FINITE ELEMENT ANALYSIS OF FLOW BETWEEN TWO FINITE COAXIAL ROTATING DISKS

by

Ali Cemal Benim

Submitted to the Faculty of Engineering

In Partial Fulfillment of the Requirements

for the Degree of

MASTER OF SCIENCE

in

MECHANICAL ENGINEERING

Boğaziçi University

1981

APROVED BY :

Prof.Dr.AKIN TEZEL
(Supervisor)

Doç.Dr.MUHSİN MENGÜTÜRK

Dr.SAHİM TEKELİ

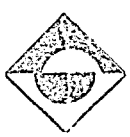


TABLE OF CONTENTS

A C K N O W L E D G E M E N T S

I would like to express my thanks to my thesis supervisor Professor Dr.Akın Tezel, to Assoc.Prof. Muhsin Mengütürk and to Dr.Sahim Tekeli for their valuable guidance.

ABSTRACT

In this study, the steady and incompressible flow between rotating, coaxial, finite disks is solved by the Finite Element Method. In the analysis velocity and pressure formulation is used. An iterative procedure which permits linearization of the governing equations is applied. Axisymmetrical-ring elements with triangular cross sections are used to discretize the domain and the Galerkin Method is applied to derive the element equations. A computer program is developed and the flow between a rotating disk and a stationary shroud and the flow inside a cylinder rotating about its axis of symmetry is investigated. It is observed that secondary flows exhibit a single-cell structure for the flow between a rotating disk an a stationary shroud and a double-cell structure for the flow inside a cylinder which rotates about its axis of symmetry.

ÖZET

Bu çalışmada, eksenleri çakışık olarak dönen iki sonlu diskin arasındaki, zamana bağımsız ve sıkıştırılamayan akışın, sonlu elemanlar metodu ile nümerik çözümü amaçlanmıştır. Hız ve basınç formülasyonu kullanılmış ve denklemler iteratif bir çözüm metodu uygulanarak lineerleştirilmiştir. Çözüm alanı üçgen kesitli, eksensel simetrik elemanlara bölünmüş ve eleman denklemleri Galerkin Metodu uygulanarak bulunmuştur. Bir kompüter programı geliştirilmiş, dönen bir disk ile duran bir muhafaza arasındaki akış ve dönen bir disk ile dönen bir muhafaza arasındaki akış problemleri incelenmiştir. İkincil akışların dönen bir disk ile duran bir muhafaza arasındaki akış için tek hücreli, simetri ekseni etrafında dönen bir silindir içindeki akış için ise iki hücreli bir yapı gösterdiği izlenmiştir.

LIST OF FIGURES

PAGE

NOMENCLATURE

d	Disk spacing
H_k	Interpolation functions for pressure
L_i	Natural area coordinates
Ni	Interpolation functions for velocity
p	Pressure
p'	$=p/\rho r_o^2 \Omega^2$, Nondimensional pressure
r_o	Disk radius
Re	$=r^2\Omega/\nu$, Reynolds number based on disk radius
Re_d	$= d^2\Omega/\nu$, Reynolds number based on disk spacing
u	Velocity component in r-direction
u'	$=u/r_o\Omega$, Nondimensional Velocity component in r-direction
v	Velocity component in θ-direction
v'	$=v/r_o\Omega$, Nondimensional velocity component in -direction
w	Velocity component in z-direction
w'	$=w/d\Omega$, Nondimensional velocity component in z-direction
ϕ	Any field variable
ϕ_i	Nodal values of the field variable
ν	Dynamic viscosity
Ω	Rotational velocity of the disks
ρ	Density

I

INTRODUCTION

Due to its applications in turbines and lubrication of thrust bearings, the flow between two rotating disks has attracted continued attention.

A survey of the available literature shows that considerable information exists on fluid flow induced by a rotating infinite disk in an infinite medium, but less information is available about a rotating disk enclosed in a chamber of finite dimensions.

A. Free Disk

Fluid flow adjacent to a rotating disk is a subject of widespread practical interest in connection with steam and gas turbines, pumps and other rotating fluid machinery. The modeling of the rotating elements as a plane circular disk frequently has been found to be useful. The motion associated with rotation of the disk in an infinite medium -the so called- "free-disk" is an example for which analytical solutions of the Navier-Stokes equations are possible. For this reason, ample attention has been given to the analysis of the free disk flow.

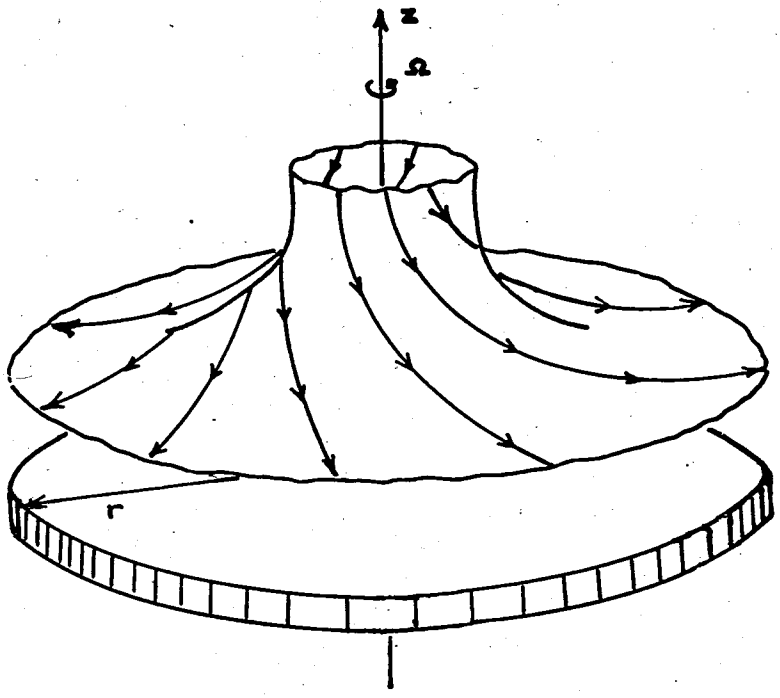


Figure I.1- Flow in the Neighbourhood of a Disk Rotating in a Fluid at Rest.

The first solution was presented by Von Kármán(1). It is shown that if the axial velocity of the flow is assumed to be independent of the radial distance, then the Navier-Stokes equations can be reduced to a system of nonlinear ordinary differential equations. Later the solution has been improved by some other authors(5).

B. Shrouded Rotating Disk With Radial Throughflow

A shrouded rotating disk with radial inflow has been studied by Soo (6). Later Canover(7) has studied this problem experimentally.

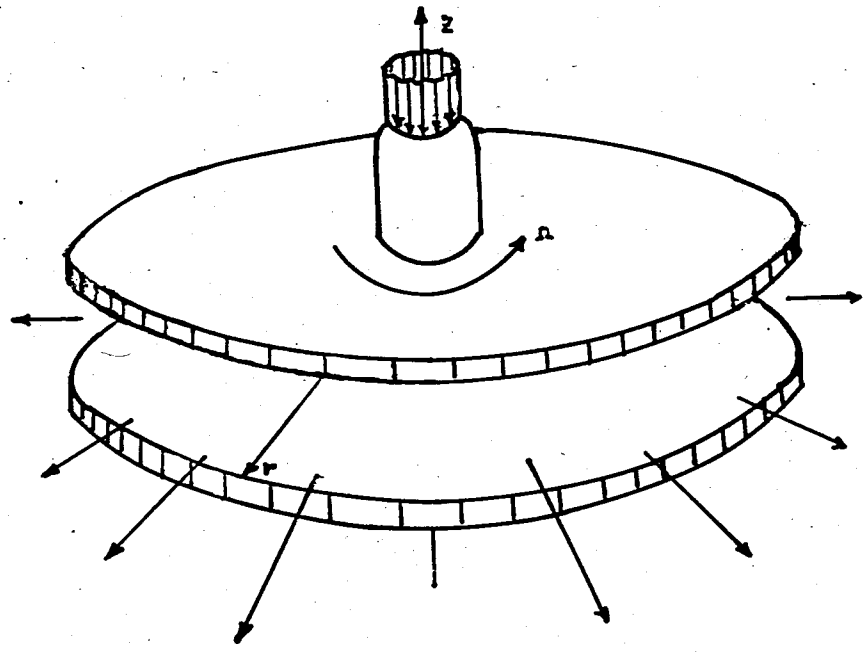


Figure I.2- Shrouded Rotating Disk With Radial Outflow

Mitchell and Metzger(8) have investigated the same problem both analytically and experimentally. Unfortunately the agreement between the experiment and the analysis was not satisfactory. Investigations regarding the case of radial outflow may be credited to Bayley and Owen(9).

C. Enclosed Disk

Since most fluid machinery have an internal geometry where non-rotating surfaces are located in close proximity to rotating surfaces, the case of a plane disk rotating in concentric cylindrical housing has received some attention. The flow was investigated by Schultz-Grunow(5) and Soo(6). Daily and Nece(10) have studied the problem both experimentally and theoretically for Reynolds numbers around 10^5 to 10^7. They also considered various types of throughflow between the shroud and the tip of the disk as well as no throughflow case.

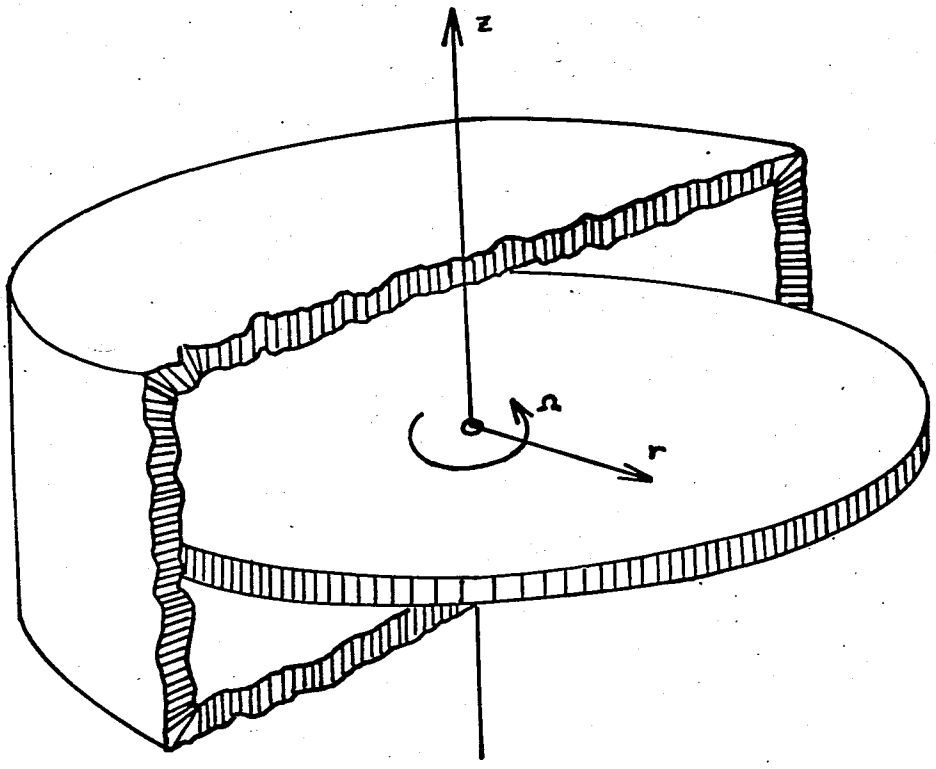


Figure.I.3- Enclosed Disk

Later the flow is studied by Senoo, Y , Hayami,H.,(II). The solutions are obtained by making use of the boundary layer theory. Mellor, et.al.(12) obtained a solution by assuming infinite disks and reducing the governing equation to a set of ordinary differential equations by using Von Karman's hpyhotesis, that the axial component of the velocity is independent of the radial distance. In this case, the Navier-Stokes equations can be reduced to a set of ordinary differential equations. Nguyen, Ribault and Florent(2), Roberts and Shipman(3) are some other authors who studied the problem of two infinite rotating disks.

