117900
C          C-----                                        0118000
C          C-----                                        0118100
C          C-----                                        0118200
C          SUBROUTINE CALCTE                               0118300
C          CA*****                                       0118400
C          CHAPTER 0 0 0 0 0 0 0 0 0 PRELIMINARIES 0 0 0 0 0 0 0 0 0 0118500
C          C-----                                        0118600
C          COMMON                                          0118700
C          1/TEN/RESORK,NSWPK,URFK                        0118800
C          */VAR/U(48,24), V(48,24), W(48,24), P(48,24), PP(48,24), TE(48,24), 0118900
C          *ED(48,24), STFN(48,24), YSTLN(48,24), STVAL(24), USTAR(48,24), 0119000
C          *VSTAR(48,24), WSTAR(48,24), PSTAR(48,24), TESTAR(48,24), YSTLND(48,24) 0119100
C          #,VISTAR(48,24)                                0119200
C          1/ALL/IT, JT, NI, NJ, NIM1, NJM1, GREAT, JMAX(48), JMAXP1(48) 0119300
C          1/GEOM/INDCOS, X(48), Y(24), DXEP(48), DXPW(48), DYNP(24), DYPS(24), 0119400
C          1 SNS(24), SEW(48), XU(48), YV(24), R(24), RV(24), 0119500
C          # WFN(24), WFS(24), WFE(48), WFW(48), RCV(24), XND(48), XUND(48), 0119600
C          #YND(24), YVND(24)                             0119700
C          COMMON                                          0119800
C          1/FLUPR/URFVIS, VISCOS, DENSIT, PRANDT, DEN(48,24), VIS(48,24) 0119900
C          C-----                                        0120000

```

```

1/COEF/AP(48,24),AN(48,24),AS(48,24),AE(48,24),AW(48,24),SU(48,24), 0120100
1 SP(48,24) 0120200
1/TURB/GEN(48,24),CD,CMU,C1,C2,CAPPA,ELOG,PRED,PRTE 0120300
1/WALLF/YPLUSN(48),XPLUSW(24),TAUN(48),TAUW(24) 0120400
1/KASE T1/UIN,TEIN,EDIN,FLOWIN,ALAMDA, 0120500
2 RSMALL,RLARGE,AL1,AL2,JSTEP,ISTEP,JSTP1,JSTM1,ISTP1,ISTM1 0120600
1/SUSP/SUKD(48,24),SFKD(48,24) 0120700
C 0120800
CHAPTER 1 1 1 1 1 1 ASSEMBLY OF COEFFICIENTS 1 1 1 1 1 1 0120900
C 0121000
PRTE=1.0 0121100
DO 100 I=2,NIM1 0121200
DO 101 J=2,NJM1 0121300
C-----COMPUTE AREAS AND VOLUME 0121400
AREAN=RV(J+1)*SEW(I) 0121500
AREAS=RV(J)*SEW(I) 0121600
AREAEW=RCV(J)*SNS(J) 0121700
VOL=RCV(J)*SNS(J)*SEW(I) 0121800
C-----CALCULATE CONVECTION COEFFICIENTS 0121900
GN=0.5*(DEN(I,J)+DEN(I,J+1))*V(I,J+1) 0122000
GS=0.5*(DEN(I,J)+DEN(I,J-1))*V(I,J) 0122100
GE=0.5*(DEN(I,J)+DEN(I+1,J))*U(I+1,J) 0122200
GW=0.5*(DEN(I,J)+DEN(I-1,J))*U(I,J) 0122300
CN=GN*AREAN 0122400
CS=GS*AREAS 0122500
CE=GE*AREAEW 0122600
CW=GW*AREAEW 0122700
C-----CALCULATE DIFFUSION COEFFICIENTS 0122800
GAMN=0.5*(VIS(I,J)+VIS(I,J+1))/PRTE 0122900
GAMS=0.5*(VIS(I,J)+VIS(I,J-1))/PRTE 0123000
GAME=0.5*(VIS(I,J)+VIS(I+1,J))/PRTE 0123100
GAMW=0.5*(VIS(I,J)+VIS(I-1,J))/PRTE 0123200
PN=GAMN*AREAN/DYNP(J) 0123300
DS=GAMS*AREAS/DYPS(J) 0123400
DE=GAME*AREAEW/DXEP(I) 0123500
DW=GAMW*AREAEW/DXPW(I) 0123600
C-----SOURCE TERMS 0123700
SMP=CN-CS+CE-CW 0123800
CP=AMAX1(0.0,SMP) 0123900
CPO=CP 0124000
DUDX=(U(I+1,J)-U(I,J))/SEW(I) 0124100
DV DY=(V(I,J+1)-V(I,J))/SNS(J) 0124200
DUDY=((U(I,J)+U(I+1,J)+U(I,J+1)+U(I+1,J+1))/4.-(U(I,J)+U(I+1,J)+
1U(I,J-1)+U(I+1,J-1))/4.)/SNS(J) 0124300
DVDX=((V(I,J)+V(I,J+1)+V(I+1,J)+V(I+1,J+1))/4.-(V(I,J)+V(I,J+1)+V(
1I-1,J)+V(I-1,J+1))/4.)/SEW(I) 0124400
DWDY=(W(I,J+1)-W(I,J-1))/(DYNP(J)+DYPS(J))-W(I,J)/R(J) 0124500
DWDX=(W(I+1,J)-W(I-1,J))/(DXPW(I)+DXEP(I)) 0124600
GEN(I,J)=(2.*(DUDX**2+DV DY**2)+(DUDY+DVDX)**2)*VIS(I,J) 0124700
