A VELOCITY-PRESSURE INTEGRATED, MIXED INTERPOLATION, GALERKIN FINITE ELEMENT METHOD FOR HIGH REYNOLDS NUMBER LAMINAR FLOWS

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

Final Report

Prepared for
NASA-Marshall Space Flight Center
Marshall Space Flight Center, Alabama 35812
A velocity-pressure integrated, mixed interpolation, Galerkin finite element method for the Navier-Stokes equations is presented. In the method, the velocity variables have been interpolated using complete quadratic shape functions and the pressure has been interpolated using linear shape functions. For the two-dimensional case, the pressure is defined on a triangular element which is contained inside the complete bi-quadratic element for velocity variables; and for the three-dimensional case, the pressure is defined on a tetrahedral element which is again contained inside the complete tri-quadratic element. Thus the pressure is discontinuous across the element boundaries. Example problems considered include: a cavity flow for Reynolds number of 400 through 10,000; a laminar backward-facing step flow; and a laminar flow in a square duct of strong curvature. The computational results compared favorably with those of the finite difference methods as well as experimental data available. It was found that: the present method could capture the delicate pressure driven recirculation zones; the method yielded accurate velocity and pressure distributions; and the method required much fewer number of grid points than the finite difference methods to obtain comparable computational results. A finite element computer program (NSFLOW/L) for incompressible, laminar flows is also presented in this report.
ACKNOWLEDGMENTS

The author would like to express his thanks to Dr. C.-P. Chen of the University of Alabama in Huntsville for bringing the author's attention to many references for this work. Thanks are also due to Dr. Rand Decker for many suggestions in preparing the report.
# TABLE OF CONTENTS

I. INTRODUCTION ........................................................ 1

II. FINITE ELEMENT EQUATIONS ........................................ 2
   2.1 The Navier-Stokes Equations .................................... 2
   2.2 Method of Weighted Residuals and the Galerkin Finite Element Method ........................................... 3
   2.3 Mixed Interpolation Methods for Velocity and Pressure ................................................... 7

III. EXAMPLE PROBLEMS ................................................ 10
   3.1 Lid-Driven Cavity Flow ........................................ 10
   3.2 Backward-Facing Step Flow .................................... 19
   3.3 Laminar Flow in a Square Duct of Strong Curvature ............ 24

IV. CONCLUSIONS AND DISCUSSION .................................... 31

REFERENCES ................................................................. 32

APPENDIX I. Finite Element Computer Program (NSFLOW/L)  
for Incompressible, Laminar Flows ................................... 35

APPENDIX II. Input Data for NSFLOW/L ............................. 73
   A.2.1 Cavity Flow for Re = 10,000 ............................ 76
   A.2.2 Backward-Facing Step Flow ............................... 78
   A.2.3 Laminar Flow in a Square Duct of Strong Curvature .... 81

APPENDIX III. Description of the Subroutines ....................... 91
<table>
<thead>
<tr>
<th>Figure</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Flow element</td>
<td>8</td>
</tr>
<tr>
<td>2.</td>
<td>Configuration, coordinates, and nomenclature of cavity flow</td>
<td>10</td>
</tr>
<tr>
<td>3.</td>
<td>Discretization of the cavity flow</td>
<td>12</td>
</tr>
<tr>
<td>4.</td>
<td>Velocity vectors for cavity flow</td>
<td>13</td>
</tr>
<tr>
<td>5.</td>
<td>Streamlines for cavity flow</td>
<td>14</td>
</tr>
<tr>
<td>6.</td>
<td>Pressure contours for cavity flow</td>
<td>15</td>
</tr>
<tr>
<td>7.</td>
<td>Horizontal velocity profiles for cavity flow at x = 0.5</td>
<td>16</td>
</tr>
<tr>
<td>8.</td>
<td>Configuration, coordinates, and nomenclature of backward-facing step flow</td>
<td>19</td>
</tr>
<tr>
<td>9.</td>
<td>Discretization of the backward-facing step flow</td>
<td>20</td>
</tr>
<tr>
<td>10.</td>
<td>Velocity vectors for the backward-facing step flow</td>
<td>21</td>
</tr>
<tr>
<td>11.</td>
<td>Streamlines for the backward-facing step flow</td>
<td>22</td>
</tr>
<tr>
<td>12.</td>
<td>Pressure contours for the backward-facing step flow</td>
<td>23</td>
</tr>
<tr>
<td>13.</td>
<td>Reattachment length versus Reynolds number</td>
<td>25</td>
</tr>
<tr>
<td>14.</td>
<td>Wall pressure for backward-facing step flow</td>
<td>26</td>
</tr>
<tr>
<td>15.</td>
<td>Configuration of the laminar flow in a square duct of strong curvature</td>
<td>27</td>
</tr>
<tr>
<td>16.</td>
<td>Discretization of the flow domain</td>
<td>28</td>
</tr>
<tr>
<td>17.</td>
<td>Velocity vectors on the curved section</td>
<td>29</td>
</tr>
<tr>
<td>18.</td>
<td>Secondary recirculation flows</td>
<td>30</td>
</tr>
</tbody>
</table>
## LIST OF TABLES

<table>
<thead>
<tr>
<th>Table</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Streamline Contour Label for Cavity Flow</td>
<td>17</td>
</tr>
<tr>
<td>2.</td>
<td>Pressure Contour Label for Cavity Flow</td>
<td>17</td>
</tr>
<tr>
<td>3.</td>
<td>Stream Function Values at the Center of Vortices for Cavity Flow</td>
<td>18</td>
</tr>
<tr>
<td>4.</td>
<td>Streamline Contour Label for Backward-Facing Step Flow</td>
<td>24</td>
</tr>
<tr>
<td>5.</td>
<td>Pressure Contour Label for Backward-Facing Step Flow</td>
<td>24</td>
</tr>
</tbody>
</table>
CONTRACTOR REPORT

A VELOCITY-PRESSURE INTEGRATED, MIXED INTERPOLATION, GALERKIN FINITE ELEMENT METHOD FOR HIGH REYNOLDS NUMBER LAMINAR FLOWS

I. INTRODUCTION

The finite element methods for the Navier-Stokes equations using the primitive variables of velocity and pressure can be categorized into three groups, in general. These are the velocity-pressure integrated mixed interpolation methods, the penalty methods, and the segregated velocity-pressure solution methods.

In the velocity-pressure integrated, mixed interpolation methods, the order of interpolating polynomial for velocity is chosen to be one order higher than that of pressure. Unfortunately, many velocity-pressure integrated, mixed interpolation methods yield inaccurate pressure which becomes more severe as the Reynolds number is increased. Details of the method can be found in Taylor and Hughes [11] and the references therein.

In the penalty method, the pressure is pre-eliminated from the Navier-Stokes equations by penalizing the conservation of mass equation, and hence, the continuity condition is approximately satisfied. If necessary, pressure can be recovered using the penalized conservation of mass equation in the post process. Details of various penalty methods and the computational results can be found in Zienkiewicz et al. [21], Engelman et al. [31], Kikuchi et al. [41], and Heinrich and Marshall [51], among many others.

Due to the shortcomings of these two classes of methods, and partly influenced by the success of the finite difference methods based on segregated formulation of the Navier-Stokes equations, such as the SIMPLE (Semi-Implicit Method for Pressure-Linked Equations) algorithms [61], a few finite element methods adopting the segregated formulation of the Navier-Stokes equations began to appear in recent years [7,8,9]. But the computational results thus obtained did not show significant improvement in accuracy over the previous two classes of methods.

The finite element method has certain advantages over the finite difference methods. These advantages are: capability to model complicated domain more precisely, capability to include various boundary conditions more naturally, and capability to show convergence nature of the method mathematically. There exists a number of example cases for which the finite element method yielded more accurate computational results than the finite difference methods for low Reynolds number flows [10]. However, for high Reynolds number flows, especially when pressure driven recirculation zones exist in the flow field, the finite difference methods yielded more accurate computational results than the finite element methods. Thus, the advantages of the finite element method have not been well demonstrated for high Reynolds number viscous flows, as yet.

In this report, a velocity-pressure integrated, mixed interpolation, Galerkin finite element method for high Reynolds number flows is presented. The finite element system of equations for the Navier-Stokes equations has been obtained using the Galerkin finite element method, and the system of equations has been solved using a
frontal solver [1,11]. The present method yielded accurate computational results for low Reynolds number flows as well as high Reynolds number flows. It was also found that the method could capture subtle pressure driven recirculation zones, and that the computational results were free of numerical wiggles for high Reynolds number flows.

II. FINITE ELEMENT EQUATIONS

A finite element method for two- and three-dimensional steady, incompressible, laminar flows is described below. The method is based on the velocity-pressure integrated formulation of the Navier-Stokes equations using a mixed interpolation of velocity and pressure.

In the following discussions, consistent notations have been used throughout, and repeated indices imply summation over the indices, unless otherwise specified.

2.1 The Navier-Stokes Equations

The form of the Navier-Stokes equations used herein are given as:

\[
\begin{align*}
\rho \frac{\partial u_i}{\partial x_j} - \frac{\partial \tau_{ij}}{\partial x_j} &= b_i \\
\frac{\partial u_j}{\partial x_j} &= 0
\end{align*}
\]

in \( \Omega \)  

where

\[
\tau_{ij} = 2\mu \varepsilon_{ij} - \nabla \delta_{ij}
\]

\[
\varepsilon_{ij} = \frac{1}{2} \left( \frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right)
\]

\( \Omega \) is the open bounded domain of the problem, the subscripts \( i \) and \( j \) denote the coordinate directions, \( \rho \) is the density, \( u_i \) is the velocity component in the \( i \)-th coordinate direction, \( p \) is the pressure, \( \nu \) is the molecular viscosity of the fluid, \( b_i \) is the body force in the \( i \)-th coordinate direction, \( \tau_{ij} \) is the stress tensor, \( \varepsilon_{ij} \) is the strain rate tensor, and \( \delta_{ij} \) 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:
\[ u = u_0(\bar{x}) \quad \text{for} \quad \bar{x} \in \partial \Omega_1 \]  
\[ T_1 = \tau_{ij} n_j \quad \text{for} \quad \bar{x} \in \partial \Omega_2 \]  

where \( \bar{x} = (x,y) \) for 2-D case and \( \bar{x} = (x,y,z) \) for 3-D case, \( \partial \Omega_1 \) is part of the boundary on which Dirichlet boundary condition is specified, \( \partial \Omega_2 \) is the rest of the boundary on which Neumann boundary condition is specified, and \( T_1 \) is the surface traction.

### 2.2 Method of Weighted Residuals and Galerkin Finite Element Method

In the context of the method of weighted residuals, the test functions for the momentum equation and the conservation of mass equation are denoted as \( W_u(\bar{x}) \) and \( W_p(\bar{x}) \), respectively. The weak form of the Navier-Stokes equations are:

\[
\int_\Omega W_u \left( \rho u_j \frac{\partial u_i}{\partial x_j} - \frac{\partial \tau_{ij}}{\partial x_j} - b_i \right) d\bar{x} = 0
\]  
\[
\int_\Omega W_p \left( \frac{\partial u_i}{\partial x_j} \right) d\bar{x} = 0
\]

Integrating by parts of the stress tensor term in equation (7) yields:

\[
\int_\Omega \left( W_u \rho u_j \frac{\partial u_i}{\partial x_j} + \frac{\partial W_u}{\partial x_j} \tau_{ij} - W_u b_i \right) d\bar{x} - \int_{\partial \Omega_2} W_u \tau_{ij} n_j d\bar{s} = 0
\]  

where \( \int_{\partial \Omega_1} W_u \tau_{ij} n_j d\bar{s} \) term has been dropped out since the test function \( W_u(\bar{x}) \) should vanish on the boundary \( \partial \Omega_1 \).

Introducing the finite element discretization into equations (9) and (8) yields:

\[
\sum_{e=1}^{E} \int_{\Omega_e} \left[ W_u \left( \rho u_j \frac{\partial u_i}{\partial x_j} \right) + \frac{\partial W_u}{\partial x_j} \tau_{ij} - W_u b_i \right] d\bar{x} - \sum_{e=1}^{E} \int_{\partial \Omega_{e2}} W_u \tau_{ij} n_j d\bar{s} = 0
\]
\[ E \sum_{e=1}^{\Omega_e} \int_{\Omega_e} \left( W_p \frac{\partial u_j}{\partial x_j} \right) \, dx = 0 \]  

where \( \Omega_e \) is a finite element, \( E \) is the total number of elements, and \( \partial \Omega_{e2} \) is the boundary of an element \( (\Omega_e) \) which lie on the part of the boundary \( \partial \Omega_2 \) for which the flux boundary condition has been specified.

The flux through inter-element boundary should be continuous and the normal vectors at the interface of the two adjacent elements are in the opposite directions. Therefore, all the inter-element fluxes in equation (10) cancel each other, and only the prescribed flux on the boundary \( \partial \Omega_2 \) contributes to the final system of equations. The consequence of eliminating the fluxes across inter-element boundaries in the finite element method is equivalent to enforcing the flux continuity condition across the inter-element boundaries.

Inserting equations (3), (4), and (6) into equation (10) yields:

\[ \sum_{e=1}^{E} \int_{\Omega_e} \left[ W_u \delta_{ij} \left( \frac{\partial u_j}{\partial x_j} + \frac{\partial u_i}{\partial x_i} \right) - \frac{\partial W_u}{\partial x_j} p \delta_{ij} - W_u b_i \right] \, dx \]

\[ - \sum_{e=1}^{E} \int_{\partial \Omega_{e2}} W_u T_i \, ds = 0 \]  

The global finite element system of equations are obtained by assembling the element system of equations. Therefore, the discrete finite element system of equations are derived for an arbitrary element in the following discussions.