Solution for the flow between a rotating disk and a stationary casing for intermediate Reynolds numbers, using full Navier-Stokes equations,could not be found in the avaliable literature.

In this work, laminar flow due to a rotating disk enclosed in a casing without throughflow is investigated. A mathematical model of the problem is developed without the use of any boundary-layer approximations and the resulting formulation is presented in chapter II. The equations are solved numerically by using the Finite Element method. The Finite Element formulation is discussed in Chapter III.

II

FORMULATION

A. Governing Equations

For axisymmetrical and steady flow of an incompressible fluid, the Navier-Stokes equations and the continuity equation become(14):

$$u\frac{\partial u}{\partial r} - \frac{v^2}{r} + w\frac{\partial u}{\partial z} = -\frac{1}{\rho}\frac{\partial p}{\partial r} + \nu\{\frac{\partial^2 u}{\partial r^2} + \frac{\partial}{\partial r}(\frac{u}{r}) + \frac{\partial^2 u}{\partial z^2}\} \qquad (1.a)$$

$$u\frac{\partial v}{\partial r} + \frac{uv}{r} + w\frac{\partial v}{\partial z} = \nu\{\frac{\partial^2 v}{\partial r^2} + \frac{\partial}{\partial r}(\frac{v}{r}) + \frac{\partial^2 v}{\partial z^2}\} \qquad (1.b)$$

$$u\frac{\partial w}{\partial r} + w\frac{\partial w}{\partial z} = -\frac{1}{\rho}\frac{\partial p}{\partial z} + \nu\{\frac{\partial^2 w}{\partial r^2} + \frac{1}{r}\frac{\partial w}{\partial r} + \frac{\partial^2 w}{\partial z^2}\} \qquad (1.c)$$

$$\frac{\partial u}{\partial r} + \frac{u}{r} + \frac{\partial w}{\partial z} = 0 \qquad (1.d)$$

where u, v, w are components of velocity in r, θ, and z directions respectively and p is the pressure.

The Navier-Stokes equations are non-dimensionalized as:

$$u' \frac{\partial u'}{\partial r'} - \frac{v'^2}{r'} + w' \frac{\partial u'}{\partial z'} = - \frac{\partial p'}{\partial r'} + \frac{1}{Re} \{ \frac{\partial^2 u'}{\partial r'^2} + \frac{\partial}{\partial r'} (\frac{u'}{r'}) + (\frac{r_o}{d})^2 \frac{\partial^2 u'}{\partial z'^2} \} \quad (2.a)$$

$$u' \frac{\partial v'}{\partial r'} + \frac{u'v'}{r} + w' \frac{\partial v'}{\partial z'} = \frac{1}{Re} \{ \frac{\partial^2 v'}{\partial r'^2} + \frac{\partial}{\partial r'} (\frac{v'}{r'}) + (\frac{r_o}{d})^2 \frac{\partial^2 v'}{\partial z'^2} \} \quad (2.b)$$

$$u' \frac{\partial w'}{\partial r'} + w' \frac{\partial w'}{\partial z'} = -(\frac{r_o}{d})^2 \frac{\partial p'}{\partial z'} + \frac{1}{Re} \{ \frac{\partial^2 w'}{\partial r'^2} + \frac{1}{r'} \frac{\partial w'}{\partial r'} + (\frac{r_o}{d})^2 \frac{\partial^2 w'}{\partial z'^2} \} \quad (2.c)$$

with the non-dimensional quantities:

$$r' = \frac{r}{r_o} , \; z' = \frac{z}{d} , \; u' = \frac{u}{r_o \Omega} , \; v' = \frac{v}{r_o \Omega} , \; w' = \frac{w}{d\Omega} , \; p' = \frac{p}{\rho r_o^2 \Omega^2} \quad (3)$$

The following independent parameters in the non-dimensional Navier-Stokes equations effect the flow characteristics.

1- Reynolds Number ($Re = r_o(r_o\Omega)/\nu$) = A flow parameter

2- The ratio $(\frac{r_o}{d})$ = A geometrical parameter.

Where, r_o = disk radius, d = disk spacing, Ω = rotational velocity of the disk (See Fig.II.1).

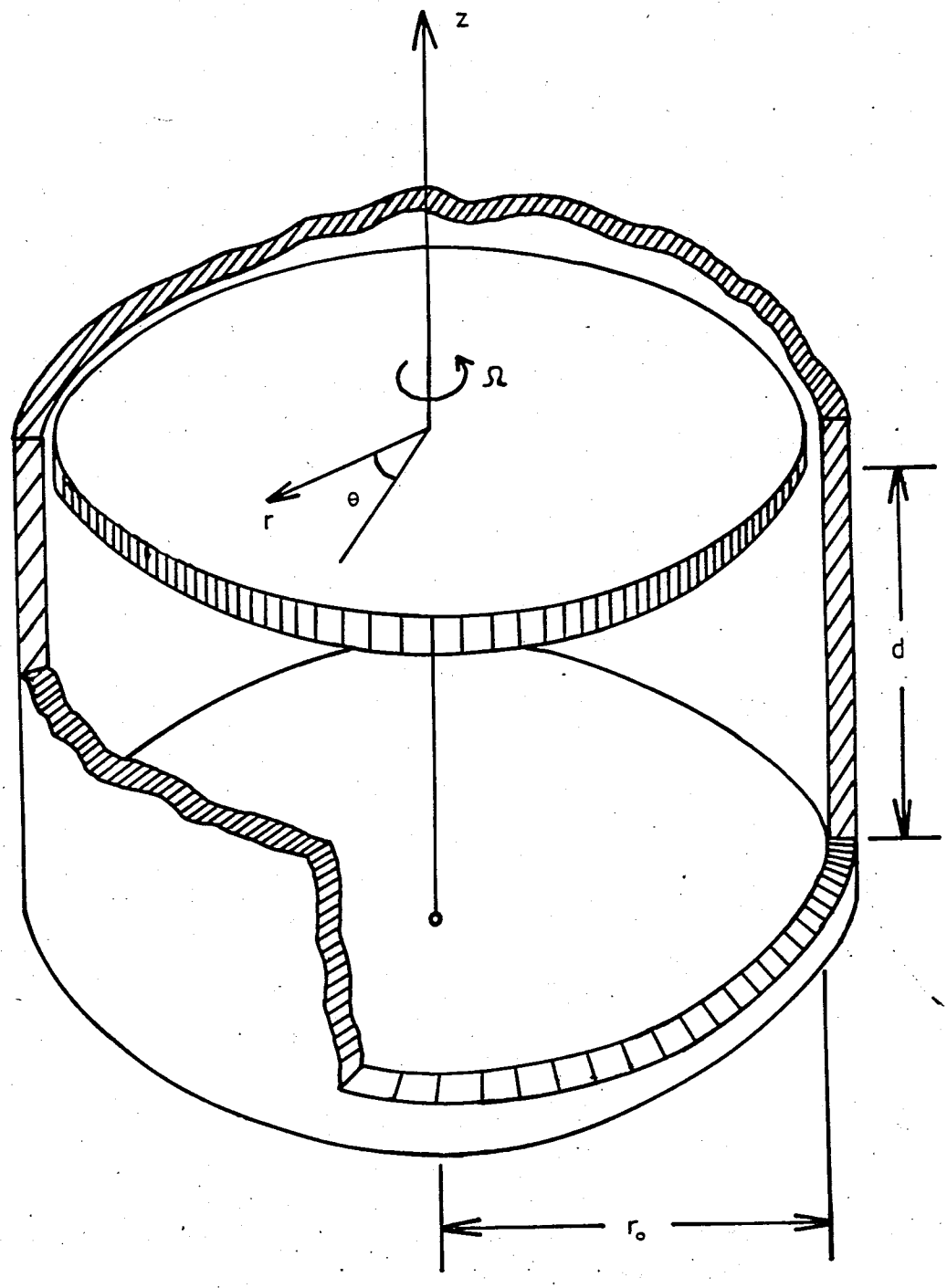


Fig.II.1. A Rotating Finite Disk Enclosed in a Casing

B. Boundary Conditions

In this work, two different problems are treated

Problem (a) : Rotation of a disk enclosed in a stationary housing

Problem (b) : Rotation of a circular cylinder about its axis of symmetry.

Since second order partial derivatives of u, v, and w with respect to r and z appear in the equations, one needs to prescribe two boundary conditions in both r and z directions for each velocity component.

The boundary conditions for rotation of a disk enclosed in a stationary housing are :

Boundary Conditions in r-direction

$$\frac{\partial u}{\partial r} = \frac{\partial v}{\partial r} = \frac{\partial w}{\partial r} = 0 \qquad \text{at } r = 0 \qquad (4a)$$

$$u = w = 0 \; ; \; v = 0 \qquad \text{at } r = r_o \qquad (4b,c)$$

Boundary Conditions in z-direction

$$u = w = 0 \; ; \; v = 0 \qquad \text{at } z = 0 \qquad (4d,e)$$

$$u = w = 0 \; ; \; v = \Omega r \qquad \text{at } z = d \qquad (4f,g)$$

Prescribed velocity conditions are the so-called no slip conditions, whereas the conditions involving first order derivatives are due to the symmetry about the axis of rotation.

