NASA CONTRACTOR
REPORT

NASA CR-179357

VELOCITY-PRESSURE INTEGRATED VERSUS PENALTY FINITE ELEMENT METHODS FOR HIGH REYNOLDS NUMBER FLOWS

By Sang-Wook Kim
Universities Space Research Association
Structures and Dynamics Laboratory
Science and Engineering Directorate

Final Report

April 1988

(NASA-CR-179357) VELOCITY-PRESSURE INTEGRATED VERSUS PENALTY FINITE ELEMENT METHODS FOR HIGH REYNOLDS NUMBER FLOWS Final Report (Universities Space Research Association) 62 p CSCL 20D G3/34

N88-26635

Unclas
0156152

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

TECHNICAL REPORT STANDARD TITLE PAGE

1. REPORT NO.	2. GOVERNMENT ACCESSION NO.	3. RECIPIENT'S CATALOG NO.
NASA CR-179357		

4. TITLE AND SUBTITLE

Velocity-Pressure Integrated Versus Penalty Finite Element Methods for High Reynolds Number Flows

5. REPORT DATE

April 1988

6. PERFORMING ORGANIZATION CODE

7. AUTHOR(S)

Sang-Wook Kim *

8. PERFORMING ORGANIZATION REPORT #

9. PERFORMING ORGANIZATION NAME AND ADDRESS

George C. Marshall Space Flight Center
Marshall Space Flight Center, Alabama 35812

10. WORK UNIT NO.

11. CONTRACT OR GRANT NO.

NAS8-35918

12. SPONSORING AGENCY NAME AND ADDRESS

National Aeronautics and Space Administration
Washington, D.C. 20546

13. TYPE OF REPORT & PERIOD COVERED

Final
Contractor Report

14. SPONSORING AGENCY CODE

15. SUPPLEMENTARY NOTES *Universities Space Research Association.
Prepared by Fluids Dynamics Branch, Structures and Dynamics Laboratory, Science and Engineering Directorate.

16. ABSTRACT

Velocity-pressure integrated and consistent penalty finite element computations of high Reynolds number, laminar flows are presented in this report. In both of the methods, the pressure has been interpolated using linear shape functions for a triangular element. The triangular element is contained inside the bi-quadratic isoparametric element. It has been reported previously that the pressure interpolation method, when used in the velocity-pressure integrated method, yielded accurate computational results for high Reynolds number flows. It is shown in this report that use of the same pressure interpolation method in the consistent penalty finite element method yielded accurate velocity and pressure fields which were comparable to those obtained by using the velocity-pressure integrated method. Accuracy of the two finite element methods has been demonstrated by comparing the computational results with available experimental data and/or fine-grid finite difference computational results. Advantages and disadvantages of the two finite element methods are discussed on the basis of accuracy and convergence nature. Example problems considered include a lid-driven cavity flow for Reynolds number of 10,000, a laminar backward-facing step flow, a laminar flow through a nest of cylinders, and a channel flow with an internal blockage. A finite element computer program [NSFLOW/P] for the two-dimensional, incompressible Navier-Stokes equations is also presented in this report.

17. KEY WORDS

Velocity-Pressure Integrated Finite Element Method, Consistent Penalty Finite Element Method, Laminar Backward-Facing Step Flow, Lid-Driven Cavity Flow, Flow through a nest of cylinders, Channel flow with an internal blockage

18. DISTRIBUTION STATEMENT

Unclassified – Unlimited

19. SECURITY CLASSIF. (of this report)	20. SECURITY CLASSIF. (of this page)	21. NO. OF PAGES	22. PRICE
Unclassified	Unclassified	96	NTIS

TABLE OF CONTENTS

PRECEDING PAGE BLANK NOT FILMED

LIST OF ILLUSTRATIONS

LIST OF TABLES

CONTRACTOR REPORT

VELOCITY-PRESSURE INTEGRATED VERSUS PENALTY FINITE ELEMENT METHODS FOR HIGH REYNOLDS NUMBER FLOWS

I. INTRODUCTION

Various finite element methods for the Navier-Stokes equations have been proposed during the last decades. These finite element methods may be categorized into three classes based on the way pressure has been treated; these are the velocity-pressure integrated mixed interpolation methods [1-3], the penalty methods [3-5], and the velocity-pressure segregated methods [6-8].

The velocity-pressure integrated, mixed interpolation methods do not require any approximation at the differential equation level. Whereas simplified pressure and/or pressure correction equations are used in the velocity-pressure segregated methods, and the penalized conservation of mass equation is used in the penalty methods. Conceptually, the velocity-pressure integrated methods would satisfy the conservation of mass equation most rigorously in a sense that no approximation has to be made at the level of differential equation. The most classical velocity-pressure integrated method is based on an 8-node velocity, 4-node pressure flow element. Unfortunately, this element yielded inaccurate pressure which became more severe as the Reynolds number was increased. It has been shown in Reference 2 that a 9-node velocity, 3-node pressure flow element, when used in the velocity-pressure integrated finite element method, yielded accurate velocity and pressure for high Reynolds number flows. It has also been shown that no upwinding technique was necessary to obtain computational results which were free of numerical wiggles for high Reynolds number flows.

The velocity-pressure segregated methods have been motivated by the success of the finite difference computational methods based on segregated formulation of the Navier-Stokes equations, such as the SIMPLE (Semi-Implicit Method for Pressure-Linked Equations) algorithm [9]. The computational results obtained by using the segregated finite element methods have not shown, as yet, any significant improvement in accuracy compared with the other finite element methods. However, significant computational efficiency in computer time and storage can be achieved by the velocity-pressure segregated methods.

In the penalty method, the pressure is pre-eliminated from the Navier-Stokes equations by penalizing the conservation of mass equation. The conservation of mass constraint could be satisfied rigorously as the penalty number approaches infinity. However, the frequently used penalty number has been limited to 10^6 through 10^{10} times of the kinematic viscosity in order to avoid ill-conditioned matrix [10]. The influence of the penalty number on the converged solution can be found in References 3 to 5, among many others. The consistent penalty finite element method [4] has been used in the present study. The improvements realized by using the new pressure interpolation polynomials in the consistent penalty finite element method is discussed in detail. It is shown that the consistent penalty method and the velocity-pressure integrated method yielded comparable computational results in accuracy and convergence rate.

The nonlinear, finite element system of equations has been solved by the direct (Picard) iteration method using a frontal solver [1,10]. It is intended to extend the present finite element code to solve turbulent flows. For turbulent flows, a strongly convergent solution technique, which may require severe under-relaxation, need to be used to obtain convergent solutions [11-13]. Thus, inclusion of the Newton-Raphson method into the present finite element code has not been considered.

II. FINITE ELEMENT EQUATIONS

A finite element system of equations for two-dimensional, laminar, steady, incompressible flows is described below. The method is based on the standard Galerkin finite element method. In the following discussion, repeated indices imply summation over the indices, unless otherwise specified.

The Navier-Stokes equations are given as:

$$\left.\begin{aligned} &\rho u_j \frac{\partial u_i}{\partial x_j} - \frac{\partial}{\partial x_j}\left\{\mu\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right) - p\delta_{ij}\right\} = b_i \qquad (1) \\ &\frac{\partial u_j}{\partial x_j} = 0 \qquad (2) \end{aligned}\right\} \text{ in } \Omega$$

where Ω is the open bounded domain, the subscripts i and j denote the coordinate directions, ρ is the density, u_i is the velocity component in the i-th coordinate direction, p is the pressure, μ is the molecular viscosity of the fluid, b_i is the body force in the i-th coordinate direction, and δ_{ij} is the Kronecker delta such that $\delta_{ij} = 1$ for $i = j$ and $\delta_{ij} = 0$ for $i \neq j$. The boundary conditions are given as:

$$\left.\begin{aligned} &\underset{\sim}{u} = \underset{\sim}{u}_o(\underset{\sim}{x}) && \text{for } \underset{\sim}{x} \in \partial\Omega_1 \\ &T_i = \tau_{ij} n_j && \text{for } \underset{\sim}{x} \in \partial\Omega_2 \end{aligned}\right\} \qquad (3)$$

where $\underset{\sim}{x} = (x,y)$, $\partial\Omega_1$ is part of the boundary on which Dirichlet boundary condition is specified, $\partial\Omega_2$ is the rest of the boundary on which Newmann boundary condition is specified, T_i is the surface traction, and τ_{ij} is the stress tensor given as $\tau_{ij} = \mu(\partial u_i/\partial x_j + \partial u_j/\partial x_i) - p\delta_{ij}$.

In the penalty method, the conservation of mass equation is expressed as:

$$\frac{\partial u_j}{\partial x_j} = - \frac{1}{\lambda} p \qquad (4)$$

where λ is the penalty number. The finite element system of equations for the consistent penalty method is described below.

The finite element system of equations has been obtained by the standard Galerkin method [14]. In the method: the flow domain is discretized into a number of elements; the Navier-Stokes equations are multiplied by appropriate test functions; the second order stress tensor term is integrated by parts using the Green-Gauss theorem; the continuous flow variables are interpolated using the nodal values of these variables and the appropriate interpolation polynomials; the weak form Navier-Stokes equations are integrated over each element to obtain the discrete, element system of equations; and the element system of equations are assembled to obtain the global system of equations. Detailed derivation of the finite element system of equations can be found in References 1, 2, and 15, among many others.

The system of equations for an element (Ω_e) is given as, in matrix form:

$$\left(\begin{bmatrix} \underset{\approx}{C} & \underset{\approx}{0} \\ (\text{sym}) & \underset{\approx}{C} \end{bmatrix} + \begin{bmatrix} \underset{\approx}{K}_0 & \underset{\approx}{0} \\ (\text{sym}) & \underset{\approx}{K}_0 \end{bmatrix} + \begin{bmatrix} \underset{\approx}{K}_{11} & \underset{\approx}{K}_{12} \\ \underset{\approx}{K}_{21} & \underset{\approx}{K}_{22} \end{bmatrix} \right) \begin{Bmatrix} \underset{\sim}{u}_1 \\ \underset{\sim}{u}_2 \end{Bmatrix} - \begin{bmatrix} \underset{\approx}{Q}_1 \\ \underset{\approx}{Q}_2 \end{bmatrix} \{ \underset{\sim}{p} \} = \begin{Bmatrix} \underset{\sim}{f}_1 \\ \underset{\sim}{f}_2 \end{Bmatrix} + \{ \text{b.c.} \} \qquad (5)$$

$$[\underset{\approx}{Q}_1^T \quad \underset{\approx}{Q}_2^T] \begin{Bmatrix} \underset{\sim}{u}_1 \\ \underset{\sim}{u}_2 \end{Bmatrix} = - \frac{1}{\lambda} [\underset{\approx}{M}_p] \qquad (6)$$

where

$$\underset{\approx}{C} = \int_{\Omega_e} \underset{\sim}{\phi}^T \rho \, (\underset{\sim}{u}_j^T \, \underset{\sim}{\phi}) \frac{\partial \underset{\sim}{\phi}}{\partial x_j} \, d\underset{\sim}{x} \quad , \qquad (7)$$

$$\underset{\approx}{K}_0 = \int_{\Omega_e} \frac{\partial \underset{\sim}{\phi}^T}{\partial x_j} \mu \frac{\partial \underset{\sim}{\phi}}{\partial x_j} \, d\underset{\sim}{x} \quad , \qquad (9)$$

$$\underset{\approx}{K}_{ij} = \int_{\Omega_e} \frac{\partial \underset{\sim}{\phi}^T}{\partial x_j} \mu \frac{\partial \underset{\sim}{\phi}}{\partial x_i} \, d\underset{\sim}{x} \quad , \qquad (9)$$

$$\underset{\approx}{Q}_i = \int_{\Omega_e} \frac{\partial \underset{\sim}{\phi}^T}{\partial x_i} \underset{\sim}{\psi} \, d\underset{\sim}{x} \quad , \tag{10}$$

$$\underset{\approx}{f}_i = \int_{\Omega_e} \underset{\sim}{\phi}^T b_i \, d\underset{\sim}{x} \quad , \tag{11}$$

$$\underset{\approx}{M}_p = \int_{\Omega_e} \underset{\sim}{\psi}^T \underset{\sim}{\psi} \, d\underset{\sim}{x} \quad , \tag{12}$$

$\underset{\sim}{u}_i$ is a column vector of nodal values of the velocity component u_i, $\underset{\sim}{p}$ is a column vector of nodal pressure, $\underset{\sim}{\phi}$ is a column vector of interpolating polynomials for velocity, $\underset{\sim}{\psi}$ is a column vector of interpolating polynomials for pressure, {b.c.} is a column vector of the flux boundary condition, and the subscripts i and j denote the spatial dimensions. The integrations in equations (7) thorugh (12) have been evaluated using the Gauss numerical quadrature method with three Gauss points in each coordinate direction.

In the velocity-pressure integrated method, the right hand side of equation (6) has been replaced by a null column vector, and the element system of equations given as equations (5) and (6) were assembled to obtain the global system of equations. In the penalty finite element method, equation (6) has been inverted to obtain a column vector of the nodal pressure and the result has been substituted into equation (5) to obtain:

$$\left(\begin{bmatrix} \underset{\approx}{C} & \underset{\approx}{0} \\ (\text{sym}) & \underset{\approx}{C} \end{bmatrix} + \begin{bmatrix} \underset{\approx}{K}_0 & \underset{\approx}{0} \\ (\text{sym}) & \underset{\approx}{K}_0 \end{bmatrix} + \begin{bmatrix} \underset{\approx}{K}_{11} & \underset{\approx}{K}_{12} \\ \underset{\approx}{K}_{21} & \underset{\approx}{K}_{22} \end{bmatrix} \right) \begin{Bmatrix} \underset{\sim}{u}_1 \\ \underset{\sim}{u}_2 \end{Bmatrix}$$

$$+ \lambda \begin{bmatrix} \underset{\approx}{Q}_1 \\ \underset{\approx}{Q}_2 \end{bmatrix} [\underset{\approx}{M}_p] \, [\underset{\approx}{Q}_1^T \; \underset{\approx}{Q}_2^T] \begin{Bmatrix} \underset{\sim}{u}_1 \\ \underset{\sim}{u}_2 \end{Bmatrix} = \begin{Bmatrix} \underset{\sim}{f}_1 \\ \underset{\sim}{f}_2 \end{Bmatrix} + \{\text{b.c.}\} \quad . \tag{13}$$

The flow element used in the present study is briefly described below. The velocities were interpolated using the bi-quadratic shape functions and the pressure was interpolated using the linear shape functions defined on a triangular element. The triangular pressure element is contained inside the quadratic isoparametric element [2]. The three pressure nodes are located at the three Gauss points of the three-point Gauss quadrature rule for quadrilateral elements [16]. The coordinates of the pressure nodes on the computational element are given as:

$$\underset{\sim}{\xi}_n = \begin{cases} (\ 0 \quad \sqrt{2}/\sqrt{3}\ ) & \text{for } n = 1 \\ (-1/\sqrt{2}\ ,\ -1/\sqrt{6}\ ) & \text{for } n = 2 \\ (\ 1/\sqrt{2}\ ,\ -1/\sqrt{6}\ ) & \text{for } n = 3 \end{cases} \tag{14}$$

where $\underset{\sim}{\xi}_n = (\xi_n, \eta_n)$, and n denotes the pressure node number. The shape function for each pressure node is given as:

$$\left.\begin{aligned} \psi_1 &= \frac{1}{3} + \frac{\sqrt{2}}{\sqrt{3}}\,\eta \\ \psi_2 &= \frac{1}{3} - \frac{1}{\sqrt{2}}\,\xi - \frac{1}{\sqrt{6}}\,\eta \\ \psi_3 &= \frac{1}{3} + \frac{1}{\sqrt{2}}\,\xi - \frac{1}{\sqrt{6}}\,\eta \end{aligned}\right\} \tag{15}$$

Slightly different pressure interpolation methods have also been tested in the present study. These pressure interpolation polynomials are given as [4]:

$$\underset{\sim}{\psi}^T = (1,\ x,\ y) \tag{16}$$

and

$$\underset{\sim}{\psi}^T = (1,\ \xi,\ \eta)\quad . \tag{17}$$

These three sets of pressure interpolation polynomials, equations (15) to (17), belong to the same approximation space if rectangular elements are used; and equations (15) and (17) belong to the same approximation space if arbitrary distorted quadrilateral elements are used. Thus, any pressure interpolation polynomials which belong to the same approximation space should yield identical computational results. Any difference in the computational results has to be related to the matrix condition and the computer round off error [2]. The performance of these three sets of slightly different pressure interpolation polynomials, when used in the consistent penalty method, are discussed in the following section.

The nonlinear system of equations has been solved by a direct (Picard) iteration method using a frontal solver. The solutions have been updated using an under-relaxation method given as:

$$a_j^* = \alpha\, a_j^n + (1 - \alpha)\, a_j^{n-1} \tag{18}$$

where a_j represents any degree of freedom, α is the under-relaxation number, the superscripts n and n-1 denote the iteration levels, and a_j^* is the updated solution. $\alpha = 0.8$ and $\alpha = 1$ have been used for the velocities and the pressure, respectively.

III. EXAMPLE PROBLEMS

The finite element methods described in the previous section were tested by solving a lid-driven cavity flow [15, 17-21], a laminar backward-facing step flow [22-24], a laminar flow through a nest of cylinders [25-27], and a channel flow with an internal blockage. For the cavity flow, sharp boundary layers develop along all the boundary edges of the cavity at high Reynolds numbers. For the backward-facing step flow, the flow expands abruptly at the convex corner of the step. These flows contain subtle pressure driven recirculation zones. Inside the pressure driven recirculation zones, the local Reynolds number may become vanishingly small. Due to these reasons, obtaining convergent solutions with any iterative numerical method can be quite difficult [23]. Therefore, these two flows provide serious test cases for any numerical methods. The laminar flow through a nest of cylinders has been considered to investigate the convergence nature of the two finite element methods. The channel flow with an internal blockage has been included herein to investigate the source of numerical wiggles for high Reynolds number flows.

The error norm (ε_i) has been defined as:

$$\varepsilon_i = \max_{j=1,N} \left(\left| 1 - \frac{a_{i,j}^{\,n}}{a_{i,max}^{\,n}} \right| \right) \tag{19}$$

where the subscript i (i = u, v, or p) denotes each component of the flow variables, $a_{i,j}$ denotes the nodal value of the i-th flow variable for the j-th node, $a_{i,max}$ denotes the maximum value of the i-th flow variable in the previous iteration level, and N is the total number of nodes. Solving the coupled system of equations once has been counted as an iteration.

In the penalty method, the pressure is recovered in the post process using the penalized conservation of mass equation. The quality of the recovered pressure depends on the velocity. In the present study, the pressure has been recovered at the end of each iteration. The purpose has been to provide some insight into the convergence nature of the pressure in the penalty methods. This information may be helpful in selecting the convergence criterion in application situations. Note that the required number of iterations to obtain a convergent solution depends on the prescribed convergence criterion in nonlinear problems.

The pressure is discontinuous across element boundaries. Thus, the nodal pressure at the velocity node has been obtained by averaging all the pressure contributions made by the elements containing the node; and each contribution has been evaluated using an equation given as:

(20)

$$p = p_j \psi_j \qquad (20)$$

where ψ_j is the pressure interpolation polynomial and p_j is the corresponding pressure degree of freedom.

A penalty number of $(\mu/\rho) \times 10^{10}$ has been used in the present study.

3.1 Lid-Driven Cavity Flow

A lid-driven cavity flow for Reynolds number of 10,000 is considered below. The fine grid finite difference computational results of the flow can be found in References 18 and 19. The no slip boundary condition ($u = v = 0$) has been applied at all the boundaries except at $y = 1$ where $u = 1$ and $v = 0$. A fixed pressure boundary condition has been prescribed at an arbitrary pressure node inside the domain. The Reynolds number is defined as $R_e = \rho UL/\mu$, where $U = 1$ is the velocity of the lid, $L = 1$ is the reference length, and μ is the molecular viscosity of the fluid. The computational domain has been discretized by unequally spaced 32 x 32 quadratic elements [2]. The trivial solution ($u = v = p = 0$) has been used as an initial guess for all the cases.

The streamline contours and the normalized pressure contours obtained by using the penalty method with the pressure interpolation polynomials given as equations (15) and (16) are shown in Figures 1 and 2, respectively. The normalized pressure (P) has been obtained from the static pressure (p) using a relationship given as [20]:

$$P = pL/U/\mu \quad . \qquad (21)$$

The streamline and the pressure contour labels are given in Table 1. In Figures 1 and 2, the reference pressure of $p = 0$ has been assigned at the middle of the bottom wall.

TABLE 1. CONTOUR LABEL FOR CAVITY FLOW

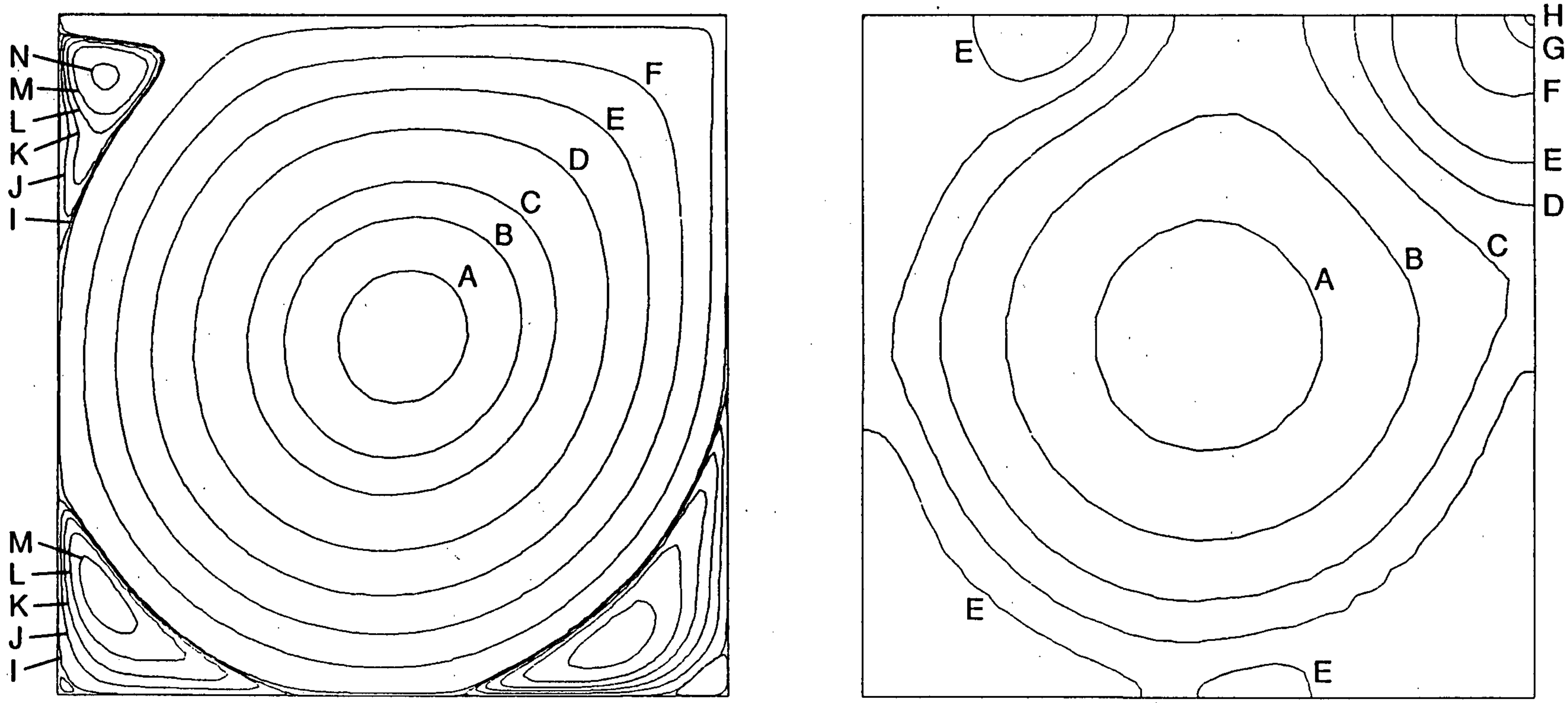
(a) Streamline Contour

Label	ψ*	Label	ψ	Label	ψ
A	-0.11	F	-0.03	K	$2.X10^{-4}$
B	-0.10	G	-0.01	L	$5.X10^{-4}$
C	-0.09	H	$-1.X10^{-10}$	M	$1.X10^{-3}$
D	-0.07	I	$1.X10^{-6}$	N	$2.X10^{-3}$
E	-0.05	J	$5.X10^{-5}$		

(b) Pressure Contour

Label	P	Label	P	Label	P
A	-900.	D	-200.	G	1000.
B	-650.	E	0.	H	3000.
C	-400.	F	400.		

* ψ: Stream Function



(a) (b)

Figure 1. Cavity flow for Re = 10,000, penalty method with equation (15), (a) streamline, and (b) pressure.

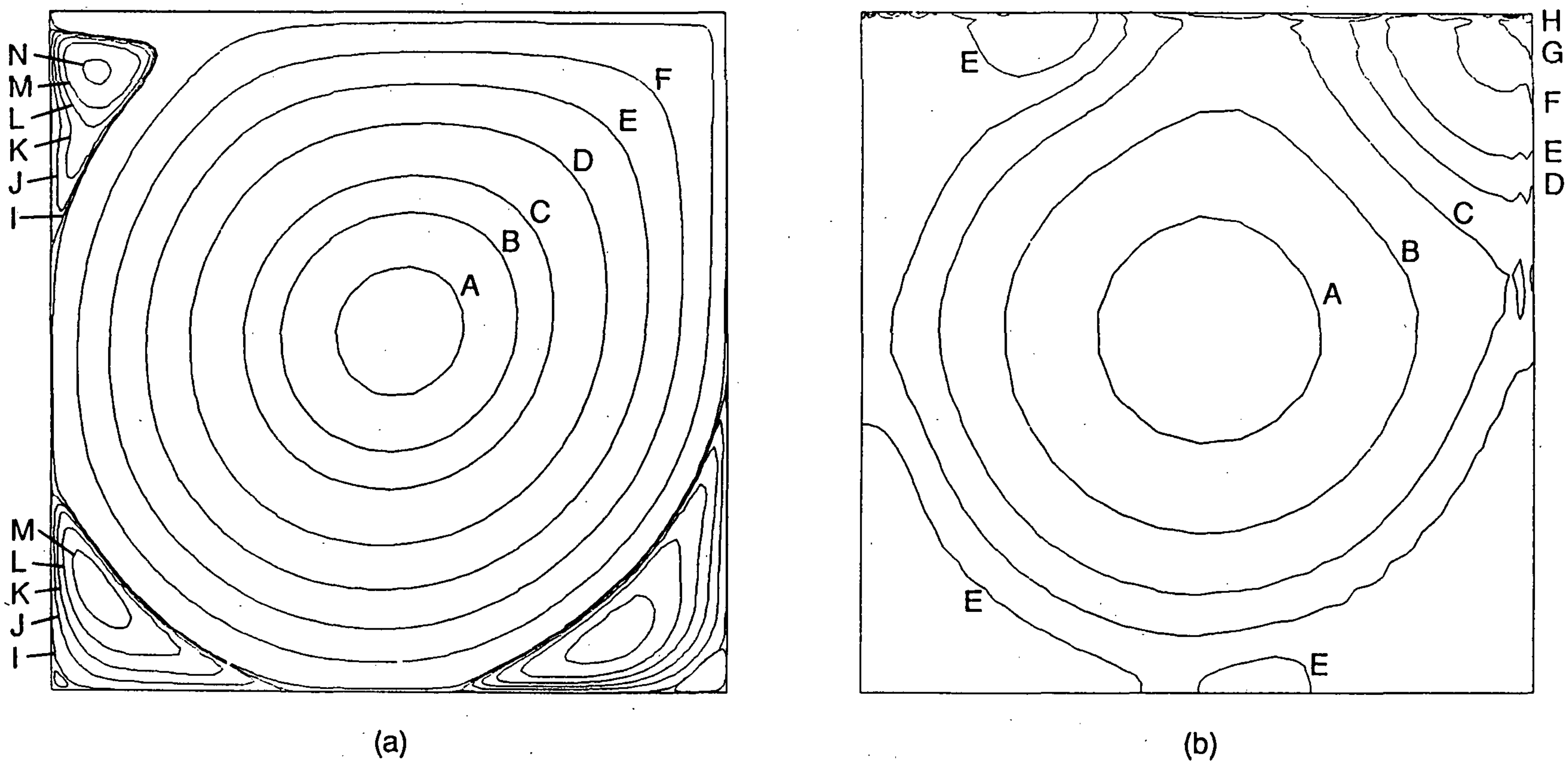


Figure 2. Cavity flow for Re = 10,000, penalty method with equation (16), (a) streamline and (b) pressure.

The penalty method with the pressure interpolation polynomials of the form (1,x,y) yielded slightly distorted pressure contour lines near the top region of the cavity [Fig. 2(b)]. It was found that the distorted pressure contour lines had been caused by the ill-conditioned pressure matrix ($\underset{\approx}{M}_p$) and the coefficients of the pressure interpolation polynomials. The entries of the pressure matrix were different by several orders of magnitude and the coefficients of the pressure interpolation polynomials were approximately 10 orders of magnitude different for the high aspect ratio fine grids located along the boundary of the cavity. However, these distorted pressure contour lines may disappear if a different pressure averaging technique and/or a different plotting package are used.

The error norm (ε_i) versus number of iterations for each flow variable is shown in Figure 3. It can be seen in Figure 3 that the velocity-pressure integrated method and the penalty method with the pressure interpolation polynomials given as equations (15) and (17) yielded almost identical convergence rate for both velocity and pressure. But the penalty method with the pressure interpolation polynomials of the form (1,x,y) exhibited significantly degenerated convergence rate for pressure. It was found that the ill-conditioned pressure matrix ($\underset{\approx}{M}_p$) and the computer round-off error were responsible for the degenerated convergence rate for pressure. For this case, the pressure does not seem to converge at all as the number of iterations were increased behond approximately 10 (Fig. 3).