Let \( \psi_m \) and \( \psi_n \) be sets of basic polynomials to interpolate velocities and pressure respectively, i.e.,

\[ u_i = u_{im} \psi_m , \quad \sum \text{on } m, \quad m = 1,M \]

\[ p = p_n \psi_n , \quad \sum \text{on } n, \quad n = 1,N \]

where \( u_{im} \) denotes the \( m \)-th nodal value of the velocity component \( u_i \), \( p_n \) denotes the \( n \)-th nodal value of pressure, and \( M \) and \( N \) are the number of velocity nodes and the number of pressure nodes in an element, respectively.

In the Galerkin finite element method, the test functions are selected from the same space of interpolating polynomials as the trial functions. Then \( W_u \) and \( W_p \) can be expanded as:
\[ W_u = a_k \phi_k \text{, sum on } k, \ k = 1, M \]  
\[ W_p = b_\ell \psi_\ell \text{, sum on } \ell, \ \ell = 1, N \]  

where \( a_k \) and \( b_\ell \) should vanish if a Dirichlet boundary condition has been specified for the corresponding degree of freedom; otherwise, \( a_k \) and \( b_\ell \) are arbitrary constants.

The finite element system of equations for an element is obtained by substituting equations (13) and (14) into equations (12) and (11), and is given as:

\[
\int_{\Omega} \left[ \phi_k \partial_{x_j} u_{i_q} \frac{\partial \phi_m}{\partial x_j} u_{i_m} + \frac{\partial \phi_k}{\partial x_j} u_{i_m} + \frac{\partial \phi_q}{\partial x_i} u_{j_q} - \frac{\partial \phi_k}{\partial x_j} \psi_n \delta_{i_j} - \phi_k b_i \right] dx
\]

\[- \int_{\partial \Omega} \phi_k T_i ds = 0 \quad (15)\]

\[
\int_{\Omega} \psi_\ell \frac{\partial \phi_k}{\partial x_j} u_{j_q} dx = 0 \quad (16)
\]

where the subscript \( q \) ranges from 1 to \( M \), and the arbitrariness of the coefficients \( a_k \) and \( b_\ell \) have been made use of in deriving equations (15) and (16).

In matrix form, equations (15) and (16) are given as, for the three-dimensional case:

\[
\begin{bmatrix}
C & 0 & 0 \\
0 & C & 0 \\
0 & 0 & C
\end{bmatrix}
\begin{bmatrix}
K_{11} & K_{12} & K_{13} \\
K_{21} & K_{22} & K_{23} \\
K_{31} & K_{32} & K_{33}
\end{bmatrix}
\begin{bmatrix}
\psi_1 \\
\psi_2 \\
\psi_3
\end{bmatrix}
\]

\[- \begin{bmatrix}
Q_{x1} \\
Q_{x2} \\
Q_{x3}
\end{bmatrix}
\{p\} = \begin{bmatrix}
f_{x1} \\
f_{x2} \\
f_{x3}
\end{bmatrix} + \{ b.c. \} \quad (17)
\]
\[
\begin{bmatrix}
Q_1^T & Q_2^T & Q_3^T
\end{bmatrix}
\begin{bmatrix}
u_1 \\
u_2 \\
u_3
\end{bmatrix} = \{Q\}
\]

where

\[
\tilde{C} = \int_{\Omega_e} \phi^T \rho (u_j T \phi) \frac{\partial \phi}{\partial x_j} \text{d}x ,
\]

\[
\tilde{K}\phi = \int_{\Omega_e} \frac{\partial \phi}{\partial x_j} u \frac{\partial \phi}{\partial x_j} \text{d}x ,
\]

\[
\tilde{K}_{ij} = \int_{\Omega_e} \frac{\partial \phi}{\partial x_j} u \frac{\partial \phi}{\partial x_i} \text{d}x ,
\]

\[
\tilde{Q}_i = \int_{\Omega_e} \frac{\partial \phi}{\partial x_i} \tilde{\psi} \text{d}x ,
\]

\[
\tilde{f}_i = \int_{\Omega_e} \phi^T b_i \text{d}x ,
\]

\(u_i\) is a column vector of nodal values of the velocity component \(u_j\), \(P\) is a column vector of nodal pressure, \(\phi\) is a column vector of interpolating polynomials for velocity, \(\tilde{\psi}\) is a column vector of interpolating polynomials for pressure, \((b.c.)\) is a column vector contributed by the specified flux boundary condition, and the subscripts \(i\) and \(j\) range from one to the number of spatial dimensions, respectively.

For the two-dimensional case, the element system of equations can be obtained by deleting the appropriate third row and column sub-matrices from equations (17) and (18).

The integrations in equations (19) through (23) were evaluated using the Gauss numerical quadrature method with three Gauss points for each coordinate direction. The assembled global system of equations was solved by a direct (Picard) iteration method using a frontal solver [1,12], and the solutions were updated using an under-relaxation method given as:

\[
a_j^* = \alpha a_j^n + (1 - \alpha) a_j^{n-1}
\]
where \( a_j \) represents any degree of freedom, \( \alpha \) is the under-relaxation number, the superscripts \( n \) and \( n-1 \) denote iteration levels, and \( a_j^* \) is the updated solution. No under-relaxation was necessary for low Reynolds number flows. For high Reynolds number flows, \( \alpha = 0.8 \) and \( \alpha = 1 \) have been used for the velocities and the pressure, respectively.

2.3 Mixed Interpolation Methods for Velocity and Pressure

The pressure interpolation polynomials used in the present study are introduced in this section. For the two-dimensional case, the velocities are interpolated using the bi-quadratic shape functions and the pressure is interpolated using the linear shape functions defined on a triangular element which is contained inside the quadratic element, as shown in Figure 1(a). The three pressure nodes are located at the three Gauss points of the three-point Gauss quadrature rule for quadrilateral elements [13], i.e., the same locations as those used in the Reduced Integration Penalty (RIP) method. The coordinates of the pressure nodes on the computational element are given as:

\[
\xi_n = \begin{cases} 
(0, \sqrt{2}/3) & \text{for } n = 1 \\
(-1/\sqrt{2}, -1/\sqrt{6}) & \text{for } n = 2 \\
(1/\sqrt{2}, -1/\sqrt{6}) & \text{for } n = 3 
\end{cases}
\]  

(25)

where \( \xi_n = (\xi_n^1, \eta_n^1) \) for the two-dimensional case, and \( n \) denotes the pressure node number. The shape functions for each of the nodes are given as:

\[
\begin{align*}
\psi_1 &= \frac{1}{3} + \frac{\sqrt{2}}{3} \eta \\
\psi_2 &= \frac{1}{3} - \frac{1}{\sqrt{2}} \xi - \frac{1}{\sqrt{6}} \eta \\
\psi_3 &= \frac{1}{3} + \frac{1}{\sqrt{2}} \xi - \frac{1}{\sqrt{6}} \eta
\end{align*}
\]  

(26)

For the three-dimensional case, the velocities are interpolated by the tri-quadratic shape functions and the pressure is interpolated using the linear shape functions defined on a tetrahedral element which is contained inside the tri-quadratic brick element, as shown in Figure 1(b). The coordinates of the pressure nodes on the computational element are given as:
Figure 1. Flow elements. (a) Two-dimensional element, (b) three-dimensional element, x: velocity nodes, o: pressure nodes.
where $\xi_n = (\xi_n, n_n, \zeta_n)$ for the three-dimensional case, and $n$ denotes the node numbers of the pressure nodes. The shape functions for each of the pressure nodes are given as:

$$
\psi_1 = \frac{1}{4} + \frac{1}{2} \frac{\sqrt{3}}{\sqrt{2}} n - \frac{\sqrt{3}}{4} \zeta \\
\psi_2 = \frac{1}{4} - \frac{1}{2} \frac{\sqrt{3}}{\sqrt{2}} n - \frac{\sqrt{3}}{4} \zeta \\
\psi_3 = \frac{1}{4} + \frac{1}{2} \frac{\sqrt{3}}{\sqrt{2}} \xi + \frac{\sqrt{3}}{4} \zeta \\
\psi_4 = \frac{1}{4} - \frac{1}{2} \frac{\sqrt{3}}{\sqrt{2}} \xi + \frac{\sqrt{3}}{4} \zeta
$$

The shape functions given in equations (26) and (30) satisfy the relationship:

$$
\psi_k(\xi_n) = \begin{cases} 
1 & \text{if } k = n \\
0 & \text{if } k \neq n
\end{cases}
$$

where $\psi_k$ is the shape function for the $k$-the pressure node and $\xi_n$ denotes the coordinates of the $n$-th pressure node. An additional pressure interpolation method tested for two-dimensional flows was $\psi_T = \{1, x, y\}$. This pressure interpolation method yielded the same computational results, up to several significant digits, as equation (26). However, the pressure interpolation polynomials given in equation (26) exhibited better convergence behavior than the other method as the number of iterations were increased, see Reference 14.
III. EXAMPLE PROBLEMS

The finite element method described in the previous sections has been tested and validated by solving a few example problems for which a vast amount of computational results and/or experiment data were available. These are: a lid driven cavity flow [5,15-19], a laminar backward-facing step flow [20-22], and a laminar flow in a square duct of strong curvature [24,25].

In the following discussions, solving the coupled momentum equation and the conservation of mass equation once is counted as an iteration. The convergence criterion used was

\[ |1 - \frac{a_{i,j}^n}{A_{i}^{n-1}}| < \epsilon , \quad i = \{u, v, w, \text{ or } p\} \text{ and } j = 1,T \]  

where \(a_{i,j}^n\) denotes the \(i\)-th flow variable for the \(j\)-th node; \(A_{i}^{n-1}\) is the maximum nodal value of the \(i\)-th flow variable in the previous iteration; \(T\) is the total number of nodes; and \(\epsilon\) is the convergence criterion. \(\epsilon = 1 \times 10^{-4}\) and \(\epsilon = 1 \times 10^{-3}\) have been used for the two- and three-dimensional flows, respectively.

For the mixed interpolation methods used herein, 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 of the contributions has been computed using equation (13).

3.1 Lid Driven Cavity Flow

The cavity flow is described in Figure 2. The no slip boundary condition, i.e., \(u = v = 0\), has been applied at all the boundaries except at \(y = 1\) where \(u = 1\) and \(v = 0\). A Dirichlet pressure boundary condition has been specified at an arbitrary

![Figure 2. Configuration, coordinates, and nomenclature of cavity flow.](image-url)
pressure node in the flow domain. The Reynolds numbers considered were 400, 1000, 3200, 5000, 7500, and 10,000; where the Reynolds number is defined as \( \text{Re} = \frac{\rho UL}{\mu} \), where \( U = 1 \) is the velocity of the lid, and \( L = 1 \) is the reference length. The various Reynolds numbers have been obtained by varying the molecular viscosity \( (\mu) \) of the fluid, with the rest of the parameters kept as constants. Two different grids, as shown in Figure 3, were used.

Computation of the cavity flow was carried out in the order of increasing the Reynolds number. Uniform zero values for both velocity and pressure were used as an initial guess for \( \text{Re} = 400 \), the converged solution for \( \text{Re} = 400 \) was used as an initial guess for \( \text{Re} = 1000 \), and so on. The required number of iterations were 17, 19, 21, 21, 26, and 30, for Reynolds numbers of 400, 1000, 3200, 5000, 7500, and 10,000, respectively.

For the coarse grid case (Grid A), convergent solutions were obtained up to Reynolds number of 3200. But the coarse grid could not resolve the smallest eddies (vortices 4 and 5 in Figure 2), and thus the grid was not tested for \( \text{Re} > 3200 \). Velocity vectors obtained by using the coarse grid for \( \text{Re} \leq 1000 \) are shown in Figures 4(a) and 4(b), respectively. The streamline contours for \( \text{Re} \leq 1000 \) obtained by using the coarse grid were qualitatively the same as those obtained by using the fine grid (Grid B), but the magnitude of the minimum stream function values at the center of the primary vortex was a few percent smaller than those obtained by using the fine grid.

The velocity vectors (\( \text{Re} \geq 3200 \)) and the streamline contours obtained by using the fine grid are shown in Figures 4 and 5, respectively. The streamline contour labels are given in Table 1. In Figure 5, it can be seen that the sizes of vortices 4 and 5 of the present computational results are smaller than those of Schreiber and Keller [16] and Ghia et al. [17].

The normalized pressure contours for all the Reynolds number cases obtained by using the fine grid are shown in Figure 6, and the pressure contour labels are given in Table 2. The normalized pressure \( (P) \) has been obtained from the static pressure \( (p) \) using a relationship given as \( P = \frac{pL_{\text{ref}}}{V_{\text{ref}}/\mu} \), where \( L_{\text{ref}} = 1 \) is the reference length, \( V_{\text{ref}} = 1 \) is the reference velocity, and \( \mu \) is the molecular viscosity of the fluid.

The horizontal velocity profiles at \( x = 0.5 \) are compared with various computational results in Figure 7. For \( \text{Re} = 400 \), the magnitude of the minimum horizontal velocity due to Burggraf [15] was approximately 12 percent smaller than that of Ghia et al. [17], and the same velocity due to Bercovier and Engelman [19] was approximately 25 percent smaller than that of Ghia et al. [17]. For \( \text{Re} = 1000 \), the horizontal velocity profile due to Bercovier and Engelman [19] further deviated from that of Ghia et al. or the present computational results. In general, the present computational results compared more favorably with those of Ghia et al. than those of Burggraf [15] and Bercovier and Engelman [19], as shown in Figure 7.