IF(INDCOS .EQ. 2) GEN(I,J)=GEN(I,J)+VIS(I,J)*(DWDY**2+DWDX**2) 0124800
IF(RV(J) .EQ. 0.) GO TO 110 0124900
VDR=V(I,J)/RV(J) 0125000
IF(INDCOS .EQ. 2) GEN(I,J)=GEN(I,J)+VIS(I,J)*.5* 0125100
*(VDR+V(I,J+1)/RV(J+1))**2 0125200
GO TO 120 0125300
110 IF(INDCOS .EQ. 2) GEN(I,J)=GEN(I,J)+VIS(I,J)* 0125400
*.5*(V(I,J+1)/RV(J+1))**2 0125500
120 CONTINUE 0125600
C-----ASSEMBLE MAIN COEFFICIENTS 0125700
AN(I,J)=AMAX1(ABS(0.5*CN),DN)-0.5*CN 0125800
0125900
0126000

```

```

AS(I,J)=AMAX1(ABS(0.5*CS),DS)+0.5*CS      0126100
AE(I,J)=AMAX1(ABS(0.5*CE),DE)-0.5*CE     0126200
AW(I,J)=AMAX1(ABS(0.5*CW),DW)+0.5*CW     0126300
SU(I,J)=CPO*TE(I,J)                      0126400
SUKD(I,J)=SU(I,J)                        0126500
SU(I,J)=SU(I,J)+GEN(I,J)*VOL             0126600
SP(I,J)=-CP                               0126700
SPKD(I,J)=SP(I,J)                        0126800
SP(I,J)=SP(I,J)-CD*CMU*DEN(I,J)**2*TE(I,J)*VOL/VIS(I,J) 0126900
101 CONTINUE                              0127000
100 CONTINUE                              0127100
C                                           0127200
CHAPTER 2 2 2 2 2 2 PROBLEM MODIFICATIONS 2 2 2 2 2 2 0127300
C                                           0127400
      CALL PROMOD (6)                      0127500
C                                           0127600
CHAPTER 3 FINAL COEFFICIENT ASSEMBLY AND RESIDUAL SOURCE CALCULATION 3 0127700
C                                           0127800
      RESORK=0.0                          0127900
      DO 300 I=2,NIM1                     0128000
      DO 301 J=2,NJMI                     0128100
      AP(I,J)=AN(I,J)+AS(I,J)+AE(I,J)+AW(I,J)-SP(I,J) 0128200
      RESOR=AN(I,J)*TE(I,J+1)+AS(I,J)*TE(I,J-1)+AE(I,J)*TE(I+1,J) 0128300
      1 +AW(I,J)*TE(I-1,J)-AP(I,J)*TE(I,J)+SU(I,J) 0128400
      VOL=RCV(J)*SNS(J)*SEW(I)           0128500
      SORVOL=GREAT*VOL                   0128600
      IF(-SP(I,J).GT.0.5*SORVOL) RESOR=RESOR/SORVOL 0128700
      RESORK=RESORK+ABS(RESOR)           0128800
C-----UNDER-RELAXATION                 0128900
      AP(I,J)=AP(I,J)/URFK               0129000
      SU(I,J)=SU(I,J)+(1.-URFK)*AP(I,J)*TE(I,J) 0129100
      301 CONTINUE                       0129200
      300 CONTINUE                       0129300
C                                           0129400
CHAPTER 4 4 4 4 4 SOLUTION OF DIFFERENCE EQUATIONS 4 4 4 4 4 0129500
C                                           0129600
      DO 400 N=1,NSWPK                   0129700
      400 CALL LISOLV(2,2,NI,JMAX,IT,JT,TE,6) 0129800
      RETURN                              0129900
      END                                 0130000
C-----                                0130100
C-----                                0130200
C                                           0130300
      SUBROUTINE CALCED                   0130400
CA*****                                0130500
C                                           0130600
CHAPTER 0 0 0 0 0 0 0 0 0 PRELIMINARIES 0 0 0 0 0 0 0 0 0 0130700
C                                           0130800
      COMMON                              0130900
      1/TDIS/RESORE,NSWPK,URFE           0131000
      1/ALL/IT,JT,NI,NJ,NIM1,GREAT,JMAX(48),JMAXP1(48) 0131100
      1/GEOM/INDCOS,X(48),Y(24),DXEP(48),DXPW(48),DYNP(24),DYPS(24), 0131200
      1 SNS(24),SEW(48),XU(48),YV(24),R(24),RV(24), 0131300
      # WFN(24),WFS(24),WFE(48),WFW(48),RCV(24),XND(48),XUND(48), 0131400