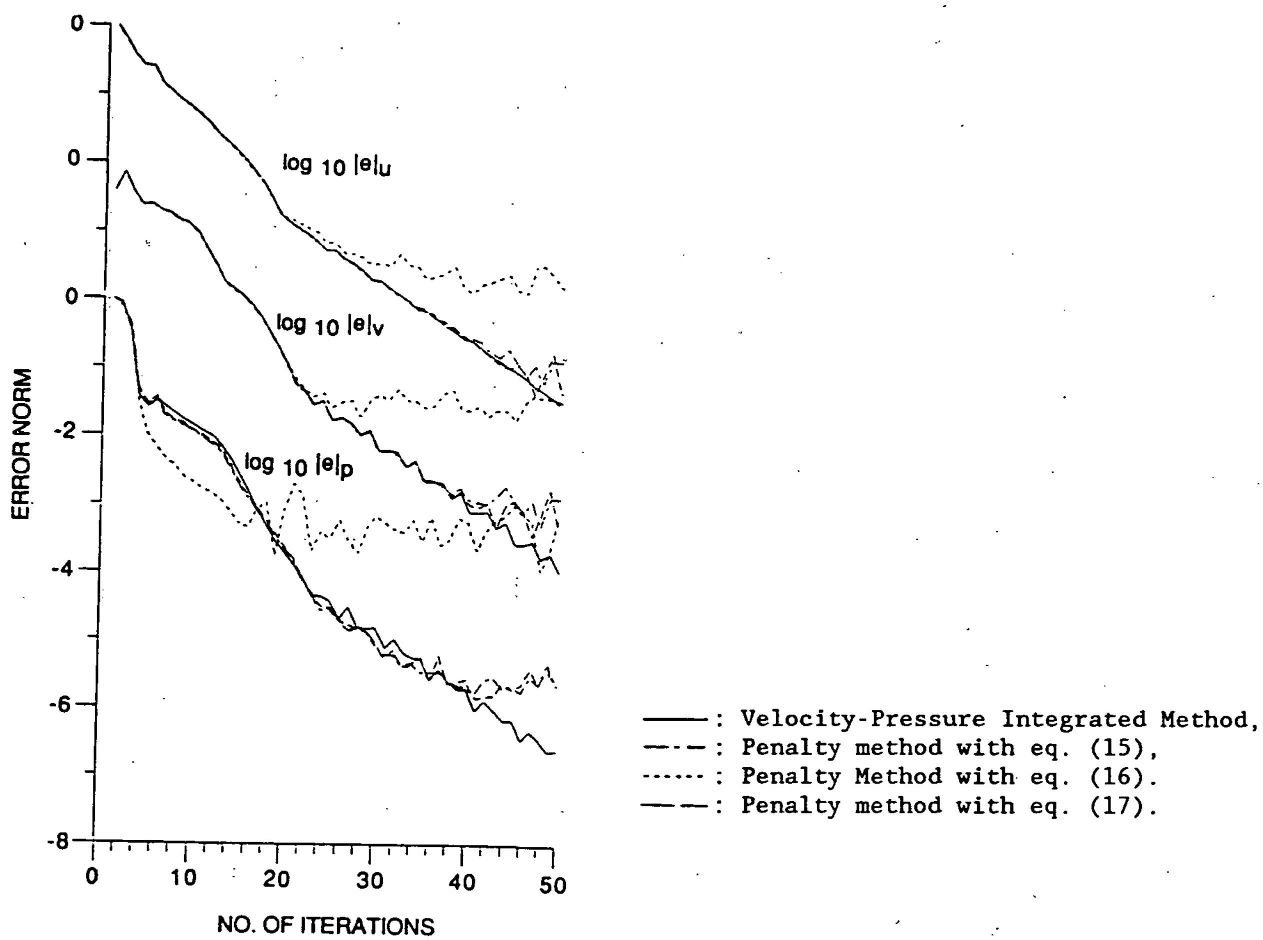


Figure 3. Error norm versus number of iterations for cavity flow.

All the methods yielded almost identical velocity vectors, which can be seen from the stream line contours shown in Figures 1 and 2. The horizontal velocity profiles at x = 0.5, obtained by using the velocity-pressure integrated method, compared favorably with those of Schreiber and Keller [18] and Ghia et al. [19] (see Reference 2). To solve the same cavity flow for Reynolds number of 10,000, 180 x 180 grid points have been used in Schreiber and Keller [18]; and 129 x 129 grid points, in Ghia et al. [19].

3.2 Backward-Facing Step Flow

A laminar backward-facing step flow is considered below. The experimental data can be found in Armaly et al. [22]. In the following discussion, the Reynolds number is based on the hydraulic diameter (D = 0.0104 m) and the bulk velocity (V = 0.6667 m/sec) at the inlet. The experimental data showed that there exists only one recirculation zone at the down-stream region of the backward-facing step for the Reynolds number less than approximately 450. As the Reynolds number was increased beyond approximately 450, a second pressure driven recirculation zone appeared at the top wall of the channel.

The Reynolds numbers considered in the present study were 410, 420, 430, 440, and 500. These various Reynolds numbers have been obtained by varying the fluid viscosity. The velocity profile of a fully developed channel flow has been applied at the inlet boundary and the vanishing normal stress boundary condition has been prescribed at the exit boundary. The trivial solution (u = v = p = 0) has been used as an initial guess for all the Reynolds number cases considered. The quality of the solutions for all the methods remained unchanged after 50 iterations. A complete set of computational results obtained by using the velocity-pressure integrated method for the Reynolds number of 100 through 900 can be found in Reference 2. The finite difference computations of the same backward-facing step flow can be found in Armaly et al. [22] and Kim and Moin [23], among many others.

All the methods considered herein yielded almost identical velocity vectors for Reynolds numbers of 430 and 440, respectively, and predicted the existence of the pressure driven recirculation zone at the top wall for Re $\geq$ 440. The streamline contours for Reynolds numbers of 430 and 440 are shown in Figure 4. The stream function (ψ) has been normalized using a relationship given as [24]:

$$\Psi = \psi/(U_{max} h) \quad , \tag{22}$$

where Ψ is the normalized stream function, U_{max} is the maximum velocity at the inlet, and h is the step height. The streamline contour labels are given in Table 2.

The streamline and pressure contours for Re = 500 obtained by using the penalty method with the pressure interpolation polynomials given in equations (15) and (16) are shown in Figures 5 and 6, respectively. The pressure (p) has been normalized using a relationship given as [24]:

$$P = p/(\rho U_{max}^2/2) \tag{23}$$

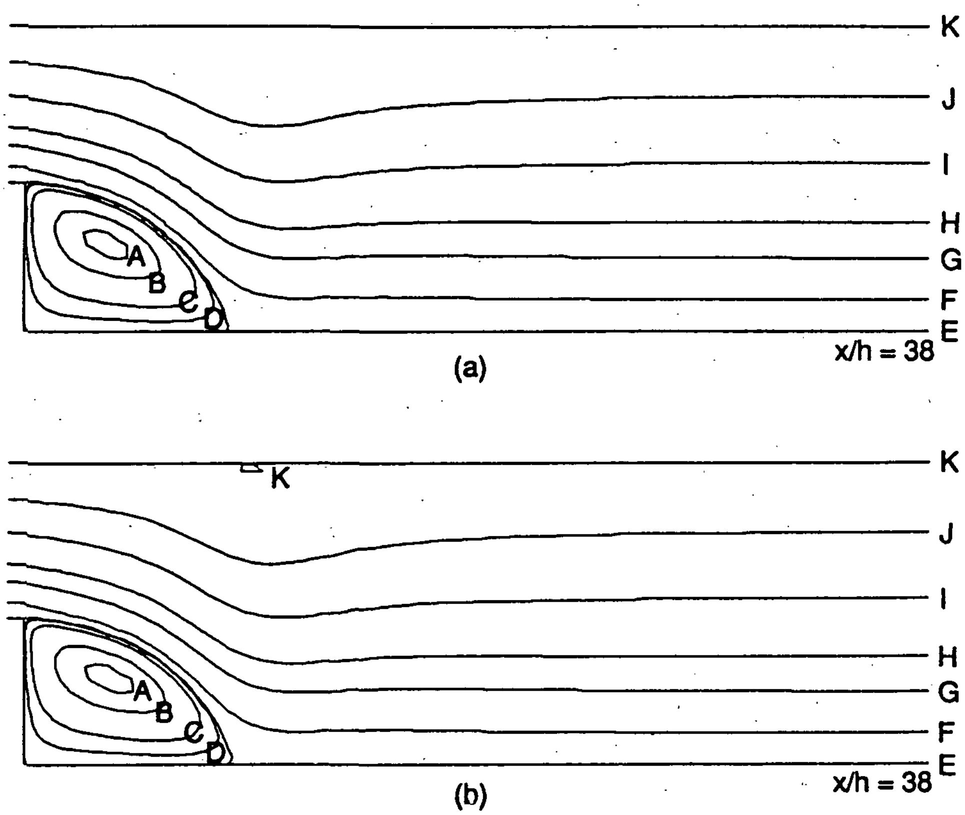


Figure 4. Streamline contours for backward-facing step flow, (a) Re = 430, and (b) Re = 440.

TABLE 2. CONTOUR LABEL FOR BACKWARD-FACING STEP FLOW

(a) Streamline Contour

Label	Ψ	Label	Ψ	Label	Ψ
A	-0.0408	E	0.0000	I	0.4082
B	-0.0305	F	0.0204	J	0.6122
C	-0.0102	G	0.1020	K	0.707483
D	-0.0020	H	0.2041		

(b) Pressure Contour

Label	P	Label	P	Label	P
A	-0.0027	F	0.1333	K	0.2044
B	0.0017	G	0.1556	L	0.2058
C	0.0111	H	0.1778	M	0.2069
D	0.0314	I	0.1955		
E	0.0755	J	0.2011		

where P is the normalized pressure. The pressure contour labels are given in Table 2. The reference pressure of p = 0 has been assigned at the concave corner of the step. The streamline contours obtained by using the velocity-pressure integrated method and the penalty method with the pressure interpolation polynomials given as equation (17) were identical to the streamline contour shown in Figure 5. The penalty method with the pressure interpolation polynomials of the form (1,x,y) yielded a severely distorted pressure contour (Fig. 6).

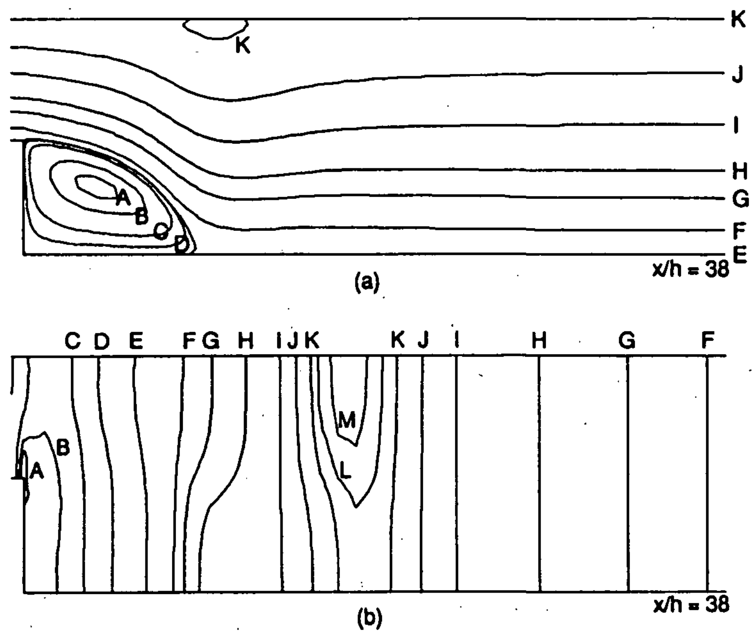


Figure 5. Backward-facing step flow, penalty method with equation (15), (a) streamline, and (b) pressure.

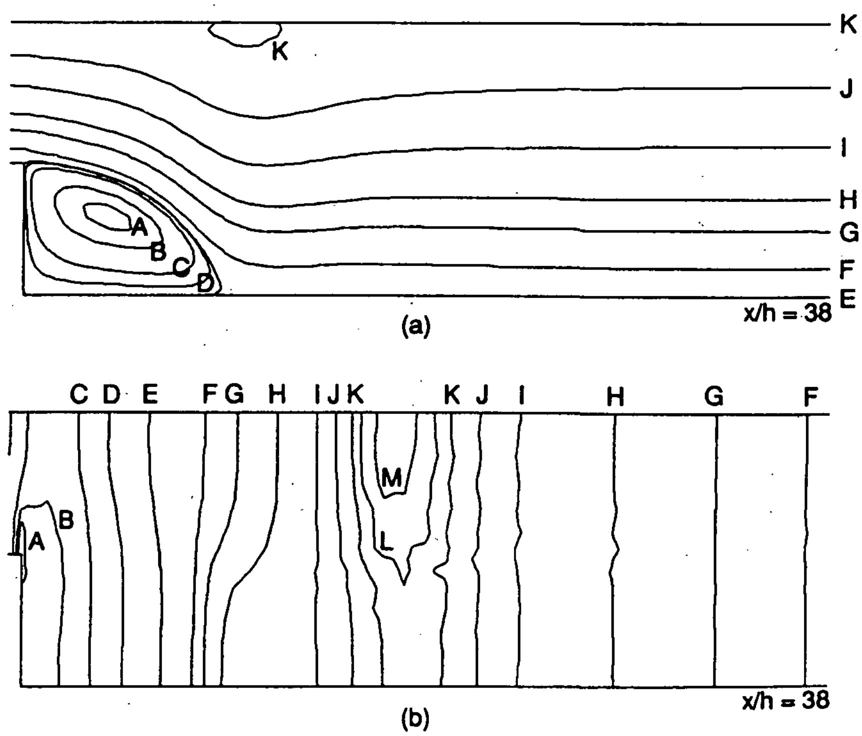


Figure 6. Backward-facing step flow, penalty method with equation (16), (a) streamline, and (b) pressure.

The error norm versus number of iterations for each flow variable is shown in Figure 7. Both the velocity-pressure integrated method and the penalty method with the pressure interpolation polynomials given as equations (15) and (17) yielded rapidly convergent solutions. The pressure interpolation polynomials of the form (1,x,y) exhibited degenerated convergence rate for pressure, due to the same reasons as have been discussed in the previous cavity flow case.

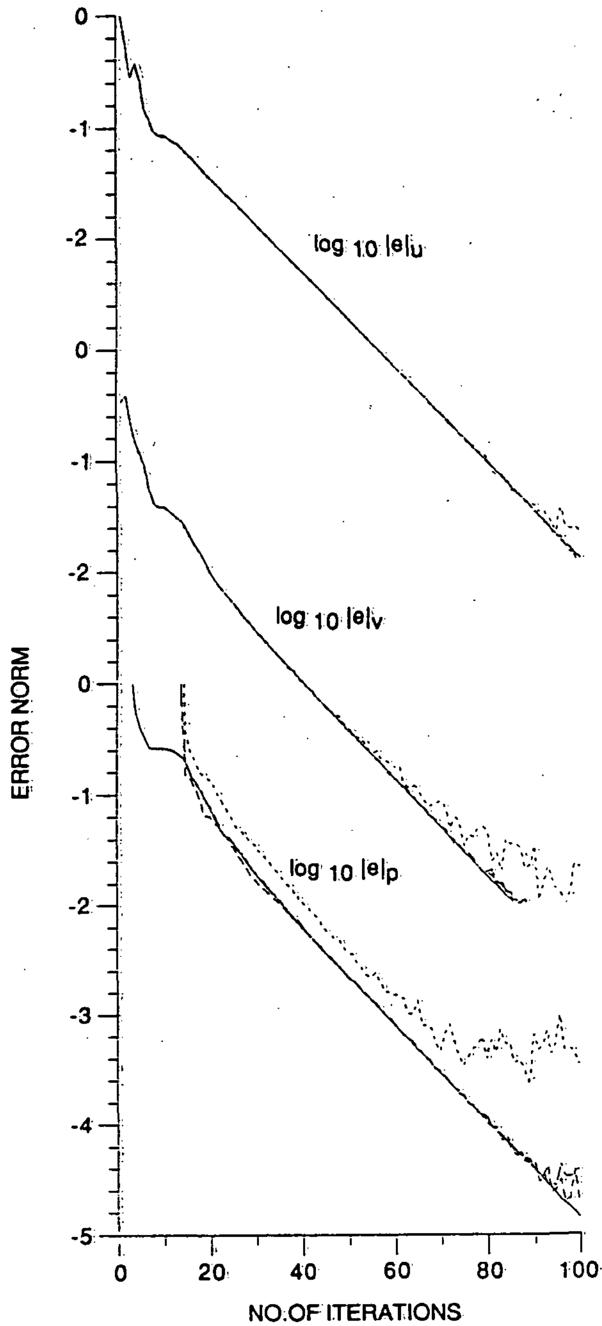


Figure 7. Error norm versus number of iterations for backward-facing step flow, notations are the same as in Figure 3.

The present computational results compared favorably with the experimental data [22] as well as the finite difference computational results of Kim and Moin [23], see Reference 2. In Kim and Moin [23], the exit boundary has been located at 30 step heights downstream of the expansion corner and 101 x 101 grid points have been used.

3.3 Flow Through a Nest of Cylinders

Flows through a nest of cylinders can be found in a number of engineering applications such as the Space Shuttle Main Engine - Main Injector Assembly (SSME-MIA) and the heat exchangers in nuclear reactors (see Reference 25 for more details). However, these flows began to be solved numerically only very recently. These are a finite element computation of a two-dimensional laminar flow through a nest of cylinders [26] and a body-fitted grid finite difference computation of a three-dimensional laminar flows through a nest of cylinders [27]. Neither experimental data nor detailed computational results are available for these flows as yet.

A laminar flow through a nest of cylinders at a Reynolds number of 40 is considered below (Fig. 8). The Reynolds number is defined as $Re = \rho UD/\mu$, where $U = 1$ is the free stream velocity and $D = 1$ is the diameter of a cylinder. The inlet boundary has been located at three diameters upstream of the forward stagnation point of the first column of cylinders; and the exit boundary at 41 diameters downstream of the inlet boundary. A uniform velocity profile has been used as the inlet boundary condition. The vanishing normal stress boundary condition has been prescribed at the exit boundary; and the symmetric boundary condition, at the top and the bottom of the computational domain. The computational domain has been discretized by 1024 quadratic elements with 4369 nodes. The finite element mesh in the vicinity of the nest of cylinders is shown in Figure 8. The trivial solution ($u = v = p = 0$) has been used as an initial guess.

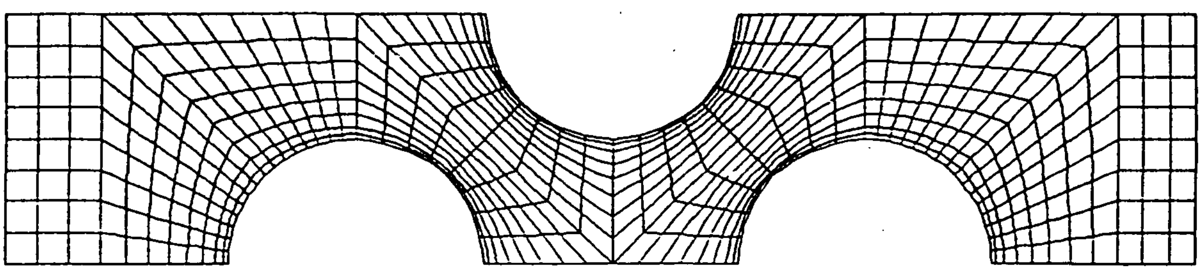


Figure 8. Flow through a nest of cylinders, grid in the vicinity of the nest of cylinders.

The streamline and pressure contours obtained by using the velocity-pressure integrated method and the penalty method with the pressure interpolation polynomials given as equations (15) and (16) are shown in Figures 9, 10, and 11, respectively. The pressure has been normalized using a relationship given as $P = p/(\rho U^2/2)$, where $U = 1$ is the reference velocity at the inlet boundary. An arbitrary reference pressure ($p = 0.0$) has been assigned at the forward stagnation point of the first column of cylinders. The streamline contour label is given in Table 3. In Figures 9 through 11, the minimum and maximum normalized pressures (P) are -20.0 and 0.0, respectively; and the incremental normalized pressure (ΔP) between the contour lines is

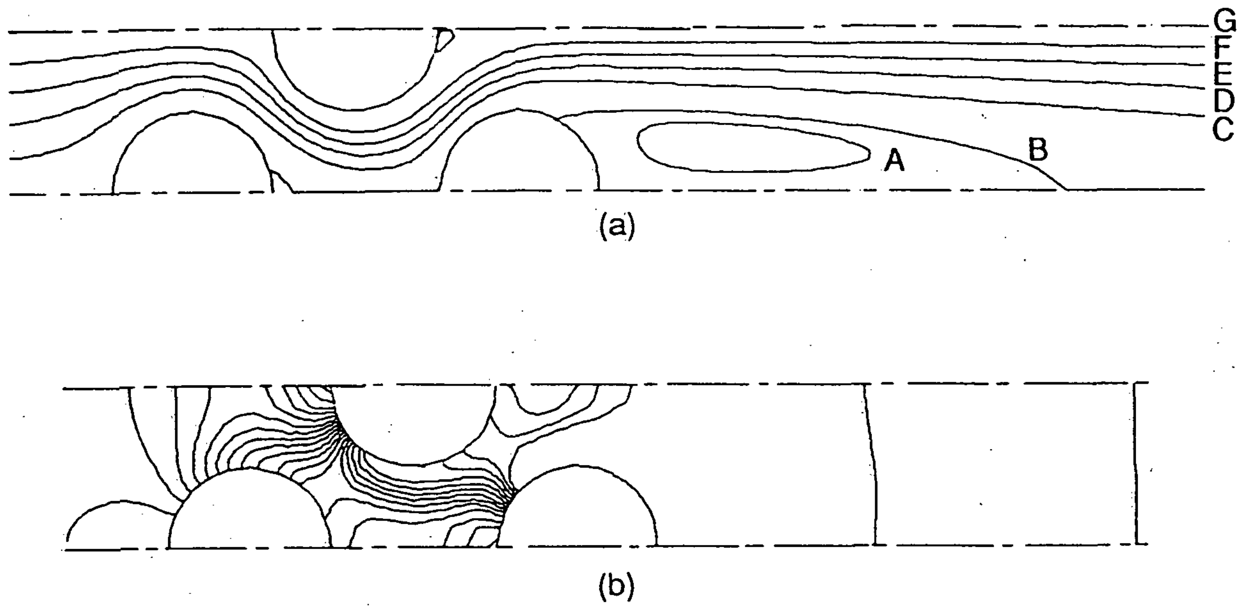


Figure 9. Flow through a nest of cylinders, velocity-pressure integrated method, (a) streamline, and (b) pressure.

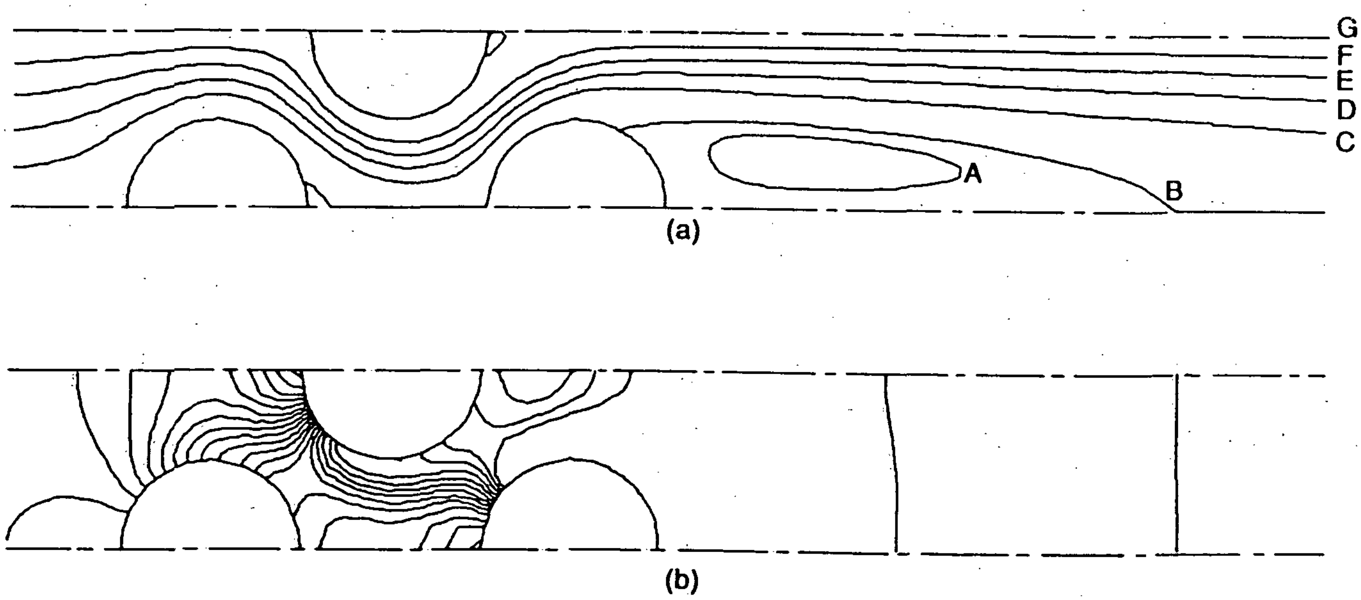


Figure 10. Flow through a nest of cylinders, penalty method with equation (15), (a) streamline, and (b) pressure.

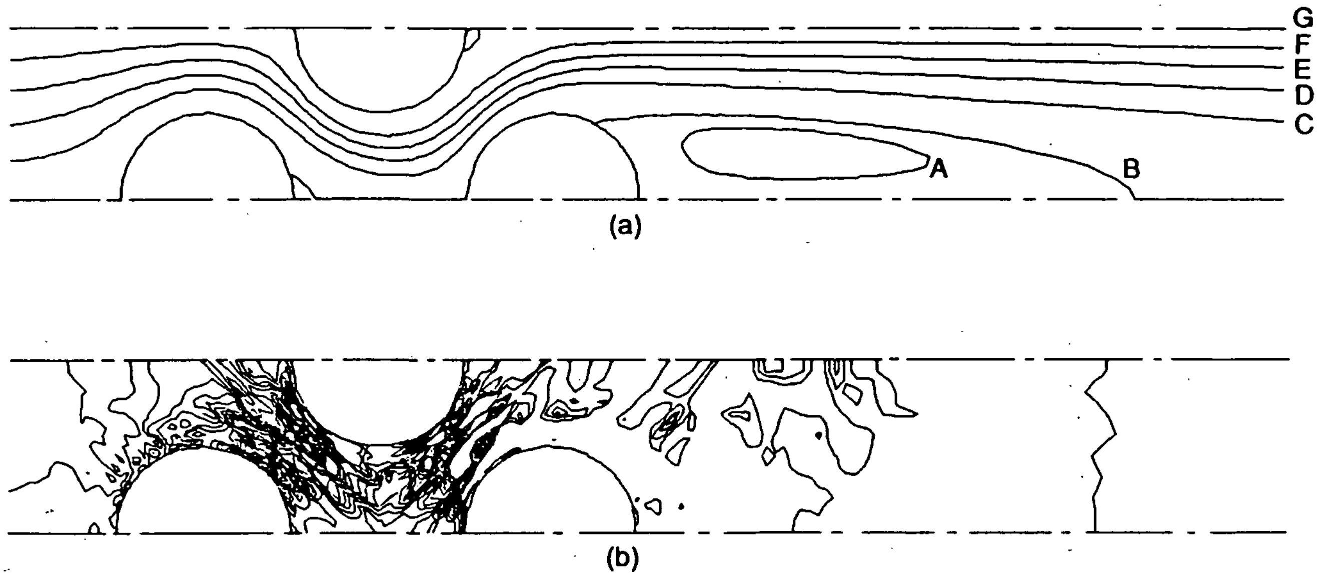


Figure 11. Flow through a nest of cylinders, penalty method with equation (16), (a) streamline, and (b) pressure.

TABLE 3. STREAMLINE CONTOUR LABEL FOR FLOW THROUGH A NEST OF CYLINDERS

Label	ψ	Label	ψ	Label	ψ
A	-0.04	D	0.40	G	1.0
B	0.00	E	0.60		
C	0.20	F	0.80		

equal to 1.0. The streamline and pressure contours obtained by using the penalty method with the pressure interpolation polynomials given as equation (17) was identical to those shown in Figures 9 and 10. The penalty method with the pressure interpolation polynomials given as equation (16) yielded severely distorted pressure contour lines for the same reasons as have been listed previously (Fig. 11).

The error norm versus number of iterations for each flow variable is shown in Figure 12. The practically convergent solutions have been obtained after approximately 15 iterations for all the cases. The velocity-pressure integrated method yielded uniformly convergent solution as before. The penalty method yielded rapidly convergent solutions as the velocity-pressure integrated method at earlier iterations. For the arbitrary distorted quadrilateral elements with high aspect ratio, the adverse effect of the ill-conditioned pressure matrix and the computer round-off error became so severe that only the velocity-pressure integrated method yielded uniformly convergent pressure as the number of iterations was increased.