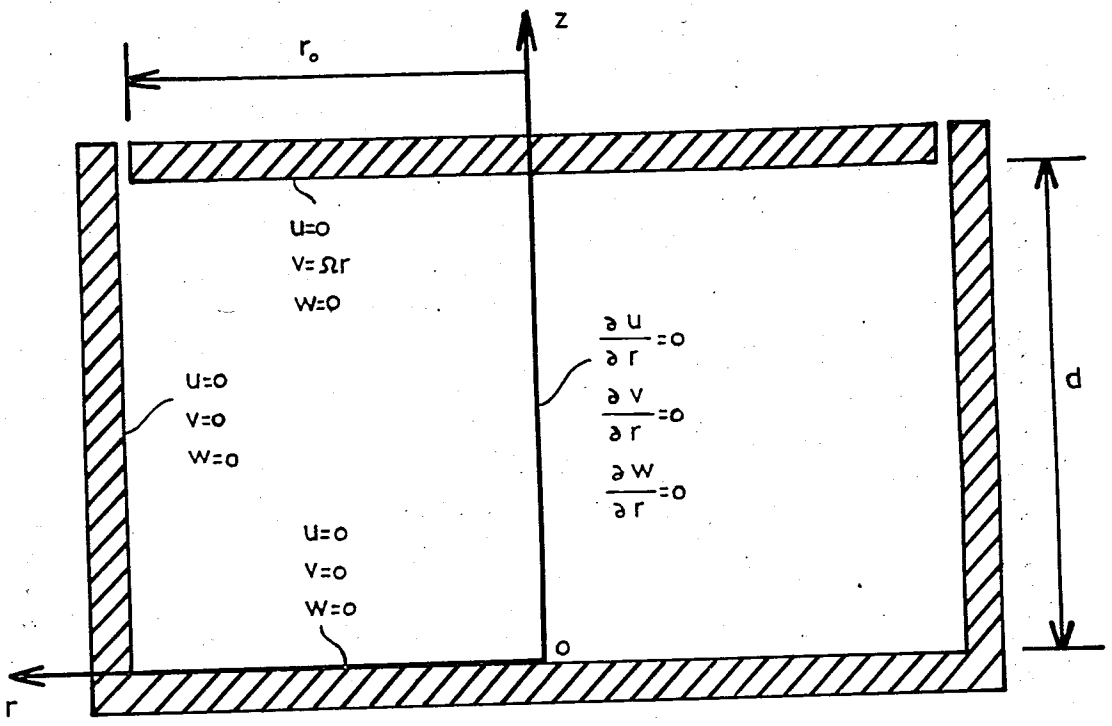


Figure II.2- Boundary Conditions for Rotation of a Disk Enclosed in a Stationary Housing

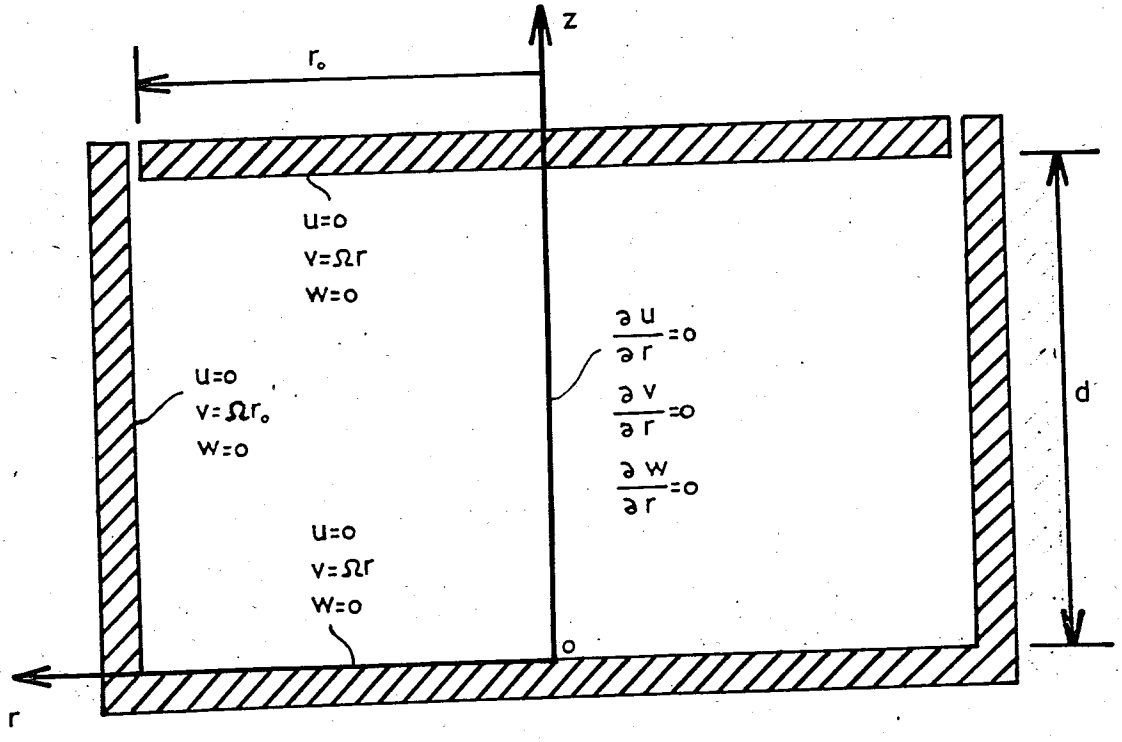


Figure II.3- Boundary Conditions for Rotation of a Circular Cylinder

For Problem (b) the conditions (4c) and (4e) become (see Fig.II.c) :

$v = \Omega r_o$ at $r = r_o$ (4h)

$v = \Omega r$ at $z = 0$ (4g)

In both of the above formulations, centerline conditions are based on symmetry considerations. The boundary conditions along the axis are of the Neumann's type (Eq.4.a). Alternately, Dirichlet's type of conditions may be prescribed, along the axis. Axial symmetry necessitates that

$u = v = 0$ at $r=0$

and evaluation of the continuity equation (Eq.1d) along the axis gives

$\frac{\partial w}{\partial z} = 0$ at $r=0$

Which implies that

w = constant along the axis of rotation

Since we have the boundary conditions

$w = 0$ at $z=0$ and $z=d$

This constant must necessarily be equal to zero.

Therefore, the Neumann's type of boundary conditions of the above formulation can be replaced by the following set of Dirichlet's type of boundary conditions

$u = v = w = 0$ at $r=0$ (4h)

Eq. (4h) will be used in the present study

Now, the formulation of the problem is complete. This type of formulation which has the primary variables velocity and pressure as dependent variables is known as the velocity-pressure formulation. It is possible to develop an alternate formulation by introducing a stream function and the vorticity. This kind of formulation is called the stream function-vorticity formulation and supplies two coupled equations in terms of the stream function and the vorticity. In this work, the velocity-pressure formulation is preferred since the corresponding equations are of lower order. Another advantage of the velocity-pressure formulation is that the pressure field is obtained directly from the coupled equations, whereas in the stream function-vorticity formulation, the pressure field must be evaluated by solving a Poisson equation.

III

FINITE ELEMENT FORMULATION

The governing equations are solved numerically with the Finite Element Method. The concepts of discretizing the continuum into a finite number of elements, and expressing the variation of the field variable within each element through interpolation functions are two basic ideas of the finite element method.

A. Domain Discretization

In the Finite Element Method, the continuum is divided into a finite number of elements. Each element has associated nodes and the values of the field variables at these nodes are searched for. For the problem of rotating disks, the domain is axisymmetric. Therefore, axisymmetric-ring elements with triangular cross-sections are used. The elements are interconnected at the nodal points. The number of nodes can increase with the order of approximation accordingly (See Fig. III.1 and 2).

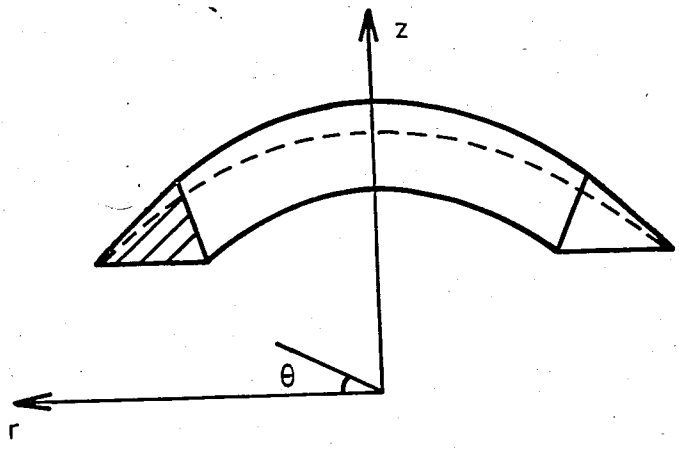


Figure III.1- A Ring Element With Triangular Cross-Section

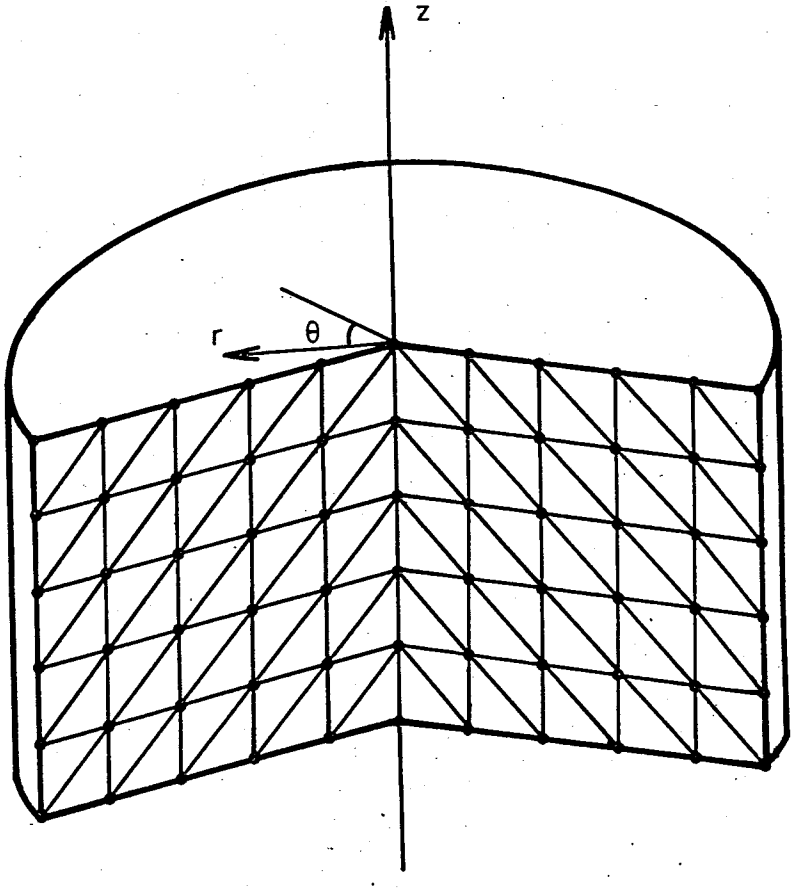


Figure III.2- Discretized Domain

B. Interpolation Functions

In the Finite Element method, the unknown field variable is approximated by:

$$\phi^{(e)} = \sum_{i=1}^{m} N_i \phi_i \qquad (5)$$

Where Ni are the selected interpolation functions, ϕ_i are the nodal values of the unknown function, and m is the number of nodes associated to that element.