      %YND(24),YVND(24)                  0131500
      1/FLUPR/URFVIS,VISCOS,DENSIT,PRANDT,DEN(48,24),VIS(48,24) 0131600
      1/COEF/AP(48,24),AN(48,24),AS(48,24),AE(48,24),AW(48,24),SU(48,24), 0131700
      1 SP(48,24)                        0131800
      %VAR/U(48,24),V(48,24),W(48,24),P(48,24),PP(48,24),TE(48,24), 0131900
      *ED(48,24),STFN(48,24),YSTLN(48,24),STVAL(24),USTAR(48,24), 0132000

```

```

*VSTAR(48,24),WSTAR(48,24),PSTAR(48,24),TESTAR(48,24),YSTLND(48,24) 0132100
#,VISTAR(48,24) 0132200
COMMON 0132300
1/TURB/GEN(48,24),CD,CMU,C1,C2,CAPPA,ELOG,PRED,PRTE 0132400
1/WALLF/YPLUSN(48),XPLUSW(24),TAUN(48),TAUN(24) 0132500
1/SUSP/SUKD(48,24),SPKD(48,24) 0132600
1/KASE T1/UIN,TEIN,EDIN,FLOWIN,ALAMDA, 0132700
2 RSMALL,RLARGE,AL1,AL2,JSTEP,ISTEP,JSTP1,JSTM1,ISTP1,ISTMI 0132800
C 0132900
CHAPTER 1 1 1 1 1 1 ASSEMBLY OF COEFFICIENTS 1 1 1 1 1 1 0133000
C 0133100
DO 100 I=2,NIM1 0133200
JJ=JMAX(I) 0133300
DO 101 J=2,JJ 0133400
C-----COMPUTE AREAS AND VOLUME 0133500
AREAN=RV(J+1)*SEW(I) 0133600
AREAS=RV(J)*SEW(I) 0133700
AREAEW=RCV(J)*SNS(J) 0133800
VOL=RCV(J)*SNS(J)*SEW(I) 0133900
C-----CALCULATE CONVECTION COEFFICIENTS 0134000
GN=0.5*(DEN(I,J)+DEN(I,J+1))*V(I,J+1) 0134100
GS=0.5*(DEN(I,J)+DEN(I,J-1))*V(I,J) 0134200
GE=0.5*(DEN(I,J)+DEN(I+1,J))*U(I+1,J) 0134300
GW=0.5*(DEN(I,J)+DEN(I-1,J))*U(I,J) 0134400
CN=GN*AREAN 0134500
CS=GS*AREAS 0134600
CE=GE*AREAEW 0134700
CW=GW*AREAEW 0134800
C-----CALCULATE DIFFUSION COEFFICIENTS 0134900
GAMN=0.5*(VIS(I,J)+VIS(I,J-1))/PRED 0135000
GAMS=0.5*(VIS(I,J)+VIS(I,J+1))/PRED 0135100
GAME=0.5*(VIS(I,J)+VIS(I+1,J))/PRED 0135200
GAMW=0.5*(VIS(I,J)+VIS(I-1,J))/PRED 0135300
DN=GAMN*AREAN/DYNP(J) 0135400
DS=GAMS*AREAS/DYPS(J) 0135500
DE=GAME*AREAEW/DXEP(I) 0135600
DW=GAMW*AREAEW/DXPW(I) 0135700
C-----SOURCE TERMS 0135800
SMP=CN-CS+CE-CW 0135900
CP=AMAX1(0.0,SMP) 0136000
CPO=CP 0136100
C-----ASSEMBLE MAIN COEFFICIENTS 0136200
AN(I,J)=AMAX1(ABS(0.5*CN),DN)-0.5*CN 0136300
AS(I,J)=AMAX1(ABS(0.5*CS),DS)+0.5*CS 0136400
AE(I,J)=AMAX1(ABS(0.5*CE),DE)-0.5*CE 0136500
AW(I,J)=AMAX1(ABS(0.5*CW),DW)+0.5*CW 0136600
SU(I,J)=CPO*ED(I,J) 0136700
SUKD(I,J)=SU(I,J) 0136800
SU(I,J)=SU(I,J)+C1*CMU*GEN(I,J)*VOL*DEN(I,J)*TE(I,J)/VIS(I,J) 0136900
SP(I,J)=-CP 0137000
SPKD(I,J)=SP(I,J) 0137100
SP(I,J)=SP(I,J)-C2*DEN(I,J)*ED(I,J)*VOL/TE(I,J) 0137200
101 CONTINUE 0137300
100 CONTINUE 0137400
C 0137500
CHAPTER 2 2 2 2 2 2 PROBLEM MODIFICATIONS 2 2 2 2 2 2 0137600
C 0137700
CALL PROMOD (7) 0137800
C 0137900
CHAPTER 3 FINAL COEFFICIENT ASSEMBLY AND RESIDUAL SOURCE CALCULATION 3 0138000

```

C		0138100
	RESORE=0.0	0138200
	DO 300 I=2,NIMI	0138300
	DO 301 J=2,NJMI	0138400
	AP(I,J)=AN(I,J)+AS(I,J)+AE(I,J)+AW(I,J)-SP(I,J)	0138500
	RESOR=AN(I,J)*ED(I,J+1)+AS(I,J)*ED(I,J-1)+AE(I,J)*ED(I+1,J)	0138600
	+AW(I,J)*ED(I-1,J)-AP(I,J)*ED(I,J)+SU(I,J)	0138700
	VOL=RCV(J)*SNS(J)*SEW(I)	0138800
	SORVOL=GREAT*VOL	0138900
	IF(-SP(I,J).GT.0.5*SORVOL) RESOR=RESOR/SORVOL	0139000
	RESORE=RESORE+ABS(RESOR)	0139100
C-----	UNDER-RELAXATION	0139200
	AP(I,J)=AP(I,J)/URFE	0139300
	SU(I,J)=SU(I,J)+(1.-URFE)*AP(I,J)*ED(I,J)	0139400
	301 CONTINUE	0139500
	300 CONTINUE	0139600
C		0139700
CHAPTER	4 4 4 4 4 SOLUTION OF DIFFERENCE EQUATIONS 4 4 4 4 4	0139800
C		0139900
	DO 400 N=1,NSWPD	0140000
	400 CALL LISOLV(2,2,NI,JMAX,IT,JT,ED,7)	0140100
	RETURN	0140200
	END	0140300
C		0140400
C-----		0140500
C		0140600
	SUBROUTINE LISOLV(ISTART,JSTART,NI,JMAX,IT,JT,PHI,NCHAP)	0140700