For \( \text{Re} = 10,000 \), it can be seen that the horizontal velocity profile due to Schreiber and Keller [16] is significantly different from that of Ghia et al. [17], which shows that the global computational results can be significantly different depending on the order of difference approximation used for the boundary condition. The present computational results compared more favorably with those of Ghia et al. [17] than those of Schreiber and Keller [16].
The local maximum and minimum values of the stream function at the center of the primary vortex and the first three secondary vortices are compared with those of References 16 to 18 in Table 3. It can be seen that the present computational results, obtained by using approximately one-fourth of the number of grid points used in References 16 and 17, compared favorably with these data, in general.

Figure 3. Discretization of cavity flow.
(a) Grid A, 25x25 grids (12x12 quadratic elements).
(b) Grid B, 65x65 grids (32x32 quadratic elements).
Figure 4. Velocity vectors for cavity flow, Re: Reynolds number.
Figure 5. Streamlines for cavity flow.
Figure 6. Pressure contours for cavity flow.
Figure 7. Horizontal velocity profiles of cavity flow at x = 0.5; ———: present (coarse grid), ——: present (fine grid, O: Ghia et al. (129x129 grids), x: Bercovier and Engelman (25x25 grids), Δ: Burggraf (41x41 grids), : Heinrich and Marshall (41x41 grids), +: Schreiber and Keller (180x180 grids).
TABLE 1. STREAMLINE CONTOUR LABEL FOR CAVITY FLOW

<table>
<thead>
<tr>
<th>Label</th>
<th>$\psi^*$</th>
<th>Label</th>
<th>$\psi$</th>
<th>Label</th>
<th>$\psi$</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>-0.11</td>
<td>F</td>
<td>-0.03</td>
<td>K</td>
<td>2.10^{-4}</td>
</tr>
<tr>
<td>B</td>
<td>-0.10</td>
<td>G</td>
<td>-0.01</td>
<td>L</td>
<td>5.10^{-4}</td>
</tr>
<tr>
<td>C</td>
<td>-0.09</td>
<td>H</td>
<td>-1.10^{-10}</td>
<td>M</td>
<td>1.10^{-3}</td>
</tr>
<tr>
<td>D</td>
<td>-0.07</td>
<td>I</td>
<td>1.10^{-6}</td>
<td>N</td>
<td>2.10^{-3}</td>
</tr>
<tr>
<td>E</td>
<td>-0.05</td>
<td>J</td>
<td>5.10^{-5}</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

* $\psi$: stream function

TABLE 2. PRESSURE CONTOUR LABEL FOR CAVITY FLOW

<table>
<thead>
<tr>
<th>Label</th>
<th>Reynolds Number (Re)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>400</td>
</tr>
<tr>
<td>C</td>
<td>-20.</td>
</tr>
<tr>
<td>E</td>
<td>0.</td>
</tr>
<tr>
<td>F</td>
<td>10.</td>
</tr>
<tr>
<td>G</td>
<td>30.</td>
</tr>
<tr>
<td>H</td>
<td>50.</td>
</tr>
</tbody>
</table>
### TABLE 3. STREAM FUNCTION VALUES AT THE CENTER OF VORTICES FOR CAVITY FLOW

<table>
<thead>
<tr>
<th>Primary Vortex</th>
<th>Re</th>
<th>Schreiber* &amp; Keller</th>
<th>Chia° et. al.</th>
<th>Gresho° et. al.</th>
<th>present°</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>400</td>
<td>-0.1130</td>
<td>-0.1139</td>
<td>---</td>
<td>-0.1128</td>
</tr>
<tr>
<td></td>
<td>1000</td>
<td>-0.1160</td>
<td>-0.1179</td>
<td>-0.114</td>
<td>-0.1169</td>
</tr>
<tr>
<td></td>
<td>3200</td>
<td>---</td>
<td>-0.1204</td>
<td>-0.118</td>
<td>-0.1181</td>
</tr>
<tr>
<td></td>
<td>5000</td>
<td>---</td>
<td>-0.1190</td>
<td>-0.109</td>
<td>-0.1173</td>
</tr>
<tr>
<td></td>
<td>7500</td>
<td>---</td>
<td>-0.1200</td>
<td>-0.108</td>
<td>-0.1157</td>
</tr>
<tr>
<td></td>
<td>10000</td>
<td>-0.1028</td>
<td>-0.1197</td>
<td>-0.101</td>
<td>-0.1150</td>
</tr>
</tbody>
</table>

| Vortex 1       | 400 | 6.440x10^-4         | 6.4235x10^-4  | ---             | 6.1810x10^-4 |
|                | 1000| 1.700x10^-3         | 1.7510x10^-3  | 1.760x10^-3     | 1.6594x10^-3 |
|                | 3200| ---                 | 3.1396x10^-3  | 3.290x10^-3     | 2.6744x10^-3 |
|                | 5000| ---                 | 3.0836x10^-3  | 3.870x10^-3     | 2.7786x10^-3 |
|                | 7500| ---                 | 3.2848x10^-3  | 4.860x10^-3     | 2.7396x10^-3 |
|                | 10000| 2.960x10^-3        | 3.4183x10^-3  | 5.540x10^-3     | 2.7528x10^-3 |

| Vortex 2       | 400 | 1.450x10^-5         | 1.4195x10^-5  | ---             | 1.3577x10^-5 |
|                | 1000| 2.170x10^-4         | 2.3113x10^-4  | 2.0 x10^-4      | 2.1951x10^-4 |
|                | 3200| ---                 | 9.7820x10^-4  | 1.20 x10^-3     | 1.0465x10^-3 |
|                | 5000| ---                 | 1.3612x10^-3  | 1.490x10^-3     | 1.2675x10^-3 |
|                | 7500| ---                 | 1.4671x10^-3  | 1.750x10^-3     | 1.3697x10^-3 |
|                | 10000| ---                 | 1.5183x10^-3  | 1.930x10^-3     | 1.4055x10^-3 |

| Vortex 3       | 3200| ---                 | 7.2786x10^-4  | 5.860x10^-3     | 6.4440x10^-4 |
|                | 5000| ---                 | 1.4564x10^-3  | 1.230x10^-3     | 1.3045x10^-3 |
|                | 7500| ---                 | 2.0462x10^-3  | 1.840x10^-3     | 1.8426x10^-3 |
|                | 10000| ---                 | 2.4210x10^-3  | 2.230x10^-3     | 2.1817x10^-3 |

* 141x141 grids for Re = 400 and 1000, and 180x180 grids for Re=10000.
° 257x257 grids for Re = 400, 5000, 7500, and 10,000, and 129x129 grids for Re = 1000 and 3200.
x 51x51 grids for all Reynolds numbers.
+ 65x65 grids for all Reynolds numbers.
3.2 Backward-Facing Step Flow

A laminar backward-facing step flow with an expansion ratio of 1:1.94 is considered below. A description of the problem is shown in Figure 8, and the experimental data can be found in Armaly et al. [20]. The Reynolds number, \( \text{Re} = \frac{\rho V D}{\mu} \), was based on the hydraulic diameter \( (D = 0.0104 \text{ m}) \) and the bulk velocity \( (V = 0.6667 \text{ m/sec}) \) at the inlet. The various Reynolds numbers have been obtained by varying the molecular viscosity \( (\mu) \) of the fluid, with the rest of the parameters kept as constants. The velocity profile of a fully developed channel flow has been prescribed at the inlet; and the vanishing normal stress has been applied at the exit boundary.

![Figure 8. Configuration, coordinates, and nomenclature of backward-facing step flow; h: step height (0.0049 cm).](image)

For the Reynolds number less than approximately 450, there exists only one recirculation zone at the down-stream region of the backward-facing step. As the Reynolds number is increased beyond approximately 450, another recirculation zone appears at the top wall of the channel, the size of which grows further as the Reynolds number is increased. Experimental data showed that a third recirculation zone appears at the bottom wall for the Reynolds number greater than approximately 1000. As the Reynolds number is increased beyond approximately 600, the three-dimensional effect becomes so strong that comparison between the two-dimensional computational result and the experimental data becomes less meaningful, which is discussed in detail in Armaly et al. [20]. Therefore, the computations have been carried out up to \( \text{Re} = 900 \) starting from \( \text{Re} = 100 \), with the incremental Reynolds number of 100.

The two different grids used are shown in Figure 9. For the coarse grid (Grid A) case, the uniform zero values for both velocity and pressure were used as an initial guess for the \( \text{Re} = 100 \) case, the converged solution of \( \text{Re} = 100 \) case was used as an initial guess for the \( \text{Re} = 200 \) case, and so on. The required number of iterations were 20, 24, 33, 43, 56, 65, 72, 80, and 77 for the Reynolds numbers of 100 through 900, respectively. For the fine grid case (Grid B), the initial guess for all the Reynolds number cases were obtained by interpolating the coarse grid solutions using the multi-grid concept of Ghia et al. [17].
The computed velocity vectors and the streamline contours for $Re = 400$, 500, and 800 are shown in Figures 10 and 11, respectively. The streamline contour labels can be seen in the streamline contour plot given in Figure 11(a), yet there exists no recirculation zone for $Re = 400$ as can be confirmed from the velocity vector plot in Figure 10(a).

The computed static pressure was normalized using a relationship given as $P = pL_{ref}/V_{ref}/\mu$, where $L_{ref} = 0.0049$ m is the step height and $V_{ref}$ is the bulk velocity at the inlet. The normalized pressure contours for Reynolds numbers of 400, 500, and 800 are shown in Figure 12.

The size of recirculation zones versus Reynolds number is compared with experimental data in Figure 13. It can be seen that the computational results compare favorably with experimental data up to Reynolds number of approximately 600.

The top and bottom wall pressure for the Reynolds number of 100 through 900 are shown in Figure 14. Neither experimental data nor computational results for the wall pressure are available as yet, and no comparison could have been made.
Figure 10. Velocity vectors for backward-facing step flow.
Figure 11. Streamlines for the backward-facing step flow.
Figure 12. Pressure contours for the backward-facing step flow.
TABLE 4. STREAMLINE CONTOUR LABEL FOR BACKWARD-FACING STEP FLOW

<table>
<thead>
<tr>
<th>Label</th>
<th>( \psi )</th>
<th>Label</th>
<th>( \psi )</th>
<th>Label</th>
<th>( \psi )</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>-2.0\times10^{-4}</td>
<td>F</td>
<td>1.0\times10^{-4}</td>
<td>K</td>
<td>3.467\times10^{-3}</td>
</tr>
<tr>
<td>B</td>
<td>-1.5\times10^{-4}</td>
<td>G</td>
<td>5.0\times10^{-4}</td>
<td>L</td>
<td>3.480\times10^{-3}</td>
</tr>
<tr>
<td>C</td>
<td>-5.0\times10^{-5}</td>
<td>H</td>
<td>1.0\times10^{-3}</td>
<td>M</td>
<td>3.50\times10^{-3}</td>
</tr>
<tr>
<td>D</td>
<td>-1.0\times10^{-5}</td>
<td>I</td>
<td>2.0\times10^{-3}</td>
<td></td>
<td></td>
</tr>
<tr>
<td>E</td>
<td>0.</td>
<td>J</td>
<td>3.0\times10^{-3}</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

TABLE 5. PRESSURE CONTOUR LABEL FOR BACKWARD-FACING STEP FLOW

<table>
<thead>
<tr>
<th>Label</th>
<th>Reynolds Number (Re)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>400.</td>
</tr>
<tr>
<td>A</td>
<td>-0.74</td>
</tr>
<tr>
<td>B</td>
<td>0.36</td>
</tr>
<tr>
<td>C</td>
<td>2.35</td>
</tr>
<tr>
<td>D</td>
<td>6.68</td>
</tr>
<tr>
<td>E</td>
<td>15.90</td>
</tr>
<tr>
<td>F</td>
<td>20.78</td>
</tr>
<tr>
<td>G</td>
<td>25.44</td>
</tr>
<tr>
<td>H</td>
<td>30.11</td>
</tr>
<tr>
<td>I</td>
<td>34.77</td>
</tr>
<tr>
<td>J</td>
<td>39.65</td>
</tr>
<tr>
<td>K</td>
<td>42.40</td>
</tr>
<tr>
<td>L</td>
<td>43.46</td>
</tr>
<tr>
<td>M</td>
<td>43.72</td>
</tr>
</tbody>
</table>

Finite difference computations of the same backward-facing step flow can be found in Kim and Moin [21] and Kwak and Chang [22] among many others. The same level of agreement as the present case between the computational results and the experimental data can be found in Kim and Moin [21]. On the other hand, the top wall recirculation zone was not shown as clearly as in the present result in the streamline contour plot due to Kwak and Chang [22].

Comparison of the present computational results with those of other investigators for a backward-facing step flow with the expansion ratio of 1:2 can be found in Reference 14.

3.3 Laminar Flow in a Square Duct of Strong Curvature

A re-developing laminar flow in a square duct of strong curvature is considered below. The flow configuration is shown in Figure 15, and the experimental data can be found in Humphery et al. [24].
Figure 13. Reattachment length versus Reynolds number. ---: Present method; ---: control-volume based finite difference method [20]; x, O, Δ: Exp't $s_1$, $s_2$, and $s_3$, respectively.
Figure 14. Wall pressure for backward-facing step flow, fine grid (Grid B) solutions, Re = 100 to 900.
The Reynolds number based on the hydraulic diameter of 0.04 m and the bulk velocity of $1.98 \times 10^{-2}$ m/sec was 790. Due to the symmetry of the flow domain, only one half of the duct has been considered in the numerical analysis. The inlet plane has been located at 2.8 hydraulic diameters upstream of the curved section, and the exit plane, at 8 hydraulic diameters downstream of the end of the curved section. Discretization of the domain is shown in Figure 16. An analytical solution for the fully developed duct flow has been prescribed at the inlet boundary; and the vanishing normal stress has been prescribed at the exit boundary.