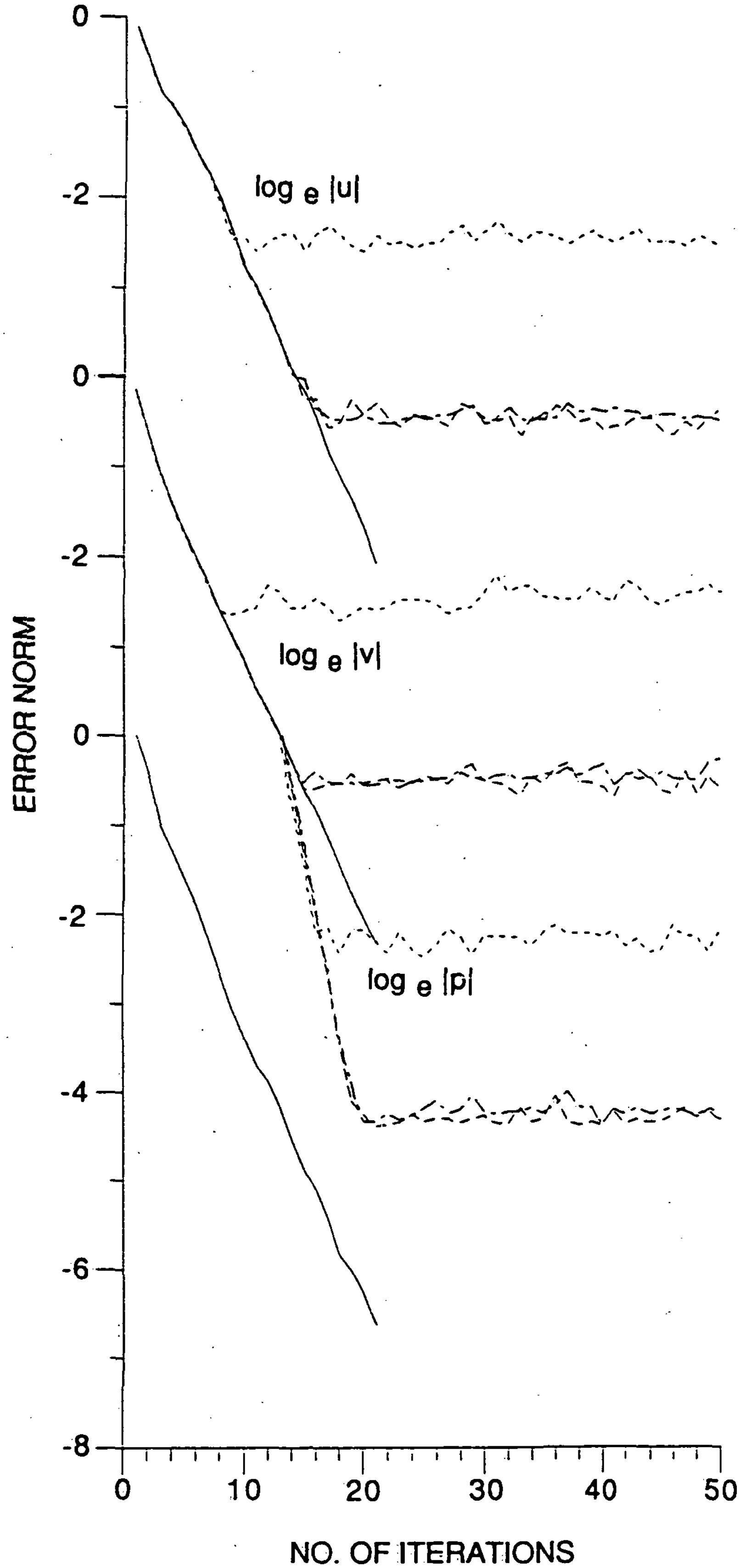


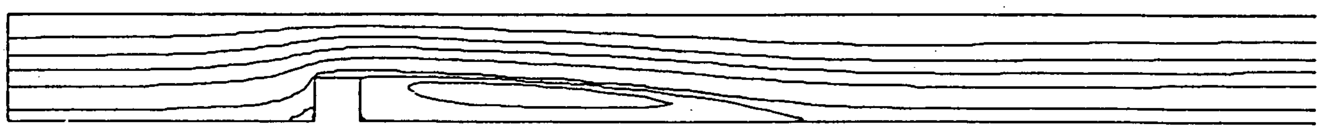
Figure 12. Error norm versus number of iterations for flow through a nest of cylinders, notations are the same as in Figure 3.

3.4 Channel Flow with an Internal Blockage

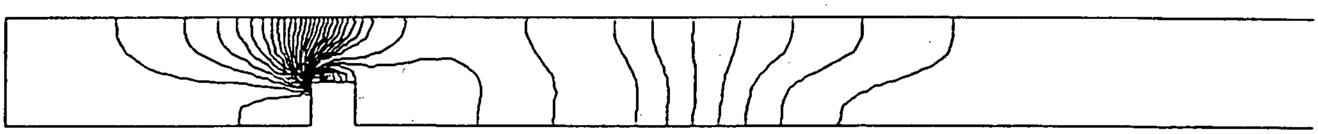
There exists a controversy over the use of upwinding techniques for convection dominated flows. Use of the upwinding techniques has been partly based on an argument that the discrete finite element system of equations become ill-conditioned as the grid Reynolds number is increased; and it has been claimed that use of such upwinding techniques is the best approach to suppress the numerical wiggles when coarse grid is used [28]. The opponents claimed that the numerical diffusion introduced by use of such upwinding techniques may obscure the physical diffusion process and the computational results may not be accurate [29].

A channel flow with an internal blockage is considered below to further investigate the cause of numerical wiggles. The finite element mesh for the full computational domain is shown in Figure 13. The velocity profile of a fully developed channel flow has been used as the inlet boundary condition; and the vanishing normal stress has been prescribed at the exit boundary. The trivial solution (u = v = p = 0) has been used as an initial guess. The streamline and pressure contours obtained by using the velocity-pressure integrated method and the penalty method with the pressure interpolation polynomials given as equations (15) and (16) are shown in Figures 14, 15, and 16, respectively. The convergence history for each of the flow variables is given in Figure 17. Again, only the velocity-pressure integrated method yielded uniformly convergent solution as the number of iterations was increased.

Figure 13. Computational domain and finite element grid for channel flow with an internal blockage.



(a)



(b)

Figure 14. Channel flow with an internal blockage, velocity-pressure integrated method, (a) streamline, and (b) pressure.

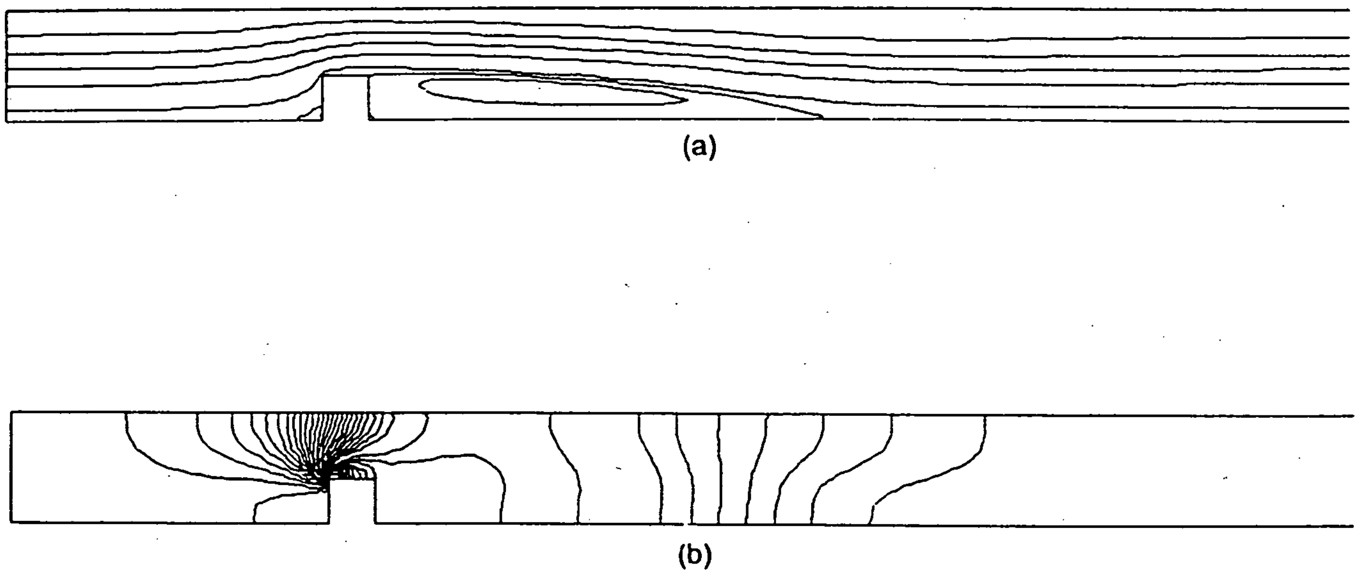


Figure 15. Channel flow with an internal blockage, penalty method with equation (15), (a) streamline, and (b) pressure.

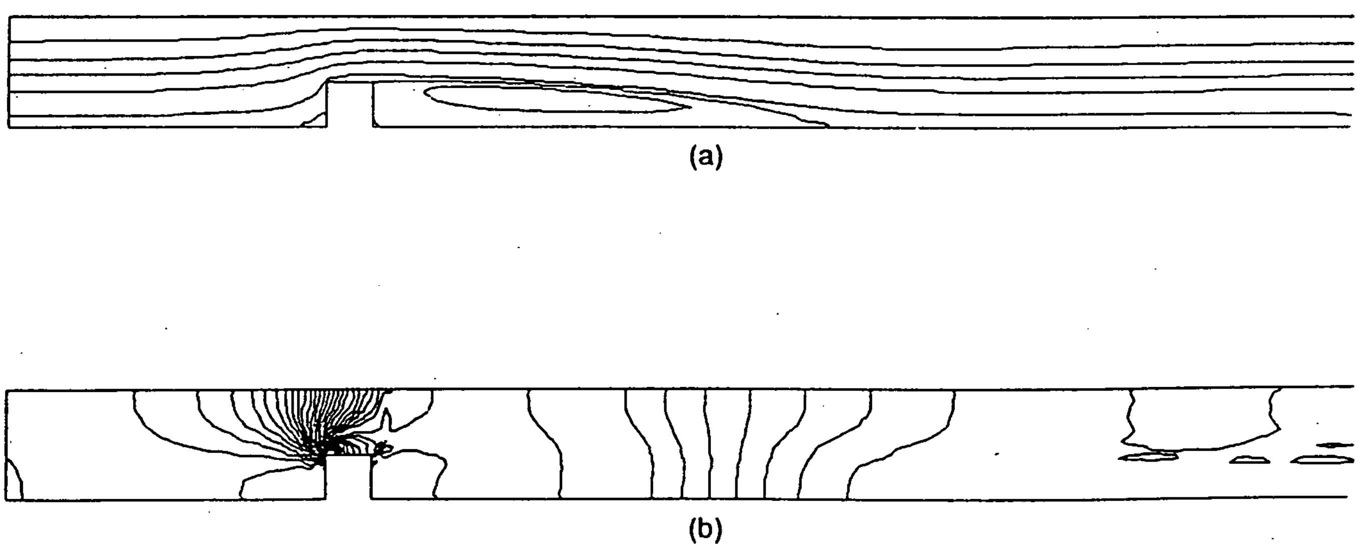


Figure 16. Channel flow with an internal blockage, penalty method with equation (16), (a) streamline, and (b) pressure.

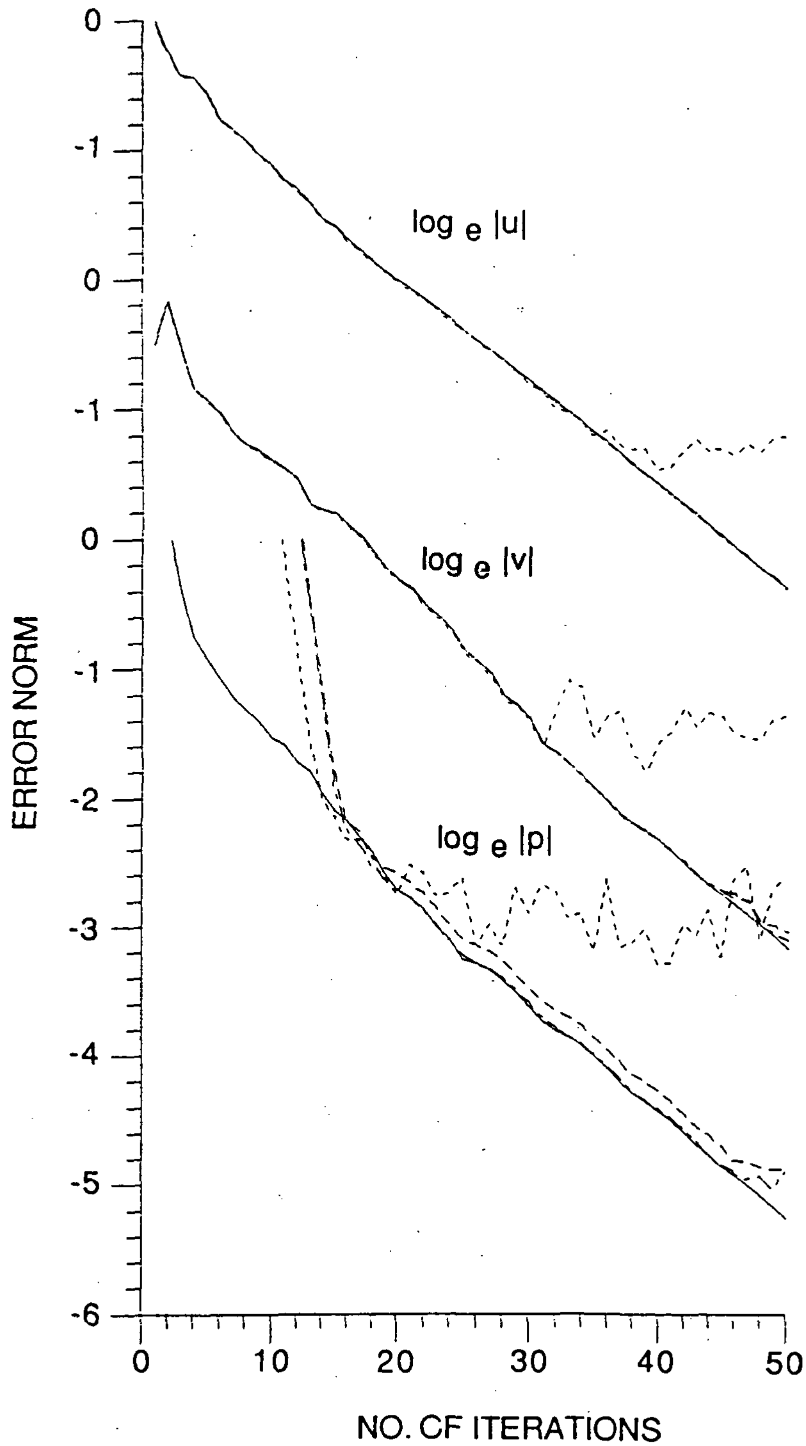


Figure 17. Error norm versus number of iterations for channel flow with an internal blockage, notations are the same as in Figure 3.

It can be seen in Figures 14 through 16 that there exists a steep pressure gradient in the forward corner region of the blockage. The same flow has been solved using four different, locally refined, coarse grids (Fig. 18). It can be seen in Figure 19(b) that the grid refinement in the local high Reynolds number region was not helpful to suppress the numerical wiggles. On the other hand, any grid refinement in the steep pressure gradient region suppressed the numerical wiggles significantly [Fig. 19(c)-(d)]. For this flow case, it can be concluded that the numerical wiggles have been caused by the coarse grids [Fig. 18(a)-(b), which could not resolve the steep pressure gradient in the forward corner region of the blockage. Note that the local Reynolds number in the high pressure gradient region is sufficiently small compared with that of the upstream region. Use of an upwinding technique has been partly justified based on the assumption that the high grid Reynolds number is responsible for numerical wiggles. However, these computational results suggest that the high grid Reynolds number was less responsible for the numerical wiggles than the steep gradient of a flow variable, which turned out to be the pressure for this flow case, in the forward corner region of the blockage.

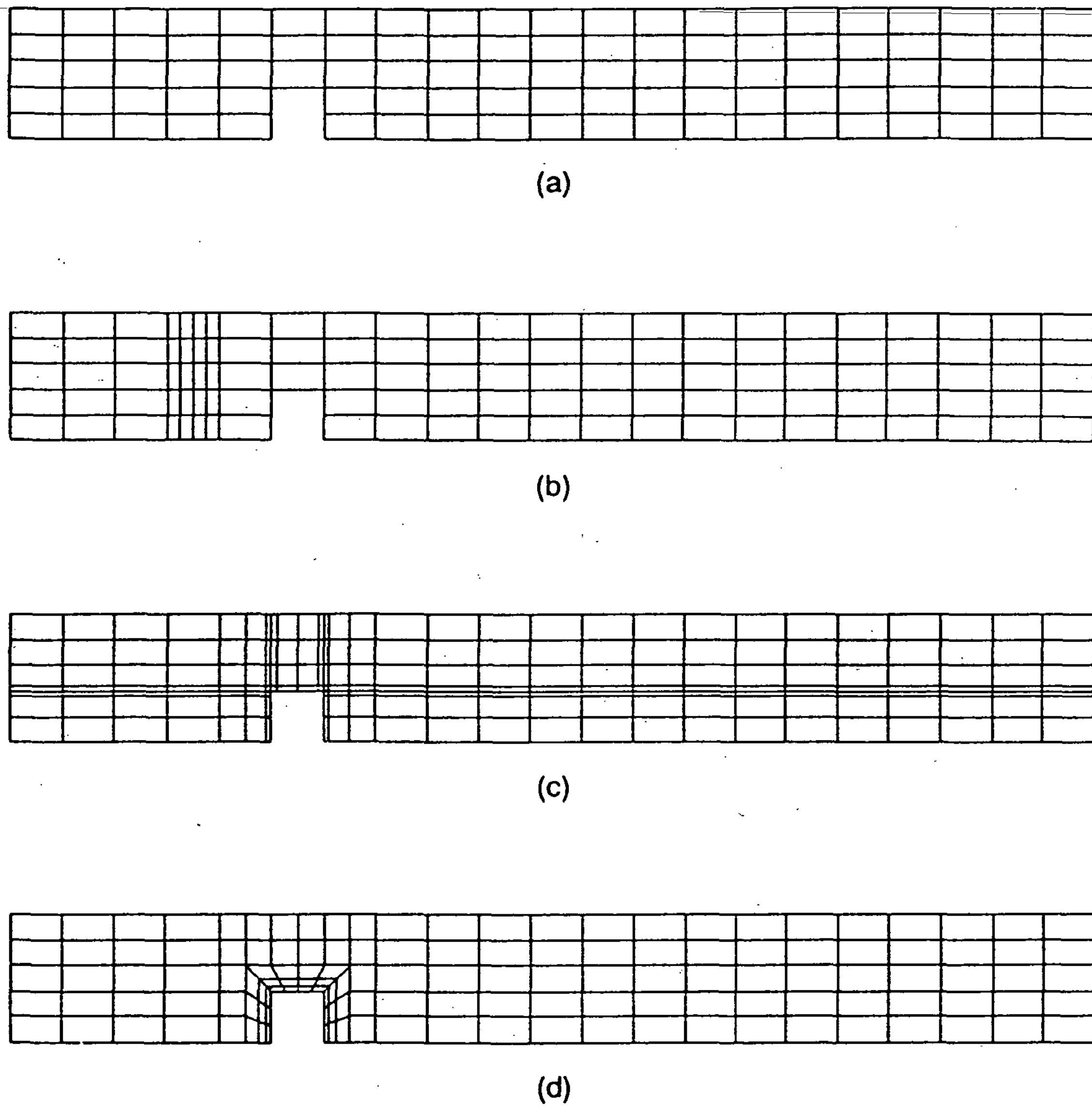


Figure 18. Local grid refinement for channel flow with an internal blockge.

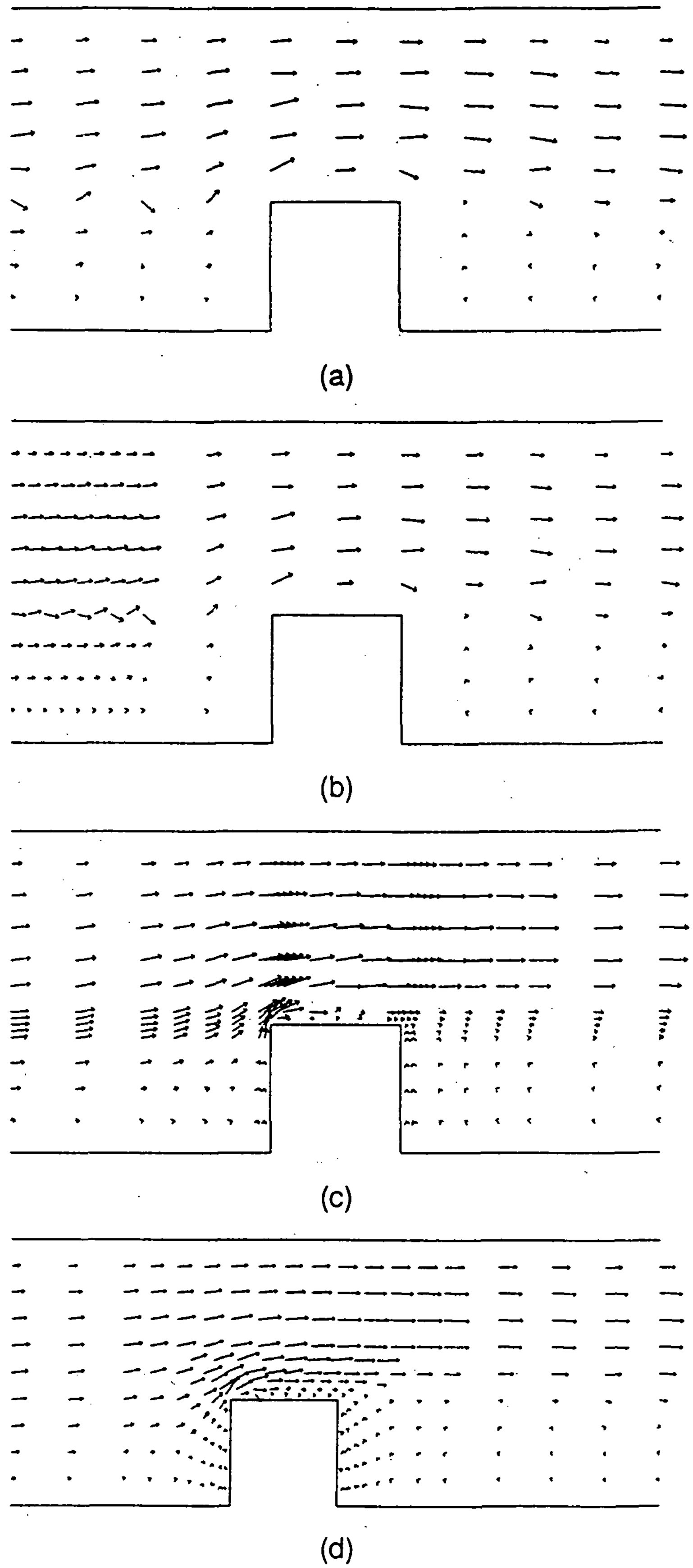


Figure 19. Velocity vectors for locally refined grids.

IV. CONCLUSIONS AND DISCUSSION

A comparative study of the velocity-pressure integrated finite element method and the consistent penalty finite element method has been presented. The penalty method with the pressure interpolation polynomials given as equations (15) and (17) yielded uniformly convergent solutions. The convergence rate was equal to that of the velocity-pressure integrated method. The penalty method with the pressure interpolation polynomials of the form (1,x,y) exhibited slightly degenerated convergence rate.

It was found that all of the methods yielded almost identical computational results for the velocity. However, the pressure interpolation polynomials of the form (1,x,y) yielded severely distorted pressure contours for the example flow cases. The distorted pressure contours had been caused by the ill-conditioned matrix of the discrete penalized conservation of mass equation. The penalty methods required slightly smaller computational time than the velocity-pressure integrated method. However, the difference was insignificant. The velocity-pressure integrated method would be preferable over the penalty methods, for its uniform convergence behavior for pressure.

It has been shown that any of the finite element methods considered in this report could capture the subtle pressure driven recirculation zones for high Reynolds number flows. The computational results compared favorably with experimental data and/or fine grid finite difference computational results.

For the example problems considered herein, a relatively small number of grid points, compared with the fine grid finite difference computations of the same example flows, were required to resolve the details of the flow field. It was found that no upwinding technique was necessary to obtain computational results which were free of numerical wiggles for high Reynolds number flows.

REFERENCES

1. Taylor, C., and Hughes, T. G.: Finite Element Programming of the Navier-Stokes Equation. Pineridge Press, Swansea, U.K., 1980.

2. S.-W. Kim: A Fine Grid Finite Element Computation of Two-Dimensional High Reynolds Number Flows. To appear in Computers and Fluids, 1987. Also available as: A Finite Element Computational Method for High Reynolds Number Flows, NASA CR-179135, 1987.

3. Zienkiewicz, O. C., Taylor, R. L., and Baynham, J. M. W.: Mixed and Irreducible Formulations in Finite Element Analysis. Eds. Atluri, S. N., et al. Hybrid and Mixed Finite Element Methods, J. Wiley and Sons, New York, 1983.

4. Engelman, M. S., Sani, R. L., Gresho, P. M., and Bercovier, M.: Consistent versus Reduced Integration Penalty Methods for Incompressible Media Using Several Old and New Elements. Int. J. Nume. Meth. Fluids, Vol. 2, 1982, pp. 25-42.

5. Kikuchi, N., Oden, J. T., and Song, Y. J.: Convergence of Modified Penalty Methods and Smoothing Schemes of Pressure for Stokes' Flow Problems. Finite Elements in Fluids, Vol. 5, Eds. Gallagher, R. H., et a., J. Wiley and Sons, New York, 1984, pp. 107-126.

6. Comini, G., and Giudice, S. D.: Finite Element Solution of the Incompressible Navier-Stokes Equations. Numerical Heat Transfer, Vol. 5, 1986, pp. 223-237.

7. Benim, A. C., and Zinser, W.: A Segregated Formulation of Navier-Stokes Equations with Finite Elements. Comput. Meth. Appl. Mech. Engrg., Vol. 57, 1986, pp. 223-237.

8. Rice, J. G., and Schnipke, R. J.: An Equal-Order Velocity-Pressure Formulation That Does Not Exhibit Spurious Pressure Modes. Comput. Meth. Appl. Mech. Engrg., Vol. 58, 1986, pp. 135-149.

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

10. Irons, B., and Ahmad, S.: Techniques of Finite Elements. J. Wiley and Sons, New York, 1980.

11. Kim, S.-W., and Chen, C.-P.: A Multiple-Time-Scale Turbulence Model Based on Variable Partitioning of the Turbulent Kinetic Energy Spectrum. AIAA Paper 88-0221, 1988.

12. Kim, S.-W., and Chen, Y.-S.: A Finite Element Computation of Turbulent Boundary Layer Flows with an Algebraic Stress Turbulence Model. Comput. Meth. Appl. Mech. Engrg., Vol. 66, 1988, pp. 45-63. Also available as NASA CR-178967, 1987.

13. Thomas, C. E., Morgan, K., and Taylor, C.: A Finite Element Analysis of Flow over a Backward-Facing Step. Computers and Fluids, Vol. 9, 1981, pp. 265-278.

14. Oden, J. T., and Reddy, J. N.: An Introduction to the Mathematical Theory of Finite Elements. J. Wiley and Sons, New York, 1976.

15. Gresho, P. M., Chan, S. T., Lee, R. L., and Upson, C. D.: A Modified Finite Element Method for Solving the Time-Dependent Incompressible Navier-Stokes Equations, Part 2. Applications. Int. J. Nume. Meth. Fluids, Vol. 4, 1984, pp. 619-640.

16. Dhatt, G. and Touzot, G.: The Finite Element Method Displayed. Translated by Cantin, G., J. Wiley and Sons, New York, 1984.

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

18. Schreiber, R., and Keller, H. B.: Driven Cavity Flows by Efficient Numerical Techniques. J. Comput. Physics, Vol. 49, 1983, pp. 310-333.

19. Ghia, U., Ghia, K. N., and Shin, C. T.: High-Re Solutions for Incompressible Flow Using the Navier-Stokes Equations and a Multigrid Method. J. Comput. Physics, Vol. 48, 1982, pp. 387-411.

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

21. Bercovier, M., and Engelman, M.: A Finite Element for Numerical Solution of Viscous Incompressible Flows. J. Comput. Physics, Vol. 30, 1979, pp. 181-201.

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

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

24. Morgan, K., Periaux, J., and Thomasset, F., Eds.: Analysis of Laminar Flow Over a Backward-Facing Step. A GAMM-Workshop, Friedr Vieweg & Sohn, Germany, 1984.

25. Kim, S.-W.: A Critical Evaluation of Various Methods for the Analysis of Flow-Solid Interaction in a Nest of Cylinders Subjected to Cross-Flows. NASA CR-178996, 1987.

26. Zienkiewicz, O. C., Loehner, R., Morgan, K., and Nakazawa, S.: Finite Elements in Fluid Mechanics - A Decade of Progress. In Gallagher, R. H., Oden, J. T., Zienkiewicz, O. C., Kawai, T., and Kawahara, M., Eds., Finite Elements in Fluids, Vol. 5, J. Wiley and Sons, New York, 1984, pp. 1-26.

27. Rogers, S. E., Kaul, U. K., and Kwak, D.: A Numerical Study of Three-Dimensional Incompressible Flow Around Multiple Posts. AIAA Paper 86-0353, 1986.

28. Brooks, A. N., and Hughes, T. J. R.: Streamline Upwind/Petrov-Galerkin Formulations for Convection Dominated Flows with Particular Emphasis on the Incompressivle Navier-Stokes Equation. Comput. Meth. Applied Mech. Engrg., Vol. 32, 1982, pp. 199-259.

29. Gresho, P. M., and Lee, R. L.: Don't Suppress the Wiggles - They are Telling You Something. Finite Element Methods for Convection Dominated Flows, AMD Vol. 34, ASME, New York, 1979, pp. 37-61.