CA*****		0140800
C		0140900
CHAPTER	0 0 0 0 0 0 0 0 0 PRELIMINARIES 0 0 0 0 0 0 0	0141000
C		0141100
	DIMENSION PHI(IT,JT),A(48),B(48),C(48),D(48),JMAX(IT)	0141200
	COMMON	0141300
	1/COEF/AP(48,24),AN(48,24),AS(48,24),AE(48,24),AW(48,24),SU(48,24),	0141400
	1 SP(48,24)	0141500
	1/KASE T1/UITN,TEIN,EDIN,FLOWIN,ALAMDA,	0141600
	2 RSMALL,RLARGE,AL1,AL2,JSTEP,ISTEP,JSTP1,JSTM1,ISTP1,ISTM1	0141700
	JSM1=JSTART-1	0141800
	NIMI=NI-1	0141900
	A(JSM1)=0.0	0142000
C-----	COMMENCE W-E SWEEP	0142100
	DO 100 I=ISTART,NIMI	0142200
	C(JSM1)=PHI(I,JSM1)	0142300
C-----	COMMENCE S-N TRAVERSE	0142400
	JJ=JMAX(I)	0142500
	IF(NCHAP.EQ.2) JJ=JMAX(I-1)	0142600
	DO 101 J=JSTART,JJ	0142700
C-----	ASSEMBLE TDMA COEFFICIENTS	0142800
	A(J)=AN(I,J)	0142900
	B(J)=AS(I,J)	0143000
	C(J)=AE(I,J)*PHI(I+1,J)+AW(I,J)*PHI(I-1,J)+SU(I,J)	0143100
	D(J)=AP(I,J)	0143200
C-----	CALCULATE COEFFICIENTS OF RECURRENCE FORMULA	0143300
	TERM=1./(D(J)-B(J)*A(J-1))	0143400
	A(J)=A(J)*TERM	0143500
	C(J)=(C(J)+B(J)*C(J-1))*TERM	0143600
	101 CONTINUE	0143700
C-----	OBTAIN NEW PHI'S	0143800
	DO 102 JJ=JSTART,JJ	0143900
	J=JJ+1+JSM1-JJ	0144000

102 PHI(I,J)=A(J)*PHI(I,J+1)+C(J)	0144100
100 CONTINUE	0144200
RETURN	0144300
END	0144400
C-----	0144500
C	0144600
C	0144700
SUBROUTINE PRINT(ISTART,JSTART,NI,NJ,IT,JT,X,Y,PHI,HEAD)	0144800
CA*****	0144900
C	0145000
DIMENSION PHI(IT,JT),X(IT),Y(JT),HEAD(9),STORE(48)	0145100
ISKIP=1	0145200
JSKIP=1	0145300
WRITE(6,110)HEAD	0145400
ISTA=ISTART-12	0145500
100 CONTINUE	0145600
ISTA=ISTA+12	0145700
IEND=ISTA+11	0145800
IF(NI.LT.IEND)IEND=NI	0145900
WRITE(6,111)(I,I=ISTA,IEND,ISKIP)	0146000
WRITE(6,114)(X(I),I=ISTA,IEND,ISKIP)	0146100
WRITE(6,112)	0146200
DO 101 JJ=JSTART,NJ,JSKIP	0146300
J=JSTART+NJ-JJ	0146400
DO 120 I=ISTA,IEND	0146500
A=PHI(I,J)	0146600
IF(ABS(A).LT.1.E-20) A=0.0	0146700
STORE(I)=A	0146800
120 101 WRITE(6,113)J,Y(J),(STORE(I),I=ISTA,IEND,ISKIP)	0146900
IF(IEND.LT.NI)GO TO 100	0147000
RETURN	0147100
110 FORMAT(1H0,17(2HX-),7X,9A4,7X,17(2H*))	0147200
111 FORMAT(1H0,13H I = ,I2,11T9)	0147300
112 FORMAT(8H0 J Y)	0147400
113 FORMAT(I3,0PF8.5,1X,1P12E9.2)	0147500
114 FORMAT(11H X = ,F8.5,11F9.5)	0147600
END	0147700
C-----	0147800
C	0147900
C	0148000
SUBROUTINE PROMOD (NCHAP)	0148100
CA*****	0148200
C	0148300
CHAPTER 0 0 0 0 0 0 0 0 PRELIMINARIES 0 0 0 0 0 0 0	0148400
C	0148500
COMMON	0148600
1/UVEL/RESORU,NSWPU,URFU,DXEPU(48),DXPWU(48),SEWU(48)	0148700
1/VVEL/RESORV,NSWPV,URFV,DYNPV(24),DYPSV(24),SNSV(24)	0148800
*WVEL/RESORW,NSWPW,URFW	0148900
*VAR/U(48,24),V(48,24),W(48,24),P(48,24),PP(48,24),TE(48,24),	0149000
*ED(48,24),STFN(48,24),YSTLN(48,24),STVAL(24),USTAR(48,24),	0149100
*VSTAR(48,24),WSTAR(48,24),PSTAR(48,24),TESTAR(48,24),YSTLND(48,24)	0149200
*,VISTAR(48,24)	0149300
1/PCOR/RESORM,NSWPP,URFP,DU(48,24),DV(48,24),IPREF,JPREF	0149400
1/ALL/IT,JT,NI,NJ,NIM1,NJM1,GREAT,JMAX(48),JMAXP1(48)	0149500
1/GEOM/INDCOS,X(48),Y(24),DXEP(48),DXPW(48),DYNP(24),DYPS(24),	0149600
1 SNS(24),SEW(48),XU(48),YV(24),R(24),RV(24),	0149700
*WFN(24),WFS(24),WFE(48),WFW(48),RCV(24),XND(48),XUND(48),	0149800
*YND(24),YVND(24)	0149900
COMMON	0150000

```

1/FLUPR/URFVIS,VISCOS,DENSIT,PRANDT,DEN(48,24),VIS(48,24) 0150100
1/KASE T1/UIN,TEIN,EDIN,FLOWIN,ALAMDA, 0150200
2 RSMALL,RLARGE,AL1,AL2,JSTEP,ISTEP,JSTP1,JSTM1,ISTP1,ISTM1 0150300
1/SUSP/SUKD(48,24),SPKD(48,24) 0150400
1/COEF/AP(48,24),AN(48,24),AS(48,24),AE(48,24),AW(48,24),SU(48,24), 0150500
1 SP(48,24) 0150600