The computed velocity vectors at the three planes of the curved section are shown in Figure 17. It can be seen in Figure 17(c) that there exists a weak recirculation zone extending up to $\theta = 40^\circ$ of the curved section. Computational results due to Humphery et al. [24] showed that the same recirculation zone extended beyond $\theta = 12^\circ$, and the computational results due to Rhie [25] showed that the recirculation zone extended beyond $\theta = 30^\circ$. There does not exist fine grid computational results for the recirculation zone as yet. The secondary recirculation flows at three cross-sections are shown in Figure 18. The computational results showed that the grid used was not fine enough to resolve all the details of the flow field. It was also found that the frontal solver used in this study was not adequate to pursue further grid refinement.

![Figure 15. Configuration of the laminar flow in a square duct of strong curvature.](image-url)
Figure 16. Discretization of the flow domain. (a) Discretization of the cross-section, and (b) grid in the flow direction.
Figure 17. Velocity vectors on the curved section.
Figure 18. Secondary recirculation flows.
IV. CONCLUSIONS AND DISCUSSION

A velocity-pressure integrated, mixed interpolation, Galerkin finite element method for the Navier-Stokes equations has been presented. The method has been tested for high Reynolds number cavity flows, a backward-facing step flow, and a laminar flow in a square duct of strong curvature.

For the cavity flows, the present computational results compared favorably with those of Schreiber and Keller [16] and Ghia et al. [17], which were obtained by using approximately four times more grids than the present case.

For the backward-facing step flow, it has been shown that the present computational method could capture the subtle pressure driven recirculation zone at the top wall of the channel for Reynolds numbers greater than 500. The size of the recirculation zone also compared favorably with the experimental data.

For the three-dimensional curved duct flow, grid refinement was not feasible due to the computer limitation and the frontal solver used in the present study.

It is usually known that the Bubnov-Galerkin method [26] (i.e., the test functions are selected from the same space as that of the trial functions, as in the present case) yields oscillatory solutions for high Reynolds number flows. An extensive discussion on the finite element upwinding technique can be found in Brooks and Hughes [26], among many others. For the example problems considered herein, no upwinding was necessary to obtain accurate solutions which were free of numerical wiggles.
REFERENCES


APPENDIX I

Finite Element Computer Program (NSFLOW/L) for Incompressible, Laminar Flows
C
C**********************************************************************5****
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,MXDOF,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
   CONTINUE
103 CONTINUE
104 WRITE(6,602)
602 FORMAT(2X,'TERMINATED IN MAIN PROGRAM 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,MXDOF,MXFRON)
   GO TO 102
C
C  PREPARE INPUT DATA
C
2 CONTINUE
   CALL PREP(MAXNOD,MAXELM,MXDOF,MXFRON)
   GO TO 102
C
3 CONTINUE
   GO TO 104
C
C  FLOW SOLVER - MAIN PROCESSOR
C
4 CONTINUE
   CALL PROCES(MAXNOD,MAXELM,MXDOF,MXFRON)
   GO TO 101
C
5 CONTINUE
   GO TO 101
C
6 CONTINUE
SUBROUTINE INITAL(MAXNOD,MAXELM,MAXDOF,MXFRON)

C
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 40 KPROB=1,10
DO 40 KNODE=1,MAXNOD
IBCA(KNODE,KPROB)= 0
ADBC(KNODE,KPROB)= 0.
A(KNODE,KPROB)= 0.
40 CONTINUE

C
RETURN
END

C
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/ INDEXF(27,3,15), INDEXF(27,15)
COMMON /CITER/ CNVCF(10), ERROF(10), RELAX(10), ITERE, MAXIT
COMMON /CLSCF/ XKSNOID(27,3,11), TM(4,4)
COMMON /CMATE/ BFX(3), DENSY, VISCY, PECLET
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,
C

DATA (XKSNOD(K,1,6),K=1,4) / -1., 1., 1., 1., 1., 1./,
   (XKSNOD(K,2,6),K=1,4) / -1., -1., 1., 1., 1., 1./,
   (XKSNOD(K,3,6),K=1,4) / -1., -1., -1., 1., 1., 1./,
   (XKSNOD(K,4,6),K=1,4) / -1., -1., -1., -1., 1., 1./,
   (XKSNOD(K,5,6),K=1,4) / -1., -1., -1., -1., -1., 1./,
   (XKSNOD(K,6,6),K=1,4) / -1., -1., -1., -1., -1., -1./,
C

C --- IFLOW=2 FOR 2-D INTEGRATED METHOD ---
DATA (INDXF(KPE,1,2),KPE=1,9) / 1, 3, 5, 7, 9, 11, 13, 15, 17/,
   (INDXF(KPE,2,2),KPE=1,9) / 2, 4, 6, 8, 10, 12, 14, 16, 18/,
   (INDXF(KPRE,2),KPRE=1,3) / 19, 20, 21/
C

C --- IFLOW=3 FOR 2-D INTEGRATED METHOD ---
DATA (INDXF(KPE,1,3),KPE=1,9) / 1, 3, 5, 7, 9, 11, 13, 15, 17/,
   (INDXF(KPE,2,3),KPE=1,9) / 2, 4, 6, 8, 10, 12, 14, 16, 18/,
   (INDXF(KPRE,3),KPRE=1,3) / 19, 20, 21/
C

C --- IFLOW=11 FOR 3-D INTEGRATED METHOD ---
DATA (INDXF(KPE,1,11),KPE=1,27)
   /1, 4, 7, 10, 13, 16, 19, 22, 25, 28, 31, 34, 37, 40, 43, 46, 49, 52,
   55, 58, 61, 64, 67, 70, 73, 76, 79/,
   (INDXF(KPE,2,11),KPE=1,27)
   /2, 5, 8, 11, 14, 17, 20, 23, 26, 29, 32, 35, 38, 41, 44, 47, 50, 53,
   56, 59, 62, 65, 68, 71, 74, 77, 80/,
   (INDXF(KPE,3,11),KPE=1,27)
   /3, 6, 9, 12, 15, 18, 21, 24, 27, 30, 33, 36, 39, 42, 45, 48, 51, 54,
   57, 60, 63, 66, 69, 72, 75, 78, 81/,
   (INDXF(KPRE,11),KPRE=1,4) / 82, 83, 84, 85/
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,NPRE,MGAUS,MGAUS,MGAUS,MAXIT /7*0/
DATA VISCY,DENSY,PECLET /3*0./
DATA (IA(K),K=1,10) /10*0/
DATA ERROR/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),NGAUS
COMMON /CPROB/ IA(10),IPILOT
DIMENSION LIBELF(15),LIBNPE(11),LBNPRE(15)
C
DATA (LIBELF(IFL),IFL=1,15) /0,8,8,0,0, 0,0,0,0, 11,0,0,0,0/,
   - (LIBNPE(IEL),IEL=1,11) /0,0,0,0,0, 4,8,9,0,0, 27/,
   - (LBNPRE(IFL),IFL=1,15) /0,3,3,0,0, 0,0,0,0, 4,0,0,0,0/
C
IELF=LIBELF(IFLOW)
NPRE=LIBNPE(IFLOW)
NPE =LBNPRE(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(1,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
      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********1********2********3********4********5**************6***
SUBROUTINE PREP(MAXNOD,MAXELM,MDOF,MXFRON)
C*X- IMPLICIT REAL*8 (A-H,O-Z)
CHARACTER*4 TITLE,ICNTRL,IWORD
COMMON /CDESC/ NNODE,NELEM,NPE,NPRE,NDIM,NEDOF,IFLOW,
COMMON /CFLOW/ A(4227,10),ADBC(4227,10),IBCA(4227,10)
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 /GRID/ 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 /CPROB/ IA(10),IPLON
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', 'IA05',
 'IA06', 'IA07', 'IA08', 'IA09', 'IA10',
 ' ****', 'EXAM', ' ****', ' ****', ' ****',
 ' ****', ' INCL', ' ****', ' ****', ' END '/

101 CONTINUE
READ(5,501) ICNTRL,TITLE
WRITE(6,601) ICNTRL,TITLE
501 FORMAT(20A4)
601 FORMAT(/2X,20A4)

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

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

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-=',I5)

CALL DATLIB
WRITE(6,606) IELF,NPE,NPRE,MGAUS
606 FORMAT(4X,'IELF-=',I2, 2X,'NPE-=',I2, 2X,'NPRE-=',I2,
  2X,'MGAUS-=',I2)
WRITE(6,607)
DO 40 LGAS=1,MGAUS
WRITE(6,608) (EXKS(KDIM,LGAUS),KDIM=1,NDIM),EW(LGAUS)

40 CONTINUE
607 FORMAT(4X,'NUMERICAL QUADRATURE DATA EXKS AND EW')
608 FORMAT(5X,4E12.4)

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, /4X,'RELAX=',5E12.4,
 - /8X,5E12.4,
 - /4X,'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
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(K),K=1,2)=',216)
  GO TO 101
C
15 CONTINUE
  GO TO 101
C
16 CONTINUE
  GO TO 101
C
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
23 CONTINUE
GO TO 101

C 24 CONTINUE
GO TO 101

C 25 CONTINUE
GO TO 101

C 26 CONTINUE
GO TO 101

C INCLUDE RE-START DATA

C 27 CONTINUE
CALL FEMDAT(A,ADBC,X,PBCDAT,NODES,IBCA,IPNOD,NPE,NNODE,
             -NELEM,MAXNOD,MAXEML)
GO TO 101

C-------1--------2----------3--------4--------5--------6------

28 CONTINUE
GO TO 103

C-------1--------2----------3--------4--------5--------6------

C 29 CONTINUE
RETURN

C-------1--------2----------3--------4--------5--------6------

30 CONTINUE

C RETURN
END

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)

DO 15 KLINE=1,NLINE
DO 15 JLINE=1,MLINE
DO 15 ILINE=1,LLINE
X(KNODE,l)=DELX(ILINE,1)
X(KNODE,2)=DELX(JLINE,2)
X(KNODE,3)=DELX(KLINE,3)
GO TO 7000

C
2000 CONTINUE
READ(5,501) TITLE
WRITE(6,601) TITLE
READ(5,*) ((KNOD(KPE,KDIM),KDIM=1,NDIM),KPE-1,NPE)
DO 22 KPE=1,NPE
WRITE(6,610) KPE,(KNOD(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')

LINE=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, 36,37,38,31,31, 41), IELF
31 CONTINUE
WRITE(6,630) IELF
630 FORMAT(2X,'SUB-RNODE       IELF =',I2)
STOP
36 CALL SHAP21(PHI,DPHI,CXKS,NPE)
GO TO 42
37 CALL SHAP22(PHI,DPHI,CXKS,NPE)
GO TO 42
38 CALL SHAP23(PHI,DPHI,CXKS,NPE)
GO TO 42
41 CALL SHAP33(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
NODS(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)
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
620 FORMAT(4X,'IEL1=',I6, 2X,'NODES(KPE,IEL1)=',8I6, 
           3(/15X,'NODES(KPE,IEL1)=',8I6))
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
C
RETURN
END
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
RETURN
END

C
C********1*********2*********3***4*********5*********6***
SUBROUTINE RBC1(DBCH,LDBC,MAXNOD,NNODE)
CHARACTER*4 TITLE
DIMENSION DBCH(MAXNOD),LDBC(MAXNOD),TITLE(15)
   DO 10 KNODE=1,NNODE
   LDBC(KNODE)=0
   DBCH(KNODE)=0.
10 CONTINUE

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),
       60 CONTINUE
       DBCH(KNODE)

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,15,5X,15,5X,15,5X,E11.4)
605 FORMAT(5X,'NODE=',15,5X,'LDBC=',15,5X,'DATDBC=',E11.4)
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)

READ(4,501) TITLE

FORMAT(15A4)

FORMAT(2X,15A4)

DO 10 KNOD=1,NNODE
READ(4,*), (X(KNODE,KDUM),KDUM=1,3)
CONTINUE

READ(4,501) TITLE

DO 20 KEL=1,NELEM
READ(4,*), (NODES(KPE,K),KPE=1,NPE)
CONTINUE

READ(4,501) TITLE

DO 30 KNOD=1,NNODE
READ(4,*), (IBCA(KNODE,K),K=1,3)
CONTINUE

READ(4,501) TITLE

READ(4,*) (IPNOD(K),K=1,2), PBCDAT

READ(4,501) TITLE

READ(4,*), (IPNOD(K),K=1,4)

READ(4,501) TITLE

READ(4,*), (PBCDAT(K),K=1,2)

READ(4,*), (PBCDAT(K),K=1,3)

DO 57 KNODE=1,NNODE
A(KNODE,4)=0.
CONTINUE

RETURN
END

SUBROUTINE ISOPEL(IFLOW, IELF, IAXS, NPE, NPRE, NDIM)

COMMON /CELEM/ EX(27,3), EA(27,10), NODEL(27), IELEM

COMMON /CWIND/ WHI(27), DWHX(27,3)

COMMON /ESHAP/ PHI(27), DPHX(27,3), PSI(27), DXDK(3,3),
DIMENSION DPHI(27,3), DPSI(27,3), XGS(3)

GO TO (1,1,1,1, 6,7,8,1,1, 11), IELF
1 CONTINUE
WRITE(6,608) IFLOW, IELF
608 FORMAT(2X,'SUB-ISOPEL IFLOW=',I2, 2X,'IELF=',I2)
STOP

CALL SHAP21(PHI, DPHI, CXKS, NPE)
GO TO 15

CALL SHAP22(PHI, DPHI, CXKS, NPE)
GO TO 15

CALL SHAP23(PHI, DPHI, CXKS, NPE)
GO TO 15

CALL SHAP33(PHI, DPHI, CXKS, NPE)