The domain is discretized with axisymmetrical-ring elements having triangular cross sections. The triangular cross sections are selected with six nodes, three on the corners, three in the middle of the sides (See Fig. III.3). The velocity is approximated quadratically by considering six of the nodes and the pressure is approximated linearly by considering the corner nodes only.

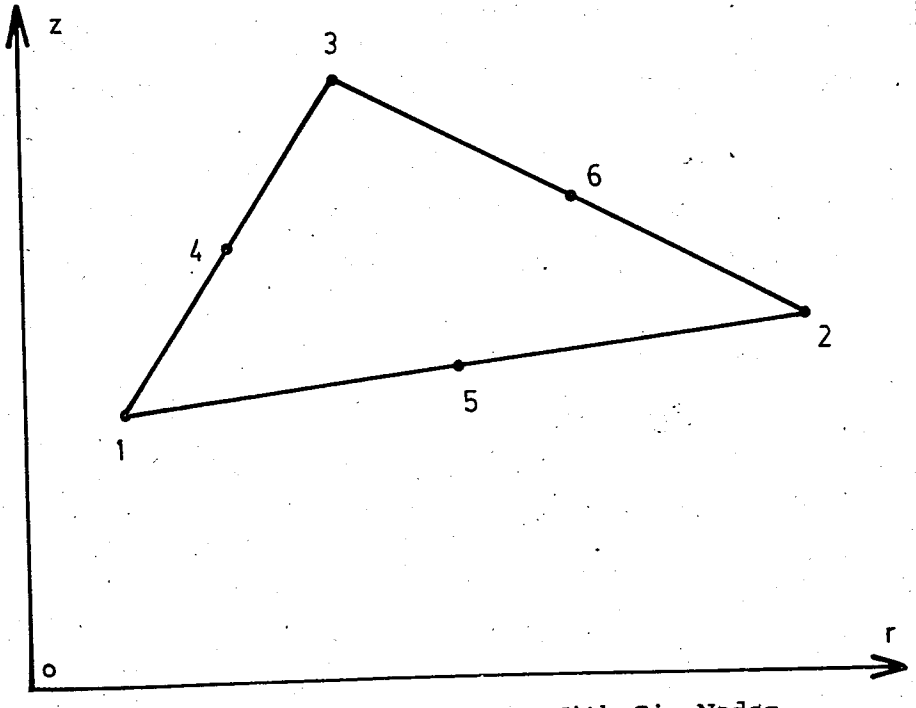


Figure III.3- A Triangular Element With Six Nodes

The use of natural coordinates in deriving the interpolation functions is advantegous because special closed form integration formulas can be used to evaluate the integrals in the element equations. Detailed information about the natural coordinates and the selected interpolation functions are given in Appendix A.

C. Element Equations

The field variables, u, v, w, p are approximated as follows :

$$\begin{aligned} u &= N_j u_j \\ v &= N_j v_j \\ w &= N_j w_j \\ p &= H_k p_k \end{aligned} \qquad (6)$$

Where u_j, v_j, w_j and p_k are unknown nodal values of the functions and N_j and H_k are interpolation functions. N_j are quadratic functions whereas H_k are linear.

To approximate the non-linear convective terms, an iterative procedure is used. The convective terms are approximated by using the previous solution and the iteration is carried until the solution converges. To apply this procedure Equations (2a,b,c) are cast into the following form :

$$u^*\frac{\partial u}{\partial r} - \frac{v^*v}{r} + w^*\frac{\partial u}{\partial z} = -\frac{\partial \rho}{\partial z} + \frac{1}{Re}\left\{\frac{\partial^2 u}{\partial r^2} + \frac{\partial}{\partial r}\left(\frac{u}{r}\right) + \left(\frac{r_o}{d}\right)^2 \frac{\partial^2 u}{\partial z^2}\right\}$$

$$u^*\frac{\partial v}{\partial r} + \frac{uv^*}{r} + w^*\frac{\partial v}{\partial z} = \frac{1}{Re}\left\{\frac{\partial^2 v}{\partial r^2} + \frac{\partial}{\partial r}\left(\frac{v}{r}\right) + \left(\frac{r_o}{d}\right)^2 \frac{\partial^2 v}{\partial z^2}\right\} \qquad (7)$$

$$u^*\frac{\partial w}{\partial r} + w^*\frac{\partial w}{\partial z} = -\left(\frac{r_o}{d}\right)^2 \frac{\partial \rho}{\partial z} + \frac{1}{Re}\left\{\frac{\partial^2 w}{\partial r^2} + \frac{1}{r}\frac{\partial w}{\partial r} + \left(\frac{r_o}{d}\right)^2 \frac{\partial^2 w}{\partial z^2}\right\}$$

Where primes denoting nondimensional quantities are dropped for convenience and u*, v*, w* are the velocity components corresponding to the last iteration. u*, v*, w* are assumed to vary linearly over an element as follows:

$$u^* = u_1L_1 + u_2L_2 + u_3L_3$$

$$v^* = v_1L_1 + v_2L_2 + v_3L_3 \qquad (7a)$$

$$w^* = w_1L_1 + w_2L_2 + w_3L_3$$

Where u_1, u_2, u_3 ..... etc. are the velocities at corner nodes and L_1, L_2, L_3 are the natural coordinates. The element equations are obtained by applying Galerkin's method and by integration by parts of the terms containing the highest order derivatives of the field variable. Detailes of this derivation is explained in Appendix B. The element equations can be put in a matrix form which is called as element matrix

$$
\left[\begin{array}{c|c|c|c|c|c}
\begin{array}{c} 6 \times 6 \\ KP_{ij} + C_{ij} + \\ UNMij \end{array} & \begin{array}{c} 6 \times 6 \\ A2_{ij} \end{array} & \begin{array}{c} 6 \times 6 \\ 0 \end{array} & \begin{array}{c} 6x3 \\ A4_{ik} \end{array} & u_j & R_i^1 \\
\hline
\begin{array}{c} 6 \times 6 \\ -A2_{ij} \end{array} & \begin{array}{c} 6 \times 6 \\ KP_{ij} + C_{ij} + \\ UNM_{ij} \end{array} & \begin{array}{c} 6 \times 6 \\ 0 \end{array} & \begin{array}{c} 6x3 \\ 0 \end{array} & v_j & R_i^2 \\
\hline
\begin{array}{c} 6 \times 6 \\ 0 \end{array} & \begin{array}{c} 6 \times 6 \\ 0 \end{array} & \begin{array}{c} 6 \times 6 \\ KP_{ij} + C_{ij} \end{array} & \begin{array}{c} 6x3 \\ A3_{ik} \end{array} & W_j & R_i^3 \\
\hline
\begin{array}{c} 3 \times 6 \\ A5_{kj} \end{array} & \begin{array}{c} 3 \times 6 \\ 0 \end{array} & \begin{array}{c} 3 \times 6 \\ A6_{kj} \end{array} & 3x3 & P_k & 0
\end{array}\right] \qquad (8)
$$

$(2l \times 2l)$ $(2lx1)$ $(2lx1)$

The elements of the submatrices are given by the following integrals

$$KP_{ij} = \frac{1}{Re} \int_{D(e)} \left[\frac{\partial N_i}{\partial r} \frac{\partial N_j}{\partial r} + \left(\frac{r_o}{d} \right)^2 \frac{\partial N_i}{\partial z} \frac{\partial N_i}{\partial z} \right] r \, dr \, dz \qquad (8a)$$

$$C_{ij} = \int_{D(e)} \left[N_i \frac{\partial N_j}{\partial r} u^* r + N_i \frac{\partial N_j}{\partial z} w^* r \right] dr \, dz \qquad (8b)$$

$$UNM_{ij} = \frac{1}{Re} \int_{D(e)} \frac{N_i N_j}{r} \, dr \, dz \qquad (8c)$$

$$A2_{ij} = -\int_{D(e)} N_i N_j \, v^* \, dr \, dz \qquad (8d)$$

$$A4_{ik} = \int_{D(e)} N_i \frac{\partial H_k}{\partial r} r \, dr \, dz \qquad (8e)$$

$$A3_{ik} = \left(\frac{r_o}{d}\right)^2 \int_{D(e)} N_i \frac{\partial H_k}{\partial z} r \, dr \, dz \qquad (8f)$$

$$A6_{kj} = \int_{D(e)} H_k \frac{\partial N_j}{\partial z} r \, dr \, dz \qquad (8g)$$

$$A5_{kj} = \int_{D(e)} \left[H_k \frac{\partial N_j}{\partial r} r + H_k N_j \right] dr \, dz \qquad 8(h)$$

$$R_i^1 = \int_{S(e)} N_i r \frac{\partial u}{\partial r} dz + \left(\frac{r_o}{d}\right)^2 \int_{S(e)} N_i r \frac{\partial u}{\partial z} dr \qquad 8(i)$$

$$R_i^2 = \int_{S(e)} N_i r \frac{\partial v}{\partial r} dz + \left(\frac{r_o}{d}\right)^2 \int_{S(e)} N_i r \frac{\partial v}{\partial z} dr \qquad 8(j)$$

$$R_i^3 = \int_{S(e)} N_i r \frac{\partial w}{\partial r} dz + \left(\frac{r_o}{d}\right)^2 \int_{S(e)} N_i r \frac{\partial w}{\partial z} dr \qquad 8(k)$$

To evaluate these integrals over triangular elements, the following formula is used :

$$\int_{A(e)} L_1^{\alpha} L_2^{\beta} L_3^{\gamma} \, dA^{(e)} = \frac{\alpha! \, \beta! \, \gamma!}{(\alpha + \beta + \gamma + 2)!} 2A \qquad (9)$$

Where 2A is the determinant of the Jacobian of transformation from the global coordinates to the area coordinates (Ref 13).

Since the integrals involved in UNM_{ij} do not fit the formula above, these integrals are evaluated numerically by Newton-Cotes formulas (Ref.13). The formula used in

this case is :

$$\int_{A(e)} f(L_1,L_2,L_3)dA^{(e)} = \frac{1}{60}\Big[27f(1/3,1/3,1/3) + 8(f(1/2,1/2,0) + f(1/2,0,1/2) + f(0,1/2,1/2)) + 3(f(1,0,0) + f(0,1,0) + f(0,0,1))\Big] \quad (10)$$

D. Solution Procedure

To solve the system of equations (Eqs 8) a FORTRAN program is developed. The program comprises the following steps.