APPENDIX I

FINITE ELEMENT COMPUTER PROGRAM (NSFLOW/P) FOR INCOMPRESSIBLE, LAMINAR FLOWS

PRECEDING PAGE BLANK NOT FILMED

```
******************************** TOP OF DATA ********************************
C
C********1*********2*********3*********4*********5*********6***
C     PROGRAM MAIN
C-X-  IMPLICIT REAL*8 (A-H,O-Z)
      CHARACTER*4 TITLE,IWORD
      DIMENSION TITLE(15),IWORD(10)
      DATA IWORD / 'INIT',  'PREP',  '****',  'PROC',  'CONT',
     -             '****',  '****',  '****',  '****',  'END '/
      DATA MAXNOD,MAXELM,MAXDOF,MXFRON /4227, 1027, 11527, 167/
C
  101 CONTINUE
      READ(5,501)  TITLE
      WRITE(6,601) TITLE
  501 FORMAT(20A4)
  601 FORMAT(/2X,20A4)
C
  102 CONTINUE
      READ(5,501)  TITLE
      WRITE(6,601) TITLE
      DO 103 K=1,10
      IF(TITLE(2).EQ.IWORD(K)) GO TO 105
  103 CONTINUE
  104 WRITE(6,602)
  602 FORMAT(2X,'TERMINATED IN SUB-PREP FOR INPUT DATA ERROR')
      STOP
C
  105 CONTINUE
      GO TO (1,2,3,4,5,  6,7,8,9,10), K
C
C     INITIALIZE DIMENSIONED VARIABLES
C
    1 CONTINUE
      CALL INITAL(MAXNOD,MAXELM,MAXDOF,MXFRON)
      GO TO 102
C
C     PREPARE INPUT DATA
C
    2 CONTINUE
      CALL PREP(MAXNOD,MAXELM,MAXDOF,MXFRON)
      GO TO 102
C
    3 CONTINUE
      GO TO 104
C
C     UNSTEADY FLOW SOLVER
C
    4 CONTINUE
      CALL PROCES(MAXNOD,MAXELM,MAXDOF,MXFRON)
      GO TO 101
C
    5 CONTINUE
      GO TO 101
C
```

```
    6 CONTINUE
    7 CONTINUE
    8 CONTINUE
    9 CONTINUE
   10 CONTINUE
      STOP
      END
C
C********1*********2*********3*********4*********5*********6***
      SUBROUTINE INITAL(MAXNOD,MAXELM,MAXDOF,MXFRON)
C-X-  IMPLICIT REAL*8 (A-H,O-Z)
      COMMON /CPRS/  PELEM(4,1027),PBCDAT,IPNOD(2),IPDOF
      COMMON /CGRID/ X(4227,3),NODES(27,1027)
      COMMON /CFLOW/ A(4227,10),ADBC(4227,10),IBCA(4227,10)
C
      DO 10 KDIM=1,3
      DO 10 KNODE=1,MAXNOD
      X(KNODE,KDIM)    = 0.
   10 CONTINUE
C
      DO 30 KELEM=1,MAXELM
      DO 30 KPE=1,27
      NODES(KPE,KELEM)=0
   30 CONTINUE
C
      DO 50 KPROB=1,10
      DO 50 KNODE=1,MAXNOD
      IBCA(KNODE,KPROB) = 0
      ADBC(KNODE,KPROB) = 0.
      A(KNODE,KPROB)= 0.
   50 CONTINUE
C
      RETURN
      END
C
C********1*********2*********3*********4*********5*********6
      BLOCK DATA BLKDAT
C-X-  IMPLICIT REAL*8 (A-H,O-Z)
      COMMON /CDESC/ NNODE,NELEM,NPE,NPRE,NDIM,NEDOF,IFLOW,
     -               IAXSY,IELF
      COMMON /CGAUL/ CLXKS(4,4),CLW(4,4),NGAUS
      COMMON /CGAUS/ EXKS(3,64),EW(64),MGAUS
      COMMON /CINDX/ INDXF(27,3,15),INDXP(27,15)
      COMMON /CITER/ CNVCF(10),ERROF(10),RELAX(10),ITERE,MAXIT
      COMMON /CLSCF/ XKSNOD(27,3,11),TM(4,4)
      COMMON /CMATE/ BFX(3),DENSY,VISCY,PECLET
      COMMON /CPNLT/ CXKPN(4,3,15),CWPN(4,15),PTNUM
      COMMON /CPROB/ IA(10),IPLOT
C
      DATA CLXKS(1,1)          /0./,
     -     CLW(1,1)            /2./,
     -     (CLXKS(K,2),K=1,2) /-0.5773502692, 0.5773502692/,
     -     (CLW(K,2),K=1,2)   / 1.,           1./,
     -     (CLXKS(K,3),K=1,3) /-0.7745966692, 0., 0.7745966692/,
```

```
     -      (CLW(K,3),K=1,3)/0.5555555556,0.8888888889,0.5555555556/,
     -      (CLXKS(K,4),K=1,4) /-0.8611363116, -0.3399810436,
     -                           0.3399810436,  0.8611363116/,
     -      (CLW(K,4),K=1,4)   / 0.3478548451,  0.6521451549,
     -                           0.6521451549,  0.3478548451/
C
      DATA (CXKPN(K,1,4),K=1,3) /0.,         -0.70710678, 0.70710678/,
     -      (CXKPN(K,2,4),K=1,3) /0.81649658,-0.40824829,-0.40824829/,
     -      (CWPN(K,4),K=1,3) /1.33333333, 1.33333333, 1.33333333/
C
      DATA (XKSNOD(K,1,6),K=1,4) / -1., 1., 1., -1./,
     -      (XKSNOD(K,2,6),K=1,4) / -1., -1., 1., 1./,
     -      (XKSNOD(K,1,8),K=1,9) / -1.,0.,1.,1.,1.,0.,-1.,-1.,0./,
     -      (XKSNOD(K,2,8),K=1,9) / -1.,-1.,-1.,0.,1.,1.,1.,0.,0./
C
C --- IFLOW=5 FOR 2-D CONSISTENT PENALTY METHOD ----
      DATA (INDXF(KPE,1,5),KPE=1,9) / 1, 3, 5, 7, 9,11,13,15,17/,
     -      (INDXF(KPE,2,5),KPE=1,9) / 2, 4, 6, 8,10,12,14,16,18/
C --- IFLOW=6 FOR 2-D IMPROVED CONSISTENT PENALTY METHOD ----
      DATA (INDXF(KPE,1,6),KPE=1,9) / 1, 3, 5, 7, 9,11,13,15,17/,
     -      (INDXF(KPE,2,6),KPE=1,9) / 2, 4, 6, 8,10,12,14,16,18/
C
      DATA ((TM(I,J),J=1,4),I=1,4) / 0.25, 0.25,  0.25, 0.25,
     -     -0.433012701,-0.433012701, 0.433012701, 0.433012701,
     -     -0.433012701, 0.433012701,-0.433012701, 0.433012701,
     -      0.75,        -0.75,        -0.75,        0.75/
C
      DATA IFLOW,IAXSY,IPLOT /3*0/,
     -      NPRE,MGAUS,NGAUS,MAXIT  /4*0/
      DATA VISCY,DENSY,PECLET /3*0./
      DATA XKAPA,EWALL,CMUF,TKRWAL,TKREXT,XMLTH /6*0./
      DATA (IA(K),K=1,10) /10*0/
      DATA ERROF/10*0./
      END
C
C********1*********2*********3*********4*********5*********6***
      SUBROUTINE DATLIB
C-X-  IMPLICIT REAL*8 (A-H,O-Z)
      COMMON /CDESC/ NNODE,NELEM,NPE,NPRE,NDIM,NEDOF,IFLOW,
     -               IAXSY,IELF
      COMMON /CGAUL/ CLXKS(4,4),CLW(4,4),NGAUS
      COMMON /CGAUS/ EXKS(3,64),EW(64),MGAUS
      COMMON /CPNLT/ CXKPN(4,3,15),CWPN(4,15),PTNUM
      COMMON /CPROB/ IA(10),IPLOT
      DIMENSION LIBELF(15),LIBELH(6),LIBNPE(11),LBNPRE(15)
C
      DATA (LIBELF(IFL),IFL=1,15) /0,0,0,0,8, 8,0,0,0,0, 0,0,0,0,0/,
     -      (LIBNPE(IEL),IEL=1,11) /0,0,0,0,0, 4,8,9,0,0, 0/,
     -      (LBNPRE(IFL),IFL=1,15) /0,0,0,0,3, 3,0,0,0,0,  0,0,0,0,0/
C
      IELF=LIBELF(IFLOW)
      NPRE=LBNPRE(IFLOW)
      NPE =LIBNPE(IELF)
C
```

```
      LGAUS=0
      MGAUS=NGAUS**NDIM
      GO TO (10,20,30), NDIM
   10 WRITE(6,601) NDIM
      STOP
  601 FORMAT(2X,'TERMINATED AT SUB-DATLIB    NDIM=',I2)
C
   20 CONTINUE
      DO 2 I=1,NGAUS
      DO 2 J=1,NGAUS
      LGAUS=LGAUS+1
      EXKS(1,LGAUS)=CLXKS(I,NGAUS)
      EXKS(2,LGAUS)=CLXKS(J,NGAUS)
      EW(LGAUS)    =CLW(I,NGAUS)*CLW(J,NGAUS)
    2 CONTINUE
      GO TO 100
C
   30 CONTINUE
      DO 3 I=1,NGAUS
      DO 3 J=1,NGAUS
      DO 3 K=1,NGAUS
      LGAUS=LGAUS+1
      EXKS(1,LGAUS)=CLXKS(I,NGAUS)
      EXKS(2,LGAUS)=CLXKS(J,NGAUS)
      EXKS(3,LGAUS)=CLXKS(K,NGAUS)
      EW(LGAUS)=CLW(I,NGAUS)*CLW(J,NGAUS)*CLW(K,NGAUS)
    3 CONTINUE
C
  100 CONTINUE
C
      RETURN
      END
C
C********1*********2*********3*********4*********5*********6***
      SUBROUTINE PREP(MAXNOD,MAXELM,MAXDOF,MXFRON)
C-X-  IMPLICIT REAL*8 (A-H,O-Z)
      CHARACTER*4    TITLE,ICNTRL,IWORD
      COMMON /CDESC/ NNODE,NELEM,NPE,NPRE,NDIM,NEDOF,IFLOW,
     -               IAXSY,IELF
      COMMON /CFLOW/ A(4227,10),ADBC(4227,10),IBCA(4227,10)
      COMMON /CFRON/ MFRONF
      COMMON /CGAUL/ CLXKS(4,4),CLW(4,4),NGAUS
      COMMON /CGAUS/ EXKS(3,64),EW(64),MGAUS
      COMMON /CGRID/ X(4227,3),NODES(27,1027)
      COMMON /CINDX/ INDXF(27,3,15),INDXP(27,15)
      COMMON /CITER/ CNVCF(10),ERROF(10),RELAX(10),ITERE,MAXIT
      COMMON /CMATE/ BFX(3),DENSY,VISCY,PECLET
      COMMON /CPNLT/ CXKPN(4,3,15),CWPN(4,15),FTNUM
      COMMON /CPROB/ IA(10),IPLOT
      COMMON /CPRS/  PELEM(4,1027),PBCDAT,IPNOD(2),IPDOF
      COMMON /CWIND/ WHI(27),DWHX(27,3)
      DIMENSION TITLE(15),IWORD(30)
      DATA IWORD / 'DESC',  'CNTL',  'ELEM',  'NODE',  'MATE',
                   '****',  '****',  'ITER',  '****',  '****',
```

```
     -              'IA01',  'IA02',  'IA03',  'IA04',  '****',
     -              '****',  '****',  '****',  '****',  '****',
     -              '****',  '****',  '****',  '****',  '****',
     -              '****',  'INCL',  '****',  '****',  'END '/
C
  101 CONTINUE
      READ(5,501)  ICNTRL,TITLE
      WRITE(6,601) ICNTRL,TITLE
  501 FORMAT(20A4)
  601 FORMAT(/2X,20A4)
C
      DO 102 K=1,30
      IF(ICNTRL.EQ.IWORD(K)) GO TO 105
  102 CONTINUE
  103 WRITE(6,602)
      WRITE(6,601) ICNTRL,TITLE
  602 FORMAT(2X,'TERMINATED IN SUB-PREP FOR INPUT DATA ERROR')
      STOP
C
  105 CONTINUE
      GO TO (1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,
     -       21,22,23,24,25,26,27,28,29,30), K
C
    1 CONTINUE
      READ(5,501)  TITLE
      WRITE(6,603) TITLE
      READ(5,*)    IFLOW,NDIM,NGAUS,MFRONF
      WRITE(6,605) IFLOW,NDIM,NGAUS,MFRONF
      IF(MFRONF.GT.MXFRON) GO TO 103
  603 FORMAT(2X,20A4)
  605 FORMAT(4X,'IFLOW=',I2,   2X,'NDIM =',I2,  2X,'NGAUS=',I2,
     -       2X,'MFRONF=',I4)
C
      CALL DATLIB
      WRITE(6,606) IELF,NPE,NPRE,MGAUS
  606 FORMAT(4X,'IELF=',I2,  2X,'NPE=',I2,  2X,'NPRE=',I2,
     -       2X,'MGAUS=',I2)
      GO TO 101
C
    2 CONTINUE
      READ(5,501)  TITLE
      WRITE(6,603) TITLE
      READ(5,*) NNODE,NELEM,IAXSY,IPLOT
      WRITE(6,610) NNODE,NELEM,IAXSY,IPLOT
      IF(NNODE.GT.MAXNOD.OR.NELEM.GT.MAXELM) THEN
        WRITE(6,611) NNODE,NELEM,MAXNOD,MAXELM
        STOP
      ENDIF
  610 FORMAT(2X,'NNODE=',I5, 2X,'NELEM=',I5, 2X,'IAXSY=',I2,
     -       2X,'IPLOT=',I2)
  611 FORMAT(2X,'TERMINATED IN SUB-PREP FOR NNODE=',I6,
     -       2X,'NELEM=',I6,  2X,'MAXNOD=',I6,  2X,'MAXELM=',I6)
      GO TO 101
C
```

```
    3 CONTINUE
      CALL RELEM(NODES,NELEM,NPE,MAXELM)
      GO TO 101
C
    4 CONTINUE
      CALL RNODE(X,NNODE,NPE,IELF,NDIM,MAXNOD)
      GO TO 101
C
    5 CONTINUE
      READ(5,501) TITLE
      WRITE(6,603) TITLE
      READ(5,*) VISCY,DENSY,(BFX(K),K=1,NDIM)
      WRITE(6,613) VISCY,DENSY,(BFX(K),K=1,3)
  613 FORMAT(2X,'VISCY=',E12.4,      2X,'DENSY=',E12.4,
     -       /2X,'BFX=',3E12.4)
      GO TO 101
C
    6 CONTINUE
      GO TO 101
C
    7 CONTINUE
      GO TO 101
C
    8 CONTINUE
      READ(5,501)  TITLE
      WRITE(6,603) TITLE
      READ(5,*) MAXIT,(RELAX(K),K=1,10),(CNVCF(K),K=1,10)
      WRITE(6,626) MAXIT,(RELAX(K),K=1,10),(CNVCF(K),K=1,10)
  626 FORMAT(2X,'MAXIT=',I5,
     -       /2X,'RELAX=',5E12.4,  /8X,5E12.4,
     -       /2X,'CNVCF=',5E12.4,  /8X,5E12.4)
      GO TO 101
C
    9 CONTINUE
      GO TO 101
C
   10 CONTINUE
      GO TO 101
C
   11 CONTINUE
      CALL RINIT(A(1,1),NNODE,MAXNOD)
      CALL RBC1(ADBC(1,1),IBCA(1,1),MAXNOD,NNODE)
      GO TO 101
C
   12 CONTINUE
      CALL RINIT(A(1,2),NNODE,MAXNOD)
      CALL RBC1(ADBC(1,2),IBCA(1,2),MAXNOD,NNODE)
      GO TO 101
C
   13 CONTINUE
      CALL RINIT(A(1,3),NNODE,MAXNOD)
      CALL RBC1(ADBC(1,3),IBCA(1,3),MAXNOD,NNODE)
      GO TO 101
C
```

```
C--------1---------2---------3---------4---------5---------6---
   14 CONTINUE
      READ(5,*)     PBCDAT,(IPNOD(K),K=1,2)
      WRITE(6,635) PBCDAT,(IPNOD(K),K=1,2)
  635 FORMAT(2X,'PRESSURE B.C. DATA  PBCDAT=',E12.4,
     -       2X,'IPNOD(1-2)=',2I7)
      GO TO 101
C--------1---------2---------3---------4---------5---------6
C
   15 CONTINUE
      GO TO 101
C
C--------1---------2---------3---------4---------5---------6
   16 CONTINUE
      GO TO 101
C
C--------1---------2---------3---------7---------5---------6
   17 CONTINUE
      GO TO 101
C
   18 CONTINUE
      GO TO 101
C
   19 CONTINUE
      GO TO 101
C
   20 CONTINUE
      GO TO 101
C
   21 CONTINUE
      GO TO 101
C
   22 CONTINUE
      GO TO 101
C
C     TURBULENCE DATA
C
   23 CONTINUE
      GO TO 101
C
   24 CONTINUE
      GO TO 101
C
   25 CONTINUE
      GO TO 101
C
   26 CONTINUE
      GO TO 101
C
C     INCLUDE RE-START DATA
C
   27 CONTINUE
      CALL FEMDAT(A,ADBC,X,PBCDAT,NODES,IBCA,IPNOD,NPE,NNODE,
     -            NELEM,MAXNOD,MAXELM)
```

```
      GO TO 101
C--------1---------2---------3---------4---------5---------6---
   28 CONTINUE
      GO TO 103
C--------1---------2---------3---------4---------5---------6---
   29 CONTINUE
      RETURN
C--------1---------2---------3---------4---------5---------6---
   30 CONTINUE
C
      PTNUM=(VISCY/DENSY)*1.E+10
C
      RETURN
      END
C
C********1*********2*********3*********4*********5*********6***
      SUBROUTINE RNODE(X,NNODE,NPE,IELF,NDIM,MAXNOD)
C-X-  IMPLICIT REAL*8 (A-H,O-Z)
      CHARACTER*4 TITLE
      DIMENSION X(MAXNOD,3),DELX(197,3),XNOD(27,3),NKS(3),CXKS(3),
     -          PHI(27),DPHI(27,3),WHI(27),DWHI(27,3),TITLE(15)
C
      READ(5,501) TITLE
      WRITE(6,601) TITLE
  501 FORMAT(20A4)
  601 FORMAT(2X,15A4)
C
      READ(5,*) NBLOC
      WRITE(6,605) NBLOC
  605 FORMAT(2X,'SUB-RNODE         NBLOC=',I2)
C
      DO 7000 IBLOC=1,NBLOC
      READ(5,501) TITLE
      WRITE(6,601) TITLE
      READ(5,*) METHOD
      WRITE(6,606) METHOD
  606 FORMAT(4X,'GRID GENERATION METHOD=',I3)
      GO TO (1000,2000) METHOD
C
 1000 CONTINUE
      READ(5,501) TITLE
      WRITE(6,601) TITLE
      READ(5,*) NODG1,INCRX,INCRY,INCRZ
      WRITE(6,640) NODG1,INCRX,INCRY,INCRZ
  640 FORMAT(4X,I5,  2X,I3,  2X,I3,  2X,I3)
      READ(5,501) TITLE
      WRITE(6,601) TITLE
      DO 10 KDIM=1,3
      READ(5,*) NDAT,(DELX(IKE,KDIM),IKE=1,NDAT)
      WRITE(6,642) NDAT,(DELX(IKE,KDIM),IKE=1,NDAT)
      NKS(KDIM)=NDAT
      IF(NDAT.GT.197) THEN
        WRITE(6,645)
        STOP
```

```
      ENDIF
   10 CONTINUE
  642 FORMAT(2X,'NDAT=',I5,  20(/4X,5F10.7))
  645 FORMAT(2X,'INPUT DATA ERROR FOR NDAT IN SUB-RNODE')
      LLINE=NKS(1)
      MLINE=NKS(2)
      NLINE=NKS(3)
C
      DO 15 KLINE=1,NLINE
      DO 15 JLINE=1,MLINE
      DO 15 ILINE=1,LLINE
      KNODE=NODG1+(ILINE-1)*INCRX+(JLINE-1)*INCRY+(KLINE-1)*INCRZ
      X(KNODE,1)=DELX(ILINE,1)
      X(KNODE,2)=DELX(JLINE,2)
      X(KNODE,3)=DELX(KLINE,3)
   15 CONTINUE
      GO TO 7000
C
 2000 CONTINUE
      READ(5,501) TITLE
      WRITE(6,601) TITLE
      READ(5,*) ((XNOD(KPE,KDIM),KDIM=1,NDIM),KPE=1,NPE)
      DO 22 KPE=1,NPE
      WRITE(6,610) KPE,(XNOD(KPE,KDIM),KDIM=1,NDIM)
   22 CONTINUE
  610 FORMAT(4X,'KPE=',I2, 2X,3E12.4)
      READ(5,501) TITLE
      WRITE(6,601) TITLE
      READ(5,*) NODG1,INCRX,INCRY,INCRZ
      WRITE(6,612) NODG1,INCRX,INCRY,INCRZ
  612 FORMAT(4X,I5,  2X,I3,  2X,I3,  2X,I3)
      READ(5,501) TITLE
      WRITE(6,601) TITLE
      DO 25 KDIM=1,3
      READ(5,*) NDAT,(DELX(IKE,KDIM),IKE=1,NDAT)
      WRITE(6,614) NDAT,(DELX(IKE,KDIM),IKE=1,NDAT)
      NKS(KDIM)=NDAT
      IF(NDAT.GT.197) THEN
        WRITE(6,620)
        STOP
      ENDIF
   25 CONTINUE
  614 FORMAT(2X,'NDAT=',I5,  10(/4X,10F5.2))
  620 FORMAT(2X,'INPUT DATA ERROR FOR NDAT IN SUB-RNODE')
      LLINE=NKS(1)
      MLINE=NKS(2)
      NLINE=NKS(3)
C
      DO 45 KLINE=1,NLINE
      DO 45 JLINE=1,MLINE
      DO 45 ILINE=1,LLINE
      CXKS(1)=DELX(ILINE,1)
      CXKS(2)=DELX(JLINE,2)
      CXKS(3)=DELX(KLINE,3)
```

```
C
      GO TO (31,31,31,31,31,  31,31,38,31,31,  31), IELF
   31 CONTINUE
      WRITE(6,630) IELF
  630 FORMAT(2X,'SUB-RNODE    IELF =',I2)
      STOP
C
   38 CALL SHAP23(PHI,DPHI,CXKS,NPE)
C
   42 CONTINUE
      KNODE=NODG1+(ILINE-1)*INCRX+(JLINE-1)*INCRY+(KLINE-1)*INCRZ
      DO 44 KDIM=1,NDIM
      X(KNODE,KDIM)=0.
      DO 44 KPE=1,NPE
      X(KNODE,KDIM)=X(KNODE,KDIM)+XNOD(KPE,KDIM)*PHI(KPE)
   44 CONTINUE
   45 CONTINUE
C
 7000 CONTINUE
      RETURN
      END
C
C********1*********2*********3*********4*********5*********6***
      SUBROUTINE RELEM(NODES,NELEM,NPE,MAXELM)
C-X-  IMPLICIT REAL*8 (A-H,O-Z)
      CHARACTER*4 TITLE
      DIMENSION NODES(27,MAXELM),NEL(3),INCREL(3),INCNOD(27,3),
     -          TITLE(15)
C
      DO 1 KELEM=1,NELEM
      DO 1 KPE=1,NPE
      NODES(KPE,KELEM)=0
    1 CONTINUE
C
      READ(5,501) TITLE
      WRITE(6,601) TITLE
  501 FORMAT(20A4)
  601 FORMAT(2X,15A4)
      READ(5,*) NBLOC
      WRITE(6,610) NBLOC
  610 FORMAT(4X,'NBLOC=',I6)
C
      DO 100 IBLOC=1,NBLOC
      READ(5,501) TITLE
      WRITE(6,601) TITLE
      READ(5,*)    IEL1,(NODES(IPE,IEL1),IPE=1,NPE)
      WRITE(6,620) IEL1,(NODES(IPE,IEL1),IPE=1,NPE)
  620 FORMAT(4X,'IEL1=',I6, 2X,'NODES(KPE,IEL1)=',8I6,
     -       3(/15X,'NODES(KPE,IEL1)=',8I6))
      READ(5,501) TITLE
      WRITE(6,601) TITLE
      DO 10 KDIM=1,3
      READ(5,*) NEL(KDIM),INCREL(KDIM),(INCNOD(K,KDIM),K=1,NPE)
   10 CONTINUE
```

```
      NELX=NEL(1)
      NELY=NEL(2)
      NELZ=NEL(3)
C
      DO 50 IELZ=1,NELZ
      DO 50 IELY=1,NELY
      DO 50 IELX=1,NELX
      KELEM=IEL1+(IELX-1)*INCREL(1)+(IELY-1)*INCREL(2)
     -                                +(IELZ-1)*INCREL(3)
      DO 30 KPE=1,NPE
      NODES(KPE,KELEM)=NODES(KPE,IEL1)+(IELX-1)*INCNOD(KPE,1)
     -          +(IELY-1)*INCNOD(KPE,2)+(IELZ-1)*INCNOD(KPE,3)
   30 CONTINUE
   50 CONTINUE
C
  100 CONTINUE
C
      RETURN
      END
C
C********1*********2*********3*********4*********5*********6***
      SUBROUTINE RINIT(AINIT,NNODE,MAXNOD)
C-X-  IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION AINIT(MAXNOD),TITLE(15)
C
      READ(5,501) TITLE
      WRITE(6,601) TITLE
  501 FORMAT(15A4)
  601 FORMAT(/2X,15A4)
      READ(5,*) NREC
      WRITE(6,610) NREC
      IF(NREC.LE.0) RETURN
  610 FORMAT(2X,'SUB-RINIT      NREC=',I5)
C
      DO 20 IREC=1,NREC
      READ(5,*) N1,N2,INCNOD,ADATA
      WRITE(6,620) N1,N2,INCNOD,ADATA
  620 FORMAT(5X,'N1=',I6,  5X,'N2=',I6,   5X,'INCNOD=',I6,
     -          5X,'ADATA=',E12.4)
      DO 10 N=N1,N2,INCNOD
      AINIT(N)=ADATA
   10 CONTINUE
   20 CONTINUE
C
      RETURN
      END
C
C********1*********2*********3*********4*********5*********6***
      SUBROUTINE RBC1(DBCH,LDBC,MAXNOD,NNODE)
C-X-  IMPLICIT REAL*8 (A-H,O-Z)
      CHARACTER*4 TITLE
      DIMENSION DBCH(MAXNOD),LDBC(MAXNOD),TITLE(15)
C
      DO 10 KNODE=1,NNODE
```

```
      LDBC(KNODE)=0
      DBCH(KNODE)=0.
   10 CONTINUE
C
C     DBC DATA
C
      READ(5,501) TITLE
      WRITE(6,601) TITLE
      READ(5,*) NREC
      WRITE(6,602) NREC
      IF(NREC.EQ.0) RETURN
C
      WRITE(6,603)
      DO 40 IREC=1,NREC
      READ(5,*) N1,N2,INCR,DUM
      WRITE(6,604) N1,N2,INCR,DUM
      DO 30 K=N1,N2,INCR
      LDBC(K)=1
      DBCH(K)=DUM
   30 CONTINUE
   40 CONTINUE
C
      WRITE(6,607)
      DO 60 KNODE=1,NNODE
      IF(LDBC(KNODE).NE.0) WRITE(6,605) KNODE,LDBC(KNODE),
     -                                  DBCH(KNODE)
   60 CONTINUE
C
      RETURN
  501 FORMAT(20A4)
  601 FORMAT(/2X,20A4)
  602 FORMAT(5X,'NO. OF INPUT DATA RECORD FOR DBC, NREC=',I5)
  603 FORMAT(5X,'  N1-NODE    N2-NODE   INCREMENT    DBC-DATA')
  604 FORMAT(5X,I5,5X,I5,5X,I5,5X,E11.4)
  605 FORMAT(5X,'NODE=',I5,5X,'LDBC=',I3,5X,'DATDBC=',E11.4)
  607 FORMAT(/2X,'LIST OF D.B.C. DATA FROM SUB-RBC1')
      END
C
C********1*********2*********3*********4*********5*********6***
      SUBROUTINE FEMDAT(A,ADBC,X,PBCDAT,NODES,IBCA,IPNOD,NPE,
     -                  NNODE,NELEM,MAXNOD,MAXELM)
C-X-  IMPLICIT REAL*8 (A-H,O-Z)
      CHARACTER*4 TITLE
      DIMENSION A(MAXNOD,10),ADBC(MAXNOD,10),X(MAXNOD,3),
     -          NODES(27,MAXELM),IBCA(MAXNOD,10),IPNOD(2),TITLE(15)
C
      READ(4,501) TITLE
      DO 50 KNOD=1,NNODE
      READ(4,*) KNODE,(X(KNODE,KDUM),KDUM=1,3)
   50 CONTINUE
  501 FORMAT(15A4)
C
      READ(4,501) TITLE
      DO 52 KEL=1,NELEM
```

```
      READ(4,*) KELEM,(NODES(KPE,KELEM),KPE=1,NPE)
   52 CONTINUE
C
      READ(4,501) TITLE
      DO 54 KNOD=1,NNODE
      READ(4,*) KNODE,(IBCA(KNODE,K),K=1,3)
   54 CONTINUE
      READ(4,501) TITLE
      READ(4,*) PBCDAT,(IPNOD(K),K=1,2)
C
      READ(4,501) TITLE
      DO 55 KNOD=1,NNODE
      READ(4,*) KNODE,(A(KNODE,K),K=1,4)
   55 CONTINUE
C
      READ(4,501) TITLE
      DO 57 KNOD=1,NNODE
      READ(4,*) KNODE,(ADBC(KNODE,K),K=1,3)
   57 CONTINUE
C
      DO 60 KNODE=1,NNODE
      A(KNODE,4)=0.
   60 CONTINUE
C
      RETURN
      END
C
C********1*********2*********3*********4*********5*********6
      SUBROUTINE ISOPEL(IFLOW,IELF,IAXSY,NPE,NPRE,NDIM)
C-X-  IMPLICIT REAL*8 (A-H,O-Z)
      COMMON /CELEM/ EX(27,3),EA(27,10),ENUT(64),ENUT1(8),
     -               NODEL(27),IELEM
      COMMON /CWIND/ WHI(27),DWHX(27,3)
      COMMON /ESHAP/ PHI(27),DPHX(27,3),PSI(27),DXDK(3,3),
     -        DKDX(3,3),CXKS(3),DETJB,RADUS,IGAUS,JGAUS,LGAUS
      DIMENSION DPHI(27,3),DPSI(27,3),XGS(3)
C
      GO TO (1,1,1,1,1,  1,1,8,1,1,  1), IELF
    1 CONTINUE
      WRITE(6,608) IFLOW,IELF
  608 FORMAT(2X,'SUB-ISOPEL  IFLOW=',I2,  2X,'IELF=',I2)
      STOP
C
    8 CALL SHAP23(PHI,DPHI,CXKS,NPE)
C
      DO 25 IDIM=1,NDIM
      DO 25 ICE=1,NDIM
      DXDK(IDIM,ICE)=0.
      DO 24 KPE=1,NPE
      DXDK(IDIM,ICE) = DXDK(IDIM,ICE)
     -               + EX(KPE,IDIM)*DPHI(KPE,ICE)
   24 CONTINUE
   25 CONTINUE
C
```

```
      GO TO (31,32,33), NDIM
   31 CONTINUE
      DETJB=DXDK(1,1)
      GO TO 37
   32 CONTINUE
      DETJB=DXDK(1,1)*DXDK(2,2)-DXDK(1,2)*DXDK(2,1)
      GO TO 37
   33 CONTINUE
      DETJB =  DXDK(1,1)*DXDK(2,2)*DXDK(3,3)
     -       + DXDK(1,2)*DXDK(2,3)*DXDK(3,1)
     -       + DXDK(2,1)*DXDK(3,2)*DXDK(1,3)
     -       - DXDK(1,1)*DXDK(2,3)*DXDK(3,2)
     -       - DXDK(2,2)*DXDK(1,3)*DXDK(3,1)
     -       - DXDK(3,3)*DXDK(2,1)*DXDK(1,2)
C
   37 CONTINUE
      IF(DETJB.LE.1.E-15) THEN
         WRITE(6,610) IELEM,IELF,NPE,NPRE
         DO 17 KPE=1,NPE
   17    WRITE(6,615) KPE,NODEL(KPE),(EX(KPE,IDIM),IDIM=1,NDIM)
         WRITE(6,620) (PHI(K),K=1,NPE)
         DO 18 IDIM=1,NDIM
         WRITE(6,625) (DPHI(K,IDIM),K=1,NPE)
   18    CONTINUE
         WRITE(6,630) DETJB,((DXDK(I,J),J=1,NDIM),I=1,NDIM)
         STOP
      ENDIF
  610 FORMAT(2X,'PROGRAM RUN TERMINATED AT SUB-ISOPEL DUE TO ',
     -            'SMALL DETJB',  /4X,'IELEM=',I5, 2X,'IELF=',I2,
     -         2X,'NPE=',I2,   2X,'NPRE=',I2)
  615 FORMAT(3X,'IPE =',I2, 2X,'INODE=',I5, 2X,'XDAT=',3E12.4)
  620 FORMAT(4X,'PHI =',5F10.4,   5(/5X,'PHI =',5F10.4))
  625 FORMAT(4X,'DPHI=',5F10.4,   5(/4X,'DPHI=',5F10.4))
  630 FORMAT(2X,'DETJB=',E11.4,/2X,'DXDK=',3(/5X,3E12.4))
C
      GO TO (41,42,43), NDIM
   41 CONTINUE
      DKDX(1,1)=1./DETJB
      GO TO 45
   42 CONTINUE
      DKDX(1,1)=DXDK(2,2)/DETJB
      DKDX(1,2)=-DXDK(1,2)/DETJB
      DKDX(2,1)=-DXDK(2,1)/DETJB
      DKDX(2,2)= DXDK(1,1)/DETJB
      GO TO 45
C
   43 CONTINUE
      WRITE(6,635) NDIM
      STOP
  635 FORMAT(2X,'TERMINATED IN SUB-ISOPEL FOR NDIM=',I3)
C
   45 CONTINUE
C
C     CALCULATE GLOBAL DERIVATIVES
```

```
C
      DO 47 IDIM=1,NDIM
      DO 47 IPE=1,NPE
      DPHX(IPE,IDIM)=0.0
      DO 47 ICE=1,NDIM
      DPHX(IPE,IDIM) = DPHX(IPE,IDIM)
     -                 + DPHI(IPE,ICE)*DKDX(ICE,IDIM)
   47 CONTINUE
C
      GO TO (50,50,50,50,52,   53,50,50,50,50,   50,50,50,50,50),
     -       IFLOW
   50 CONTINUE
      WRITE(6,650) IFLOW
      STOP
  650 FORMAT(2X,'TERMINATED AT SUB-ISOPEL FOR IFLOW=',I5)
C
   51 CALL SHAP01(PSI,DPSI,CXKS,NPRE)
      GO TO 85
C
   52 CONTINUE
      DO 61 KDIM=1,NDIM
      XGS(KDIM)=0.
      DO 61 KPE=1,NPE
      XGS(KDIM)=XGS(KDIM)+EX(KPE,KDIM)*PHI(KPE)
   61 CONTINUE
      PSI(1)=1.
      DO 62 KDIM=1,NDIM
   62 PSI(KDIM+1)=XGS(KDIM)
      GO TO 85
C
   53 CALL SHAP02(PSI,DPSI,CXKS,NPRE)
C
   85 CONTINUE
      IF(IAXSY.EQ.1) THEN
        RADUS=0.
        DO 90 KPE=1,NPE
        RADUS=RADUS+EX(KPE,2)*PHI(KPE)
   90   CONTINUE
      ENDIF
C
      RETURN
      END
C
C********1*********2*********3*********4*********5*********6
      SUBROUTINE LSHP1(PLK,DPLK,S)
C-X-  IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION PLK(3),DPLK(3)
C
      PLK(1)=(1.-S)/2.
      PLK(2)=(1.+S)/2.
      DPLK(1)=-0.5
      DPLK(2)= 0.5
      RETURN
      END
```

```
C
C*********1*********2*********3*********4*********5*********6
      SUBROUTINE LSHP2(PNK,DPNK,S)
C-X-  IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION PNK(3),DPNK(3)
C
C     1-D QUADRATIC ELEMENT  (NE= 1    2    3)
C                            (S =-1.  0.   1.)
      PNK(1)=S*(S-1.)/2.
      PNK(2)=1.-S**2
      PNK(3)=S*(1.+S)/2.
C
      DPNK(1)=S-0.5
      DPNK(2)=-2.*S
      DPNK(3)=S+0.5
      RETURN
      END
C
C*********1*********2*********3*********4*********5*********6***
      SUBROUTINE SHAP01(SHP,DSHP,CXKS,NPE)
C-X-  IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION SHP(27),DSHP(27,3),CXKS(3)
C
      SHP(1)=1.
      RETURN
      END
C
C*********1*********2*********3*********4*********5*********6***
      SUBROUTINE SHAP02(SHP,DSHP,CXKS,NPE)
C-X-  IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION SHP(27),DSHP(27,3),CXKS(3)
      SHP(1)=0.333333333+0.816496582*CXKS(2)
      SHP(2)=0.333333333-0.707106781*CXKS(1)-0.408248291*CXKS(2)
      SHP(3)=0.333333333+0.707106781*CXKS(1)-0.408248291*CXKS(2)
      RETURN
      END
C
C*********1*********2*********3*********4*********5*********6***
      SUBROUTINE SHAP23(SHP,DSHP,CXKS,NPE)
C-X-  IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION SHP(27),DSHP(27,3),PNK(3),
     -          DPNK(3),PNE(3),DPNE(3),INDK(9),INDE(9),CXKS(3)
      DATA (INDK(KPE),KPE=1,9) /1,2,3,  3,3,2,  1,1,2/,
     -     (INDE(KPE),KPE=1,9) /1,1,1,  2,3,3,  3,2,2/
C                                 7  6  5
C     9 NODE QUADRATIC ELEMENT    8  9  4
C                                 1  2  3
      CALL LSHP2(PNK,DPNK,CXKS(1))
      CALL LSHP2(PNE,DPNE,CXKS(2))
C
      DO 10 KPE=1,NPE
      IPE=INDK(KPE)
      JPE=INDE(KPE)
      SHP(KPE)=PNK(IPE)*PNE(JPE)
```

```
      DSHP(KPE,1)=DPNK(IPE)*PNE(JPE)
      DSHP(KPE,2)=PNK(IPE)*DPNE(JPE)
   10 CONTINUE
      RETURN
      END
C
C********1*********2*********3*********4*********5*********6
      SUBROUTINE PROCES(MAXNOD,MAXELM,MAXDOF,MXFRON)
C-X-  IMPLICIT REAL*8 (A-H,O-Z)
      COMMON /CDESC/ NNODE,NELEM,NPE,NPRE,NDIM,NEDOF,IFLOW,
     -               IAXSY,IELF
      COMMON /CFRON/ MFRONF
      COMMON /CGRID/ X(4227,3),NODES(27,1027)
      COMMON /CPROB/ IA(10),IPLOT
      COMMON /CFLOW/ A(4227,10),ADBC(4227,10),IBCA(4227,10)
C
      CALL PFRONT(NODES,NNODE,NELEM,NPE,MAXELM)
      CALL SHPLIB
C
      CALL SFLOW(MAXNOD,MAXELM,MAXDOF,MXFRON)
      CALL PFLOW(MAXNOD,MAXELM,MAXDOF)
      IF(IPLOT.GE.1) CALL PLSDAT(MAXNOD,MAXELM,MAXDOF)
C
      RETURN
      END
C
C********1*********2*********3*********4*********5*********6
      SUBROUTINE PFRONT(NODES,NNODE,NELEM,NPE,MAXELM)
C-X-  IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION NODES(27,MAXELM)
C
C     FIND LAST APPEARENCE OF EACH NODE AT FIRST ITERATION ONLY
C
      DO 30 INODE=1,NNODE
      LASTE=0
      DO 20 KELEM=1,NELEM
      DO 10 IPE=1,NPE
      INODP=ABS(NODES(IPE,KELEM))
      IF(INODP.NE.INODE) GO TO 10
      LASTE=KELEM
      LASTN=IPE
      GO TO 20
   10 CONTINUE
   20 CONTINUE
      NODES(LASTN,LASTE)=-INODE
   30 CONTINUE
C
      RETURN
      END
C
C********1*********2*********3*********4*********5*********6***
      SUBROUTINE SHPLIB
C-X-  IMPLICIT REAL*8 (A-H,O-Z)
      COMMON /CDESC/ NNODE,NELEM,NPE,NPRE,NDIM,NEDOF,IFLOW,
```

```
     -                IAXSY,IELF
      COMMON /CELEM/ EX(27,3),EA(27,10),ENUT(64),ENUT1(8),
     -                NODEL(27),IELEM
      COMMON /CGAUS/ EXKS(3,64),EW(64),MGAUS
      COMMON /CSHAP/ APHI(27,64),APHX(27,3,64),APSI(8,64),AREA(64),
     -                ARADUS(64)
      COMMON /ESHAP/ PHI(27),DPHX(27,3),PSI(27),DXDK(3,3),
     -         DKDX(3,3),CXKS(3),DETJB,RADUS,IGAUS,JGAUS,LGAUS
C
      REWIND 2
C
      DO 100 IELEM=1,NELEM
      CALL EXDAT
C
      DO 50 LGAUS=1,MGAUS
      DO 10 KDIM=1,NDIM
   10 CXKS(KDIM)=EXKS(KDIM,LGAUS)
      CALL ISOPEL(IFLOW,IELF,IAXSY,NPE,NPRE,NDIM)
      DO 30 KPE=1,NPE
      APHI(KPE,LGAUS)=PHI(KPE)
      DO 20 KDIM=1,NDIM
      APHX(KPE,KDIM,LGAUS)=DPHX(KPE,KDIM)
   20 CONTINUE
   30 CONTINUE
      IF(NPRE.GT.0) THEN
        DO 40 KPRE=1,NPRE
        APSI(KPRE,LGAUS)=PSI(KPRE)
   40   CONTINUE
      ENDIF
      AREA(LGAUS)=DETJB*EW(LGAUS)