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

GO TO (31,32,33), NDIM
31 CONTINUE
DETB = DXDK(1,1)
GO TO 37
32 CONTINUE
DETB = DXDK(1,1)*DXDK(2,2) - DXDK(1,2)*DXDK(2,1)
GO TO 37
33 CONTINUE
DETB = 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)

IF(DETB.LE.1.E-15) THEN
  WRITE(6,610) IELEM, IELF, NPE, NPRE
  DO 17 KPE=1, NPE
    WRITE(6,620) (PHI(K), K=1, NPE)
  DO 18 IDIM=1, NDIM
    WRITE(6,625) (DPHI(K, IDIM), K=1, NPE)
  17 WRITE(6,615) KPE, NODEL(KPE), (EX(KPE, IDIM), IDIM=1, NDIM)
  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))
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
43 CONTINUE
  DKDX(1,1)=(DXDK(2,2)*DXDK(3,3)-DXDK(3,2)*DXDK(2,3))/DETJB
  DKDX(1,2)=(DXDK(1,3)*DXDK(3,2)-DXDK(1,2)*DXDK(3,3))/DETJB
  DKDX(1,3)=(DXDK(1,2)*DXDK(2,3)-DXDK(2,2)*DXDK(1,3))/DETJB
  DKDX(2,1)=(DXDK(2,3)*DXDK(3,1)-DXDK(2,1)*DXDK(3,3))/DETJB
  DKDX(2,2)=(DXDK(1,1)*DXDK(3,3)-DXDK(3,1)*DXDK(1,3))/DETJB
  DKDX(2,3)=(DXDK(2,1)*DXDK(1,3)-DXDK(1,1)*DXDK(2,3))/DETJB
  DKDX(3,1)=(DXDK(2,1)*DXDK(3,2)-DXDK(2,2)*DXDK(3,1))/DETJB
  DKDX(3,2)=(DXDK(1,2)*DXDK(3,1)-DXDK(1,1)*DXDK(3,2))/DETJB
  DKDX(3,3)=(DXDK(1,1)*DXDK(2,2)-DXDK(2,1)*DXDK(1,2))/DETJB
45 CONTINUE
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 IF(IFLOW.EQ.0) GO TO 85
GO TO (51,52,53,85,85,85,85,61,85,85,85,85), IFLOW
C 51 CALL SHAPOl(PSI,DPSI,CXKS,NPRE)
GO TO 85
C 52 CALL SHAP02(PSI,DPSI,CXKS,NPRE)
GO TO 85

51
CONTINUE
DO 57 KDIM=1,NDIM
XGS(KDIM)=0.
DO 57 KPE=1,NPE
XGS(KDIM)=XGS(KDIM)+EX(KPE,KDIM)*PHI(KPE)
57 CONTINUE
PSI(1)=1.
DO 58 KDIM=1,NDIM
58 PSI(KDIM+1)=XGS(KDIM)
GO TO 85
CALL SHAP03(Psi,Dpsi,Cxks,Npre)

CONTINUE
IAXSY.EQ.1 THEN
RADUS=0.
DO 90 KPE=1,NPE
RADUS=RADUS+EX(KPE,2)*PHI(KPE)
90 CONTINUE
ENDIF
RETURN
END

SUBROUTINE LSHPl(PLK,DPLK,S)
DIMENSION PLK(3),DPLK(3)
IMPLICIT REAL*8 (A-H,O-Z)

PLK(1)=(1.-S)/2.
PLK(2)=(1.+S)/2.
DPLK(1)=-0.5
DPLK(2)= 0.5
RETURN
END

SUBROUTINE LSHP2(PNK,DPNK,S)
DIMENSION PNK(3),DPNK(3)
IMPLICIT REAL*8 (A-H,O-Z)

1-D QUADRATIC ELEMENT (NE= 1 2 3)
(S =-1. 0. 1.)
PNK(1)=S*(S-1.)/2.
PNK(2)=1.-S**2
PNK(3)=S*(1.+S)/2.

DPNK(1)=S-0.5
DPNK(2)=-2.*S
DPNK(3)=S+0.5
RETURN
END
SUBROUTINE SHAPO1(SHP,DSHP,CXKS,NPE)
DIMENSION SHP(27),DSHP(27,3),CXKS(3)
SHP(1)=1.
RETURN
END
C*X- IMPLICIT REAL*8 (A-H,O-Z)

SUBROUTINE SHAPO2(SHP,DSHP,CXKS,NPE)
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*X- IMPLICIT REAL*8 (A-H,O-Z)

SUBROUTINE SHAPO3(SHP,DSHP,CXKS,NPE)
DIMENSION SHP(27),DSHP(27,3),CXKS(3)
SHP(1)=0.25+0.612372435*CXKS(2)-0.433012701*CXKS(3)
SHP(2)=0.25-0.612372435*CXKS(2)-0.433012701*CXKS(3)
SHP(3)=0.25+0.612372435*CXKS(1)+0.433012701*CXKS(3)
SHP(4)=0.25-0.612372435*CXKS(1)+0.433012701*CXKS(3)
RETURN
END
C*X-

SUBROUTINE SHAP21(SHP,DSHP,CXKS,NPE)
DIMENSION SHP(27),DSHP(27,3),CXKS(3)
IMPLICIT REAL*8 (A-H,O-Z)
4-NODE LINEAR ELEMENT COUNTER-CLOCKWISE NODE NUMBERING
S=CXKS(1)
T=CXKS(2)
ST=S*T
SHP(1)=0.25*(1.-S-T+ST)
SHP(2)=0.25*(1.+S-T-ST)
SHP(3)=0.25*(1.+S+T+ST)
SHP(4)=0.25*(1.-S+T-ST)
DSHP(1,1)=0.25*(-1.+T)
DSHP(2,1)=0.25*(-1.-T)
DSHP(3,1)=0.25*(-1.+S)
DSHP(4,1)=0.25*(-1.-S)
DSHP(1,2)=0.25*(-1.+T)
DSHP(2,2)=0.25*(-1.-T)
DSHP(3,2)=0.25*(-1.+S)
DSHP(4,2)=0.25*(-1.-S)
SUBROUTINE SHAP23(SHP, DSHP, CXKS, NPE)
DIMENSION SHP(27), DSHP(27, 3), CXKS(3)

C
S = CXKS(1)
T = CXKS(2)
ST = S*T
SS = S*S
TT = T*T
SST = SS*T
STT = S*TT
S2 = 2.*S
T2 = 2.*T
ST2 = 2.*ST

SHP(1) = 0.25*(-1.+ST+SS+TT-SST-STT)
SHP(2) = 0.50*(-1.-T-SS+SST)
SHP(3) = 0.25*(-1.-ST+SS+TT+SST+STT)
SHP(4) = 0.50*(1.+S-TT-STT)
SHP(5) = 0.25*(-1.+ST+SS+TT+STT)
SHP(6) = 0.50*(1.-T-SS-SST)
SHP(7) = 0.25*(-1.-ST+SS+TT-SST+STT)
SHP(8) = 0.50*(1.-S- TT+ STT)

DSHP(1,1) = 0.25*(T+S2-ST2-TT)
DSHP(2,1) = -S-ST
DSHP(3,1) = 0.25*(-T+S2-ST2+TT)
DSHP(4,1) = 0.50*(1.-TT)
DSHP(5,1) = 0.25*(T+S2+ST2+TT)
DSHP(6,1) = -S-ST
DSHP(7,1) = 0.25*(-T+S2+ST2-TT)
DSHP(8,1) = 0.50*(-1.+TT)

DSHP(1,2) = 0.25*(S+T2-SS-ST2)
DSHP(2,2) = 0.50*(-1+SS)
DSHP(3,2) = 0.25*(-S+T2-SS+ST2)
DSHP(4,2) = -T-ST
DSHP(5,2) = 0.25*(S+T2+SS+ST2)
DSHP(6,2) = 0.50*(1.-SS)
DSHP(7,2) = 0.25*(-S+T2+SS-ST2)
DSHP(8,2) = -T+ST
RETURN
END
**NODE QUADRATIC ELEMENT**

```fortran
CALL LSHP2(PNK, DPNK, CXKS(1))
CALL LSHP2(PNE, DPNE, CXKS(2))

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
```

**SUBROUTINE SHAP33(SHP, DSHP, CXKS, NPE)**

```fortran
IMPLICIT REAL*8 (A-H,O-Z)
DIMENSION SHP(27), DSHP(27, 3), PNK(3),
          DPNK(3), PNE(3), DPNE(3), PNZ(3), DPNZ(3), INDK(27),
          INDE(27), INDZ(27), CXKS(3)
DATA (INDK(K), K = 1, 27), (INDE(K), K = 1, 27), (INDZ(K), K = 1, 27)
          / 1, 2, 3, 3, 2, 1, 1, 1, 2, 3, 3, 2, 1, 1, 1, 2, 3, 3, 2, 1, 1, 1, 2, 3, 3, 2, 2, 2,
          - 1, 1, 1, 2, 3, 3, 3, 2, 1, 1, 2, 3, 3, 3, 2, 1, 1, 2, 3, 3, 3, 2, 2, 2,
          - 1, 1, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 2, 3, 3, 3, 3, 3, 3, 1, 3, 2/

CALL LSHP2(PNK, DPNK, CXKS(1))
CALL LSHP2(PNE, DPNE, CXKS(2))
CALL LSHP2(PNZ, DPNZ, CXKS(3))

DO 10 K = 1, NPE
  IPE = INDK(K)
  JPE = INDE(K)
  KPE = INDZ(K)
  SHP(K) = PNK(IPE) * PNE(JPE) * PNZ(KPE)
  DSHP(K, 1) = DPNK(IPE) * PNE(JPE) * PNZ(KPE)
  DSHP(K, 2) = PNK(IPE) * DPNE(JPE) * PNZ(KPE)
  DSHP(K, 3) = PNK(IPE) * PNE(JPE) * DPNZ(KPE)
10 CONTINUE
RETURN
END
```

**SUBROUTINE PROCES(MAXNOD, MAXELM, MAXDOF, MXFRON)**

```fortran
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 /CPROB/ IA(10), IPLOT
```
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)
RETURN
END

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 APPEARANCE 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
30 CONTINUE
C
RETURN
END

C********1*********2*********3*********4*********5*********6***
SUBROUTINE SHPLIB
C-X- IMPLICIT REAL*8 (A-H,O-Z)
COMMON /CDESC/ NNOD,NELEM,NPE,NPRE,NDIM,NEDOF,IFLOW,
- IAXSY,IElf
COMMON /CELEM/ EX(27,3),EA(27,10),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 K1,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)
   IF(IAXSY.EQ.1) WRITE(2) (ARADUS(LGAUS),LGAUS=1,MGAUS)
90 CONTINUE
C
   ENTRY SHPDAT
   DO 95 G1,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 SIFLOW(NODES,NNODE,NELEM,NPE,
   -.X- IMPLICIT REAL*8 (A-H,O-Z)
COMMON /CDOF/ A1(11527),IDBC(11527),LDGF(4227),LDOF(4227)
DIMENSION NODES(27,MAXELM),LIBDOF(15),LFLEL(27,15)
C
DATA (LIBDOF(IFL),IFL=1,15) / 0,21,21,0,0, 0,0,0,0, 85, 0,0,0,0/
DATA (LFLEL(KPE,2),KPE=1,9) / 2,2,2,2,2, 2,2,2,5/,
   -(LFLEL(KPE,3),KPE=1,9) / 2,2,2,2,2, 2,2,2,5/,
   -(LFLEL(KPE,11),KPE=1,27)/3,3,3,3,3, 3,3,3,3,3, 3,3,3,3,3,
C
   DO 10 KELEM=1,NELEM
   DO 10 KPE =1,NPE
      LDOF(Abs(NODES(KPE,KELEM))) = LFLEL(KPE,IFLOW)
10 CONTINUE
C LIDO(1)=1
DO 30 INODE=2,NNODE
LIDO(INODE)=LIDO(INODE-1)+LIDO(INODE-1)
30 CONTINUE
NEDOF=LIBDOF(IFLOW)
NTDOF=LIDO(NNODE)+LIDO(NNODE)-1
C RETURN
END

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),LIDO(4227)
COMMON /CFLOW/ A(4227,10),ADBC(4227,10),IBCA(4227,10)
COMMON /CFRON/ MFRONF
COMMON /CINDX/ INDEXF(27,3,15),INDEXP(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=LIDO(KNODE)-1+KDIM
IDBC(KDOF)=IBCA(KNODE,KDIM)
A1(KDOF) =ADBC(KNODE,KDIM)
15 CONTINUE
20 CONTINUE
C
IF(IPNOD(1).LE.0) GO TO 50
KDOF=LIDO(IPNOD(1))+NDIM
IDBC(KDOF)=1
A1(KDOF) =PBCDAT
C
50 CONTINUE
KDBC=0
DO 70 IDOF=1,NTDOF
IF(ABS(IDBC(IDOF)).NE.0) KDBC=KDBC+1
70 CONTINUE
NDBC=KDBC
C
RETURN
END
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, NNODE, NELEM, NEDOF, NPE, NPRE, 
- NDIM, IFLOW, IAXSY, IELF, 
- MXFRON, MAXNOD, MAXELM, MAXDOF)

C
DO 120 KPROB = 1, 4
IF(ERROF(KPROB) .GT. CNVCF(KPROB)) GO TO 1001
120 CONTINUE
IF(ERROF(4) .GT. CNVCF(4)) GO TO 1001

C
2001 CONTINUE
IF(IFLOW .GT. 0) CALL SPRS(A(1,4), IBCA(1,4), NODES, NNODE, NELEM, 
- NPE, NPRE, NDIM, IFLOW, IELF, MAXNOD, MAXELM, MAXDOF)