1- System topology, code-numbers, necessary constants and boundary conditions are read as input.

2- The linear part of the element equations for all elements are formed.

3- The iteration number is set to one and all nodal values of the velocity field are initialized to zero.

4- Using the velocity field u*, v*, and w* the convective part of the element equations is formed.

5- The linear part is added to the convective part to form the element equations.

6- The system matrix is assembled.

7- The system matrix is modified by applying the boundary conditions.

8- The system of equations is solved by using library program LEQTIF.

9- New values for the velocity field is compared with the previous one. If the convergence criterion is not satisfied, the new values of the velocity field is replaced by the old ones and set to be new u*, v* and w* and iterative procedure continues from item 4 and repeats until the convergence criterion is satisfied

A description of the program is presented in Appendix C.

IV

RESULTS AND DISCUSSION

A - Verification of the Model

The Finite Element solution of the Navier-Stokes equations is obtained for laminar flow between two disks at low and intermediate Reynolds numbers. To test the validity of the results obtained by the present method, the results have been compared with the solutions available at certain limiting conditions.

One of the limiting cases is the low Reynolds number flow (Stokes' flow) where the inertial forces are negligible compared to the viscous forces. In the limit, the Reynolds number goes to zero and the motion of the fluid is governed by viscous forces alone. Since the convective terms are negligible, the equations become linear and the θ component of the momentum equation is uncoupled.

Stokes' flow solution of the problem of flow inside of a rotating cylinder (Problem b) is compared with the analytical solution, which indicates that the fluid assumes a solid body rotation, i.e. $u=w=0$, $v=wr$ throughout the domain. The computer code results are in a good agreement with this solution.

For the case of Stokes' flow between a rotating disk and a stationary shroud, the motion is imposed only by the rotating disk. The boundary conditions are given

in Equation (4). The computer results are shown in Fig.IV.1 (a) and Fig.IV.1 (b). For a constant z, tangential velocity of the fluid increases as r increases, due to the rotating disk but after a point it starts to decrease in order to satisfy the zero boundary condition on the shroud. As z increases (i.e. as the rotating disk is approached) the peak tangential velocity occurs nearer to the outer casing.

The present model is not applicable for the other limiting case of $Re \to \infty$, where the resulting flow is of boundary layer type and consequently the governing equations are parabolic. The additional fact that high Reynolds number flows between disks are generally turbulent, makes the present model unrealistic for such an application.

For comparison with Schlichting's solution of flow near an infinite rotating disk, a finite portion of the flow geometry lying near the rotating disk is considered. The conditions along the boundaries of the selected subregion are barrowed from Schlighting's(14) exact solution and are used in the present model to calculate the velocities in the subregion. The results are in good agreement.

The velocity field solution obtained by the model is shown in Figure IV.2 and Figure IV.3. In Figure IV.3, the results are compared with the exact solution.

In Figure IV.2 as well as in Figures IV.5-9,18, and 19, the flow field on θ plane is represented by nondimensional velocity vectors, which are the resultant of u' and w' (defined by Eq.3).

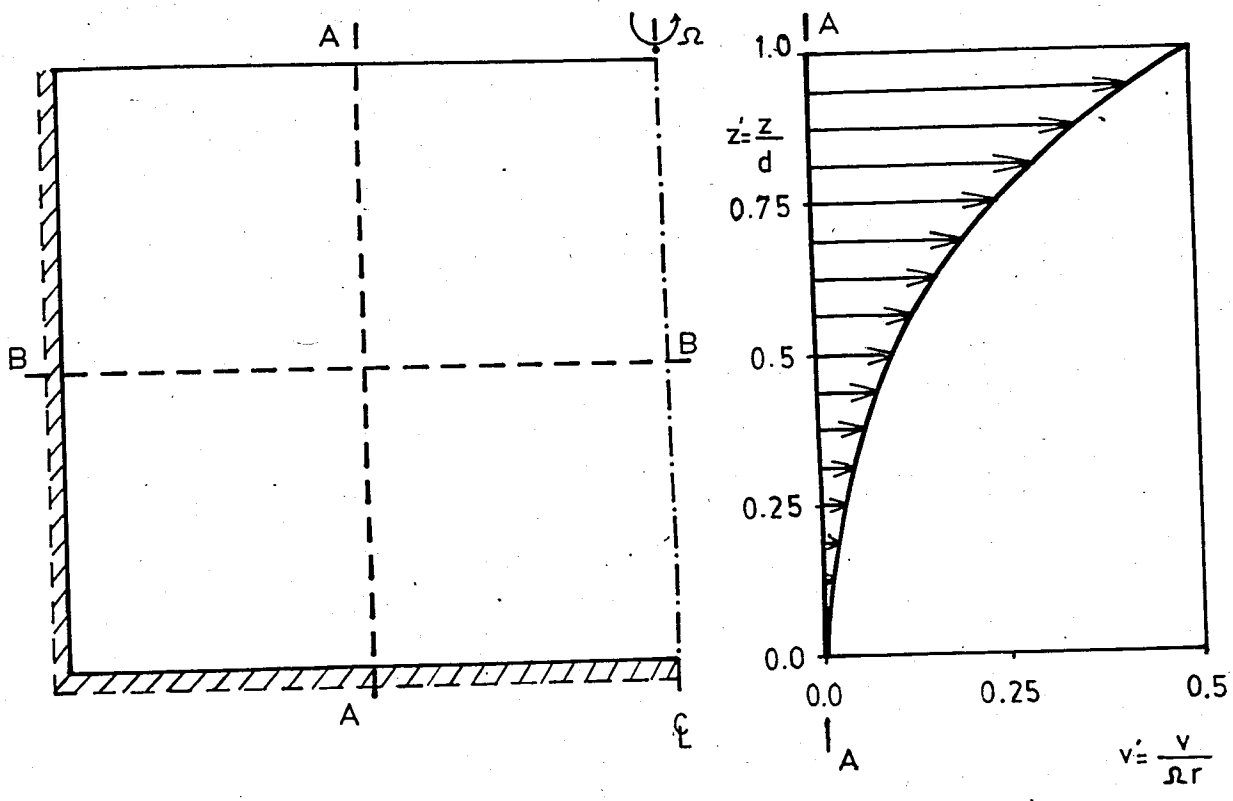


Figure IV.1 (a)- Tangential Velocity Profile at Section A-A for Stokes Flow of an Enclosed Rotating Disk

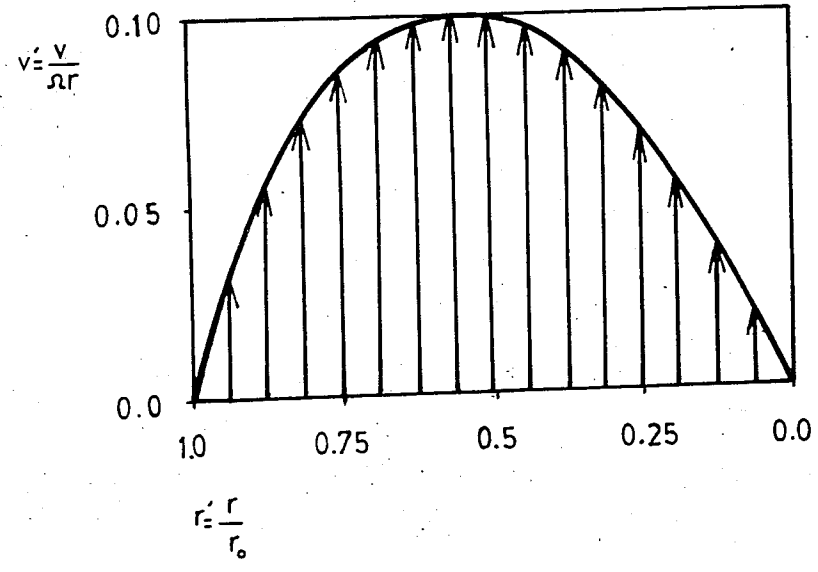


Figure IV.1 (b)- Tangential Velocity Profile at Section B-B for Stokes Flow of an Enclosed Rotating Disk

Nondimensional Velocity Scale = 1 cm = 0.1

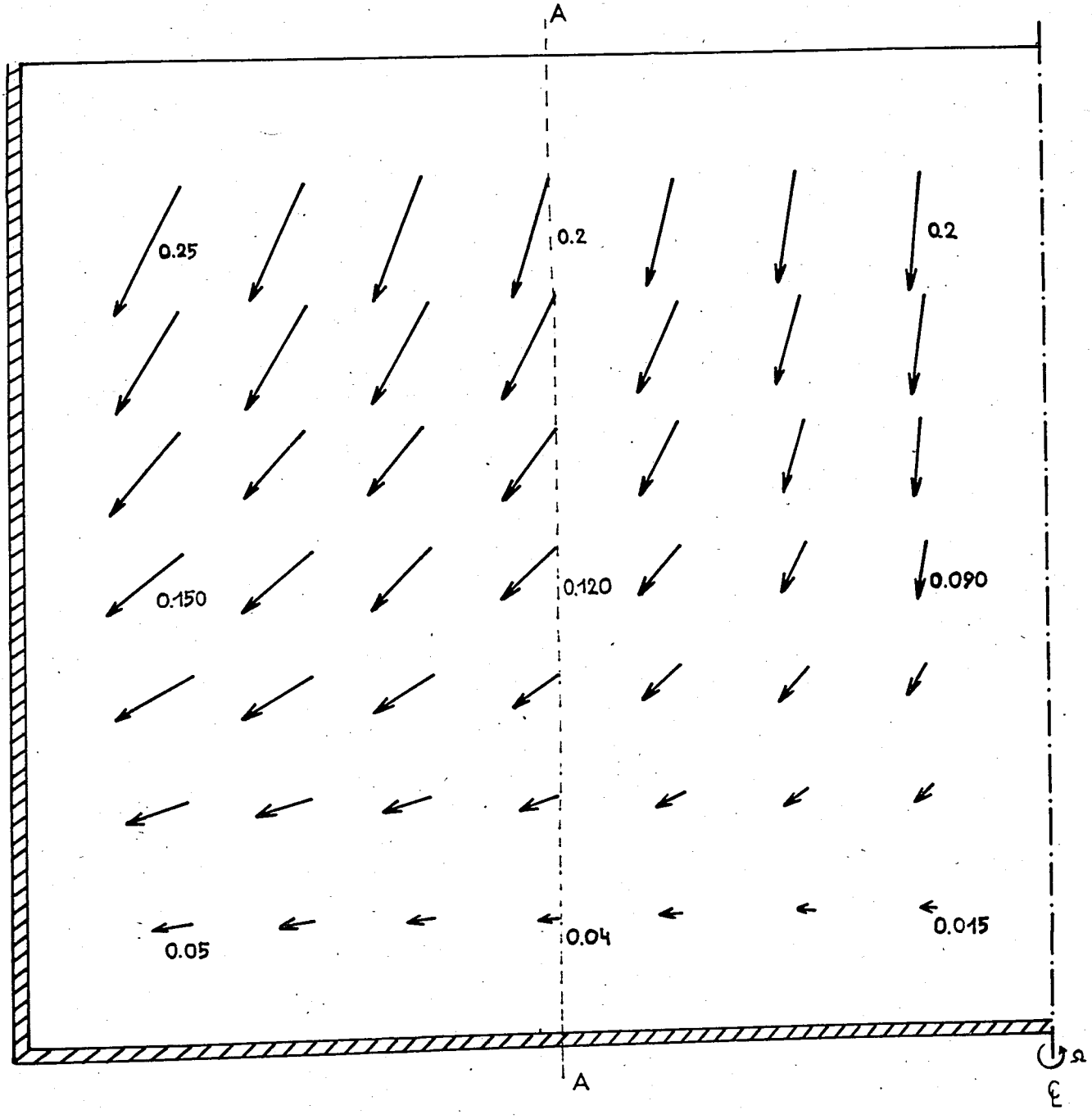


Figure IV.2 - Velocity Field for Flow Near a Rotating Infinite Disk (Re = 4, $r_o/d = 1$).