      IF(IAXSY.EQ.1) AREA(LGAUS)=RADUS*AREA(LGAUS)
      ARADUS(LGAUS)=RADUS
   50 CONTINUE
C
      DO 90 L=1,MGAUS
      WRITE(2) (APHI(KPE,L),KPE=1,NPE),((APHX(KPE,K,L),KPE=1,NPE),
     -         K=1,NDIM),AREA(L)
      IF(NPRE.GT.0) WRITE(2) (APSI(KPRE,L),KPRE=1,NPRE)
   90 CONTINUE
      IF(IAXSY.EQ.1) WRITE(2) (ARADUS(LGAUS),LGAUS=1,MGAUS)
  100 CONTINUE
      RETURN
C
C--------1---------2---------3---------4---------5---------6---
      ENTRY SHPDAT
      DO 95 L=1,MGAUS
      READ(2) (APHI(KPE,L),KPE=1,NPE),((APHX(KPE,K,L),KPE=1,NPE),
     -         K=1,NDIM),AREA(L)
      IF(NPRE.GT.0) READ(2) (APSI(KPRE,L),KPRE=1,NPRE)
   95 CONTINUE
      IF(IAXSY.EQ.1) READ(2) (ARADUS(LGAUS),LGAUS=1,MGAUS)
      RETURN
      END
C
```

```
C********1*********2*********3*********4*********5*********6***
      SUBROUTINE S1FLOW(NODES,NNODE,NELEM,NPE,
     -                  NEDOF,NTDOF,IFLOW,MAXNOD,MAXELM,MAXDOF)
C-X-  IMPLICIT REAL*8 (A-H,O-Z)
      COMMON /CDOF/  A1(11527),IDBC(11527),LDOF(4227),L1DOF(4227)
      DIMENSION NODES(27,MAXELM),LIBDOF(15),LFLEL(27,15)
C
      DATA (LIBDOF(IFL),IFL=1,15) / 0, 0, 0, 0,18,  18, 0, 0, 0, 0,
     -                              0, 0, 0, 0, 0/
      DATA (LFLEL(KPE,5),KPE=1,9) /2,2,2,2,2,  2,2,2,2/,
     -     (LFLEL(KPE,6),KPE=1,9) /2,2,2,2,2,  2,2,2,2/
C
      DO 10 KELEM=1,NELEM
      DO 10 KPE  =1,NPE
      LDOF(ABS(NODES(KPE,KELEM))) = LFLEL(KPE,IFLOW)
   10 CONTINUE
C
      L1DOF(1)=1
      DO 30 INODE=2,NNODE
      L1DOF(INODE)=L1DOF(INODE-1)+LDOF(INODE-1)
   30 CONTINUE
      NEDOF=LIBDOF(IFLOW)
      NTDOF=L1DOF(NNODE)+LDOF(NNODE)-1
C
      IF(MAXDOF.GE.NTDOF) RETURN
      WRITE(6,610) IFLOW,IPROB,MAXDOF,NTDOF
      STOP
  610 FORMAT(2X,'TERMINATED IN SUB-S1FLOW FOR IFLOW=',I2,
     -       2X,'IPROB=',I2,  2X,'MAXDOF=',I6,  2X,'NTDOF=',I6)
      END
C
C********1*********2*********3*********4*********5*********6***
      SUBROUTINE SEQVFL(IFLOW,MFRON,NTDOF,NDBC,NDIM,NNODE,
     -                  MAXNOD,MAXELM,MAXDOF)
C-X-  IMPLICIT REAL*8 (A-H,O-Z)
      COMMON /CDOF/  A1(11527),IDBC(11527),LDOF(4227),L1DOF(4227)
      COMMON /CFLOW/ A(4227,10),ADBC(4227,10),IBCA(4227,10)
      COMMON /CFRON/ MFRONF
      COMMON /CINDX/ INDXF(27,3,15),INDXP(27,15)
      COMMON /CPRS/  PELEM(4,1027),PBCDAT,IPNOD(2),IPDOF
C
      MFRON=MFRONF
C
      DO 10 KDOF=1,NTDOF
      IDBC(KDOF)=0
      A1(KDOF)  =0.
   10 CONTINUE
C
      DO 20 KNODE=1,NNODE
      DO 15 KDIM=1,NDIM
      IF(IBCA(KNODE,KDIM).EQ.0) GO TO 15
      KDOF=L1DOF(KNODE)-1+KDIM
      IDBC(KDOF)= IBCA(KNODE,KDIM)
      A1(KDOF)  = ADBC(KNODE,KDIM)
```

```
   15 CONTINUE
   20 CONTINUE
C
      KDBC=0
      DO 30 IDOF=1,NTDOF
      IF(ABS(IDBC(IDOF)).NE.0) KDBC=KDBC+1
   30 CONTINUE
      NDBC=KDBC
C
      RETURN
      END
C
C********1*********2*********3*********4*********5*********6
      SUBROUTINE SFLOW(MAXNOD,MAXELM,MAXDOF,MXFRON)
C-X-  IMPLICIT REAL*8 (A-H,O-Z)
      COMMON /CDESC/ NNODE,NELEM,NPE,NPRE,NDIM,NEDOF,IFLOW,
     -               IAXSY,IELF
      COMMON /CFLOW/ A(4227,10),ADBC(4227,10),IBCA(4227,10)
      COMMON /CFRON/ MFRONF
      COMMON /CGRID/ X(4227,3),NODES(27,1027)
      COMMON /CITER/ CNVCF(10),ERROF(10),RELAX(10),ITERE,MAXIT
      COMMON /CPROB/ IA(10),IPLOT
C
      ITERE=0
 1001 CONTINUE
      ITERE=ITERE+1
      IF(ITERE.GT.MAXIT) GO TO 2001
C
      CALL FRONTS(NODES,IELIB,NNODE,NELEM,NEDOF,NPE,NPRE,
     -            NDIM,IFLOW,IAXSY,IELF,MXFRON,MAXNOD,MAXELM,
     -            MAXDOF)
C
      CALL SPRS4(A(1,4),IBCA(1,4),ERROF(4),MAXNOD,MAXELM,
     -           MAXDOF)
C
      DO 120 IPROB=1,3
      IF(ERROF(IPROB).GT.CNVCF(IPROB)) GO TO 1001
  120 CONTINUE
      IF(ERROF(4).GT.CNVCF(4)) GO TO 1001
C
 2001 CONTINUE
      CALL SPRS4(A(1,4),IBCA(1,4),ERROF(4),MAXNOD,MAXELM,
     -           MAXDOF)
      IF(ITERE.LE.MAXIT) RETURN
      CALL PLSDAT(MAXNOD,MAXELM,MAXDOF)
C
      WRITE(6,688) MAXIT,ITERE
  688 FORMAT(2X,'SOLUTION HAS FAILED TO CONVERGE',
     -      /4X,'MAXIT=',I5, 2X,'ITERE=',I5)
      STOP
      END
C
C********1*********2*********3*********4*********5*********6***
      SUBROUTINE FRONTS(NODES,IELIB,NNODE,NELEM,NEDOF,NPE,NPRE,
```

```
     -              NDIM,IFLOW,IAXSY,IELF,MXFRON,MAXNOD,MAXELM,
     -              MAXDOF)
C-X-  IMPLICIT REAL*8 (A-H,O-Z)
      COMMON /CDOF/  A1(11527),IDBC(11527),LDOF(4227),L1DOF(4227)
      COMMON /CELEM/ EX(27,3),EA(27,10),ENUT(64),ENUT1(8),
     -               NODEL(27),IELEM
      DIMENSION NODES(27,MAXELM),GK(167,167),GF(11527),
     -          PNORM(167),LHEAD(167),EK(85,85),EF(85),
     -          LOCEL(85),NDEST(85)
C
      CALL S1FLOW(NODES,NNODE,NELEM,NPE,
     -                  NEDOF,NTDOF,IFLOW,MAXNOD,MAXELM,MAXDOF)
      CALL SEQVFL(IFLOW,MFRON,NTDOF,NDBC,NDIM,NNODE,MAXNOD,
     -            MAXELM,MAXDOF)
C
      REWIND 1
      REWIND 2
C
C     INITIALIZE HEADING AND GRAND FLUID MATRIX
C
      NCRIT=MFRON-NEDOF
      NFRON=0
      DO 10 JFRON=1,MFRON
      DO 10 IFRON=1,MFRON
      GK(IFRON,JFRON)=0.
   10 CONTINUE
      DO 20 IDOF=1,MAXDOF
      GF(IDOF)=0.
   20 CONTINUE
C
      IELEM=0
   30 CONTINUE
      IELEM=IELEM+1
      CALL ELEMFL(EK,EF,NPE,NPRE,NDIM,NEDOF,IFLOW,
     -            IAXSY,IELF,MAXNOD,MAXELM,MAXDOF)
C
C     CREATE GLOBAL DOF ARRAY FOR EACH ELEMENT DOF
C
      IDOF=0
      DO 70 IPE=1,NPE
      INODE=NODES(IPE,IELEM)
      N1DOF=L1DOF(IABS(INODE))
      NDOF=LDOF(IABS(INODE))
      DO 70 KDOF=1,NDOF
      IDOF=IDOF+1
      LOCEL(IDOF)=N1DOF+KDOF-1
      IF(INODE.LT.0) LOCEL(IDOF)=-LOCEL(IDOF)
   70 CONTINUE
C
C     CONTRACT D.B.C. FOR ELEMENT SYSTEM OF EQUATIONS
C
      KDOF   = 0
      NEWDOF= NEDOF
      DO 90 KDUM=1,NEDOF
```

```
      KDOF = KDOF + 1
      IEQ  = ABS(LOCEL(KDOF))
      IF(IDBC(IEQ).EQ.0) GO TO 90
C
      IF(KDOF.EQ.1) GO TO 81
      DO 80 IDOF=1,KDOF-1
      EF(IDOF) = EF(IDOF)-EK(IDOF,KDOF)*A1(IEQ)
      IF(KDOF.EQ.NEWDOF) GO TO 80
      DO 71 JDOF=KDOF+1,NEWDOF
      EK(IDOF,JDOF-1)=EK(IDOF,JDOF)
   71 CONTINUE
   80 CONTINUE
C
   81 CONTINUE
      IF(KDOF.EQ.NEWDOF) GO TO 86
      DO 85 IDOF=KDOF+1,NEWDOF
      EF(IDOF-1) = EF(IDOF)-EK(IDOF,KDOF)*A1(IEQ)
      IF(KDOF.EQ.1) GO TO 83
      DO 82 JDOF=1,KDOF-1
      EK(IDOF-1,JDOF) = EK(IDOF,JDOF)
   82 CONTINUE
C
   83 CONTINUE
      DO 84 JDOF=KDOF+1,NEWDOF
      EK(IDOF-1,JDOF-1) = EK(IDOF,JDOF)
   84 CONTINUE
   85 CONTINUE
C
   86 CONTINUE
      DO 87 IDOF=1,NEWDOF
      EK(IDOF,NEWDOF) = 0.
      EK(NEWDOF,IDOF) = 0.
      EF(NEWDOF)      = 0.
   87 CONTINUE
      IF(KDOF.EQ.NEWDOF) GO TO 89
      DO 88 IDOF=KDOF+1,NEWDOF
      LOCEL(IDOF-1) = LOCEL(IDOF)
   88 CONTINUE
C
   89 CONTINUE
      KDOF   = KDOF   - 1
      NEWDOF = NEWDOF - 1
   90 CONTINUE
C
C     FIT EACH DOF INTO THE FRONT WIDTH EXTENDING IF NECESSARY
C
      DO 120 IDOF=1,NEWDOF
      IEQ=LOCEL(IDOF)
      IF(NFRON.EQ.0) GO TO 95
      DO 94 IFRON=1,NFRON
      KFRON=IFRON
      IF(IABS(IEQ).EQ.IABS(LHEAD(KFRON))) GO TO 110
   94 CONTINUE
   95 CONTINUE
```

```
C
      NFRON=NFRON+1
      IF(NFRON.LE.MFRON) GO TO 100
      WRITE(6,637) MXFRON,MFRON,NFRON,NCRIT,IELEM
      WRITE(6,638) (LHEAD(KFRON),KFRON=1,NFRON)
  637 FORMAT(/2X,'SUB-FRONTS --- FRONT WIDTH TO SMALL',
     - /4X,'MXFRON=',I5,  2X,'MFRON=',I5,  2X,'NFRON=',I5,
     - /4X,'NCRIT=',I5,   2X,'IELEM=',I5,
     - /2X,'LIST OF LHEAD DATA')
  638 FORMAT(2X,10I5)
      STOP
C
  100 CONTINUE
      NDEST(IDOF)=NFRON
      LHEAD(NFRON)=IEQ
      GO TO 120
  110 CONTINUE
      NDEST(IDOF)=KFRON
      LHEAD(KFRON)=IEQ
  120 CONTINUE
C
C     ASSEMBLE AN ELEMENT SYS. OF EQS. INTO A GLOBAL SYS. EQS.
C
      DO 130 IDOF=1,NEWDOF
      IEQ=ABS(LOCEL(IDOF))
      GF(IEQ)=GF(IEQ)+EF(IDOF)
      IFRON=NDEST(IDOF)
      DO 130 JDOF=1,NEWDOF
      JFRON=NDEST(JDOF)
      GK(JFRON,IFRON)=GK(JFRON,IFRON)+EK(JDOF,IDOF)
  130 CONTINUE
      IF(NFRON.LT.NCRIT.AND.IELEM.LT.NELEM) GO TO 30
C
C     CHECK THE LAST APPEARENCE OF EACH DOF
C
  140 CONTINUE
      PIVOT=0.0
      DO 170 IFRON=1,NFRON
      IF(LHEAD(IFRON).GE.0) GO TO 170
      PIVOG=GK(IFRON,IFRON)
      IF(ABS(PIVOG).LT.ABS(PIVOT)) GO TO 170
      PIVOT=PIVOG
      LPIVOT=IFRON
  170 CONTINUE
C
      IEQ=IABS(LHEAD(LPIVOT))
      IF(ABS(PIVOT).GT.1.E-10) GO TO 180
         WRITE(6,650) IPROB,IEQ,PIVOT,NCRIT,NFRON,IELEM
         DO 171 IEQ=1,NEWDOF
         WRITE(6,652) IEQ,EF(IEQ)
         WRITE(6,654) (EK(IEQ,JEQ),JEQ=1,NEWDOF)
  171    CONTINUE
         WRITE(6,656)
         WRITE(6,657) (NDEST(JDOF),JDOF=1,NEWDOF)
```

```
      WRITE(6,658) (LHEAD(IFRON),IFRON=1,NFRON)
      WRITE(6,659) LPIVOT,NFRON,GF(LPIVOT)
      WRITE(6,654) (GK(LPIVOT,JFRON),JFRON=1,NFRON)
      WRITE(6,654) (GK(IFRON,LPIVOT),IFRON=1,NFRON)
      STOP
  650 FORMAT(/2X,'PROGRAM TERMINATED --- ILL-CONDITIONED MATRIX',
     -        /4X,'IPROB=',I2,  2X,'IEQ=',I6,  2X,'PIVOT=',E12.4,
     -        /4X,'NCRIT=',I5,  2X,'NFRON=',I5,  2X,'IELEM=',I5,
     -        /4X,'CURRENT ELEMENT IN PROCESS    IELEM=',I5)
  652 FORMAT(4X,'IEQ=',I2,  2X,'EF(IEQ)=',E12.4, 2X,'EK-DATA')
  654 FORMAT(4X,5E12.4)
  656 FORMAT(2X,'CURRENT DATA IN THE GLOBAL MATRIX')
  657 FORMAT(2X,'NDEST-DATA',20(/4X,20I3))
  658 FORMAT(2X,'LHEAD-DATA',25(/4X,10I6))
  659 FORMAT(2X,'LPIVOT=',I6,  2X,'NFRON=',I5,  2X,'GF=',E12.4,
     -       /2X,'LIST OF PIVOTAL ROW AND COLUMN')
C
  180 CONTINUE
C
      DO 190 IFRON=1,NFRON
      PNORM(IFRON)=GK(LPIVOT,IFRON)/PIVOT
  190 CONTINUE
      RHSID=GF(IEQ)/PIVOT
      GF(IEQ)=RHSID
C
      IF(LPIVOT.EQ.1) GO TO 250
      DO 240 IFRON=1,LPIVOT-1
      FACTOR=GK(IFRON,LPIVOT)
C
C     UNDERFLOW MAY OCCUR IN THE FOLLOWING DO-200-LOOP IF
C     FACTOR IS SMALL. THE FOLLOWING STATEMENT NEED TO BE
C     CHANGED FOR DIFFERENT COMPUTERS.
C
      DO 200  JFRON=1,LPIVOT-1
      GK(IFRON,JFRON)=GK(IFRON,JFRON)-FACTOR*PNORM(JFRON)
  200 CONTINUE
C
  210 CONTINUE
      IF(LPIVOT.EQ.NFRON) GO TO 230
      DO 220 JFRON=LPIVOT+1,NFRON
      GK(IFRON,JFRON-1)=GK(IFRON,JFRON)-FACTOR*PNORM(JFRON)
  220 CONTINUE
  230 CONTINUE
      ITOTV=IABS(LHEAD(IFRON))
      GF(ITOTV)=GF(ITOTV)-FACTOR*RHSID
  240 CONTINUE
C
  250 CONTINUE
      IF(LPIVOT.EQ.NFRON) GO TO 300
      DO 290 IFRON=LPIVOT+1,NFRON
      FACTOR=GK(IFRON,LPIVOT)
      IF(LPIVOT.EQ.1) GO TO 270
      DO 260 JFRON=1,LPIVOT-1
      GK(IFRON-1,JFRON)=GK(IFRON,JFRON)-FACTOR*PNORM(JFRON)
```

```
  260 CONTINUE
  270 CONTINUE
      DO 280 JFRON=LPIVOT+1,NFRON
      GK(IFRON-1,JFRON-1)=GK(IFRON,JFRON)-FACTOR*PNORM(JFRON)
  280 CONTINUE
      ITOTV=IABS(LHEAD(IFRON))
      GF(ITOTV)=GF(ITOTV)-FACTOR*RHSID
  290 CONTINUE
  300 CONTINUE
C
C     WRITE OUT NON-FIXED PIVOTAL EQUATION ON TAPE
C
      WRITE(1) NFRON,LPIVOT,(LHEAD(IFRON),PNORM(IFRON),IFRON=1,NFRON)
C
      DO 320 IFRON=1,NFRON
      GK(IFRON,NFRON)=0.0
      GK(NFRON,IFRON)=0.0
  320 CONTINUE
      IF(LPIVOT.EQ.NFRON) GO TO 340
      DO 330 IFRON=LPIVOT,NFRON-1
      LHEAD(IFRON)=LHEAD(IFRON+1)
  330 CONTINUE
  340 CONTINUE
      NFRON=NFRON-1
C
C     ASSEMBLE, ELIMINATE, OR BACK-SUBSTITUTION
C
      IF(NFRON.GT.NCRIT) GO TO 140
      IF(IELEM.LT.NELEM) GO TO 30
      IF(NFRON.GT.0)     GO TO 140
C
C     BACK-SUBSTITUTION
C
      DO 370 ITOTV=1,NTDOF-NDBC
      BACKSPACE 1
      READ(1) NFRON,LPIVOT,(LHEAD(IFRON),PNORM(IFRON),IFRON=1,NFRON)
      IEQ=IABS(LHEAD(LPIVOT))
      TEMPR=0.0
      PNORM(LPIVOT)=0.0
      DO 360 IFRON=1,NFRON
      TEMPR=TEMPR-PNORM(IFRON)*A1(IABS(LHEAD(IFRON)))
  360 CONTINUE
      A1(IEQ)=GF(IEQ)+TEMPR
C
      BACKSPACE 1
  370 CONTINUE
C
      CALL SCNVFL(NODES,NNODE,NELEM,NPE,NPRE,NDIM,IFLOW,
     -                 MAXNOD,MAXELM,MAXDOF)
C
      RETURN
      END
C
C*********1*********2*********3*********4*********5*********6***
```

```
      SUBROUTINE ELEMFL(EK,EF,NPE,NPRE,NDIM, NEDOF,IFLOW,IAXSY,
     -                  IELF,MAXNOD,MAXELM,MAXDOF)
C-X-  IMPLICIT REAL*8 (A-H,O-Z)
      COMMON /CELEM/ EX(27,3),EA(27,10),ENUT(64),ENUT1(8),
     -               NODEL(27),IELEM
      COMMON /CGAUS/ EXKS(3,64),EW(64),MGAUS
      COMMON /CPNLT/ CXKPN(4,3,15),CWPN(4,15),PTNUM
      COMMON /CINDX/ INDXF(27,3,15),INDXP(27,15)
      COMMON /CMATE/ BFX(3),DENSY,VISCY,PECLET
      COMMON /CSHAP/ APHI(27,64),APHX(27,3,64),APSI(8,64),AREA(64),
     -               ARADUS(64)
      COMMON /CUSE2/ XK(3),XKN(3,3),XC(10),XB(3),XF(3),PROD,XWALL
      COMMON /CWIND/ WHI(27),DWHX(27,3)
      DIMENSION EK(85,85),EF(85),EQ(81,4),QE(4,81),EPM(4,4),
     -          EPINV(4,4),PIQE(4,81),IROW(3),JCOL(3),DNUM(3),
     -          DIFFU(3,3)
C
      DO 1 IDOF=1,NEDOF
      EF(IDOF)=0.
      DO 1 JDOF=1,NEDOF
      EK(IDOF,JDOF)=0.0
    1 CONTINUE
C
      DO 3 KDOF=1,NEDOF
      DO 3 KPRE=1,NPRE
      EQ(KDOF,KPRE)=0.
      QE(KPRE,KDOF)=0.
    3 CONTINUE
      DO 4 IPRE=1,NPRE
      DO 4 JPRE=1,NPRE
      EPM(IPRE,JPRE)=0.
    4 CONTINUE
C
      CALL ELMDAT
      CALL SHPDAT
C
      DO 1000 LGAUS=1,MGAUS
      XK(1)=VISCY
      DO 5 KDIM=1,NDIM
      XC(KDIM)=0.
      DO 5 KPE=1,NPE
      XC(KDIM)=XC(KDIM)+EA(KPE,KDIM)*APHI(KPE,LGAUS)
    5 CONTINUE
C
      DO 7 KPE=1,NPE
      WHI(KPE)=APHI(KPE,LGAUS)
    7 CONTINUE
      DO 10 KDIM=1,NDIM
      DO 10 KPE=1,NPE
      DWHX(KPE,KDIM)=APHX(KPE,KDIM,LGAUS)
   10 CONTINUE
C
      DO 30 IPE=1,NPE
      DO 11 KDIM=1,NDIM
```

```
      IROW(KDIM)=INDXF(IPE,KDIM,IFLOW)
      EF(IROW(KDIM))=EF(IROW(KDIM))+WHI(IPE)*BFX(KDIM)*AREA(LGAUS)
   11 CONTINUE
C
      DO 25 JPE=1,NPE
      DO 12 KDIM=1,NDIM
      JCOL(KDIM)=INDXF(JPE,KDIM,IFLOW)
   12 CONTINUE
C
        CONVC=0.
        DIFF=0.
        DO 15 IDIM=1,NDIM
        CONVC=CONVC+WHI(IPE)*DENSY*XC(IDIM)*APHX(JPE,IDIM,LGAUS)
     -             *AREA(LGAUS)
        DIFF =DIFF +DWHX(IPE,IDIM)*XK(1)*APHX(JPE,IDIM,LGAUS)
     -             *AREA(LGAUS)
        DO 14 JDIM=1,NDIM
        DIFFU(IDIM,JDIM)=DWHX(IPE,JDIM)*XK(1)*APHX(JPE,IDIM,LGAUS)
     -                  *AREA(LGAUS)
   14   CONTINUE
   15   CONTINUE
        DO 20 IDIM=1,NDIM
        EK(IROW(IDIM),JCOL(IDIM))=EK(IROW(IDIM),JCOL(IDIM))+CONVC
     -                           +DIFF
        DO 19 JDIM=1,NDIM
        EK(IROW(IDIM),JCOL(JDIM))=EK(IROW(IDIM),JCOL(JDIM))
     -                           +DIFFU(IDIM,JDIM)
   19   CONTINUE
   20   CONTINUE
C
      IF(IAXSY.EQ.1) THEN
        THETV =   2.*XK(1)*WHI(IPE)*APHI(JPE,LGAUS)
     -            /ARADUS(LGAUS)**2*AREA(LGAUS)
        EK(IROW(2),JCOL(2))=EK(IROW(2),JCOL(2))+THETV
      ENDIF
   25 CONTINUE
   30 CONTINUE
C
      DO 52 IPRE=1,NPRE
      DO 52 JPRE=1,NPRE
      EPM(IPRE,JPRE)=EPM(IPRE,JPRE)+APSI(IPRE,LGAUS)
     -                        *APSI(JPRE,LGAUS)*AREA(LGAUS)
   52 CONTINUE
      DO 53 IPE=1,NPE
      IROW(1)=INDXF(IPE,1,IFLOW)
      IROW(2)=INDXF(IPE,2,IFLOW)
      DO 53 JPRE=1,NPRE
      EQ(IROW(1),JPRE)=EQ(IROW(1),JPRE)+DWHX(IPE,1)
     -                     *APSI(JPRE,LGAUS)*AREA(LGAUS)
      EQ(IROW(2),JPRE)=EQ(IROW(2),JPRE)+DWHX(IPE,2)
     -                     *APSI(JPRE,LGAUS)*AREA(LGAUS)
      IF(IAXSY.EQ.1) EQ(IROW(2),JPRE)=EQ(IROW(2),JPRE)+WHI(IPE)
     -                  *APSI(JPRE,LGAUS)/ARADUS(LGAUS)*AREA(LGAUS)
   53 CONTINUE
```

```
      DO 54 IPRE=1,NPRE
      DO 54 JPE=1,NPE
      JCOL(1)=INDXF(JPE,1,IFLOW)
      JCOL(2)=INDXF(JPE,2,IFLOW)
      QE(IPRE,JCOL(1))=QE(IPRE,JCOL(1))+PTNUM*APSI(IPRE,LGAUS)
     -                                 *APHX(JPE,1,LGAUS)*AREA(LGAUS)
      QE(IPRE,JCOL(2))=QE(IPRE,JCOL(2))+PTNUM*APSI(IPRE,LGAUS)
     -                                 *APHX(JPE,2,LGAUS)*AREA(LGAUS)
      IF(IAXSY.EQ.1) EK(IPRE,JCOL(2))=EK(IPRE,JCOL(2)) + PTNUM
     -   *APSI(IPRE,LGAUS)*APHI(JPE,LGAUS)*AREA(LGAUS)/ARADUS(LGAUS)
   54 CONTINUE
 1000 CONTINUE
C
      DETM =    EPM(1,1)*EPM(2,2)*EPM(3,3)
     -        + EPM(1,2)*EPM(2,3)*EPM(3,1)
     -        + EPM(2,1)*EPM(3,2)*EPM(1,3)
     -        - EPM(1,1)*EPM(2,3)*EPM(3,2)
     -        - EPM(2,2)*EPM(1,3)*EPM(3,1)
     -        - EPM(3,3)*EPM(2,1)*EPM(1,2)
      EPINV(1,1)=(EPM(2,2)*EPM(3,3)-EPM(3,2)*EPM(2,3))/DETM
      EPINV(1,2)=(EPM(1,3)*EPM(3,2)-EPM(1,2)*EPM(3,3))/DETM
      EPINV(1,3)=(EPM(1,2)*EPM(2,3)-EPM(2,2)*EPM(1,3))/DETM
      EPINV(2,1)=(EPM(2,3)*EPM(3,1)-EPM(2,1)*EPM(3,3))/DETM
      EPINV(2,2)=(EPM(1,1)*EPM(3,3)-EPM(3,1)*EPM(1,3))/DETM
      EPINV(2,3)=(EPM(2,1)*EPM(1,3)-EPM(1,1)*EPM(2,3))/DETM
      EPINV(3,1)=(EPM(2,1)*EPM(3,2)-EPM(2,2)*EPM(3,1))/DETM
      EPINV(3,2)=(EPM(1,2)*EPM(3,1)-EPM(1,1)*EPM(3,2))/DETM
      EPINV(3,3)=(EPM(1,1)*EPM(2,2)-EPM(2,1)*EPM(1,2))/DETM
C
      DO 83 IPRE=1,NPRE
      DO 83 JDOF=1,NEDOF
      PIQE(IPRE,JDOF)=0.
      DO 82 KDUM=1,NPRE
      PIQE(IPRE,JDOF)=PIQE(IPRE,JDOF)+EPINV(IPRE,KDUM)
     -                                      *QE(KDUM,JDOF)
   82 CONTINUE
   83 CONTINUE
      DO 85 IDOF=1,NEDOF
      DO 85 JDOF=1,NEDOF
      DO 84 KDUM=1,NPRE
      EK(IDOF,JDOF)=EK(IDOF,JDOF)+EQ(IDOF,KDUM)*PIQE(KDUM,JDOF)
   84 CONTINUE
   85 CONTINUE
C
      RETURN
      END
C
C********1*********2*********3*********4*********5*********6***
      SUBROUTINE SCNVFL(NODES,NNODE,NELEM,NPE,NPRE,NDIM,IFLOW,
     -                  MAXNOD,MAXELM,MAXDOF)
C-X-  IMPLICIT REAL*8 (A-H,O-Z)
      COMMON /CDOF/  A1(11527),IDBC(11527),LDOF(4227),L1DOF(4227)
      COMMON /CFLOW/ A(4227,10),ADBC(4227,10),IBCA(4227,10)
      COMMON /CITER/ CNVCF(10),ERROF(10),RELAX(10),ITERE,MAXIT
```

```
      COMMON /CPRS/  PELEM(4,1027),PBCDAT,IPNOD(2),IPDOF
      DIMENSION NODES(27,MAXELM),KERR(4),DELA(4)
C
C        A1 --- NEW SOLUTION OBTAINED IN SUB-FRONTS
C
      DO 1 K=1,4
    1 ERROF(K)=0.
C
      AVELY=0.
      DO 5 KNODE=1,NNODE
      IDOF=L1DOF(KNODE)-1
      ADUM=0.
      DO 2 KDIM=1,NDIM
      ADUM=ADUM+A1(IDOF+KDIM)**2
    2 CONTINUE
      ADUM=ADUM**0.5
      IF(ADUM.GT.AVELY) AVELY=ADUM
    5 CONTINUE
C
      DO 10 KNODE=1,NNODE
      IDOF=L1DOF(KNODE)-1
C
      DO 7 KDIM=1,NDIM
      KDOF=IDOF+KDIM
      DELA(KDIM)=ABS(A1(KDOF)-A(KNODE,KDIM))/AVELY
      IF(DELA(KDIM).GT.ERROF(KDIM)) THEN
        ERROF(KDIM)=DELA(KDIM)
        KERR(KDIM)=KNODE
      ENDIF
    7 CONTINUE
   10 CONTINUE
C
      DO 15 KNODE=1,NNODE
      IDOF=L1DOF(KNODE)-1
      DO 12 KDIM=1,NDIM
      KDOF=IDOF+KDIM
      IF(IDBC(KDOF).EQ.1) THEN
        A(KNODE,KDIM)=A1(KDOF)
        ELSE
        A(KNODE,KDIM)=(1.-RELAX(KDIM))*A(KNODE,KDIM)
     -                            +RELAX(KDIM)*A1(KDOF)
      ENDIF
   12 CONTINUE
   15 CONTINUE
C
C
      WRITE(6,630) ITERE,(K,KERR(K),ERROF(K),K=1,NDIM)
  630 FORMAT(2X,'SUB-SCNVFL   ITERE=',I5,
     -    4(/4X,'KDIM=',I2, 2X,'NODE=',I5, 2X,'ERROF=',E12.4))
C
      RETURN
      END
C
C********1*********2*********3*********4*********5*********6***
```

```
      SUBROUTINE SPRS4(P,IBCP,PERROR,MAXNOD,MAXELM,MAXDOF)
C-X-  IMPLICIT REAL*8 (A-H,O-Z)
      COMMON /CDESC/ NNODE,NELEM,NPE,NPRE,NDIM,NEDOF,IFLOW,
     -               IAXSY,IELF
      COMMON /CELEM/ EX(27,3),EA(27,10),ENUT(64),ENUT1(8),
     -               NODEL(27),IELEM
      COMMON /CGAUS/ EXKS(3,64),EW(64),MGAUS
      COMMON /CLSCF/ XKSNOD(27,3,11),TM(4,4)
      COMMON /CMATE/ BFX(3),DENSY,VISCY,PECLET
      COMMON /CPNLT/ CXKPN(4,3,15),CWPN(4,15),PTNUM
      COMMON /CPROB/ IA(10),IPLOT
      COMMON /CPRS/  PELEM(4,1027),PBCDAT,IPNOD(2),IPDOF
      COMMON /CSHAP/ APHI(27,64),APHX(27,3,64),APSI(8,64),AREA(64),
     -               ARADUS(64)
      DIMENSION P(MAXNOD),IBCP(MAXNOD),EK(85,85),EF(85),EKI(4,4),
     -          PNLT(4),DADX(3),DIV(3),CXKS(3),PSIZ(27),
     -          DPSIZ(27,3),PSINOD(4,27)
C
      REWIND 2
      IPROF=0
C
      DO 8 KPE=1,NPE
      DO 1 KDIM=1,NDIM
      CXKS(KDIM)=XKSNOD(KPE,KDIM,IELF)
    1 CONTINUE
      GO TO (2,2,2,2,10, 3,2,2,2,2,  2,2,2,2,2), IFLOW
    2 CONTINUE
      WRITE(6,605) IFLOW
      STOP
  605 FORMAT(2X,'TERMINATED AT SUB-SPRS4 FOR IFLOW=',I5)
C
    3 CONTINUE
      CALL SHAP02(PSIZ,DPSIZ,CXKS,NDIM)
      DO 7 KPRE=1,NPRE
      PSINOD(KPRE,KPE)=PSIZ(KPRE)
    7 CONTINUE
    8 CONTINUE
C
   10 CONTINUE
      DO 11 KNODE=1,NNODE
      P(KNODE)   =0.
      IBCP(KNODE)=0
   11 CONTINUE
C
      PERROR=0.
      PMAX  =0.
      DO 13 KELEM=1,NELEM
      DO 12 KPRE=1,NPRE
      DUM=ABS(PELEM(KPRE,KELEM))
      IF(DUM.GT.PMAX) PMAX=DUM
   12 CONTINUE
   13 CONTINUE
C
      DO 1000 IELEM=1,NELEM
```

```
      DO 14 IPE=1,NPE
      EF(IPE)=0.
      DO 14 JPE=1,NPE
      EK(IPE,JPE)=0.
   14 CONTINUE
      CALL EXDAT
      CALL ELMDAT
      CALL SHPDAT
C
      DO 100 LGAUS=1,MGAUS
      XF=0.
      DO 16 KDIM=1,NDIM
      DADX(KDIM)=0.
      DO 15 KPE=1,NPE
      DADX(KDIM)=DADX(KDIM)+EA(KPE,KDIM)*APHX(KPE,KDIM,LGAUS)
   15 CONTINUE
      XF=XF+DADX(KDIM)
   16 CONTINUE
C
      IF(IAXSY.EQ.1) THEN
        VVELY=0.
        DO 20 KPE=1,NPE
        VVELY=VVELY+EA(KPE,2)*APHI(KPE,LGAUS)
   20   CONTINUE
        XF = XF + VVELY/ARADUS(LGAUS)
      ENDIF
C
      DO 25 IPRE=1,NPRE
      EF(IPRE)=EF(IPRE)+APSI(IPRE,LGAUS)*PTNUM*XF*AREA(LGAUS)
      DO 25 JPRE=1,NPRE
      EK(IPRE,JPRE)=EK(IPRE,JPRE)+APSI(IPRE,LGAUS)
     -                                *APSI(JPRE,LGAUS)*AREA(LGAUS)
   25 CONTINUE
  100 CONTINUE
C
      DETM=EK(1,1)*EK(2,2)*EK(3,3) + EK(1,2)*EK(2,3)*EK(3,1)
     -    +EK(2,1)*EK(3,2)*EK(1,3) - EK(1,1)*EK(2,3)*EK(3,2)
     -    -EK(2,2)*EK(1,3)*EK(3,1) - EK(3,3)*EK(2,1)*EK(1,2)
C
      EKI(1,1)=(EK(2,2)*EK(3,3)-EK(3,2)*EK(2,3))/DETM
      EKI(1,2)=(EK(1,3)*EK(3,2)-EK(1,2)*EK(3,3))/DETM
      EKI(1,3)=(EK(1,2)*EK(2,3)-EK(2,2)*EK(1,3))/DETM
      EKI(2,1)=(EK(2,3)*EK(3,1)-EK(2,1)*EK(3,3))/DETM
      EKI(2,2)=(EK(1,1)*EK(3,3)-EK(3,1)*EK(1,3))/DETM
      EKI(2,3)=(EK(2,1)*EK(1,3)-EK(1,1)*EK(2,3))/DETM
      EKI(3,1)=(EK(2,1)*EK(3,2)-EK(2,2)*EK(3,1))/DETM
      EKI(3,2)=(EK(1,2)*EK(3,1)-EK(1,1)*EK(3,2))/DETM
      EKI(3,3)=(EK(1,1)*EK(2,2)-EK(2,1)*EK(1,2))/DETM
C
      DO 50 IPRE=1,NPRE
      PNLT(IPRE)=0.
      DO 40 JPRE=1,NPRE
      PNLT(IPRE)=PNLT(IPRE) - EKI(IPRE,JPRE)*EF(JPRE)
   40 CONTINUE
```

```
   50 CONTINUE
C
      IF(PMAX.LE.1.E-6) GO TO 61
      DO 60 KPRE=1,NPRE
      DUM=ABS(PNLT(KPRE)-PELEM(KPRE,IELEM))/PMAX
      IF(DUM.GT.PERROR) THEN
        PERROR=DUM
        KPRNOD=NODEL(NPE)
      ENDIF
   60 CONTINUE
C
   61 CONTINUE
      DO 65 KPRE=1,NPRE
      PELEM(KPRE,IELEM)=PNLT(KPRE)
   65 CONTINUE
C
      GO TO (71,71,71,71,75,  76,71,71,71,71,  71,71,71,71,71),
     -                                                   IFLOW
   71 CONTINUE
      WRITE(6,672) IFLOW
      STOP
  672 FORMAT(2X,'TERMINATED AT SUB-SPRS4 FOR IFLOW=',I5)
C
C ----PRESSURE INTERPOLATION POLYNOMIALS OF THE FORM [1,X,Y] ---
   75 CONTINUE
      DO 140 KPE=1,NPE
      KNODE=NODEL(KPE)
      IBCP(KNODE)=IBCP(KNODE)+1
      PDUM = PNLT(1)
      DO 135 KDIM=1,NDIM
      PDUM = PDUM + PNLT(KDIM+1)*EX(KPE,KDIM)
  135 CONTINUE
      P(KNODE) = P(KNODE) + PDUM
  140 CONTINUE
      GO TO 145
C
C --- NEW PRESSURE INTERPOLATION POLYNOMIALS ------------
   76 CONTINUE
      DO 144 KPE=1,NPE
      DO 142 KDIM=1,NDIM
      CXKS(KDIM)=XKSNOD(KPE,KDIM,IELF)
  142 CONTINUE
      KNODE=NODEL(KPE)
      IBCP(KNODE)=IBCP(KNODE)+1
      PDUM=0.
      DO 143 KPRE=1,NPRE
      PDUM = PDUM + PNLT(KPRE)*PSINOD(KPRE,KPE)
  143 CONTINUE
      P(KNODE)=P(KNODE) + PDUM
  144 CONTINUE
  145 CONTINUE
C
 1000 CONTINUE
  687 FORMAT(2X,'SUB-SPRS   IELEM=',I5,  2X,'KPE=',I2,
```

```
     -       2X,'KNODE=',I6, /2X,'X=',E12.4,  2X,'Y=',E12.4,
     -       2X,'P=',E12.4)
C
      WRITE(6,650) KPRNOD,PERROR
  650 FORMAT(2X,'KDIM=4  KPRNOD=',I6,  2X,'PERROR=',E12.4)
C
      DO 150 KNODE=1,NNODE
      P(KNODE)=P(KNODE)/FLOAT(IBCP(KNODE))
  150 CONTINUE
C
      PREF=P(IPNOD(2))
      DO 160 KNODE=1,NNODE
      P(KNODE)=P(KNODE)-PREF+PBCDAT
  160 CONTINUE
C
      RETURN
      END
C
C********1*********2*********3*********4*********5*********6***
      SUBROUTINE PFLOW(MAXNOD,MAXELM,MAXDOF)
C-X-  IMPLICIT REAL*8 (A-H,O-Z)
      COMMON /CDESC/ NNODE,NELEM,NPE,NPRE,NDIM,NEDOF,IFLOW,
     -               IAXSY,IELF
      COMMON /CGRID/ X(4227,3),NODES(27,1027)
      COMMON /CITER/ CNVCF(10),ERROF(10),RELAX(10),ITERE,MAXIT
      COMMON /CPROB/ IA(10),IPLOT
      COMMON /CPRS/  PELEM(4,1027),PBCDAT,IPNOD(2),IPDOF
      COMMON /CFLOW/ A(4227,10),ADBC(4227,10),IBCA(4227,10)
C
      WRITE(6,650) ITERE,IFLOW
  650 FORMAT(2X,'ENTRY-PFLOW          ITERE=',I4, 2X,'IFLOW=',I2)
C
      DO 50 KNODE=1,NNODE
      WRITE(6,660) KNODE,(A(KNODE,IDUM),IDUM=1,4)
   50 CONTINUE
      RETURN
  660 FORMAT(2X,I5,5E12.4)
C
C---------1---------2---------3---------4---------5---------6---
      ENTRY PLSDAT(MAXNOD,MAXELM,MAXDOF)
      WRITE(7,605)
      DO 10 KNODE=1,NNODE
      WRITE(7,606) KNODE,(X(KNODE,KDUM),KDUM=1,3)
   10 CONTINUE
  605 FORMAT(2X,'  KNODE          X            Y            Z')
  606 FORMAT(2X,I5,2X,3E16.8)
C
      WRITE(7,612)
  612 FORMAT(2X,'NODE CONNECTIVITY DATA')
      DO 15 KELEM=1,NELEM
      WRITE(7,615) KELEM,(NODES(KPE,KELEM),KPE=1,NPE)
   15 CONTINUE
  615 FORMAT(2X,I5,2X,10I7,  2(/9X,10I7))
C
```

```
      WRITE(7,620)
      DO 20 KNODE=1,NNODE
      WRITE(7,621) KNODE,(IBCA(KNODE,KPROB),KPROB=1,3)
   20 CONTINUE
C
      WRITE(7,624)
      WRITE(7,625) PBCDAT,(IPNOD(K),K=1,2)
  620 FORMAT(2X,'IBC-DATA FOR IA=1,2,3,4,6')
  621 FORMAT(4X,I5,2X,20I3)
  624 FORMAT(2X,'PBCDAT AND IPNOD(1-2)')
  625 FORMAT(2X,E14.6,  2X,2I7)
C
      WRITE(7,630)
      DO 30 KNODE=1,NNODE
      WRITE(7,631) KNODE,(A(KNODE,K),K=1,4)
   30 CONTINUE
  630 FORMAT(2X,' KNODE    U                V              W',
     -          '          P')
  631 FORMAT(2X,I5,4E17.9)
C
      WRITE(7,640)
      DO 40 KNODE=1,NNODE
      WRITE(7,641) KNODE,(ADBC(KNODE,K),K=1,3),ADBC(KNODE,6)
   40 CONTINUE
  640 FORMAT(2X,'ADBC-DATA FOR U, V, AND W')
  641 FORMAT(2X,I5,4E17.9)
C
      RETURN
      END
C
C********1*********2*********3*********4*********5*********6
      SUBROUTINE USER
C-X-  IMPLICIT REAL*8 (A-H,O-Z)
      COMMON /CDESC/ NNODE,NELEM,NPE,NPRE,NDIM,NEDOF,IFLOW,
     -               IAXSY,IELF
      COMMON /CELEM/ EX(27,3),EA(27,10),ENUT(64),ENUT1(8),
     -               NODEL(27),IELEM
      COMMON /CGAUL/ CLXKS(4,4),CLW(4,4),NGAUS
      COMMON /CGAUS/ EXKS(3,64),EW(64),MGAUS
      COMMON /CGRID/ X(4227,3),NODES(27,1027)
      COMMON /CMATE/ BFX(3),DENSY,VISCY,PECLET
      COMMON /CSHAP/ APHI(27,64),APHX(27,3,64),APSI(8,64),AREA(64),
     -               ARADUS(64)
      COMMON /CUSE2/ XK(3),XKN(3,3),XC(10),XB(3),XF(3),PROD,XWALL
      COMMON /CFLOW/ A(4227,10),ADBC(4227,10),IBCA(4227,10)
      DIMENSION PNK(3),DPNK(3)
C
C--------1---------2---------3---------4---------5---------6---
      ENTRY EXDAT
      DO 4 KPE=1,NPE
      KNODE=ABS(NODES(KPE,IELEM))
      NODEL(KPE)=KNODE
      DO 3 KDIM=1,NDIM
      EX(KPE,KDIM)=X(KNODE,KDIM)
```

```
    3 CONTINUE
    4 CONTINUE
      RETURN
C
C---------1---------2---------3---------4---------5---------6
      ENTRY ELMDAT
      DO 6 KPE=1,NPE
      KNODE=ABS(NODES(KPE,IELEM))
      NODEL(KPE)=KNODE
      DO 5 KPROB=1,10
      EA(KPE,KPROB)=A(KNODE,KPROB)
    5 CONTINUE
    6 CONTINUE
C
      RETURN
      END
******************************** BOTTOM OF DATA ******************************
```