1/TURB/GEN(48,24),CD,CMU,C1,C2,CAPPA,ELOG,PRED,PRTE 0150700
1/WALLF/YPLUSN(48),XPLUSW(24),TAUN(48),TAUW(24) 0150800
IF(NCHAP .EQ. 2) GO TO 1150 0150900
IF(JSTEP .EQ. NJM1) GO TO 1150 0151000
C-----OUT OF RANGE VALUES 0151100
DO 1100 I=2,NI 0151200
IF(JMAX(I) .EQ. NJM1) GO TO 1150 0151300
JJ=JMAXPI(I) 0151400
DO 1100 J=JJ,NJM1 0151500
1100 SP(I,J)=-GREAT 0151600
1150 CONTINUE 0151700
GO TO (1,2,3,4,5,6,7,8),NCHAP 0151800
C 0151900
C 0152000
C 0152100
CHAPTER 1 1 1 1 1 1 1 1 1 PROPERTIES 1 1 1 1 1 1 1 1 1 0152200
C 0152300
1 CONTINUE 0152400
C-----NO MODIFICATIONS FOR THIS PROBLEM 0152500
RETURN 0152600
C 0152700
CHAPTER 2 2 2 2 2 2 2 2 2 U MOMENTUM 2 2 2 2 2 2 2 2 2 0152800
C 0152900
2 CONTINUE 0153000
C-----OUT OF RANGE VALUES 0153100
IF(JSTEP .EQ. NJM1) GO TO 202 0153200
DO 200 I=3,NI 0153300
IF(JMAX(I-1) .EQ. NJM1) GO TO 202 0153400
JJ=JMAXPI(I-1) 0153500
DO 200 J=JJ,NJM1 0153600
SP(I,J)=-GREAT 0153700
200 CONTINUE 0153800
202 CONTINUE 0153900
C-----TOP WALL 0154000
CDTERM=CMU**0.25 0154100
DO 210 I=3,NIM1 0154200
J=JMAX(I-1) 0154300
YP=YV(J+1)-Y(J) 0154400
SQRTK=SQRT(0.5*(TE(I,J)+TE(I-1,J))) 0154500
DENU=0.5*(DEN(I,J)+DEN(I-1,J)) 0154600
YPLUSA=0.5*(YPLUSN(I)+YPLUSN(I-1)) 0154700
IF(YPLUSA.LE.11.63) GO TO 211 0154800
TMULT=DENU*CDTERM*SQRTK*CAPPA/ALOG(ELOG*YPLUSA) 0154900
GO TO 212 0155000
211 TMULT=VISCOS/YP 0155100
212 CONTINUE 0155200
205 SP(I,J)=SP(I,J)-TMULT*SEWU(I)*RV(J+1) 0155300
IF(JMAX(I-1) .NE. JMAX(I)) SP(I,J)=SP(I,J)/2. 0155400
210 AN(I,J)=0. 0155500
C-----SIDE WALL 0155600
IF(JSTEP .EQ. NJM1) GO TO 214 0155700
DO 225 I=3,NIM1 0155800
IF(JMAX(I-2) .GE. JMAX(I-1)) GO TO 225 0155900
JJ=JMAXPI(I-2) 0156000

```

JI=JMAX(I-1)	0156100
DO 220 J=JJ,JI	0156200
AW(I,J)=0.	0156300
220 CONTINUE	0156400
225 CONTINUE	0156500
C-----SYMMETRY AXIS	0156600
214 CONTINUE	0156700
DO 203 I=1,NI	0156800
203 AS(I,2)=0.	0156900
C-----OUTLET	0157000
ARDENT=0.0	0157100
FLOW=0.0	0157200
DO 209 J=2,NJM1	0157300
ARDEN=0.50*(DEN(NIM1,J)+DEN(NIM1-1,J))*RCV(J)*SNS(J)	0157400
ARDENT=ARDENT+ARDEN	0157500
209 FLOW=FLOW+ARDEN*U(NIM1,J)	0157600
UINC=(FLOWIN-FLOW)/ARDENT	0157700
DO 215 J=2,NJM1	0157800
215 U(NI,J)=U(NIM1,J)+UINC	0157900
RETURN	0158000
C	0158100
CHAPTER 3 3 3 3 3 3 3 3 V MOMENTUM 3 3 3 3 3 3 3 3	0158200
C	0158300
3 CONTINUE	0158400
C-----SIDE WALL	0158500
IF(JSTEP .EQ. NJM1) GO TO 314	0158600
CDTERM=CMU**0.25	0158700
DO 325 I=2,NIM1	0158800
IF(JMAX(I-1) .GE. JMAX(I)) GO TO 325	0158900
JJ=JMAX(I-1)	0159000
JI=JMAX(I)	0159100
DO 320 J=JJ,JI	0159200
XP=X(I)-XU(I)	0159300
SQRTK=SQRT(0.5*(TE(I,J)+TE(I,J-1)))	0159400
DENV=0.5*(DEN(I,J)+DEN(I,J-1))	0159500
XPLUSA=0.5*(XPLUSW(J)+XPLUSW(J-1))	0159600
IF(XPLUSA .LE. 11.63) GO TO 311	0159700
TMULT=DENV*CDTERM*SQRTK*CAPPA/ALOG(ELOG*XPLUSA)	0159800
GO TO 312	0159900
311 TMULT=VISCOS/XP	0160000
312 CONTINUE	0160100
305 SP(I,J)=SP(I,J)-TMULT*SNSV(J)*RV(J)	0160200
IF(J .EQ. JMAX(I-1)) SP(I,J)=SP(I,J)/2.	0160300
310 AW(I,J)=0.0	0160400
320 CONTINUE	0160500
325 CONTINUE	0160600
C-----TOP WALL	0160700
314 CONTINUE	0160800
DO 313 I=2,NIM1	0160900
J=JMAX(I)	0161000
313 AN(I,J)=0.	0161100
RETURN	0161200
C	0161300
CHAPTER 4 4 4 4 4 4 PRESSURE CORRECTION 4 4 4 4 4 4 4 4	0161400
C	0161500
4 CONTINUE	0161600
C-----SIDE WALL	0161700
IF(JSTEP .EQ. NJM1) GO TO 414	0161800
DO 412 I=2,NIM1	0161900
IF(JMAX(I-1) .GE. JMAX(I)) GO TO 412	0162000

```

        JJ=JMAXP1(I-1)
        JI=JMAX(I)
        DO 410 J=JJ,JI
        AW(I,J)=0.