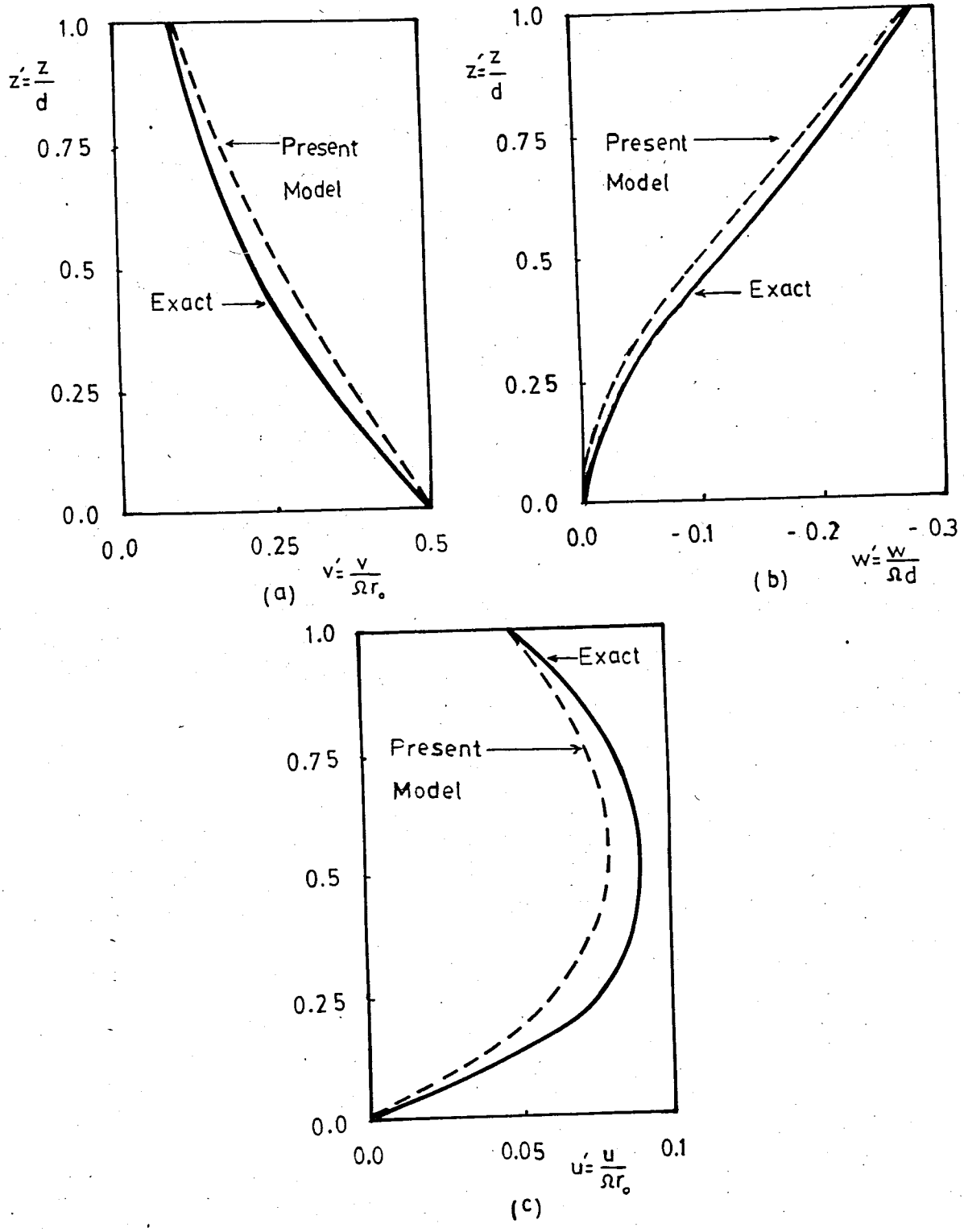


Figure IV.3- Velocity Profiles at Section A-A of Flow Near of Rotating Infinite Disk at Re = 4, (o) Tangential, (b) Axial, (c) Radial

Fig IV.4 (a,b) shows the effect of the number of elements used for the Stokes' flow inside a rotating cylinder. In Fig. IV (a) tangential velocity at $r/r_o=0.5$ is plotted against the number of elements used. The variation of percentage error between the analytical solution and the numerical solution is shown in Figure IV.4 (b). As the number of elements is increased, percentage error decreases.

B - Flow Between a Rotating Disk and a Stationary Shroud

The effect of the Reynolds number on the secondary flows(flow in r-z plane) is shown in Figs IV 5-9 for the flow between a rotating disk and a stationary shroud. It is noted that the intensity of secondary flows increase with increasing Reynolds number.

Tangential, radial and axial velocity profiles for various Reynolds numbers at different cross sections are shown in Figures IV.10, 11, 12 respectively. In Figure IV.10 it is seen that the tangential velocity profile becomes steeper as Reynolds number increases. At the same time, radial velocity profiles develop in magnitudes and gradients with increasing Reynolds number as shown in Figure IV.11.

With this model and storage capacity it becomes difficult to predict large velocity gradients and to achieve convergence. Since velocity profiles have larger gradients at higher Reynolds numbers, it is a more realistic approach to make the boundary layer approximations.

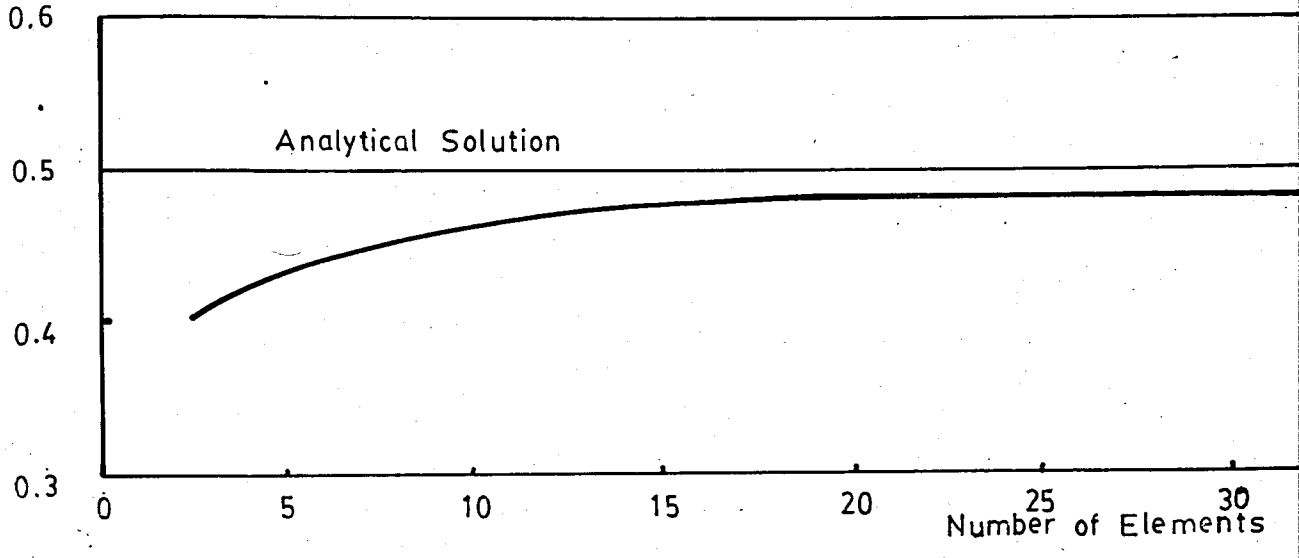


Figure IV. 4 (a)- Variation of Tangential Velocity of r/r_o = 0.5 with Number of Elements used, for Stokes' Flow of a Rotating Circular Cylinder

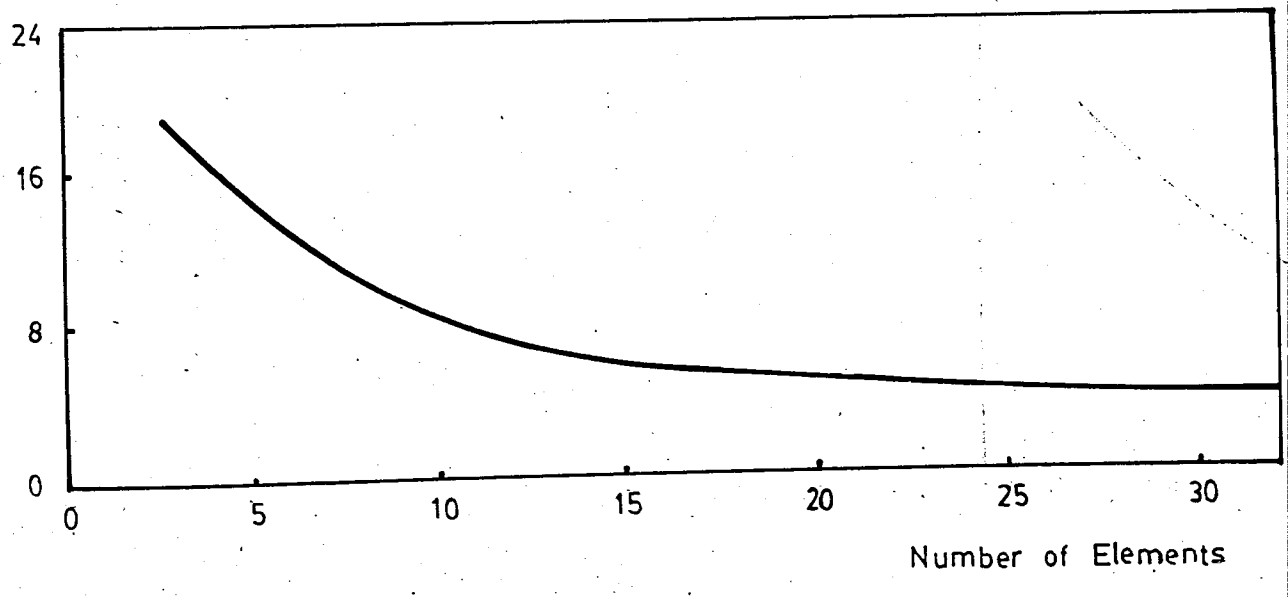


Figure IV.4 (b)- Variation of Percent Error in Tangential Velocity at r/r_o = 0.5 with Number of Elements used for Stokes' Flow of a Rotating circular Cylinder