C
IF(ITERE .LE. MAXIT) RETURN
CALL PLSDAT(MAXNOD, MAXELM, MAXDOF)
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, 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), IBCA(11527), LDOF(4227), L1DOF(4227)
COMMON /CELEM/ EX(27,3), EA(27,10), NODEL(27), IELEM
DIMENSION NODES(27, MAXELM), OK(167, 167), GF(11527), 
- PNORM(167), LHEAD(167), EK(85, 85), EF(85), 
- LOCEL(85), NDEST(85)

C
CALL SLFLOW(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 INITIALIZE HEADING AND GRAND FLUID MATRIX

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

IELEM=0
30 CONTINUE
IELEM=IELEM+1
CALL ELEMFL(EK,EF,NPE,NPRE,NDIM,NEDOF,IFLOW, 
- IAXSY,IELF,MAXNOD,MAXELM,MAXDOF)

C CREATE GLOBAL DOF ARRAY FOR EACH ELEMENT DOF
C
40 CONTINUE

IDOF=0
DO 70 IPE=1,NPE
INODE=NODES(IPE,IELEM)
N1DOF=L1DOF(IABS(INODE))
NDOF=LDOF(IABS(INODE))
DO 70 KDOF=1,NDOF
LOCEL(IDOF)=N1DOF+KDOF-1
IF(INODE.LT.0) LOCEL(IDOF)=-LOCEL(IDOF)
70 CONTINUE

C CONTRACT D.B.C. FOR ELEMENT SYSTEM OF EQUATIONS
C
KDOF = 0
NEWDOF= NEDOF
DO 90 KDUM=1,NEWDOF
KDOF = KDOF + 1
IEQ = ABS(LOCEL(KDOF))
IF(IDBC(IEQ).EQ.O) GO TO 90
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

IF(KDOF.EQ.NEWDOF) GO TO 86
DO 85 IDOF=KDOF+1,NEWDOF
85 CONTINUE
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
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
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
    CONTINUE
    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,NmON)
    FORMAT(/2X,'SUB-FRONT --- FRONT WIDTH TO SMALL',
         -'/4X,'MXFRON=',I5, 2X,'MFRON=',I5, 2X,'NFRON=',I5,
         -'/4X,'NCRIT=',I5, 2X,'IELEM=',I5,
         -'/2X,'LIST OF LHEAD DATA')
637 FORMAT(2X,10I5)
638 FORMAT(2X,10I5)
STOP
C
100 CONTINUE
    NDEST(IDOF)=NFRON
    LHEAD(NFRON)=IEQ
    GO TO 120
CONTINUE
NDEST(IDOF)=KFRON
LHEAD(KFRON)=IEQ
CONTINUE

ASSEMBLE AN ELEMENT SYST. OF EQS. INTO A GLOBAL SYST. EQS.

DO 130 IDOF=1,NEWDOF
IEQ=ABS(LOCSEL(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)
CONTINUE
IF(NFRON.LT.NCRIT.AND.IELEM.LT.NELEM) GO TO 30

CHECK THE LAST APPEARANCE OF EACH DOF

CONTINUE

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
CONTINUE

IEQ=IABS(LHEAD(LPIVOT))
IF(ABS(PIVOT).GT.1.E-14) GO TO 180
WRITE(6,650) 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)
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

PROGRAM TERMINATED --- ILL-CONDITIONED MATRIX',
       /2X,'IEQ=',I6, 2X,'PIVOT=',E12.4,
       /2X,'NCRIT=',I5, 2X,'NFRON=',I5, 2X,'IELEM=',I5,
       /2X,'CURRENT ELEMENT IN PROCESS IELEM=',I5)

CURRENT DATA IN THE GLOBAL MATRIX')
          FORMAT(2X,'CURRENT DATA IN THE GLOBAL MATRIX')
          FORMAT(2X,'NDEST-DATA',20(/4X,10I6))
          FORMAT(2X,'LHEAD-DATA',25(/4X,10I6))
          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 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.1) GO TO 230
DO 220 JFRON=LPIVOT+1,NFRON
  GK(IFRON,JFRON-1)=GK(IFRON,JFRON)-FACTO**RPNORM(JFRON)
220 CONTINUE
C 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)-FACTO**RPNORM(JFRON)
260 CONTINUE
C 250 CONTINUE
ITOTV=IABS(LHEAD(IFRON))
GF(ITOTV)=GF(ITOTV)-FACTO**RHSID
240 CONTINUE
C 270 CONTINUE
DO 280 JFRON=1,LPIVOT+1,NFRON
  GK(IFRON-1,JFRON-1)=GK(IFRON,JFRON)-FACTO**RPNORM(JFRON)
280 CONTINUE
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
320 CONTINUE
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(IIELEM.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**********************************************************************
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),NODEL(27),IELEM
COMMON /CGAUS/ EXKS(3,64),EW(64),MGAUS
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
COMMON /CWIND/ WHI(27),DWHX(27,3)
DIMENSION EK(85,85),EF(85),IROW(3),JCOL(3), -
                  DIFFU(3,3),DNUM(3),XGS(3),DPDX(3),DVDX(3)
C
   DO 2 IDOF=1,NEDOF
      EF(IDOF)=0.
DO 2 JDOF=1,NEDOF
   EK(IDOF,JDOF)=0.0
2 CONTINUE

CALL ELMDAT
CALL SHPDAT

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

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

DO 30 IPE=1,NPE
   DO 11 KDIM=1,NDIM
      IROW(KDIM)=INDXF(IPE,KDIM,IFLOW)
   DO 11 KPE=1,NPE
      EF(IROW(KDIM))=EF(IROW(KDIM))+WHI(IPE)*BFX(KDIM)*AREA(LGAUS)
11 CONTINUE

DO 25 JPE=1,NPE
   DO 12 KDIM=1,NDIM
      JCOL(KDIM)=INDXF(JPE,KDIM,IFLOW)
25 CONTINUE

CONVC=0.
   DIFF=0.
   DO 15 IDIM=1,NDIM
      CONVC=CONVC+WHI(IPE)*DENSY*XC(IDIM)*APHX(JPE,IDIM,LGAUS)
   DIFF =DIFF +DWHX(IPE,IDIM)*XX(1)*APHX(JPE,IDIM,LGAUS)
   DO 14 JDIM=1,NDIM
      DIFFU(IDIM,JDIM)=DWHX(IPE,JDIM)*XX(1)*APHX(JPE,IDIM,LGAUS)
   14 CONTINUE
15 CONTINUE

DO 20 IDIM=1,NDIM
   EK(IROW(IDIM),JCOL(IDIM))=EK(IROW(IDIM),JCOL(IDIM))+CONVC
   +DIFF
20 CONTINUE
IF(IAXSY.EQ.1) THEN
   THETV = 2.*XXK(1)*WHI(IPE)*APHI(JPE,LGAUS)
   - /ARADUS(LGAUS)**2*AREA(LGAUS)
   EK(IROW(2),JCOL(2)) = EK(IROW(2),JCOL(2)) + THETV
END IF

25 CONTINUE
30 CONTINUE
C
DO 45 IPE=1,NPE
   DO 41 KDIM=1,NDIM
      IROW(KDIM)=INDXF(IPE,KDIM,IFLOW)
      DO 45 JPRE=1,NPRE
         JCOLP=INDXP(JPRE,IFLOW)
         DO 42 KDIM=1,NDIM
            EK(IROW(KDIM),JCOLP)=EK(IROW(KDIM),JCOLP)-DWHX(IPE,KDIM)
            - *APSI(JPRE,LGAUS)*AREA(LGAUS)
            IF(IAXSY.EQ.1) EK(IROW(2),JCOLP)=EK(IROW(2),JCOLP)-WHI(IPE)
            - *APSI(JPRE,LGAUS)/ARADUS(LGAUS)*AREA(LGAUS)
         42 CONTINUE
      45 CONTINUE
   41 CONTINUE
DO 50 IPRE=1,NPRE
   IROWP=INDXP(IPRE,IFLOW)
   DO 47 JPE=1,NPE
      DO 46 KDIM=1,NDIM
         JCOL(KDIM)=INDXF(JPE,KDIM,IFLOW)
         EK(IROWP,JCOL(KDIM))-EK(IROWP,JCOL(KDIM))+APSI(IPRE,LGAUS)
         *APHX(JPE,KDIM,LGAUS)*AREA(LGAUS)
      46 CONTINUE
   47 CONTINUE
50 CONTINUE
1000 CONTINUE
C
RETURN
END
C
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
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
GO TO (101,102,102,101,101, 101,101,101,101,101,
      102,102,101,101,101,101), IFLOW
C
101 CONTINUE
WRITE(6,605) IFLOW
605 FORMAT(2X,'TERMINATED IN SUB-SCNVFL FOR IFLOW=',I5)
STOP
C
102 CONTINUE
PMAX=0.
DO 40 KELEM=1,NELEM
   KNODE=ABS(NODES(NPE,KELEM))
   N1P=L1DOF(KNODE)+NDIM-1
   DO 35 IPRE=1,NPRE
      KDOFP=N1P+IPRE
      PDUM=ABS(A1(KDOFP))
      IF(PDUM.GT.PMAX) PMAX=PDUM
35 CONTINUE

67
C
DO 50 KELEM=1,NELEM
KNODE=ABS(NODES(NPE,KELEM))
N1P=L1DOF(KNODE)+NDIM-1
DO 45 IPRE=1,NPRE
KDOFP=N1P+IPRE
DELA(4)=ABS(A1(KDOFP)-PELEM(IPRE,KELEM))/PMAX
IF(DELA(4).GT.ERROF(4)) THEN
  ERROF(4)=DELA(4)
  KERR(4)=KNODE
ENDIF
PELEM(IPRE,KELEM)=A1(KDOFP)
45 CONTINUE
50 CONTINUE
C
WRITE(6,630) ITERE,(K,KERR(K),ERROF(K),K=1,4)
630 FORMAT(2X,'SUB-SCNVFL ITERE-',I5,
- 4(/4X,'KDIM-',I2, 2X,'NODE=',I5, 2X,'ERROF=',E12.4))
RETURN
END
C
C*********1*********2*********3*********4*********5*********6***
SUBROUTINE SPRS(P,IBCP,NODES,NNODE,NELEM,NPE,NPRE,NDIM,
- IFLOW,IELF,MAXNOD,MAXELM,MAXDOF)
C-X- IMPLICIT REAL*8 (A-H,O-Z)
COMMON /CELEM/
- EX(27,3),EA(27,10),NODEL(27),IELEM
COMMON /CGAUS/ EXKS(3,64),EW(64),MGAUS
COMMON /CLSCF/ XKSNOD(27,3,11),TM(4,4)
COMMON /CPRS/
- PELEM(4,1027),PBCDAT,IPNOD(2),IPDOF
COMMON /ESHAP/
- PHI(27),DPHX(27,3),PSI(27),DXDK(3,3),
- DPSI(27,3),PSINOD(27,27)
COMMON /CELEM/ EX(27,3),EA(27,10),NODEL(27),IELEM
COMMON /CGAUS/ EXKS(3,64),EW(64),MGAUS
COMMON /CLSCF/ XKSNOD(27,3,11),TM(4,4)
COMMON /CPRS/ PELEM(4,1027),PBCDAT,IPNOD(2),IPDOF
COMMON /ESHAP/ PHI(27),DPHX(27,3),PSI(27),DXDK(3,3),
- DKDX(3,3),CXS(3),DETJB,RADIUS,IGAUS,JGAUS,LGAUS
DIMENSION P(MAXNOD),IBCP(MAXNOD),NODES(27,MAXELM),
- DPSI(27,3),PSINOD(27,27)
C
DO 10 KNODE=1,NNODE
IBCP(KNODE) = 0
P(KNODE) = 0.
10 CONTINUE
C
GO TO (11,1001,1002,11,11, 11,11,11,11,11, 1001,11,11,11,11),
- IFLOW
11 CONTINUE
WRITE(6,610) IFLOW
STOP
610 FORMAT(2X,'TERMINATED IN SUB-SPRS FOR IFLOW=',I5)
C
1001 CONTINUE
DO 100 KPE=1,NPE
DO 20 KDIM=1,NDIM
20 CXKS(KD1M)=XKSNOD(KPE,KDIM,IELF)
C
GO TO (1,2,2,11,11, 11,11,11,11, 3,3,11,11,11, IFLOW
C
CONTINUE
CALL SHAPO1(PSI, DPSI, CXKS, NDIM)
GO TO 50
CONTINUE
CALL SHAPO2(PSI, DPSI, CXKS, NDIM)
GO TO 50
CONTINUE
CALL SHAPO3(PSI, DPSI, CXKS, NDIM)

CONTINUE
DO 60 KPRE=1,NPRE
PSINOD(KPRE,KPE) = PSI(KPRE)
60 CONTINUE

CONTINUE
DO 200 KELEM=1,NELEM
DO 200 KPE=1,NPE
KNODE=ABS(NODES(KPE,KELEM))
PDUM=0.
DO 70 KPRE=1,NPRE
PDUM=PDUM+PELEM(KPRE,KELEM)*PSINOD(KPRE,KPE)
70 CONTINUE
IBCP(KNODE)=IBCP(KNODE)+1
P(KNODE) = P(KNODE)+PDUM
200 CONTINUE
GO TO 1005

CONTINUE
DO 300 IELEM=1,NELEM
CALL EXDAT
DO 140 KPE=1,NPE
KNODE=NODEL(KPE)
PDUM = PELEM(1,IELEM)
DO 75 KDIM=1,NDIM
PDUM=PDUM+PELEM(KDIM+1,IELEM)*EX(KPE,KDIM)
75 CONTINUE
IBCP(KNODE)=IBCP(KNODE)+1
P(KNODE) = P(KNODE)+PDUM
140 CONTINUE
300 CONTINUE