APPENDIX II

INPUT DATA FOR NSFLOW/P

The required input data to solve the incompressible, laminar flows is described below. The computational sequence is controlled by the macro-instruction data [27] in the main program. These macro-instruction data are "INIT," "PREP," "PROC," "CONT," AND "END;" and these data have to start from the fifth column of each card. The function of these data are described below.

"INIT" – Initialize dimensioned variables.

"PREP" – Call the SUBROUTINE PREP to read in the descriptive data for each flow problem.

"PROC" – Call the SUBROUTINE PROCES to solve the Navier-Stokes equations.

"CONT" – Continue computation for the next flow problem.

"END" – Terminate the computation.

The descriptive data for a specific flow case are read into the computer program in the SUBROUTINE PREP. The sequence to read in the various descriptive data is also controlled by the macro-instruction data. The macro-instruction data used in the SUBROUTINE PREP are listed below. The function for each of these macro-instruction data and a set of specific data followed by each of these macro-instruction data are described below. The macro-instruction data used in the SUBROUTINE PREP have to start from the first column of each card. In most of the cases, a comment card has been used to clarify the input data to be prepared.

1. "DESC" – Read in the general descriptive data.

   IFLOW – = 5, Solve two-dimensional flows using the pressure interpolation polynomials of the form (1,x,y); = 6, Solve two-dimensional flows using the new pressure interpolation method.

   NDIM – Dimension of the problem.

   NGAUS – Number of Gauss points in each coordinate direction. (Ngaus = 3 has been tested).

   MFRONF – Frontal width.

2. "CNTL" – Control parameters.

   NNODE – Number of nodes.

   NELEM – Number of elements.

   IAXSY – = 0 for two-dimensional case, and = 1 for axisymmetric case.

   IPLOT – = 1 to write the computational results on a disk file.

3. "ELEM" – Call the SUBROUTINE ELEM to generate the node connectivity data. The input data for the subroutine is described below. Again, some of the data are followed by a comment card.

NBLOC – Number of blocks to generate the node connectivity data.

IEL1 – The first element number in each block.

(NODES(IPE,IEL1),IPE=1,NPE) – Node connectivity data for the first element in each block. NPE is the number of nodes in an element.

NEL(KDIM) – Number of elements in each coordinate direction.

INCREL(KDIM) – Incremental element number in each coordinate direction.

(INCNOD(K,KDIM),K=1,NPE) – Increment of the connectivity data for each coordinate direction.

4. "NODE" – Call the SUBROUTINE RNODE to generate the grid coordinate data.

NBLOC – Number of blocks for the coordinate data generation.

METHOD – = 1, To read in the coordinate data on the physical domain;
= 2, To read in the coordinate data on the computational element. In this case, isoparametric mapping is used for grid generation.

<u>Description of the input data for METHOD=1</u>

NODG1 – The first node number in each block.

INCRX, INCRY, INCRZ – Incremental node numbers in each coordinate direction.

NDAT – Number of grid points in each coordinate direction.

(DELX(IKE,KDIM),IKE=1,NDAT) – An array of physical coordinate data in each coordinate direction.