Nondimensional Velocity Scale : 1 cm = 0.05

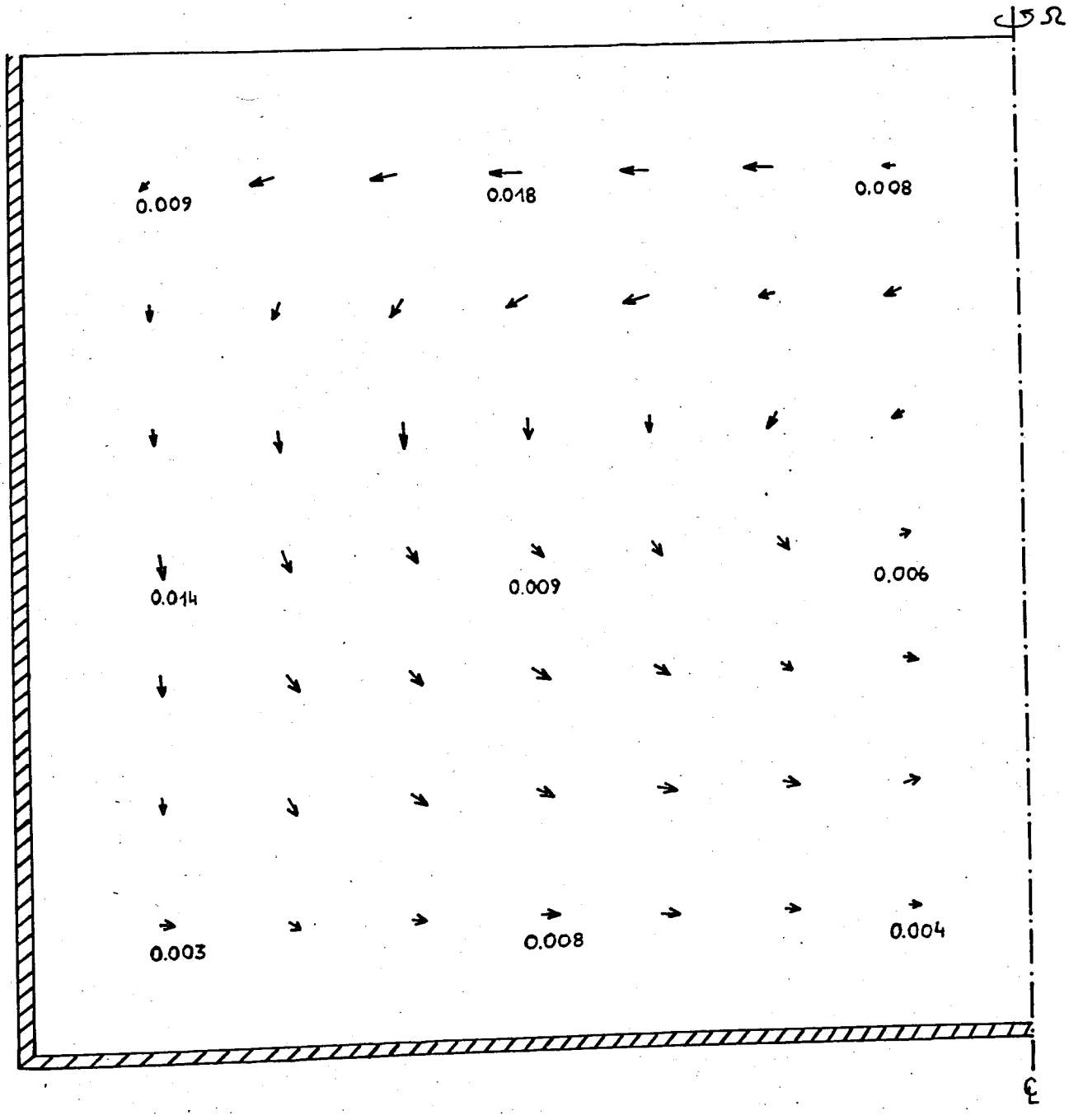


Figure IV.5- Velocity Field of Flow at Re = 10 for an Enclosed Rotating Disk ($r_o/d = 1$)

Nondimensional Velocity Scale : 1 cm = 0.05

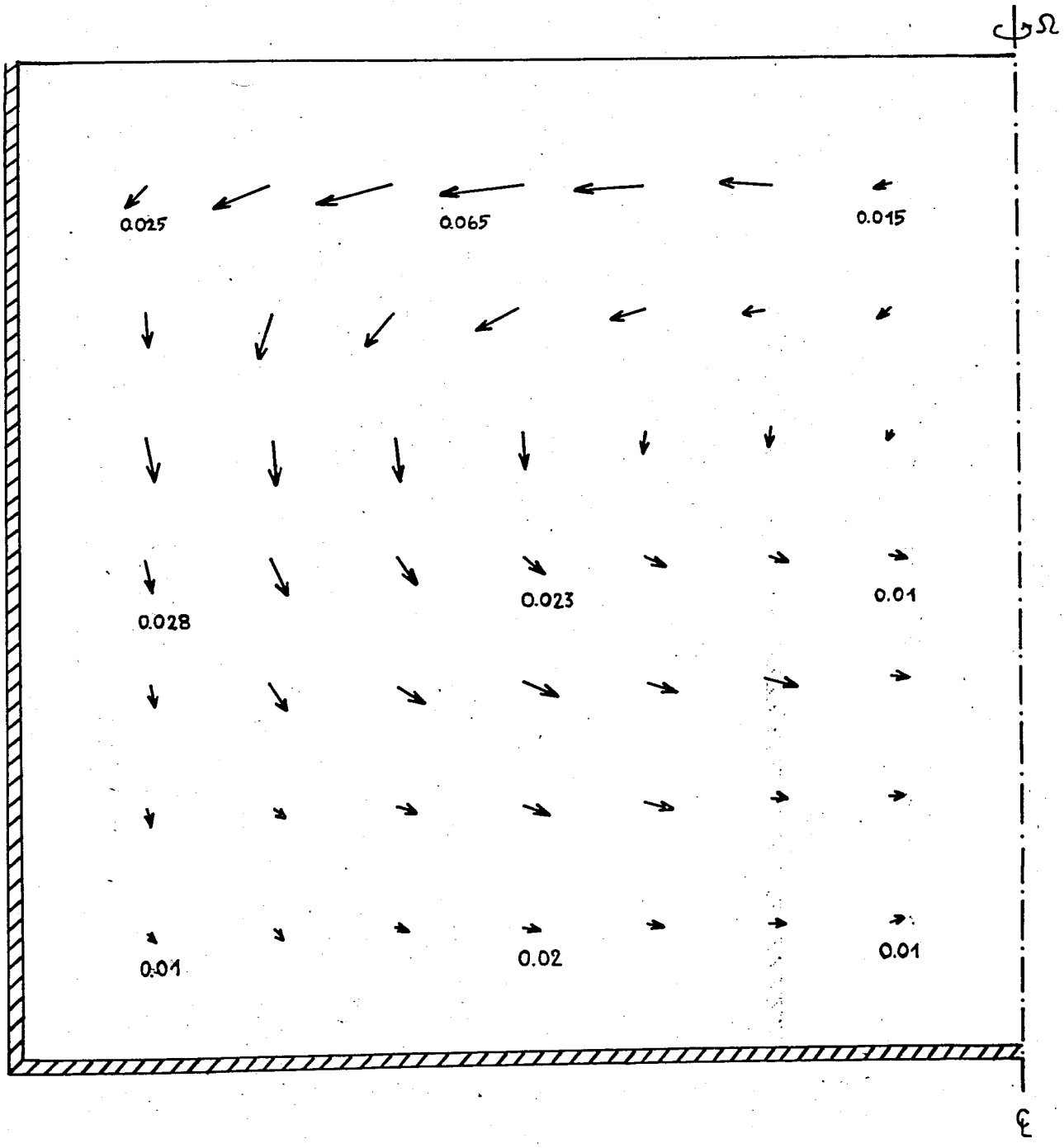


Figure IV.6- Velocity Field of Flow at Re = 40 for an Enclosed Rotating Disk ($r_o/d = 1$)