410 CONTINUE
412 CONTINUE
C-----TOP WALL
414 CONTINUE
    DO 402 I=2,NIM1
    J=JMAX(I)
402 AN(I,J)=0.0
C-----SYMMETRY AXIS
    DO 420 I=2,NIM1
    AS(I,2)=0.0
420 CONTINUE
C-----OUTLET
    DO 440 J=2,NJM1
    AE(NIM1,J)=0.0
440 CONTINUE
    RETURN
C
CHAPTER 5 5 5 5 5 5 5 THERMAL ENERGY 5 5 5 5 5 5 5 5
C
5 CONTINUE
C-----NO MODIFICATIONS FOR THIS PROBLEM
    RETURN
C
CHAPTER 6 6 6 6 6 TURBULENT KINETIC ENERGY 6 6 6 6 6 *6 6 6
C
6 CONTINUE
C-----TOP WALL
    CDTERM=CMU**0.25
    DO 610 I=2,NIM1
    J=JMAX(I)
    DWDY=(W(I,J+1)-W(I,J-1))/(DYNP(J)+DYPS(J))
    UAVG=U(I,J)*WFE(I)+(1.-WFE(I))*U(I+1,J)
    UEFF=SQRT(UAVG*UAVG + W(I,J)*W(I,J))
    YP=YV(J+1)-Y(J)
    DENU=DEN(I,J)
    SQRTK=SQRT(TE(I,J))
    VOL=RCV(J)*SNS(J)*SEW(I)
    YPLUSN(I)=DENU*SQRTK*CDTERM*YP/VISCOS
    IF(YPLUSN(I) .LE. 11.63) GO TO 608
    TMULT=DENU*CDTERM*SQRTK*CAPPA/ALOG(ELOG*YPLUSN(I))
    TAUN(I)=-TMULT*UEFF
    DITERM=DEN(I,J)*(CMU**0.75)*SQRTK*ALOG(ELOG*YPLUSN(I))/(CAPPA*YP)
    GO TO 609
608 TAURX=-VISCOS*UAVG/YP
    TAURW=VISCOS*(-W(I,J)/YP - W(I,J)/Y(J))
    TAUN(I)=SQRT(TAURX**2+TAURW**2)
    DITERM=DEN(I,J)*(CMU**0.75)*SQRTK*YPLUSN(I)/YP
609 DUDY=((U(I,J)+U(I+1,J)+U(I,J+1)+U(I+1,J+1))/4.-(U(I,J)+U(I+1,J)+U(
4I,J-1)+U(I+1,J-1))/4.)/SNS(J)
    GENCOU=TAUN(I)**2/VIS(I,J)
    GENRES=GEN(I,J)-VIS(I,J)*(DUDY**2+(DWDY-W(I,J)/Y(J))**2)
    GEN(I,J)=GENRES+GENCOU
    SU(I,J)=GEN(I,J)*VOL+SPKD(I,J)
    SP(I,J)=-DITERM*VOL+SPKD(I,J)
    AN(I,J)=0.0

```

```

0162100
0162200
0162300
0162400
0162500
0162600
0162700
0162800
0162900
0163000
0163100
0163200
0163300
0163400
0163500
0163600
0163700
0163800
0163900
0164000
0164100
0164200
0164300
0164400
0164500
0164600
0164700
0164800
0164900
0165000
0165100
0165200
0165300
0165400
0165500
0165600
0165700
0165800
0165900
0166000
0166100
0166200
0166300
0166400
0166500
0166600
0166700
0166800
0166900
0167000
0167100
0167200
0167300
0167400
0167500
0167600
0167700
0167800
0167900
0168000

```

610	CONTINUE	0168100
	TAUN(NI)=TAUN(NIMI)	0168200
C-----	SIDE WALL	0168300
	IF(JSTEP .EQ. NJM1) GO TO 614	0168400
	DO 625 I=2,NIMI	0168500
	IF(JMAX(I-1) .GE. JMAX(I)) GO TO 625	0168600
	JJ=JMAXPI(I-1)	0168700
	JI=JMAX(I)	0168800
	DO 620 J=JJ,JI	0168900
	DWDX=(W(I+1,J)-W(I-1,J))/(DXPW(I)+DXEP(I))	0169000
	VAVG=V(I,J)*WFN(J)+(1.-WFN(J))*V(I,J+1)	0169100
	VEFF=SQRT(VAVG*VAVG + W(I,J)*W(I,J))	0169200
	XP=X(I)-XU(I)	0169300
	DENV=DEN(I,J)	0169400
	SQRTK=SQRT(TE(I,J))	0169500
	VOL=RCV(J)*SNS(J)*SEW(I)	0169600
	XPLUSW(J)=DENV*SQRTK*CDTERM*XP/VISCOS	0169700
	IF(XPLUSW(J) .LE. 11.63) GO TO 621	0169800
	TMULT=DENV*CDTERM*SQRTK*CAPPA/ALOG(ELOG*XPLUSW(J))	0169900
	TAUW(J)=-TMULT*VEFF	0170000
	DITERM=DEN(I,J)*(CMU*.75)*SQRTK*ALOG(ELOG*XPLUSW(J))/(CAPPA*XP)	0170100
	GO TO 622	0170200
621	TAUXR=VISCOS*VAVG/XP	0170300
	TAUXW=VISCOS*W(I,J)/XP	0170400
	TAUW(J)=SQRT(TAUXR**2+TAUXW**2)	0170500
	DITERM=DEN(I,J)*(CMU*.75)*SQRTK*XPLUSW(J)/XP	0170600
622	DVDX=((V(I,J)+V(I,J+1)+V(I+1,J)+V(I+1,J+1))/4.-(V(I,J)+V(I,J+1)+V(I-1,J)+V(I-1,J+1))/4.)/SEW(I)	0170700
	GENCOU=TAUW(J)**2/VIS(I,J)	0170800
	GENRES=GEN(I,J)-VIS(I,J)*(DVDX**2+DWDX**2)	0170900
	GEN(I,J)=GENRES+GENCOU	0171000
	SU(I,J)=SU(I,J)+SUKD(I,J)+GEN(I,J)*VOL	0171100
	SP(I,J)=SP(I,J)+SPKD(I,J)-DITERM*VOL	0171200
	AW(I,J)=0.0	0171300
620	CONTINUE	0171400
625	CONTINUE	0171500
	TAUW(NJ)=TAUW(NJMI)	0171600
C-----	SYMMETRY AXIS	0171700
614	CONTINUE	0171800
	J=2	0171900
	DO 630 I=2,NIMI	0172000