<u>Description of input data for METHOD=2</u>

((XNOD(KPE,KDIM),KDIM=1,KPE=1,NPE) – Coordinate data of the block. The sequence of node numbers should be the same as that of the computational element.

NODG1, INCRX, INCRY, INCRZ – The same as above.

NDAT – The same as above.

(DELX(IKE,KDIM),IKE=1,NDAT) – An array of cordinate data defined on the computational element in each coordinate direction.

5. "MATE" – Material property data.

VISCY – Molecular viscosity of the fluid.

DENSY – Density of the fluid.

(BFX(K),K=1,NDIM) – Body force in each coordinate direction.

6. "ITER" – Iteration parameters.

MAXIT – Maximum number of iterations.

(RELAX(K),K=1,10) – Under-relaxation numbers; K = 1, 2, and 3 for the x-, y-, z-momentum equations, respectively; K = 4 for pressure; rest of these under-relaxation numbers are not used as yet.

(CNVCF(K),K=1,10) – Convergence criteria, use is the same as above.

7. "IA##" – Call the SUBROUTINES RINIT and RBC1 to read in the initial guess and the boundary condition data for flow equations. (## = 1, 2, and 3 for u, v, and w, respectively; IA05 through IA10 are not used as yet.)

Input data for SUBROUTINES RINIT and RBC1

NREC – Number of records.

N1 – The first node number.

N2 – The last node number.

INCNOD – Incremental node number.

ADATA – Real variable to be assigned as the initial guess.

8. "IA04" – Input data for pressure.

PBCDAT – A real variable for pressure boundary condition.

IPNOD(1) – A pressure node number for which the pressure boundary condition is specified.

IPNOD(2) – A velocity node number to prescribe a reference pressure.

9. "INCL" – Include re-start data.

10. "END" – Return the control to the main program.

A.2.1 Cavity Flow for Re = 10,000

```
******************************** TOP OF DATA ********************************
CAVITY FLOW FOR REYNOLDS NUMBER=10000 (CAVT91)
****INITIALIZE DIMENSIONED VARIABLES****
****PREPARE INPUT DATA              ****
DESCRIPTIVE DATA -------------------------------------------------
         IFLOW, NDIM, NGAUS, MFRONF,
           6,      2,    3,     165,
CNTL PARAMETERS -------------------------------------------------
         NNODE, NELEM, IAXSY, IPLOT,
          4225,  1024,    0,     1,
MATERIAL PROPERTY OF FLUID ---------------------------------------
         VISCY,         DENSY,   BFX(1-2),
         0.0001225,     1.225,    0., 0.,
ITERATION PARAMETERS ---------------------------------------------
         MAXIT, RELAX(1-10), CNVCF(1-10)
          100,
      0.8,    0.8,     1.,      1.,      1.,
      1.,     1.,      1.,      1.,      1.,
      1.E-4,     1.E-4,     1.E-4,     1.E-4,     1.E-4,
      1.E-4,     1.E-4,     1.E-4,     1.E-4,     1.E-4,
NODE COORDINATE DATA - GRID GENERATION
  NUMBER OF BLOCKS (NBLOC)
     1,
  GRID GENERATION METHOD FOR IBLOC=1 (METHOD)
      2,
     NODE COORDINATE DATA ((XNOD(KPE,KDIM),KDIM=1,NDIM),KPE=1,NPE)
        0.,  0.,      0.5,  0.,      1.,  0.,
        1.,  0.5,     1.,   1.,      0.5, 1.,
        0.,  1.,      0.,   0.5,     0.5, 0.5,
     NODG1,  INCR-X,-Y,-Z
      1,         65, 1, 0,
     DISCRETIZATION OF THE COMPUTATIONAL GRID (NDAT,DELX-ARRAY)
  65,
    -1.00, -0.995, -0.99, -0.98,  -0.97,  -0.96,  -0.95, -0.935,
    -0.92, -0.905, -0.89, -0.87,  -0.85,  -0.83,  -0.81, -0.785,
    -0.76, -0.73,  -0.70, -0.66,  -0.62,  -0.575, -0.53, -0.48,
    -0.43, -0.38,  -0.33, -0.28,  -0.23,  -0.175, -0.12, -0.06,
     0.00,  0.06,   0.12,  0.175,  0.23,   0.28,   0.33,  0.38,
     0.43,  0.48,   0.53,  0.575,  0.62,   0.66,   0.70,  0.73,
     0.76,  0.785,  0.81,  0.83,   0.85,   0.87,   0.89,  0.905,
     0.92,  0.935,  0.95,  0.96,   0.97,   0.98,   0.99,  0.995,
     1.00,
  65,
    -1.00, -0.995, -0.99, -0.98,  -0.97,  -0.96,  -0.95, -0.935,
    -0.92, -0.905, -0.89, -0.87,  -0.85,  -0.83,  -0.81, -0.785,
    -0.76, -0.73,  -0.70, -0.66,  -0.62,  -0.575, -0.53, -0.48,
    -0.43, -0.38,  -0.33, -0.28,  -0.23,  -0.175, -0.12, -0.06,
     0.00,  0.06,   0.12,  0.175,  0.23,   0.28,   0.33,  0.38,
     0.43,  0.48,   0.53,  0.575,  0.62,   0.66,   0.70,  0.73,
     0.76,  0.785,  0.81,  0.83,   0.85,   0.87,   0.89,  0.905,
     0.92,  0.935,  0.95,  0.96,   0.97,   0.98,   0.99,  0.995,
     1.00,
   1,
     0.,
```

```
ELEMENT CONNECTIVITY DATA FOR THE GLOBAL DOMAIN --------------
     NUMBER OF BLOCKS (NBLOC)
              1,
     ELEMENT NO. AND NODE CONNECTIVITY (IEL1,NODES(IEL1))
         1,   1, 66, 131,    132, 133, 68,     3, 2, 67,
     NO. OF ELEMENTS (NEL,INCREL,INCNOD)
         32,  32,  130,130,130,  130,130,130,  130,130,130,
         32,   1,   2, 2, 2,    2, 2, 2,     2, 2, 2,
          1,   0,   0, 0, 0,    0, 0, 0,     0, 0, 0,
IA01 ----------------------------------------------------------
     INITIAL GUESS FOR U (NREC)
          0,
     DBC FOR U
          4,
            1,   4161,   65,    0.,
         4161,   4224,    1,    0.,
           65,   4225    65,    1.,
            1,     64,    1,    0.,
IA02 ----------------------------------------------------------
     INITIAL GUESS FOR V (NREC)
          0,
     DBC FOR V
          4,
            1,   4161,   65,    0.,
         4161,   4224,    1,    0.,
           65,   4225,   65,    0.,
            1,     64,    1,    0.,
IA04 --(PBCDAT, IPNOD1, IPNOD2)-------------------------
          0.,    2017,    2081,
END OF INPUT DATA
****PROCESSOR FOR NAVIER-STOKES EQUATIONS ****
****END OF RUN
******************************** BOTTOM OF DATA ********************************
```

A.2.2 Backward-Facing Step Flow

```
********************************* TOP OF DATA *********************************
--- LAMINAR BACKWARD-FACING STEP FLOW (STP5K) ---
****INITIALIZE DIMENSIONED VARIABLES ****
****PREPARE INPUT DATA               ****
DESCRIPTIVE DATA -------------------------------------------------
         IFLOW, NDIM, NGAUS, MFRONF,
          6,     2,     3,     95,
CNTL PARAMETERS -------------------------------------------------
         NNODE, NELEM, IAXSY, IPLOT,
          2631,  628,    0,      1,
ELEMENT CONNECTIVITY DATA FOR THE GLOBAL DOMAIN ---------------
     NUMBER OF BLOCKS (NBLOC)
             3,
  NODE CONNECTIVITY DATA FOR IBLOC=1 (IEL1,NODES)
     1,    1, 16, 31, 32, 33, 18, 3, 2, 17,
     NEL(KDIM),INCREL(KDIM), INCNOD(KDIM)
          3,         7,        30,30,30, 30,30,30, 30,30,30,
          7,         1,         2, 2, 2,  2, 2, 2,  2, 2, 2,
          1,         0,         0, 0, 0,  0, 0, 0,  0, 0, 0,
  NODE CONNECTIVITY DATA FOR IBLOC=2 (IEL1,NODES)
     22,  91, 106, 137, 138, 139, 108, 93, 92, 107,
     NEL(KDIM),INCREL(KDIM), INCNOD(KDIM)
          1,         0,         0, 0, 0,  0, 0, 0,  0, 0, 0,
          7,         1,         2, 2, 2,  2, 2, 2,  2, 2, 2,
          1,         0,         0, 0, 0,  0, 0, 0,  0, 0, 0,
  NODE CONNECTIVITY DATA FOR IBLOC=3 (IEL1,NODES)
     29,  121, 152, 183, 184, 185, 154, 123, 122, 153,
     NEL(KDIM), INCREL(KDIM), INCNOD(KDIM)
          40,        15,        62,62,62, 62,62,62, 62,62,62,
          15,         1,         2, 2, 2,  2, 2, 2,  2, 2, 2,
          1,          0,         0, 0, 0,  0, 0, 0,  0, 0, 0,
NODE COORDINATE DATA -------------------------------------------------
  NUMBER OF BLOCKS (NBLOC)
     2,
  GRID GENERATION METHOD FOR IBLOC=1 (METHOD)
       1,
      NODG1,INCRX,INCRY,INCRZ,
       1,    15,    1,    0,
      NDAT, GRID COORDINDATE DATA
       8,    -0.0147,    -0.01274,   -0.01078,   -0.00882,   -0.00686,
             -0.0049,    -0.00294,   -0.00147,
       15,    0.0049,     0.005145,   0.00539,    0.0057085,  0.006027,
              0.0064925,  0.006958,   0.0075,     0.008042,   0.0085075,
              0.008973,   0.0092915,  0.00961,    0.009855,   0.0101,
       1,    0.,
  GRID GENERATION METHOD FOR IBLOC=2 (METHOD)
       1,
      NODG1,INCRX,INCRY,INCRZ,
      121,   31,    1,    0,
      NDAT, GRID COORDINATE DATA
       81, 0.,          0.00049,  0.00098,  0.00196,  0.00294,
           0.00441,     0.00588,  0.00784,  0.0098,   0.01225,
           0.0147,      0.01715,  0.0196,   0.02205,  0.0245,
```

```
        0.02695,  0.0294,   0.03185,  0.0343,   0.03675,
        0.0392,   0.04165,  0.0441,   0.04665,  0.049,
        0.05145,  0.0539,   0.05635,  0.0588,   0.06125,
        0.0637,   0.06615,  0.0686,   0.07105,  0.0735,
        0.07595,  0.0784,   0.08085,  0.0833,   0.08575,
        0.0882,   0.09065,  0.0931,   0.09555,  0.098,
        0.10045,  0.1029,   0.10535,  0.1078,   0.11025,
        0.1127,   0.11515,  0.1176,   0.12005,  0.1225,
        0.12495,  0.1274,   0.12985,  0.1323,   0.13475,
        0.1372,   0.13965,  0.1421,   0.14455,  0.147,
        0.14994,  0.15288,  0.15631,  0.15974,  0.16366,
        0.16758,  0.17199,  0.1764,   0.1813,   0.1862,
        0.19159,  0.19698,  0.202615, 0.20825,  0.214375,
        0.2205,
    31, 0.,       0.000196, 0.000392, 0.0006615,0.000931,
        0.001274, 0.001617, 0.0020335,0.00245,  0.0028665,
        0.003283, 0.003626, 0.003969, 0.0042385,0.004508,
        0.004704, 0.0049,   0.005145, 0.00539,  0.0057085,
        0.006027, 0.0064925,0.006958, 0.0075,   0.008042,
        0.0085075,0.008973, 0.0092915,0.00961,  0.009855,
        0.0101,
     1,   0.,
MATERIAL PROPERTY OF FLUID --- (RE=500) ------------------------
     VISCY,     DENSY,    BFX(1-2),
     0.000016986, 1.225,   0., 0.,
ITERATION PARAMETERS -------------------------------------------
    MAXIT, RELAX(1-10), CNVCF(1-10),
    100,
    0.8,    0.8,    1.,   1.,     1.,
    1.,     1.,     1.,   1.,     1.,
    1.E-4,  1.E-4,  1.E-4,  1.E-4,  1.E-4,
    1.E-4,  1.E-4,  1.E-4,  1.E-4,  1.E-4,
IA01 ----------------------------------------------------------
   INITIAL GUESS FOR U-VELOCITY (NREC)
     0,
   DBC FOR U
     20,
        1,   1,   1,      0.,
        2,   2,   1,      0.1796,
        3,   3,   1,      0.3414,
        4,   4,   1,      0.5252,
        5,   5,   1,      0.6790,
        6,   6,   1,      0.8498,
        7,   7,   1,      0.9565,
        8,   8,   1,      1.0000,
        9,   9,   1,      0.9565,
       10,  10,   1,      0.8498,
       11,  11,   1,      0.6790,
       12,  12,   1,      0.5252,
       13,  13,   1,      0.3414,
       14,  14,   1,      0.1796,
       15,  15,   1,      0.,
       16, 106,  15,      0.,
      121, 137,   1,      0.,
```

```
            121,2601,  31,      0.,
             30, 120,  15,      0.,
            151,2631,  31,      0.,
IA02 -------------------------------------------------------------
     INITIAL GUESS FOR V-VELOCITY (NREC)
         0,
     DBC FOR V
         6,
              1,  15,   1,      0.,
             16, 106,  15,      0.,
            121, 137,   1,      0.,
            121,2601,  31,      0.,
             30, 120,  15,      0.,
            151,2631,  31,      0.,
IA04 ---- (PBCDAT, IPNODE(1-2))-------------------------
             0.,      0,  2617,
END OF INPUT DATA
****PROCESSOR FOR NAVIER-STOKES EQUATIONS ****
****END OF RUN                           ****
******************************** BOTTOM OF DATA ********************************
```

A.2.3 Flow Through a Nest of Cylinders

```
********************************* TOP OF DATA *********************************
---- STEADY FLOW THROUGH A NEST OF CYLINDERS -(CNEST)-
****INITIALIZE DIMENSIONED VARIABLES****
****PREPARE INPUT DATA              ****
DESCRIPTIVE DATA ------------------------------------------------------
         IFLOW, NDIM, NGAUS, MFRONF,
           6,     2,     3,    115,
CNTL PARAMETERS -------------------------------------------------------
         NNODE, NELEM, IAXSY, IPLOT
          4369,  1024,    0,     1,
ELEMENT CONNECTIVITY DATA FOR THE GLOBAL DOMAIN --------------
     NUMBER OF BLOCKS (NBLOC)
         21,
     IBLOC=1, (IEL1, NODES(1-NPE))
         1,     1, 18, 35,     36, 37, 20,      3, 2, 19,
         NO. OF ELEMENTS (NEL,INCREL,INCNOD)
        20,    8,    34, 34, 34, 34, 34, 34, 34, 34, 34,
         8,    1,     2,  2,  2,  2,  2,  2,  2,  2,  2,
         1,    0,     0,  0,  0,  0,  0,  0,  0,  0,  0,
     IBLOC=2, (IEL1, NODES(1-NPE))
       161,  681,698,715,    716,717,700,    683,682,699,
         NO. OF ELEMENTS (NEL,INCREL,INCNOD)
         8,    8,    34, 34, 34, 34, 34, 34, 34, 34, 34,
         8,    1,     2,  2,  2,  2,  2,  2,  2,  2,  2,
         1,    0,     0,  0,  0,  0,  0,  0,  0,  0,  0,
     IBLOC=3-1, (IEL1, NODES(1-NPE))
       225,  697,714,731,    972,989,988,    987,970,971,
         NO. OF ELEMENTS (NEL,INCREL,INCNOD)
         8,    1,    34, 34, 34,  2,  2,  2,  2,  2,  2,
         1,    0,     0,  0,  0,  0,  0,  0,  0,  0,  0,
         1,    0,     0,  0,  0,  0,  0,  0,  0,  0,  0,
     IBLOC=3-2, (IEL1, NODES(1-NPE))
       233,  987,988,989,   1006,1023,1022, 1021,1004,1005,
         NO. OF ELEMENTS (NEL,INCREL,INCNOD)
         8,    1,     2,  2,  2,  2,  2,  2,  2,  2,  2,
         7,    8,    34, 34, 34, 34, 34, 34, 34, 34, 34,
         1,    0,     0,  0,  0,  0,  0,  0,  0,  0,  0,
     IBLOC=4, (IEL1, NODES(1-NPE))
       289,  1225,1226,1227, 1244,1261,1260, 1259,1242,1243,
         NO. OF ELEMENTS (NEL,INCREL,INCNOD)
         8,    1      2,  2,  2,  2,  2,  2,  2,  2,  2,
         8,    8,    34, 34, 34, 34, 34, 34, 34, 34, 34,
         1,    0,     0,  0,  0,  0,  0,  0,  0,  0,  0,
     IBLOC=5-1, (IEL1, NODES(1-NPE))
       353,  1225,1242,1259, 1546,1547,1531, 1515,1514,1530,
         NO. OF ELEMENTS (NEL,INCREL,INCNOD)
         8,    8,    34, 34, 34, 32, 32, 32, 32, 32, 32,
         1,    0,     0,  0,  0,  0,  0,  0,  0,  0,  0,
         1,    0,     0,  0,  0,  0,  0,  0,  0,  0,  0,
     IBLOC=5-2,(IEL1, NODES(1-NPE))
       354,  1515,1531,1547, 1548,1549,1533, 1517,1516,1532,
         NO. OF ELEMENTS (NEL,INCREL,INCNOD)
         8,    8,    32, 32, 32, 32, 32, 32, 32, 32, 32,
         7,    1,     2,  2,  2,  2,  2,  2,  2,  2,  2,
```

```
    1,    0,     0,  0,  0,  0,  0,  0,  0,  0,  0,
IBLOC=6-1, IEL1, NODES(1-NPE))
  417,  1497,1786,1803, 1804,1805,1788, 1771,1770,1787,
    NO. OF ELEMENTS (NEL,INCREL,INCNOD)
    1,    0,     0,  0,  0,  0,  0,  0,  0,  0,  0,
    1,    0,     0,  0,  0,  0,  0,  0,  0,  0,  0,
    1,    0,     0,  0,  0,  0,  0,  0,  0,  0,  0,
IBLOC=6-2, IEL1, NODES(1-NPE))
  425,  1803,1820,1837, 1838,1839,1822, 1805,1804,1821,
    NO. OF ELEMENTS (NEL,INCREL,INCNOD)
    7,    8,    34, 34, 34, 34, 34, 34, 34, 34, 34,
    1,    0,     0,  0,  0,  0,  0,  0,  0,  0,  0,
    1,    0,     0,  0,  0,  0,  0,  0,  0,  0,  0,
IBLOC=6-3, IEL1, NODES(1-NPE))
  418,  1771,1788,1805, 1806,1807,1790, 1773,1772,1789,
    NO. OF ELEMENTS (NEL,INCREL,INCNOD)
    8,    8,    34, 34, 34, 34, 34, 34, 34, 34, 34,
    7,    1,     2,  2,  2,  2,  2,  2,  2,  2,  2,
    1,    0,     0,  0,  0,  0,  0,  0,  0,  0,  0,
IBLOC=7-1, (IEL1, NODES(1-NPE))
  481,  1497,2058,2059, 2075,2091,2090, 1803,1786,2074,
    NO. OF ELEMENTS (NEL,INCREL,INCNOD)
    1,    0,     0,  0,  0,  0,  0,  0,  0,  0,  0,
    1,    0,     0,  0,  0,  0,  0,  0,  0,  0,  0,
    1,    0,     0,  0,  0,  0,  0,  0,  0,  0,  0,
IBLOC=7-2, (IEL1, NODES(1-NPE))
  489,  1803,2090,2091, 2107,2123,2122, 1837,1820,2106,
    NO. OF ELEMENTS (NEL,INCREL,INCNOD)
    1,    0,     0,  0,  0,  0,  0,  0,  0,  0,  0,
    7,    8,    34, 32, 32, 32, 32, 32, 34, 34, 32,
    1,    0,     0,  0,  0,  0,  0,  0,  0,  0,  0,
IBLOC=7-3, (IEL1, NODES(1-NPE))
  482,  2059,2060,2061, 2077,2093,2092, 2091,2075,2076,
    NO. OF ELEMENTS (NEL,INCREL,INCNOD)
    7,    1,     2,  2,  2,  2,  2,  2,  2,  2,  2,
    8,    8,    32, 32, 32, 32, 32, 32, 32, 32, 32,
    1,    0,     0,  0,  0,  0,  0,  0,  0,  0,  0,
IBLOC=8-1, (IEL1, NODES(1-NPE))
  545,  2041,2314,2315, 2332,2349,2348, 2347,2330,2331,
    NO. OF ELEMENTS (NEL,INCREL,INCNOD)
    1,    0,     0,  0,  0,  0,  0,  0,  0,  0,  0,
    1,    0,     0,  0,  0,  0,  0,  0,  0,  0,  0,
    1,    0,     0,  0,  0,  0,  0,  0,  0,  0,  0,
IBLOC=8-2, (IEL1, NODES(1-NPE))
  553,  2347,2348,2349, 2366,2383,2382, 2381,2364,2365,
    NO. OF ELEMENTS (NEL,INCREL,INCNOD)
    1,    0,     0,  0,  0,  0,  0,  0,  0,  0,  0,
    7,    8,    34, 34, 34, 34, 34, 34, 34, 34, 34,
    1,    0,     0,  0,  0,  0,  0,  0,  0,  0,  0,
IBLOC=8-3, (IEL1, NODES(1-NPE))
  546,  2315,2316,2317, 2334,2351,2350, 2349,2332,2333,
    NO. OF ELEMENTS (NEL,INCREL,INCNOD)
    7,    1,     2,  2,  2,  2,  2,  2,  2,  2,  2,
    8,    8,    34, 34, 34, 34, 34, 34, 34, 34, 34,
```

```
        1,   0,     0,  0,  0,  0,  0,  0,  0,  0,  0,
    IBLOC=9-1, (IEL1, NODES(1-NPE))
      609,  2601,2602,2619, 2620,2621,2604, 2599,2600,2603,
        NO. OF ELEMENTS (NEL,INCREL,INCNOD)
        1,   1,     0,  0,  0,  0,  0,  0,  0,  0,  0,
        8,   1,    -2,  2,  2,  2,  2,  2, -2, -2,  2,
        1,   0,     0,  0,  0,  0,  0,  0,  0,  0,  0,
    IBLOC=9-2, (IEL1, NODES(1-NPE))
      617,  2619,2636,2653, 2654,2655,2638, 2621,2620,2637,
        NO. OF ELEMENTS (NEL,INCREL,INCNOD)
        7,   8,    34, 34, 34, 34, 34, 34, 34, 34, 34,
        8,   1,     2,  2,  2,  2,  2,  2,  2,  2,  2,
        1,   0,     0,  0,  0,  0,  0,  0,  0,  0,  0,
    IBLOC=10-1, (IEL1, NODES(1-NPE))
      673,  2873,2874,2891, 2892,2893,2876, 2839,2856,2875,
        NO. OF ELEMENTS (NEL,INCREL,INCNOD)
        1,   0,     0,  0,  0,  0,  0,  0,  0,  0,  0,
        7,   1,   -34,  2,  2,  2,  2,  2,-34,-34,  2,
        1,   0,     0,  0,  0,  0,  0,  0,  0,  0,  0,
    IBLOC=10-2, (IEL1, NODES(1-NPE))
      680,  2635,2888,2905, 2906,2907,2890, 2585,2618,2889,
        NO. OF ELEMENTS (NEL,INCREL,INCNOD)
        1,   0,     0,  0,  0,  0,  0,  0,  0,  0,  0,
        1,   0,     0,  0,  0,  0,  0,  0,  0,  0,  0,
        1,   0,     0,  0,  0,  0,  0,  0,  0,  0,  0,
    IBLOC=10-3, (IEL1, NODES(1-NPE))
      681,  2891,2908,2925, 2926,2927,2910, 2893,2892,2909,
        NO. OF ELEMENTS (NEL,INCREL,INCNOD)
       43,   8,    34, 34, 34, 34, 34, 34, 34, 34, 34,
        8,   1,     2,  2,  2,  2,  2,  2,  2,  2,  2,
        1,   0,     0,  0,  0,  0,  0,  0,  0,  0,  0,
NODE COORDINATE DATA -----------------------------------------
  NUMBER OF BLOCKS (NBLOC)
     10,
  GRID GENERATION METHOD FOR IBLOC=1
     2,
    NODE COORDINATE DATA ((XNOD(KPE,KDIM),KDIM=1,NDIM),KPE=1,NPE)
       0.,          0.,                 1.25,         0.,
       2.5,         0.,                 2.5,          0.5,
       2.5,         1.,                 1.25,         1.,
       0.,          1.,                 0.,           0.5,
       1.25,        0.5,
    NODG1,  INCR-X,-Y,-Z
     1,         17, 1, 0,
    DISCRETIZATION OF THE COMPUTATIONAL GRID (NDAT,DELX-ARRAY)
     41,
       -1.,      -0.95,      -0.9,       -0.85,      -0.8,
       -0.75,    -0.7,       -0.65,      -0.6,       -0.55,
       -0.50,    -0.45,      -0.40,      -0.35,      -0.30,
       -0.25,    -0.20,      -0.15,      -0.10,      -0.05,
        0.,       0.05,       0.10,       0.15,       0.20,
        0.25,     0.30,       0.35,       0.40,       0.45,
        0.50,     0.55,       0.60,       0.65,       0.70,
        0.75,     0.80,       0.85,       0.90,       0.95,
```

```
      1.0,
    17,
     -1.,      -0.875,    -0.75,     -0.625,    -0.50,
     -0.375,   -0.250,    -0.125,     0.0,       0.125,
      0.25,     0.375,     0.50,      0.625,     0.750,
      0.875,    1.0,
     1,
      0.,
GRID GENERATION METHOD FOR IBLOC=2
    2,
   NODE COORDINATE DATA ((XNOD(KPE,KDIM),KDIM=1,NDIM),KPE=1,NPE)
     2.5,          0.,                    2.75,          0.,
     3.0,          0.,                    3.03806,       0.19134,
     3.14645,      0.35355,               2.82323,       0.67678,
     2.5,          1.,                    2.5,           0.5,
     2.76903,      0.34567
   NODG1,  INCR-X,-Y,-Z
    681,        17, 1, 0,
   DISCRETIZATION OF THE COMPUTATIONAL GRID (NDAT,DELX-ARRAY)
    17,
    -1.,       -0.8,      -0.6,      -0.42,     -0.24,
    -0.08,      0.08,      0.22,      0.36,      0.48,
     0.60,      0.70       0.80,      0.86,      0.92,
     0.96,      1.,
    17,
     -1.,      -0.875,    -0.75,     -0.625,    -0.50,
     -0.375,   -0.250,    -0.125,     0.0,       0.125,
      0.25,     0.375,     0.50,      0.625,     0.750,
      0.875,    1.0,
     1,
      0.,
GRID GENERATION METHOD FOR IBLOC=3
    2,
   NODE COORDINATE DATA ((XNOD(KPE,KDIM),KDIM=1,NDIM),KPE=1,NPE)
     2.5,          1.,                    2.82323,       0.67678,
     3.14645,      0.35355,               3.30866,       0.46194,
     3.5,          0.5,                   3.5,           0.75,
     3.5,          1.,                    3.,            1.,
     3.15433,      0.73097,
   NODG1,  INCR-X,-Y,-Z
    970,        1, 17, 0,
   DISCRETIZATION OF THE COMPUTATIONAL GRID (NDAT,DELX-ARRAY)
    17,
    -1.,       -0.8,      -0.6,      -0.42,     -0.24,
    -0.08,      0.08,      0.22,      0.36,      0.48,
     0.60,      0.70       0.80,      0.86,      0.92,
     0.96,      1.,
    16,
               -0.875,    -0.75,     -0.625,    -0.50,
     -0.375,   -0.250,    -0.125,     0.0,       0.125,
      0.25,     0.375,     0.50,      0.625,     0.750,
      0.875,    1.0,
     1,
      0.,
```

```
GRID GENERATION METHOD FOR IBLOC=4
  2,
   NODE COORDINATE DATA ((XNOD(KPE,KDIM),KDIM=1,NDIM),KPE=1,NPE)
     3.5,          1.,                  3.5,          0.75,
     3.5,          0.5,                 3.85355,      0.35355,
     4.,           0.,                  4.25,         0.,
     4.5,          0.,                  4.,           0.5,
     3.926775,     0.426775,
   NODG1,  INCR-X,-Y,-Z
    1225,      1, 17, 0,
   DISCRETIZATION OF THE COMPUTATIONAL GRID (NDAT,DELX-ARRAY)
    17,
     -1.,       -0.8,       -0.6,       -0.42,      -0.24,
     -0.08,      0.08,       0.22,       0.36,       0.48,
      0.60,      0.70        0.80,       0.86,       0.92,
      0.96,      1.,
    17,
      -1.0,     -0.875,     -0.75,      -0.625,     -0.50,
      -0.375,   -0.250,     -0.125,      0.0,        0.125,
       0.25,     0.375,      0.50,       0.625,      0.750,
       0.875,    1.0,
     1,
       0.,
GRID GENERATION METHOD FOR IBLOC=5
  2,
   NODE COORDINATE DATA ((XNOD(KPE,KDIM),KDIM=1,NDIM),KPE=1,NPE)
     3.5,          1.,                  4.,           0.5,
     4.5,          0.,                  4.5,          0.25,
     4.5,          0.5,                 4.14645,      0.64645,
     4.,           1.,                  3.75,         1.,
     4.07322,      0.573225,
   NODG1,  INCR-X,-Y,-Z
    1514,      16, 1, 0,
   DISCRETIZATION OF THE COMPUTATIONAL GRID (NDAT,DELX-ARRAY)
    17,
      -1.0,     -0.875,     -0.75,      -0.625,     -0.50,
      -0.375,   -0.250,     -0.125,      0.0,        0.125,
       0.25,     0.375,      0.50,       0.625,      0.750,
       0.875,    1.0,
    16,
                -0.8,       -0.6,       -0.42,      -0.24,
     -0.08,      0.08,       0.22,       0.36,       0.48,
      0.60,      0.70        0.80,       0.86,       0.92,
      0.96,      1.,
     1,
       0.,
GRID GENERATION METHOD FOR IBLOC=6
  2,
   NODE COORDINATE DATA ((XNOD(KPE,KDIM),KDIM=1,NDIM),KPE=1,NPE)
     4.5,          0.,                  5.,           0.5,
     5.5,          1.,                  5.25,         1.,
     5.,           1.,                  4.85355,      0.64645,
     4.5,          0.5,                 4.5,          0.25,
     4.926775,     0.573225,
```

```
 NODG1,  INCR-X,-Y,-Z
  1786,       17, 1, 0,
 DISCRETIZATION OF THE COMPUTATIONAL GRID (NDAT,DELX-ARRAY)
  16,
            -0.875,     -0.75,      -0.625,     -0.50,
    -0.375, -0.250,     -0.125,      0.0,        0.125,
     0.25,   0.375,      0.50,       0.625,      0.750,
     0.875,  1.0,
  17,
   -1.0,    -0.8,       -0.6,       -0.42,      -0.24,
   -0.08,    0.08,       0.22,       0.36,       0.48,
    0.60,    0.70        0.80,       0.86,       0.92,
    0.96,    1.,
   1,
     0.,
GRID GENERATION METHOD FOR IBLOC=7
  2,
 NODE COORDINATE DATA ((XNOD(KPE,KDIM),KDIM=1,NDIM),KPE=1,NPE)
    4.5,         0.,                    4.75,         0.,
    5.,          0.,                    5.14645,      0.35355,
    5.5,         0.5,                   5.5,          0.75,
    5.5,         1.,                    5.,           0.5,
    5.073225,    0.426775,
 NODG1,  INCR-X,-Y,-Z
  2058,       1, 16, 0,
 DISCRETIZATION OF THE COMPUTATIONAL GRID (NDAT,DELX-ARRAY)
  16,
            -0.8,       -0.6,       -0.42,      -0.24,
   -0.08,    0.08,       0.22,       0.36,       0.48,
    0.60,    0.70        0.80,       0.86,       0.92,
    0.96,    1.,
  17,
    -1.,    -0.875,     -0.75,      -0.625,     -0.50,
    -0.375, -0.250,     -0.125,      0.0,        0.125,
     0.25,   0.375,      0.50,       0.625,      0.750,
     0.875,  1.0,
   1,
     0.,
GRID GENERATION METHOD FOR IBLOC=8
  2,
 NODE COORDINATE DATA ((XNOD(KPE,KDIM),KDIM=1,NDIM),KPE=1,NPE)
    5.5,         1.,                    5.5,          0.75,
    5.5,         0.5,                   5.69134,      0.46194,
    5.85355,     0.35355,               6.176775,     0.676775,
    6.5,         1.,                    6.,           1.,
    5.84567,     0.73097,
 NODG1,  INCR-X,-Y,-Z
  2330,       1, 17, 0,
 DISCRETIZATION OF THE COMPUTATIONAL GRID (NDAT,DELX-ARRAY)
  17,
   -1.,     -0.8,       -0.6,       -0.42,      -0.24,
   -0.08,    0.08,       0.22,       0.36,       0.48,
    0.60,    0.70        0.80,       0.86,       0.92,
    0.96,    1.,
```

```
   16,
               -0.875,    -0.75,     -0.625,    -0.50,
     -0.375,   -0.250,    -0.125,     0.0,       0.125,
      0.25,     0.375,     0.50,      0.625,     0.750,
      0.875,    1.0,
     1,
      0.,
GRID GENERATION METHOD FOR IBLOC=9
   2,
  NODE COORDINATE DATA ((XNOD(KPE,KDIM),KDIM=1,NDIM),KPE=1,NPE)
     5.85355,     0.35355,              5.96194,     0.19134,
     6.,          0.,                   6.25,        0.,
     6.5,         0.,                   6.5,         0.5,
     6.5,         1.,                   6.176775,    0.676775,
     6.23097,     0.34567,
  NODG1,  INCR-X,-Y,-Z
   2602,     17, 1, 0,
  DISCRETIZATION OF THE COMPUTATIONAL GRID (NDAT,DELX-ARRAY)
   16,
               -0.875,    -0.75,     -0.625,    -0.50,
     -0.375,   -0.250,    -0.125,     0.0,       0.125,
      0.25,     0.375,     0.50,      0.625,     0.750,
      0.875,    1.0,
   17,
    -1.,        -0.96,     -0.92,     -0.86,     -0.80,
    -0.70,      -0.60,     -0.48,     -0.36,     -0.22,
    -0.08,       0.08,      0.24,      0.42,      0.6,
     0.8,        1.0,
    1,
     0.,
GRID GENERATION METHOD FOR IBLOC=10
   2,
  NODE COORDINATE DATA ((XNOD(KPE,KDIM),KDIM=1,NDIM),KPE=1,NPE)
     6.5,         0.,                  23.75,        0.,
    41.0,         0.,                  41.,          0.5,
    41.0,         1.,                  23.75,        1.,
     6.5,         1.,                   6.5,         0.5,
    23.75,        0.5,
  NODG1,  INCR-X,-Y,-Z
   2874,     17, 1, 0,
  DISCRETIZATION OF THE COMPUTATIONAL GRID (NDAT,DELX-ARRAY)
   88,
     -0.997,   -0.994,   -0.991,   -0.988,   -0.985,
     -0.982,   -0.979,   -0.976,   -0.973,   -0.97,
     -0.967,   -0.964,   -0.961,   -0.958,   -0.954,
     -0.95,    -0.945,   -0.94,    -0.934,   -0.928,
     -0.921,   -0.914,   -0.906,   -0.898,   -0.889,
     -0.88,    -0.87,    -0.86,    -0.848,   -0.836,
     -0.823,   -0.81,    -0.794,   -0.778,   -0.76,
     -0.742,   -0.722,   -0.702,   -0.68,    -0.658,
     -0.635,   -0.612,   -0.588,   -0.564,   -0.54,
     -0.516,   -0.492,   -0.468,   -0.443,   -0.418,
     -0.393,   -0.368,   -0.34,    -0.312,   -0.282,
     -0.252,   -0.221,   -0.19,    -0.155,   -0.12,
```

```
      -0.08,    -0.04,     0.0,      0.04,     0.08,
       0.12,     0.16,     0.20,     0.24,     0.28,
       0.32,     0.36,     0.40,     0.44,     0.48,
       0.52,     0.56,     0.60,     0.64,     0.68,
       0.72,     0.76,     0.80,     0.84,     0.88,
       0.92,     0.96,     1.,
   17,
      -1.0,    -0.875,    -0.75,     -0.625,    -0.50,
      -0.375,  -0.250,    -0.125,     0.0,       0.125,
       0.25,    0.375,     0.50,      0.625,     0.750,
       0.875,   1.0,
    1,
       0.,
MATERIAL PROPERTY OF THE FLUID --------------------------------
    VISCY(RE=40),  DENSY,   BFX(1,2)
     0.030625,     1.225,    0., 0.,
ITERATION PARAMETERS ------------------------------------------
     MAXIT, RELAX(1-10), CNVCF(1-10)
       50,
    0.8,     0.8,     1.,      1.,      1.,
    1.,      1.,      1.,      1.,      1.,
    1.E-4,   1.E-4,   1.E-4,   1.E-4,   1.E-4,
    1.E-4,   1.E-4,   1.E-4,   1.E-4,   1.E-4,
IA01 ---------------------------------------------------------
   NREC FOR INITIAL GUESS
        0,
   DBC FOR U
        9,
           1,     17,     1,     1.,
         953,    969,     1,     0.,
         986,   1241,    17,     0.,
        1258,   1513,    17,     0.,
        1529,   1785,    16,     0.,
        1802,   2057,    17,     0.,
        2073,   2329,    16,     0.,
        2346,   2601,    17,     0.,
        2602,   2857,    17,     0.,
IA02 ---------------------------------------------------------
   NREC FOR INITIAL GUESS
        0,
   DBC FOR V
       21,
           1,     17,     1,     0.,
           1,    681,    17,     0.,
          17,    697,    17,     0.,
         698,    953,    17,     0.,
         953,    969,     1,     0.,
         970,   1225,    17,     0.,
         986,   1241,    17,     0.,
        1258,   1513,    17,     0.,
        1497,   1513,     1,     0.,
        1514,   1529,     1,     0.,
        1529,   1785,    16,     0.,
        1802,   2057,    17,     0.,
```

```
          2041,  2057,    1,    0.,
          2058,  2073,    1,    0.,
          2073,  2329,   16,    0.,
          2330,  2585,   17,    0.,
          2346,  2601,   17,    0.,
          2602,  2857,   17,    0.,
          2857,  2873,    1,    0.,
          2874,  4353,   17,    0.,
          2890,  4369,   17,    0.,
IA04 --(PBCDAT, IPNOD1, IPNOD2)------------------------
            0.,   -953,   953,
END OF INPUT DATA
****PROCESSOR FOR NAVIER-STOKES EQUATIONS ****
****END                                   ****
******************************** BOTTOM OF DATA ********************************
```