CONTINUE
DO 96 KNODE=1,NNODE
P(KNODE) = P(KNODE)/FLOAT(IBCP(KNODE))
96 CONTINUE
IF(IPNOD(2).NE.0) THEN
PREF=P(IPNOD(2))
DO 97 KNODE=1,NNODE
P(KNODE)=P(KNODE)-PREF+PBCDAT
97 CONTINUE
ENDIF

RETURN
END
SUBROUTINE PFLOW(MAXNOD, MAXELM, MAXDOF)
COMMON /CDESC/ NNODE, NELEM, NPE, NPRE, NDIM, NEDOF, IFLOW,
- IAXSY, IELF
COMMON /CFLOW/ A(4227,10), ADBC(4227,10), IBCA(4227,10)
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

WRITE(6,650) ITERE, IFLOW
C
C-------1---------2---------3---------4---------5---------6-------
ENTRY PLSDAT(MAXNOD, MAXELM, MAXDOF)
WRITE(7,605)
DO 40 KNODE = 1, NNODE
WRITE(7,606) KNODE, (X(KNODE,KDUM), KDUM = 1,3)
40 CONTINUE
605 FORMAT(2X,'KNODE X Y Z')
606 FORMAT(2X,I5,2X,3E16.8)

WRITE(7,612)
612 FORMAT(2X,'NODE CONNECTIVITY DATA')
DO 42 KELEM = 1, NELEM
WRITE(7,615) KELEM, (NODES(KPE,KELEM), KPE = 1, NPE)
42 CONTINUE
615 FORMAT(2X,I5,2X,1017, 2(/9X,1017))

WRITE(7,620)
DO 45 KNODE = 1, NNODE
WRITE(7,621) KNODE, (IBCA(KNODE,KPROB), KPROB = 1,3)
45 CONTINUE

WRITE(7,624)
WRITE(7,625) (IPNOD(K), K=1,2), PBCDAT
620 FORMAT(2X,'IBC-DATA FOR IA-1,2,3,4,6')
621 FORMAT(4X,I5,2X,216,E14.6)
624 FORMAT(2X,'IPNOD(1-2) AND PBC-DATA')
625 FORMAT(2X,216,E14.6)

WRITE(7,630)
DO 50 KNODE = 1, NNODE
WRITE(7,631) KNODE, (A(KNODE,K), K = 1,4)
50 CONTINUE
630 FORMAT(2X,'KNODE U V W P')
C
WRITE(7,640)
DO 70 KNODE-1,NNODE
   WRITE(7,641) KNODE,(ADBC(KNODE,K),K=1,3),ADBC(KNODE,6)
70 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)
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
RETURN
END
*******************************************************************
BOTTOM OF DATA*******************************************************************
APPENDIX II
INPUT DATA FOR NSFLOW/L

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 = 2, Solve two-dimensional flows using the new pressure interpolation method; = 3, Solve two-dimensional flows using the pressure interpolation polynomials of the form \((1, x, y)\); = 11, Solve three-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.

**Description of the input data for METHOD=1**

- **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.

**Description of input data for METHOD=2**

- **((XNOD(KPE,KDIM),KDIM=1,NDIM),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 coordinate 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.

   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, uses are 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.

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 of the main program.
A-2-1. Cavity Flow for Re = 10,000

CAVITY FLOW FOR REYNOLDS NUMBER=10000 (CAVT91)

****INITIALIZE DIMENSIONED VARIABLES****

****PREPARE INPUT DATA  

DESCRIPTIVE DATA .............................................

IFLOW, NDIM, NGAUS, MFRONF,
2, 2, 3, 165,

CNTL PARAMETERS .............................................

NNODE, NELEM, IAXSY, IPILOT,
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.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, -0.7, -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, 0.7, 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, -0.96, -0.96, -0.95, -0.935,
-0.92, -0.905, -0.89, -0.88, -0.85, -0.83, -0.81, -0.785,
-0.76, -0.73, -0.7, -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, 0.7, 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  

76
NUMBER OF BLOCKS (NBLOC)
   1,
ELEMNT 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,

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 --(PBKDAT, 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

--- LAMINAR BACKWARD-FACING STEP FLOW (STP5) ---

****INITIALIZE DIMENSIONED VARIABLES ****

****PREPARE INPUT DATA ****

DESCRIPTIVE DATA 

IFLOW, NDIM, NGAUS, MFRONF,
2, 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,

7, 1, 2, 2, 2, 2, 2, 2, 2, 2,

1, 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,

7, 1, 2, 2, 2, 2, 2, 2, 2, 2,

1, 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,

15, 1, 2, 2, 2, 2, 2, 2, 2, 2,

1, 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 COORDINATE 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, 0.007535, 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, 0.00978, 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.,
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 Laminar Flow in a Square Duct of Strong Curvature

*----------------------------------- TOP OF DATA -----------------------------------*

---- 3-D DUCT FLOW WITH STRONG CURVATURE ----
****INITIALIZE DIMENSIONED VARIABLES ****
****PREPARE INPUT DATA ****
DESCRIPTIVE DATA ---------------------------------
IFLOW, NDIM, NGAUS, MFRONF,
11, 3, 3, 0,
CNTL PARAMETERS ---------------------------------
NNODE, NELEM, IAXSY, IPLOT,
19825, 2160, 0, 1,
NODE COORDINATE DATA --------------------------
NUMBER OF BLOCKS (NBLOC)
3,
GRID GENERATION METHOD FOR IBLOC=1 (METHOD)
2,
XNOD(KPE,KDIM) --- COORD-DATA FOR COMPUTATIONAL ELEMENT
0.,0., 0.088, 0.01, 0.088, 0.02, 0.0, 0.088,
0.02, 0.02, 0.088, 0.02, 0.04, 0.088, 0.01, 0.04, 0.088,
0.0, 0.04, 0.088, 0.0, 0.02, 0.088, 0.0, 0.144,
0.01, 0.0, 0.144, 0.02, 0.0, 0.144, 0.02, 0.02, 0.144,
0.02, 0.04, 0.144, 0.01, 0.04, 0.144, 0.0, 0.04, 0.144,
0.0, 0.02, 0.144, 0., 0., 0.2, 0.01, 0., 0.2,
0.02, 0., 0.2, 0.02, 0.02, 0.2, 0.02, 0.04, 0.2,
0.01, 0.04, 0.2, 0.0, 0.04, 0.2, 0.0, 0.02, 0.2,
0.01, 0.02, 0.088, 0.01, 0.02, 0.2, 0.01, 0.02, 0.144,
NODGl, INCrx, INCry, INCrz,
1, 1, 13, 325,
NDAT, GRID COORDINATE DATA
13, -1., -0.8, -0.6, -0.4, -0.2, 0.88, 0.94, 1.,
0., 0.2, 0.4, 0.6, 0.74, 0.97, 1.,
25, -1., -0.97, -0.94, -0.87, -0.8, -0.7, -0.6, -0.5, -0.4, -0.3, -0.2, -0.1, 0., 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7,
0.8, 0.87, 0.94, 0.97, 1.,
13, -1., -0.82, -0.64, -0.46, -0.28, -0.1, 0.08, 0.26, 0.44, 0.6,
0.76, 0.88, 1.,
GRID GENERATION METHOD FOR IBLOC=2 (METHOD)
2,
XNOD(KPE,KDIM) --- COORD-DATA FOR COMPUTATIONAL ELEMENT
0., 0.02, 0.0, 0.02, 0.0, 0.2,
0.02, 0.02, 0.2, 0.02, 0.04, 0.2, 0.01, 0.04, 0.2,
0.0, 0.04, 0.2, 0.0, 0.02, 0.2,
0., 0.03280404, 0.279195959, 0.01, 0.03280404, 0.279195959,
0.02, 0.03280404, 0.279195959, 0.02, 0.046946176, 0.265053823,
0.02, 0.061088311, 0.250911688, 0.01, 0.061088311, 0.250911688,
0., 0.061088311, 0.250911688, 0., 0.046946176, 0.265053823,
0., 0.112, 0.312,
NODG1, INCRX, INCRY, INCRZ,
3901, 1, 13, 325,
NDAT, GRID COORDINATE DATA

GRID GENERATION METHOD FOR IBLOC=3 (METHOD)

2,

XNOD(KPE,KDIM) --- COORD-DATA FOR COMPUTATIONAL ELEMENT

GRID COORDINATE DATA
ELEMENT CONNECTIVITY DATA FOR THE GLOBAL DOMAIN

NUMBER OF BLOCKS (NBLOC)
1,

NODE CONNECTIVITY DATA FOR IBLOC=1 (IEL1,NODES)
1, 1, 2, 3, 16, 29, 28, 27, 14,
326, 327, 328, 341, 354, 353, 352, 339,
651, 652, 653, 666, 679, 678, 677, 664,
15, 665, 340.

NEL(KDIM),INCREL(KDIM), INCNOD(KDIM)
6, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2,
2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
12, 6, 26, 26, 26, 26, 26, 26, 26, 26,
26, 26, 26, 26, 26, 26, 26, 26, 26,
30, 72, 650, 650, 650, 650, 650, 650, 650, 650,
650, 650, 650, 650, 650, 650, 650, 650, 650,
650, 650, 650, 650, 650, 650, 650, 650, 650,
650, 650, 650, 650, 650, 650, 650, 650, 650,
650, 650, 650, 650, 650, 650, 650, 650, 650,
650, 650, 650, 650, 650, 650, 650, 650, 650,
650, 650, 650, 650, 650, 650, 650, 650, 650,
650, 650, 650, 650, 650, 650, 650, 650, 650,
650, 650, 650, 650, 650, 650, 650, 650, 650,
650, 650, 650, 650, 650, 650, 650, 650, 650,
650, 650, 650, 650, 650, 650, 650, 650, 650,
650, 650, 650, 650, 650, 650, 650, 650, 650,
650, 650, 650, 650, 650, 650, 650, 650, 650,
650, 650, 650, 650, 650, 650, 650, 650, 650,
650, 650, 650, 650, 650, 650, 650, 650, 650,
650, 650, 650, 650, 650, 650, 650, 650, 650,
650, 650, 650, 650, 650, 650, 650, 650, 650,
650, 650, 650, 650, 650, 650, 650, 650, 650,
650, 650, 650, 650, 650, 650, 650, 650, 650,
650, 650, 650, 650, 650, 650, 650, 650, 650,
650, 650, 650, 650, 650, 650, 650, 650, 650,
650, 650, 650, 650, 650, 650, 650, 650, 650,
650, 650, 650, 650, 650, 650, 650, 650, 650,
650, 650, 650, 650, 650, 650, 650, 650, 650,
650, 650, 650, 650, 650, 650, 650, 650, 650,
650, 650, 650, 650, 650, 650, 650, 650, 650,
650, 650, 650, 650, 650, 650, 650, 650, 650,
650, 650, 650, 650, 650, 650, 650, 650, 650,
650, 650, 650, 650, 650, 650, 650, 650, 650,
650, 650, 650, 650, 650, 650, 650, 650, 650,
650, 650, 650, 650, 650, 650, 650, 650, 650,
650, 650, 650, 650, 650, 650, 650, 650, 650,
650, 650, 650, 650, 650, 650, 650, 650, 650,
<table>
<thead>
<tr>
<th></th>
<th>13</th>
<th>1</th>
<th>25</th>
<th>61</th>
<th>0</th>
<th>13</th>
<th>325</th>
<th>0</th>
<th>13</th>
<th>1</th>
<th>61</th>
<th>1</th>
<th>0</th>
<th>325</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.00000000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.00000000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.00000000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.00000000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.00000000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.00000000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.00000000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.00000000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.00000000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.00000000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.00000000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.00000000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.00000000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.14107073</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.14017388</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.13653400</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.13283643</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.12612298</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.11704388</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.10803068</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>21</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.08986184</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>22</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.06991971</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>23</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.05203716</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>24</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.02887626</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>25</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.01612664</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>26</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.00000000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>27</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.27589211</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>28</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.27410454</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>29</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.26726292</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>30</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.25945320</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>31</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.24604894</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>32</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.22792856</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>33</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.20921753</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>34</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.17372024</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>35</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.13404164</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>36</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.09858202</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>37</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.05316719</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>38</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.02887081</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>39</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.000000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>40</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.56707486</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>41</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.56323512</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>42</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.54924424</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>43</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.53174448</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>44</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.50295666</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>45</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.46408786</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>46</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.42135755</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>47</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.34826482</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>48</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.26426789</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>49</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.19028446</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>50</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.09859682</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>51</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.05204642</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>52</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.000000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