	DUDY=((U(I,J)+U(I+1,J)+U(I,J+1)+U(I+1,J+1))/4.-(U(I,J)+U(I+1,J)+U(I,J-1)+U(I+1,J-1))/4.)/SNS(J)	0172100
	VOL=RCV(J)*SNS(J)*SEW(I)	0172200
	GEN(I,J)=GEN(I,J)-VIS(I,J)*DUDY**2	0172300
	SU(I,J)=SUKD(I,J)+GEN(I,J)*VOL	0172400
630	AS(I,2)=0.0	0172500
C-----	OUTLET	0172600
	DO 640 J=2,NJMI	0172700
	AE(NIMI,J)=0.0	0172800
640	CONTINUE	0172900
	RETURN	0173000
C	CHAPTER 7 7 7 7 7 7 7 7 DISSIPATION 7 7 7 7 7 7 7 7	0173100
C		0173200
	7 CONTINUE	0173300
C-----	TOP WALL	0173400
	DO 710 I=2,NIMI	0173500
	J=JMAX(I)	0173600
	YP=YV(J+1)-Y(J)	0173700
		0173800
		0173900
		0174000


```

DO 600 J=2,JJ                                0186100
IF(STFN(I,J) .GE. STVAL(K)) GO TO 650         0186200
600 CONTINUE                                  0186300
YSTLN(I,K)=RV(IJ)                             0186400
GO TO 670                                      0186500
650 IF(J .EQ. 2) YSTLN(I,K)=0.0              0186600
IF(J .EQ. 2) GO TO 670                        0186700
SLOPE=(STVAL(K)-STFN(I,J-1))/(STFN(I,J)-STFN(I,J-1)) 0186800
YSTLN(I,K)=Y(J-1)+SLOPE*(Y(J)-Y(J-1))       0186900
670 CONTINUE                                  0187000
YSTLND(I,K)=YSTLN(I,K)/(2.*RLARGE)           0187100
700 CONTINUE                                  0187200
800 CONTINUE                                  0187300
IF(.NOT. INPLOT) GO TO 745                    0187400
N=0                                             0187500
DO 730 K=1,11,2                               0187600
N=N+1                                          0187700
DO 730 I=1,NIM1                               0187800
YSLPLT(N,I)=YSTLND(I+1,K)                   0187900
730 CONTINUE                                  0188000
DO 740 I=1,NIM1                               0188100
XUDPLT(I)=XUND(I+1)                          0188200
740 CONTINUE                                  0188300
745 NPTS=NIM1                                  0188400
RETURN                                         0188500
END                                             0188600
C                                               0188700
C                                               0188800
SUBROUTINE PLOT (X,IDIM,IMAX,XAXIS,Y,JDIM,JMAX,YAXES,SYMB L,LA) 0188900
CA*****                                     0189000
C                                               0189100
C SUBROUTINE FOR PLOTTING J CURVES OF Y(J,I) AGAINST X(I).    0189200
C                                               0189300
C X AND Y ARE ASSUMED TO BE IN ANY RANGE EXCEPT THAT NEGATIVE VALUES * 0189400
C ARE PLOTTED AS ZERO. * 0189500
C X AND Y ARE SCALED TO THE RANGE 0. TO 1. BY DIVISION BY THE MAXIMA, * 0189600
C WHICH ARE ALSO PRINTED. 0189700
C IDIM IS THE VARIABLE DIMENSION FOR X. 0189800
C IMAX IS THE NUMBER OF X VALUES. 0189900
C XAXIS STORES THE NAME OF THE X-AXIS. 0190000
C JDIM IS THE VARIABLE DIMENSION FOR Y. 0190100
C JMAX IS THE NUMBER OF CURVES TO BE PLOTTED, (UP TO 10). 0190200
C THE ARRAY YAXES(J) STORES THE NAMES OF THE CURVES. 0190300
C THE ARRAY SYMBOL(J) STORES THE SINGLE CHARACTERS USED FOR PLOTTING. * 0190400
C                                               0190500
CA*****                                     0190600
DIMENSION X(IDIM),Y(JDIM,IDIM),YAXES(JDIM),SYMB L(JDIM),
1 A(101),YMAX(10)
DATA DOT,CROSS,BLANK/1H.,1H+,1H /
C-----SCALING X ARRAY TO THE RANGE 0 TO 50
XMAX=1.E-30
DO 1 I=1,IMAX
IF(X(I).GT.XMAX) XMAX=X(I)
1 CONTINUE
DO 2 I=1,IMAX
X(I)=X(I)/XMAX*50.
IF(X(I).LT.0.) X(I)=0.