A.2.4 Channel Flow with an Internal Blockage

```
******************************** TOP OF DATA ********************************
--- CHANNEL FLOW WITH BLOCKAGE --- BLOCK5 ---
****INITIALIZE DIMENSIONED VARIABLES ****
****PREPARE INPUT DATA              ****
DESCRIPTIVE DATA ------------------------------------------------
         IFLOW, NDIM, NGAUS, MFRONF,
           6,     2,    3,     95,
CNTL PARAMETERS ------------------------------------------------
         NNODE, NELEM, IAXSY, IPLOT,
          2085   490,    0;     1,
ELEMENT CONNECTIVITY DATA FOR THE GLOBAL DOMAIN --------------
     NUMBER OF BLOCKS (NBLOC)
             13
  NODE CONNECTIVITY DATA FOR IBLOC=1 AND 2 (IEL1,NODES) - B1
     1,   1, 26, 51, 52, 53, 28, 3, 2, 27,
     NEL(KDIM),INCREL(KDIM), INCNOD(KDIM)
         10,        12,        50,50,50, 50,50,50, 50,50,50,
         12,         1,         2, 2, 2,  2, 2, 2,  2, 2, 2,
          1,         0,         0, 0, 0,  0, 0, 0,  0, 0, 0,
  NODE CONNECTIVITY DATA FOR IBLOC=3-1 (IEL1,NODES) - B2
    121, 501, 526, 541, 542, 543, 528, 503, 502, 527,
     NEL(KDIM),INCREL(KDIM), INCNOD(KDIM)
          1,         0,         0, 0, 0,  0, 0, 0,  0, 0, 0,
          7,         1,         2, 2, 2,  2, 2, 2,  2, 2, 2,
          1,         0,         0, 0, 0,  0, 0, 0,  0, 0, 0,
  NODE CONNECTIVITY DATA FOR IBLOC=3-2 (IEL1,NODES) - B3
    128,  541, 556, 571, 572, 573, 558, 543, 542, 557,
     NEL(KDIM), INCREL(KDIM), INCNOD(KDIM)
          1,         0,         0, 0, 0,  0, 0, 0,  0, 0, 0,
          7,         1,         2, 2, 2,  2, 2, 2,  2, 2, 2,
          1,         0,         0, 0, 0,  0, 0, 0,  0, 0, 0,
  NODE CONNECTIVITY DATA FOR IBLOC=4-1 (IEL1,NODES) - B4
    135,  585,586,601,  602,603,588,  555,570,587,
     NEL(KDIM), INCREL(KDIM), INCNOD(KDIM)
          1,         0,         0, 0, 0,  0, 0, 0,  0, 0, 0,
          1,         0,         0, 0, 0,  0, 0, 0,  0, 0, 0,
          1,         0,         0, 0, 0,  0, 0, 0,  0, 0, 0,
  NODE CONNECTIVITY DATA FOR IBLOC=4-2 (IEL1,NODES) - B5
    136,  555,588,603,  604,605,590,  515,540,589,
     NEL(KDIM), INCREL(KDIM), INCNOD(KDIM)
          1,         0,         0, 0, 0,  0, 0, 0,  0, 0, 0,
          1,         0,         0, 0, 0,  0, 0, 0,  0, 0, 0,
          1,         0,         0, 0, 0,  0, 0, 0,  0, 0, 0,
  NODE CONNECTIVITY DATA FOR IBLOC=5-1 (IEL1,NODES) - B6
    137,  515,590,605,  606,607,592,  517,516,591,
     NEL(KDIM), INCREL(KDIM), INCNOD(KDIM)
          1,         0,         0, 0, 0,  0, 0, 0,  0, 0, 0,
          5,         1,         2, 2, 2,  2, 2, 2,  2, 2, 2,
          1,         0,         0, 0, 0,  0, 0, 0,  0, 0, 0,
  NODE CONNECTIVITY DATA FOR IBLOC=4-3 AND 5-2 (IEL1,NODES) - B7
    142,  601,616,631,  632,633,618,  603,602,617,
     NEL(KDIM), INCREL(KDIM), INCNOD(KDIM)
          5,         7,        30,30,30, 30,30,30, 30,30,30,
          7,         1,         2, 2, 2,  2, 2, 2,  2, 2, 2,
```

```
          1,        0,         0, 0, 0,  0, 0, 0,  0, 0, 0,
  NODE CONNECTIVITY DATA FOR IBLOC=6-1 (IEL1,NODES) - B8
    177,  766,780,794,  795,796,782,  768,767,781,
     NEL(KDIM), INCREL(KDIM), INCNOD(KDIM)
          2,        7,        28,28,28, 28,28,28, 28,28,28,
          6,        1,         2, 2, 2,  2, 2, 2,  2, 2, 2,
          1,        0,         0, 0, 0,  0, 0, 0,  0, 0, 0,
  NODE CONNECTIVITY DATA FOR IBLOC=6-2 (IEL1,NODES) - B9
    183,  778,792,806,  807,753,752,  751,779,793,
     NEL(KDIM), INCREL(KDIM), INCNOD(KDIM)
          2,        7,        28,28,28, 28, 2, 2,  2,28,28,
          1,        0,         0, 0, 0,  0, 0, 0,  0, 0, 0,
          1,        0,         0, 0, 0,  0, 0, 0,  0, 0, 0,
  NODE CONNECTIVITY DATA FOR IBLOC=7-1 (IEL1,NODES) - B10
    191,  822,836,861,  862,863,838,  824,823,837,
     NEL(KDIM), INCREL(KDIM), INCNOD(KDIM)
          1,        0,         0, 0, 0,  0, 0, 0,  0, 0, 0,
          6,        1,         2, 2, 2,  2, 2, 2,  2, 2, 2,
          1,        0,         0, 0, 0,  0, 0, 0,  0, 0, 0,
  NODE CONNECTIVITY DATA FOR IBLOC=7-2 (IEL1,NODES) - B11
    197,  834,848,873,  874,875,850,  755,835,849,
     NEL(KDIM), INCREL(KDIM), INCNOD(KDIM)
          1,        0,         0, 0, 0,  0, 0, 0,  0, 0, 0,
          1,        0,         0, 0, 0,  0, 0, 0,  0, 0, 0,
          1,        0,         0, 0, 0,  0, 0, 0,  0, 0, 0,
  NODE CONNECTIVITY DATA FOR IBLOC=8-1 (IEL1,NODES) - B12
    198,  755,850,875,  876,877,852,  757,756,851,
     NEL(KDIM), INCREL(KDIM), INCNOD(KDIM)
          1,        0,         0, 0, 0,  0, 0, 0,  0, 0, 0,
          5,        1,         2, 2, 2,  2, 2, 2,  2, 2, 2,
          1,        0,         0, 0, 0,  0, 0, 0,  0, 0, 0,
  NODE CONNECTIVITY DATA FOR IBLOC=7-3 AND 8-2 (IEL1,NODES) - B13
    203,  861,886,911,  912,913,888,  863,862,887,
     NEL(KDIM), INCREL(KDIM), INCNOD(KDIM)
         24,       12,        50,50,50, 50,50,50, 50,50,50,
         12,        1,         2, 2, 2,  2, 2, 2,  2, 2, 2,
          1,        0,         0, 0, 0,  0, 0, 0,  0, 0, 0,
NODE COORDINATE DATA -----------------------------------------
  NUMBER OF BLOCKS (NBLOC)
     8,
  1. GRID GENERATION METHOD FOR IBLOC=1 (METHOD)
        2,
      GLOBAL NODE COORD. DATA (XNOD(KPE,KDIM),KDIM=1,NDIM),KPE=1,NPE)
        0.,  0.,       3.375, 0.,      6.75,  0.,
        6.75,0.625,    6.75, 1.25,     3.375, 1.25,
        0.,  1.25,     0.,    0.625,   3.375, 0.625,
      NODG1,INCRX,INCRY,INCRZ,
        1,    25,   1,    0,
      NDAT, GRID COORDINDATE DATA
        21,    -1.,      -0.852,  -0.704,   -0.555,   -0.406,
               -0.258, -0.110,    0.038,    0.186,    0.312,
                0.438,  0.534,    0.630,    0.704,    0.778,
                0.830,  0.882,    0.919,    0.956,    0.978,
                1.0,
```

```
      15,    -1.,     -0.92,    -0.84,    -0.56,    -0.28,
             -0.04,    0.20,     0.32,     0.44,     0.52,
              0.60,    0.68,     0.76,     0.88,     1.0,
       1,     0.,
2. GRID GENERATION METHOD FOR IBLOC=2 (METHOD)
      2,
    GLOBAL NODE COORD. DATA (XNOD(KPE,KDIM),KDIM=1,NDIM),KPE=1,NPE)
      0.,  1.25,     3.375,1.25,     6.75, 1.25,
      6.75,1.875,    6.75, 2.5,      3.375,2.5,
      0.,  2.5,      0.,   1.875,    3.375,1.875,
    NODG1,INCRX,INCRY,INCRZ,
      15,   25,   1,    0,
    NDAT, GRID COORDINDATE DATA
      21,    -1.,     -0.852,   -0.704,   -0.555,   -0.406,
             -0.258,  -0.110,    0.038,    0.186,    0.312,
              0.438,   0.534,    0.630,    0.704,    0.778,
              0.830,   0.882,    0.919,    0.956,    0.978,
              1.0,
      11,    -1.,     -0.88,    -0.76,    -0.52,    -0.28,
              0.0,     0.28,     0.56,     0.84,     0.92,
              1.0,
       1,     0.,
3. GRID GENERATION METHOD FOR IBLOC=3 (METHOD)
      2,
    GLOBAL NODE COORD. DATA (XNOD(KPE,KDIM),KDIM=1,NDIM),KPE=1,NPE)
      6.75,0.,       6.875,0.,       7.,   0.,
      7.,  0.5,      7.,   1.,       6.875,1.125,
      6.75,1.25,     6.75, 0.625,    6.875,0.5625,
    NODG1,INCRX,INCRY,INCRZ,
     526,   15,   1,    0,
    NDAT, GRID COORDINDATE DATA
       4,             -0.4,      0.2,      0.6,      1.0,
      15,    -1.,     -0.92,    -0.84,    -0.56,    -0.28,
             -0.04,    0.20,     0.32,     0.44,     0.52,
              0.60,    0.68,     0.76,     0.88,     1.0,
       1,     0.,
4. GRID GENERATION METHOD FOR IBLOC=4 (METHOD)
      2,
    GLOBAL NODE COORD. DATA (XNOD(KPE,KDIM),KDIM=1,NDIM),KPE=1,NPE)
      7.,     1.,        7.5,  1.,       8.,   1.,
      8.125, 1.125,      8.25, 1.25,     7.5,  1.25,
      6.75,  1.25,       6.875,1.125,    7.5,  1.125
    NODG1,INCRX,INCRY,INCRZ,
     586,   15,   1,    0,
    NDAT, GRID COORDINDATE DATA
      12,             -0.83,    -0.67,    -0.5,     -0.33,
             -0.17,    0.,       0.17,     0.33,     0.5,
              0.67,    0.83,     1.,
       5,    -1.,     -0.6,     -0.2,      0.4,      1.,
       1,     0.,
5. GRID GENERATION METHOD FOR IBLOC=5 (METHOD)
      2,
    GLOBAL NODE COORD. DATA (XNOD(KPE,KDIM),KDIM=1,NDIM),KPE=1,NPE)
      6.75,  1.25,     7.5,   1.25,      8.25,  1.25,
```

```
      8.25,  1.875,   8.25,  2.5,      7.5,   2.5,
      6.75,  2.5,     6.75,  1.875,    7.5,   1.875,
    NODG1,INCRX,INCRY,INCRZ,
     590,   15,   1,    0,
    NDAT, GRID COORDINDATE DATA
      12,            -0.83,    -0.67,    -0.5,     -0.33,
            -0.17,   0.,        0.17,     0.33,     0.5,
             0.67,   0.83,      1.,
      11,   -1.,    -0.88,    -0.76,    -0.52,    -0.28,
             0.0,    0.28,     0.56,     0.84,     0.92,
             1.0,
       1,    0.,
6. GRID GENERATION METHOD FOR IBLOC=6 (METHOD)
      2,
    GLOBAL NODE COORD. DATA (XNOD(KPE,KDIM),KDIM=1,NDIM),KPE=1,NPE)
      8.,  0.,       8.125, 0.,      8.25, 0.,
      8.25, 0.625,  8.25,  1.25,    8.125, 1.125,
      8.,  1.,       8.,   0.5,     8.125, 0.5625,
    NODG1,INCRX,INCRY,INCRZ,
     766,   14,   1,    0,
    NDAT, GRID COORDINDATE DATA
       5,   -1.,    -0.6,     -0.2,      0.4,      1.0,
      14,   -1.,    -0.92,    -0.84,    -0.56,    -0.28,
            -0.04,   0.20,     0.32,     0.44,     0.52,
             0.60,   0.68,     0.76,     0.88,
       1,    0.,
7. GRID GENERATION METHOD FOR IBLOC=7 (METHOD)
      2,
    GLOBAL NODE COORD. DATA (XNOD(KPE,KDIM),KDIM=1,NDIM),KPE=1,NPE)
      8.25,  0.,      19.125, 0.,     30.,     0.,
     30.0,   0.625,  30.0,   1.25,   19.125, 1.25,
      8.25,  1.25,    8.25,  0.625,  19.125, 0.625,
    NODG1,INCRX,INCRY,INCRZ,
     836,   25,   1,    0,
    NDAT, GRID COORDINDATE DATA
      50,   -0.993, -0.986,   -0.975,   -0.964,   -0.948,
            -0.932, -0.909,   -0.886,   -0.856,   -0.826,
            -0.787, -0.748,   -0.702,   -0.656,   -0.610,
            -0.564, -0.518,   -0.472,   -0.426,   -0.380,
            -0.334, -0.288,   -0.242,   -0.196,   -0.150,
            -0.104, -0.058,   -0.012,    0.034,    0.080,
             0.126,  0.172,    0.218.    0.264,    0.310,
             0.356,  0.402,    0.448,    0.494,    0.540,
             0.586,  0.632,    0.678,    0.724,    0.770,
             0.816,  0.862,    0.908,    0.954,    1.0,
      15,   -1.,    -0.92,    -0.84,    -0.56,    -0.28,
            -0.04,   0.20,     0.32,     0.44,     0.52,
             0.60,   0.68,     0.76,     0.88,     1.0,
       1,    0.,
8. GRID GENERATION METHOD FOR IBLOC=8 (METHOD)
      2,
    GLOBAL NODE COORD. DATA (XNOD(KPE,KDIM),KDIM=1,NDIM),KPE=1,NPE)
      8.25,  1.25,    19.125, 1.25,     30.0,   1.25,
     30.0,   1.875,   30.0,   2.5,      19.125, 2.5,
```

```
     8.25,  2.5,      8.25,  1.875,     19.125, 1.875,
   NODG1,INCRX,INCRY,INCRZ,
    850,   25,   1,    0,
   NDAT, GRID COORDINDATE DATA
     50,    -0.993, -0.986,  -0.975,  -0.964,  -0.948,
            -0.932, -0.909,  -0.886,  -0.856,  -0.826,
            -0.787, -0.748,  -0.702,  -0.656,  -0.610,
            -0.564, -0.518,  -0.472,  -0.426,  -0.380,
            -0.334, -0.288,  -0.242,  -0.196,  -0.150,
            -0.104, -0.058,  -0.012,   0.034,   0.080,
             0.126,  0.172,   0.218,   0.264,   0.310,
             0.356,  0.402,   0.448,   0.494,   0.540,
             0.586,  0.632,   0.678,   0.724,   0.770,
             0.816,  0.862,   0.908,   0.954,   1.0,
     11,    -1.,    -0.88,   -0.76,   -0.52,   -0.28,
             0.0,    0.28,    0.56,    0.84,    0.92,
             1.0,
      1,    0.,
MATERIAL PROPERTY OF FLUID --- (RE=500) ------------------------
      VISCY,     DENSY,     BFX(1-2),
      0.006125,    1.225,    0., 0.,
ITERATION PARAMETERS -------------------------------------------
    MAXIT, RELAX(1-10), CNVCF(1-10),
      50,
     0.8,     0.8,     1.,    1.,      1.,
     1.,      1.,      1.,    1.,      1.,
     1.E-6,  1.E-6,  1.E-4,  1.E-6,  1.E-4,
     1.E-4,  1.E-4,  1.E-4,  1.E-4,  1.E-4,
IA01 ----------------------------------------------------------
   INITIAL GUESS FOR U-VELOCITY (NREC)
       0,
   DBC FOR U
      35,
            1,    1,    1,      0.,
            2,    2,    1,      0.0784,
            3,    3,    1,      0.1536,
            4,    4,    1,      0.3916
            5,    5,    1,      0.5904,
            6,    6,    1,      0.7296,
            7,    7,    1,      0.84,
            8,    8,    1,      0.8844,
            9,    9,    1,      0.9216,
           10,   10,    1,      0.9424,
           11,   11,    1,      0.96,
           12,   12,    1,      0.9744,
           13,   13,    1,      0.9856,
           14,   14,    1,      0.9964,
           15,   15,    1,      1.0,
           16,   16,    1,      0.9964,
           17,   17,    1,      0.9856,
           18,   18,    1,      0.9424,
           19,   19,    1,      0.8704,
           20,   20,    1,      0.75,
           21,   21,    1,      0.5904,
```

```
          22,  22,   1,      0.3916,
          23,  23,   1,      0.1536,
          24,  24,   1,      0.0.0784,
          25,  25,   1,      0.,
           1, 501,  25,      0.,
          25, 525,  25,      0.,
         526, 571,  15,      0.,
         571, 585,   1,      0.,
         586, 751,  15,      0.,
         600, 765,  15,      0.,
         766, 779,   1,      0.,
         780, 822,  14,      0.,
         836,2061,  25,      0.,
         860,2085,  25,      0.,
IA02 ---------------------------------------------------------------
     INITIAL GUESS FOR V-VELOCITY (NREC)
         0,
     DBC FOR V
         35,
           1,   1,   1,      0.,
           2,   2,   1,      0.,
           3,   3,   1,      0.,
           4,   4,   1,      0.,
           5,   5,   1,      0.,
           6,   6,   1,      0.,
           7,   7,   1,      0.,
           8,   8,   1,      0.,
           9,   9,   1,      0.,
          10,  10,   1,      0.,
          11,  11,   1,      0.,
          12,  12,   1,      0.,
          13,  13,   1,      0.,
          14,  14,   1,      0.,
          15,  15,   1,      0.,
          16,  16,   1,      0.,
          17,  17,   1,      0.,
          18,  18,   1,      0.,
          19,  19,   1,      0.,
          20,  20,   1,      0.,
          21,  21,   1,      0.,
          22,  22,   1,      0.,
          23,  23,   1,      0.,
          24,  24,   1,      0.,
          25,  25,   1,      0.,
           1, 501,  25,      0.,
          25, 525,  25,      0.,
         526, 571,  15,      0.,
         571, 585,   1,      0.,
         586, 751,  15,      0.,
         600, 765,  15,      0.,
         766, 779,   1,      0.,
         780, 822,  14,      0.,
         836,2061,  25,      0.,
         860,2085,  25,      0.,
```

```
IA04 ---- (PBCDAT, IPNODE(1-2))------------------------
             0.,     -571,  571,
END OF INPUT DATA
****PROCESSOR FOR NAVIER-STOKES EQUATIONS ****
****END OF RUN                            ****
******************************** BOTTOM OF DATA ********************************
```

APPENDIX III

DESCRIPTION OF THE SUBROUTINES

INITAL - Initialize the dimensioned variables.

BLKDAT - Define the program control parameters, and set the Gauss numerical quadrature data in each coordinate direction.

DATLIB - Define the flow element to be used, and set the Gauss Numerical quadrature data for the computational element.

PREP - Prepare the input data.

RNODE - Generate the node coordinate data.

RELEM - Generate the node connectivity data.

RINIT - Read in the initial guess.

RBC1 - Read in the boundary condition data.

FEMDAT - Read in the re-start data.

ISOPEL - Compute the interpolation polynomials and the derivatives.

LSHP1 - Shape functions for one-dimensional linear element.

LSHP2 - Shape functions for one-dimensional quadratic element.

SHAP01 - Shape function for two-dimensional constant element.

SHAP02 - Shape functions for triangular element.

SHAP23 - Shape functions for bi-quadratic quadrilateral element.

PROCES - Processor for Navier-Stokes equations.

PFRONT - Pre-processor for the frontal solver.

SHPLIB - Save the shape functions on a disk file (logical unit = 2), and read the data whenever necessary.

S1FLOW - Create the sequential degree-of-freedom number for each flow variable, and compute the total degrees of freedom.

SEQVFL - Include boundary conditions into the global solution vector.

SFLOW - Solve the Navier-Stokes equations iteratively.

FRONTS - Frontal solver.

ELEMFL - Compute the element system of equations.

SCNVFL - Check the convergence.

SPRS4 - Compute the nodal pressure.

PFLOW - Print out the computational results.

USER - Load the coordinate data and the flow variables for each element.

APPROVAL

VELOCITY-PRESSURE INTEGRATED VERSUS PENALTY FINITE ELEMENT METHODS FOR HIGH REYNOLDS NUMBER FLOWS

By Sang-Wook Kim

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

G. F. McDONOUGH
Director, Structures and Dynamics Laboratory

☆U.S. GOVERNMENT PRINTING OFFICE: 1988 - 530-050/80082