84
<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>1</th>
<th>1</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0.82711370</th>
</tr>
</thead>
<tbody>
<tr>
<td>54</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.82128529</td>
</tr>
<tr>
<td>55</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.80078752</td>
</tr>
<tr>
<td>56</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.77350367</td>
</tr>
<tr>
<td>57</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.72988076</td>
</tr>
<tr>
<td>58</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.67109105</td>
</tr>
<tr>
<td>59</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.60404118</td>
</tr>
<tr>
<td>60</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.49696052</td>
</tr>
<tr>
<td>61</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.37226771</td>
</tr>
<tr>
<td>62</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.26427482</td>
</tr>
<tr>
<td>63</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.13406339</td>
</tr>
<tr>
<td>64</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.06993587</td>
</tr>
<tr>
<td>65</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.000000</td>
</tr>
<tr>
<td>66</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1.14840269</td>
</tr>
<tr>
<td>67</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1.13990303</td>
</tr>
<tr>
<td>68</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1.11101123</td>
</tr>
<tr>
<td>69</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1.07030254</td>
</tr>
<tr>
<td>70</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1.00692210</td>
</tr>
<tr>
<td>71</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.92179497</td>
</tr>
<tr>
<td>72</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.82193331</td>
</tr>
<tr>
<td>73</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.67226107</td>
</tr>
<tr>
<td>74</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.49695325</td>
</tr>
<tr>
<td>75</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.34826449</td>
</tr>
<tr>
<td>76</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.17373472</td>
</tr>
<tr>
<td>77</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.08987081</td>
</tr>
<tr>
<td>78</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.000000</td>
</tr>
<tr>
<td>79</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1.41539637</td>
</tr>
<tr>
<td>80</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1.40449035</td>
</tr>
<tr>
<td>81</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1.36821233</td>
</tr>
<tr>
<td>82</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1.31531694</td>
</tr>
<tr>
<td>83</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1.23434608</td>
</tr>
<tr>
<td>84</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1.12600781</td>
</tr>
<tr>
<td>85</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.99720242</td>
</tr>
<tr>
<td>86</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.81174605</td>
</tr>
<tr>
<td>87</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.59467088</td>
</tr>
<tr>
<td>88</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.41341425</td>
</tr>
<tr>
<td>89</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.20427502</td>
</tr>
<tr>
<td>90</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.10519067</td>
</tr>
<tr>
<td>91</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.000000</td>
</tr>
<tr>
<td>92</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1.63288940</td>
</tr>
<tr>
<td>93</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1.61989608</td>
</tr>
<tr>
<td>94</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1.57728044</td>
</tr>
<tr>
<td>95</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1.51381935</td>
</tr>
<tr>
<td>96</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1.41778420</td>
</tr>
<tr>
<td>97</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1.28974150</td>
</tr>
<tr>
<td>98</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1.13655992</td>
</tr>
<tr>
<td>99</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.92179717</td>
</tr>
<tr>
<td>100</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.67108599</td>
</tr>
<tr>
<td>101</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.46408975</td>
</tr>
<tr>
<td>102</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.22794528</td>
</tr>
<tr>
<td>103</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.11705506</td>
</tr>
<tr>
<td>104</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.000000</td>
</tr>
<tr>
<td>105</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1.80504454</td>
</tr>
<tr>
<td>106</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1.79031847</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>----------------</td>
<td></td>
</tr>
<tr>
<td>107</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1.74247049</td>
</tr>
<tr>
<td>108</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1.67026695</td>
</tr>
<tr>
<td>109</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1.56186391</td>
</tr>
<tr>
<td>110</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1.41778361</td>
</tr>
<tr>
<td>111</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1.24493065</td>
</tr>
<tr>
<td>112</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1.00692372</td>
</tr>
<tr>
<td>113</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.72987512</td>
</tr>
<tr>
<td>114</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.50295796</td>
</tr>
<tr>
<td>115</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.24606508</td>
</tr>
<tr>
<td>116</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.12613359</td>
</tr>
<tr>
<td>117</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.000000</td>
</tr>
<tr>
<td>118</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1.93532049</td>
</tr>
<tr>
<td>119</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1.91923665</td>
</tr>
<tr>
<td>120</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1.86730113</td>
</tr>
<tr>
<td>121</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1.78827236</td>
</tr>
<tr>
<td>122</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1.67026865</td>
</tr>
<tr>
<td>123</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1.51382045</td>
</tr>
<tr>
<td>124</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1.32590929</td>
</tr>
<tr>
<td>125</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1.07030586</td>
</tr>
<tr>
<td>126</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.77349953</td>
</tr>
<tr>
<td>127</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.53174749</td>
</tr>
<tr>
<td>128</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.25947104</td>
</tr>
<tr>
<td>129</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.13284874</td>
</tr>
<tr>
<td>130</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.000000</td>
</tr>
<tr>
<td>131</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2.02642949</td>
</tr>
<tr>
<td>132</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2.00937277</td>
</tr>
<tr>
<td>133</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1.95451373</td>
</tr>
<tr>
<td>134</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1.87060992</td>
</tr>
<tr>
<td>135</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1.74577595</td>
</tr>
<tr>
<td>136</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1.58057084</td>
</tr>
<tr>
<td>137</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1.38205629</td>
</tr>
<tr>
<td>138</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1.11414976</td>
</tr>
<tr>
<td>139</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.80361284</td>
</tr>
<tr>
<td>140</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.55159848</td>
</tr>
<tr>
<td>141</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.26870843</td>
</tr>
<tr>
<td>142</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.13747516</td>
</tr>
<tr>
<td>143</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.000000</td>
</tr>
<tr>
<td>144</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2.08031605</td>
</tr>
<tr>
<td>145</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2.06267508</td>
</tr>
<tr>
<td>146</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2.00606218</td>
</tr>
<tr>
<td>147</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1.91923715</td>
</tr>
<tr>
<td>148</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1.79032067</td>
</tr>
<tr>
<td>149</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1.61989769</td>
</tr>
<tr>
<td>150</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1.41508740</td>
</tr>
<tr>
<td>151</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1.13990685</td>
</tr>
<tr>
<td>152</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.82128186</td>
</tr>
<tr>
<td>153</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.56323865</td>
</tr>
<tr>
<td>154</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.27412290</td>
</tr>
<tr>
<td>155</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.14018670</td>
</tr>
<tr>
<td>156</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.000000</td>
</tr>
<tr>
<td>157</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2.09814840</td>
</tr>
<tr>
<td>158</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2.08031265</td>
</tr>
<tr>
<td>159</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2.02311537</td>
</tr>
<tr>
<td>160</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1.93531760</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Value</td>
<td></td>
<td></td>
</tr>
<tr>
<td>-----</td>
<td>-----</td>
<td>-----</td>
<td>-----</td>
<td>-----</td>
<td>--------</td>
<td></td>
<td></td>
</tr>
<tr>
<td>161</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0.000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>162</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0.000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>163</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0.000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>164</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0.000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>165</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0.000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>166</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0.000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>167</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0.000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>168</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0.000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>169</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0.000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>170</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0.000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>171</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0.000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>172</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0.000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>173</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0.000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>174</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0.000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>175</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0.000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>176</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0.000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>177</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0.000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>178</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0.000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>179</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0.000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>180</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0.000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>181</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0.000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>182</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0.000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>183</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0.000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>184</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0.000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>185</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0.000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>186</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0.000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>187</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0.000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>188</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0.000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>189</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0.000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>190</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0.000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>191</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0.000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>192</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0.000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>193</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0.000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>194</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0.000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>195</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0.000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>196</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0.000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>197</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0.000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>198</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0.000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>199</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0.000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>200</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0.000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>201</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0.000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>202</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0.000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>203</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0.000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>204</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0.000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>205</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0.000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>206</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0.000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>207</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0.000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>208</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0.000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>209</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0.000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>210</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0.000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>211</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0.000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>212</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0.000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>213</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0.000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>214</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0.000</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>----------------------</td>
</tr>
<tr>
<td>215</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1.24493065</td>
</tr>
<tr>
<td>216</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1.00692372</td>
</tr>
<tr>
<td>217</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.72987512</td>
</tr>
<tr>
<td>218</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.50295796</td>
</tr>
<tr>
<td>219</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.24606508</td>
</tr>
<tr>
<td>220</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.12613359</td>
</tr>
<tr>
<td>221</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.000000</td>
</tr>
<tr>
<td>222</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1.63288940</td>
</tr>
<tr>
<td>223</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1.61989608</td>
</tr>
<tr>
<td>224</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1.57728044</td>
</tr>
<tr>
<td>225</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1.51381935</td>
</tr>
<tr>
<td>226</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1.41778420</td>
</tr>
<tr>
<td>227</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1.28974150</td>
</tr>
<tr>
<td>228</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1.13655992</td>
</tr>
<tr>
<td>229</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.92179717</td>
</tr>
<tr>
<td>230</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.67108599</td>
</tr>
<tr>
<td>231</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.46408975</td>
</tr>
<tr>
<td>232</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.22794528</td>
</tr>
<tr>
<td>233</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.11705506</td>
</tr>
<tr>
<td>234</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.000000</td>
</tr>
<tr>
<td>235</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1.41539637</td>
</tr>
<tr>
<td>236</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1.40449035</td>
</tr>
<tr>
<td>237</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1.36821233</td>
</tr>
<tr>
<td>238</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1.31531694</td>
</tr>
<tr>
<td>239</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1.23434968</td>
</tr>
<tr>
<td>240</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1.12600781</td>
</tr>
<tr>
<td>241</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.99720242</td>
</tr>
<tr>
<td>242</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.81174605</td>
</tr>
<tr>
<td>243</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.59467088</td>
</tr>
<tr>
<td>244</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.41341425</td>
</tr>
<tr>
<td>245</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.20427502</td>
</tr>
<tr>
<td>246</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.10519067</td>
</tr>
<tr>
<td>247</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.000000</td>
</tr>
<tr>
<td>248</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1.14840269</td>
</tr>
<tr>
<td>249</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1.13990303</td>
</tr>
<tr>
<td>250</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1.11101123</td>
</tr>
<tr>
<td>251</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1.07030254</td>
</tr>
<tr>
<td>252</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1.00692210</td>
</tr>
<tr>
<td>253</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.92179497</td>
</tr>
<tr>
<td>254</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.82193331</td>
</tr>
<tr>
<td>255</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.67226107</td>
</tr>
<tr>
<td>256</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.49695325</td>
</tr>
<tr>
<td>257</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.34826449</td>
</tr>
<tr>
<td>258</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.17373472</td>
</tr>
<tr>
<td>259</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.08987081</td>
</tr>
<tr>
<td>260</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.000000</td>
</tr>
<tr>
<td>261</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.82711370</td>
</tr>
<tr>
<td>262</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.82128529</td>
</tr>
<tr>
<td>263</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.80078752</td>
</tr>
<tr>
<td>264</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.77350347</td>
</tr>
<tr>
<td>265</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.72988076</td>
</tr>
<tr>
<td>266</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.67109105</td>
</tr>
<tr>
<td>267</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.60404118</td>
</tr>
<tr>
<td>268</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.49696052</td>
</tr>
</tbody>
</table>
269  1  1  1  0  0  0  0.37226771
270  1  1  1  0  0  0  0.26427482
271  1  1  1  0  0  0  0.13406339
272  1  1  1  0  0  0  0.06993587
273  1  1  1  0  0  0  0.000000
274  1  1  1  0  0  0  0.56707486
275  1  1  1  0  0  0  0.56323512
276  1  1  1  0  0  0  0.54924424
277  1  1  1  0  0  0  0.53174448
278  1  1  1  0  0  0  0.50295666
279  1  1  1  0  0  0  0.46408786
280  1  1  1  0  0  0  0.42135755
281  1  1  1  0  0  0  0.34826482
282  1  1  1  0  0  0  0.26426789
283  1  1  1  0  0  0  0.19028446
284  1  1  1  0  0  0  0.09859682
285  1  1  1  0  0  0  0.05204642
286  1  1  1  0  0  0  0.000000
287  1  1  1  0  0  0  0.27589211
288  1  1  1  0  0  0  0.27410454
289  1  1  1  0  0  0  0.26726292
290  1  1  1  0  0  0  0.25945320
291  1  1  1  0  0  0  0.24604894
292  1  1  1  0  0  0  0.22792856
293  1  1  1  0  0  0  0.20921753
294  1  1  1  0  0  0  0.17372024
295  1  1  1  0  0  0  0.13404164
296  1  1  1  0  0  0  0.09858202
297  1  1  1  0  0  0  0.05316719
298  1  1  1  0  0  0  0.02887081
299  1  1  1  0  0  0  0.000000
300  1  1  1  0  0  0  0.14107073
301  1  1  1  0  0  0  0.14017388
302  1  1  1  0  0  0  0.13665340
303  1  1  1  0  0  0  0.13283643
304  1  1  1  0  0  0  0.12612298
305  1  1  1  0  0  0  0.11704388
306  1  1  1  0  0  0  0.10803068
307  1  1  1  0  0  0  0.08986184
308  1  1  1  0  0  0  0.06991971
309  1  1  1  0  0  0  0.05203716
310  1  1  1  0  0  0  0.02887626
311  1  1  1  0  0  0  0.01612664
312  1  1  1  0  0  0  0.000000
313  1  1  1  0  0  0  0.0000000
314  1  1  1  0  0  0  0.0000000
315  1  1  1  0  0  0  0.0000000
316  1  1  1  0  0  0  0.0000000
317  1  1  1  0  0  0  0.0000000
318  1  1  1  0  0  0  0.0000000
319  1  1  1  0  0  0  0.0000000
320  1  1  1  0  0  0  0.0000000
321  1  1  1  0  0  0  0.0000000
322  1  1  1  0  0  0  0.0000000
IA04 ------ PBCDAT,IPNOD(1-2) ------------------------
      0., -19657,  19657,
END OF INPUT DATA
****PROCESSOR FOR NAVIER-STOKES EQUATION ****
****END ****
************************************************************************************************* BOTTOM OF DATA ***********************************************
APPENDIX III
DESCRIPTION OF THE SUBROUTINES

INITIAL - 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.

LSHPI - 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.

SHAP03 - Shape functions for tetrahedral element.

SHAP21 - Shape functions for bi-linear quadrilateral element.

SHAP22 - Shape functions for serendipity element.

SHAP23 - Shape functions for bi-quadratic quadrilateral element.

SHAP33 - Shape functions for tri-quadratic cubic 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.

SPRS - Compute the nodal pressure.

PFLOW - Print out the computational results.

USER - Load the coordinate data and the flow variables for each element.
A VELOCITY-PRESSURE INTEGRATED, MIXED INTERPOLATION, GALERKIN FINITE ELEMENT METHOD FOR HIGH REYNOLDS NUMBER LAMINAR 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