2 CONTINUE
C-----SCALING Y ARRAY TO THE RANGE 0 TO 100
DO 3 J=1,JMAX
0190700
0190800
0190900
0191000
0191100
0191200
0191300
0191400
0191500
0191600
0191700
0191800
0191900
0192000

```

YMAX(J)=1.E-30	0192100
DO 4 I=1,IMAX	0192200
IF(Y(J,I).GT.YMAX(J)) YMAX(J)=Y(J,I)	0192300
4 CONTINUE	0192400
DO 3 I=1,IMAX	0192500
C NO Y SCALING	0192600
Y(J,I)=Y(J,I)*100.0	0192700
IF(LA.EQ.1) Y(J,I)=0.02*Y(J,I)	0192800
IF(Y(J,I).LT.0.) Y(J,I)=0.	0192900
3 CONTINUE	0193000
C-----IDENTIFYING THE VARIOUS CURVES TO BE PLOTTED	0193100
WRITE(6,103) XAXIS	0193200
WRITE(6,100) (YAXES(I),I=1,JMAX)	0193300
WRITE(6,106) (SYMB L(I),I=1,JMAX)	0193400
WRITE(6,102) (YMAX(I),I=1,JMAX)	0193500
DO 5 I=1,11	0193600
5 A(I)=0.1*(I-1)	0193700
IF(LA.EQ.1) WRITE(6,120)	0193800
IF(LA.EQ.0) WRITE(6,115)	0193900
WRITE(6,101) (A(I),I=1,11)	0194000
C-----MAIN LOOP. EACH PASS PRODUCES AN X-CONSTANT LINE.	0194100
DO 40 II=1,51	0194200
I=II	0194300
IF(I.EQ.1.OR.I.EQ.51) GO TO 32	0194400
GO TO 33	0194500
C-----ALLOCATE . OR + AS MARKER ON THE Y-AXIS	0194600
32 DO 30 K=1,101	0194700
30 A(K)=DOT	0194800
DO 31 K=11,101,10	0194900
31 A(K)=CROSS	0195000
C-----ALLOCATE . OR + MARK ON THE X-AXIS, ALSO THE APPROPRIATE	0195100
C----- X VALUE	0195200
33 A(1)=DOT	0195300
A(101)=DOT	0195400
K=I-1	0195500
46 K=K-5	0195600
IF(K)48,47,46	0195700
47 A(1)=CROSS	0195800
A(101)=CROSS	0195900
48 XL=0.02*(I-1)	0196000
C-----CHECK IF ANY Y(X(I)) VALUE LIES ON THIS X-CONSTANT LINE	0196100
C-----IF YES GO TO 41, OTHERWISE GO TO 42	0196200
DO 43 K=1,IMAX	0196300
IFIX=X(K)+1.5	0196400
IF(IFIX-I)43,41,43	0196500
C-----LOCATE Y(X(I))	0196600
41 DO 44 J=1,JMAX	0196700
NY=Y(J,K)+1.5	0196800
A(NY)=SYMB L(J)	0196900
44 CONTINUE	0197000
GO TO 42	0197100
43 CONTINUE	0197200
C-----PRINT X-CONSTANT LINE	0197300
42 CONTINUE	0197400
IF(LA.EQ.1) GO TO 51	0197500
WRITE(6,105) XL,(A(K),K=1,101),XL	0197600
GO TO 52	0197700
51 WRITE(6,107) XL,(A(K),K=1,101),XL	0197800
52 CONTINUE	0197900
C-----PUTTING BLANKS INTO X-CONSTANT LINE	0198000

DO 49 K=1,101	0198100
49 A(K)=BLANK	0198200
40 CONTINUE	0198300
DO 50 I=1,11	0198400
50 A(I)=.1*(I-1)	0198500
WRITE(6,104)(A(I),I=1,11)	0198600
WRITE(6,130)	0198700
RETURN	0198800
100 FORMAT(11H Y-AXES ARE,5X,10(1X,A10))	0198900
101 FORMAT(1H0,2X,11F10.1)	0199000
102 FORMAT(15H MAXIMUM VALUES, 10E11.3)	0199100
103 FORMAT(11HX-AXIS IS ,A3)	0199200
104 FORMAT(3X,11F10.1)	0199300
105 FORMAT(2H X,F6.2,3X,101A1,F6.2)	0199400
106 FORMAT(7H SYMBOL,11X,10(1X,A10))	0199500
107 FORMAT(2H X,F6.2,3X,101A1,F6.2)	0199600
115 F0RMAT(//,T50,'RADIAL POSITION R/D',/)	0199700
120 FJRMAT(//,T50,'RADIAL POSITION 2R/D',/)	0199800
130 F0RMAT(//,T45,'DIMENSIONLESS STREAMLINE PLOT')	0199900
RETURN	0200000
END	0200100
U VELOCITY	0200600
V VELOCITY	0200700
W VELOCITY	0200800
PRESSURE	0200900
TEMPERATURE	0201000
TURBULENCE ENERGY	0201100
TURBULENCE DISSIPATION	0201200
VISCOSITY	0201300
DIMENSIONLESS LENGTH SCALE	0201400
DIMENSIONLESS STREAM FUNCTION	0201500
RADIAL COORDINATE OF STREAMLINES	0201600
DIMENSIONLESS U VELOCITY	0201700
DIMENSIONLESS V VELOCITY	0201800
DIMENSIONLESS W VELOCITY	0201900
DIMENSIONLESS PRESSURE	0202000
DIMENSIONLESS TURBULENCE ENERGY	0202100
DIMENSIONLESS STREAMLINE COORDS	0202200
DIMENSIONLESS EFF. VISCOSITY	0202300

1. Report No. NASA CR-3442		2. Government Accession No.		3. Recipient's Catalog No.	
4. Title and Subtitle A COMPUTER CODE FOR SWIRLING TURBULENT AXISYMMETRIC RECIRCULATING FLOWS IN PRACTICAL ISOTHERMAL COMBUSTOR GEOMETRIES				5. Report Date February 1982	
				6. Performing Organization Code	
7. Author(s) D. G. Lilley and D. L. Rhode				8. Performing Organization Report No. None	
				10. Work Unit No.	
9. Performing Organization Name and Address Oklahoma State University School of Mechanical and Aerospace Engineering Stillwater, Oklahoma 74078				11. Contract or Grant No. NAG 3-74	
				13. Type of Report and Period Covered Contractor Report	
12. Sponsoring Agency Name and Address National Aeronautics and Space Administration Washington D. C. 20546				14. Sponsoring Agency Code 505-32-32	
				15. Supplementary Notes Final report. Project Manager, Cecil J. Marek, Aerothermodynamics and Fuels Division, NASA Lewis Research Center, Cleveland, Ohio 44135. The microfiche supplement at the back of this report contains the output obtained for a sample computation.	
16. Abstract A primitive pressure-velocity variable finite difference computer code has been developed to predict swirling recirculating inert turbulent flows in axisymmetric combustors in general, and for application to a specific idealized combustion chamber with sudden or gradual expansion. The technique involves a staggered grid system for axial and radial velocities, a line relaxation procedure for efficient solution of the equations, a two-equation k-ε turbulence model, a stairstep boundary representation of the expansion flow, and realistic accommodation of swirl effects. This report is a user's manual and deals with the computational problem, showing how the mathematical basis and computational scheme may be translated into a computer program. A flow chart, Fortran 4 listing, notes about various subroutines and a user's guide are supplied as an aid to prospective users of the code.					
17. Key Words (Suggested by Author(s)) Computer program Combustor Turbulent flow			18. Distribution Statement Unclassified - unlimited STAR Category 07		
19. Security Classif. (of this report) Unclassified		20. Security Classif. (of this page) Unclassified		21. No. of Pages 133	22. Price* A07

* For sale by the National Technical Information Service, Springfield, Virginia 22161