Nondimensional Velocity Scale : 1 cm = 0.05

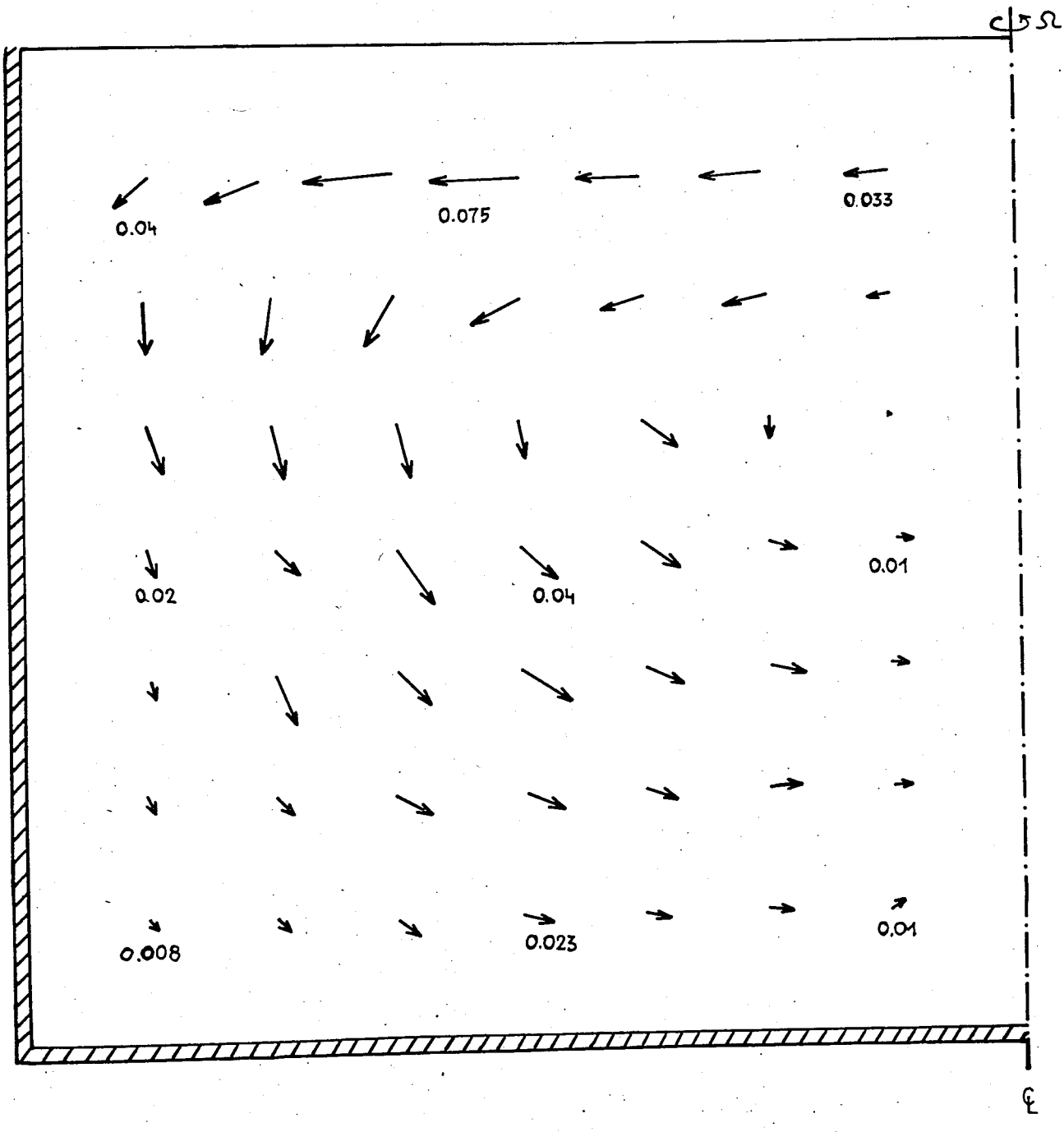


Figure IV.7- Velocity Field of Flow at Re = 60 for an Enclosed Rotating Disk ($r_o/d = 1$)

Nondimensional Velocity Scale : 1 cm = 0.05

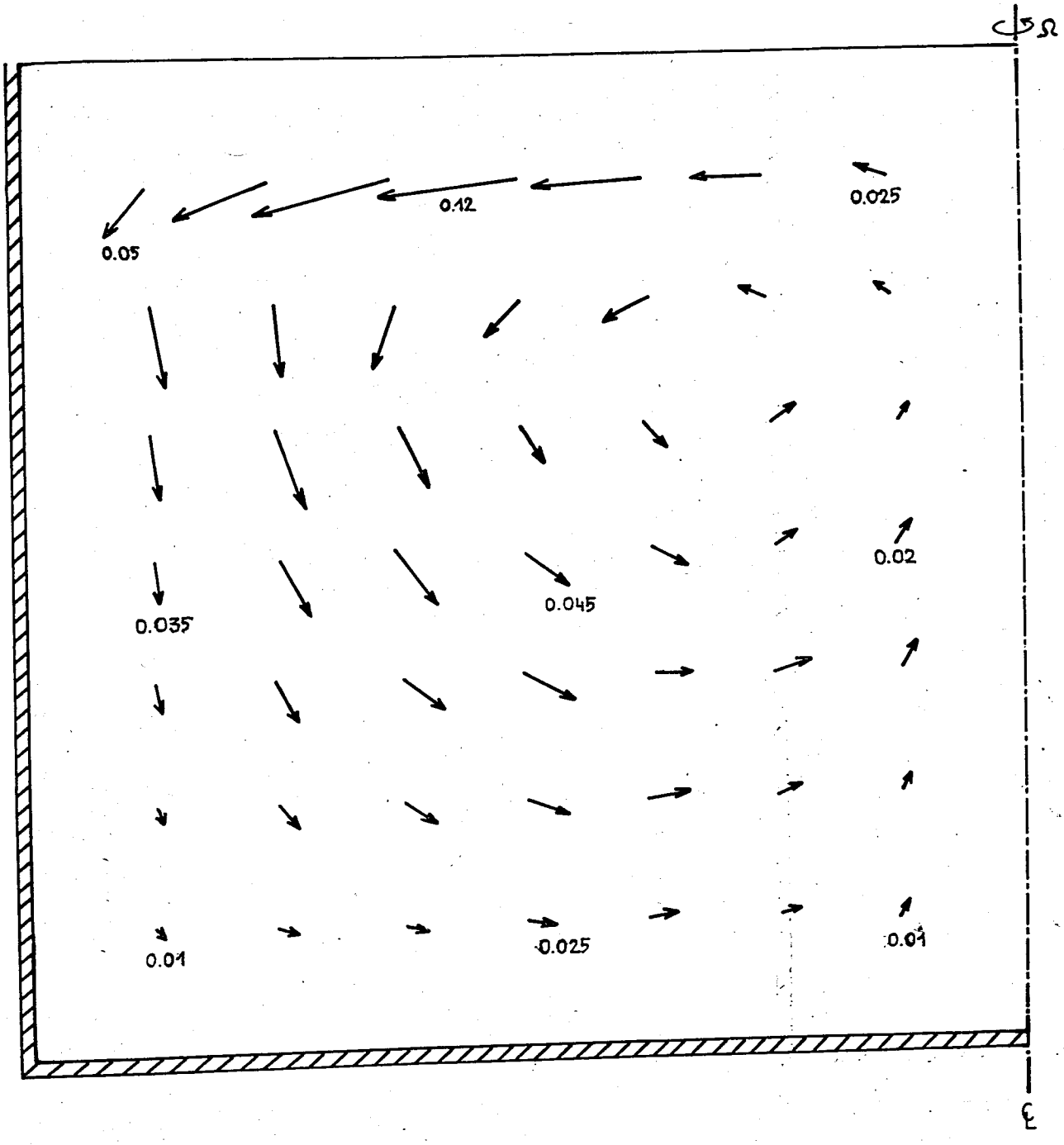


Figure IV.8- Velocity Field of Flow at Re = 100 for an Enclosed Rotating Disk ($r_o/d = 1$)

Nondimensional Velocity Scale : 1 cm = 0.05

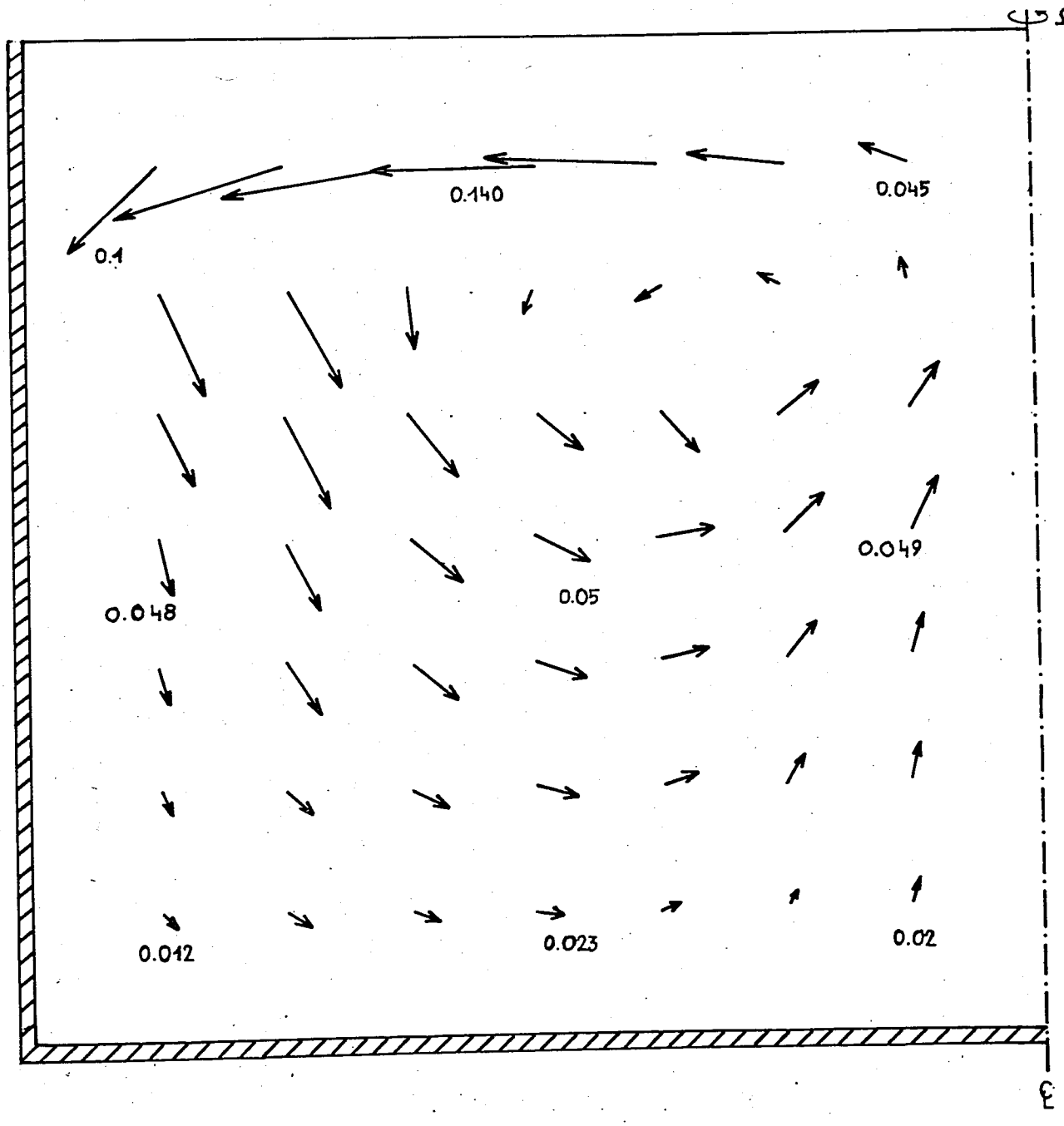


Figure IV.9- Velocity Field of Flow at Re = 200 for an Enclosed Rotating Disk ($r_o/d = 1$)

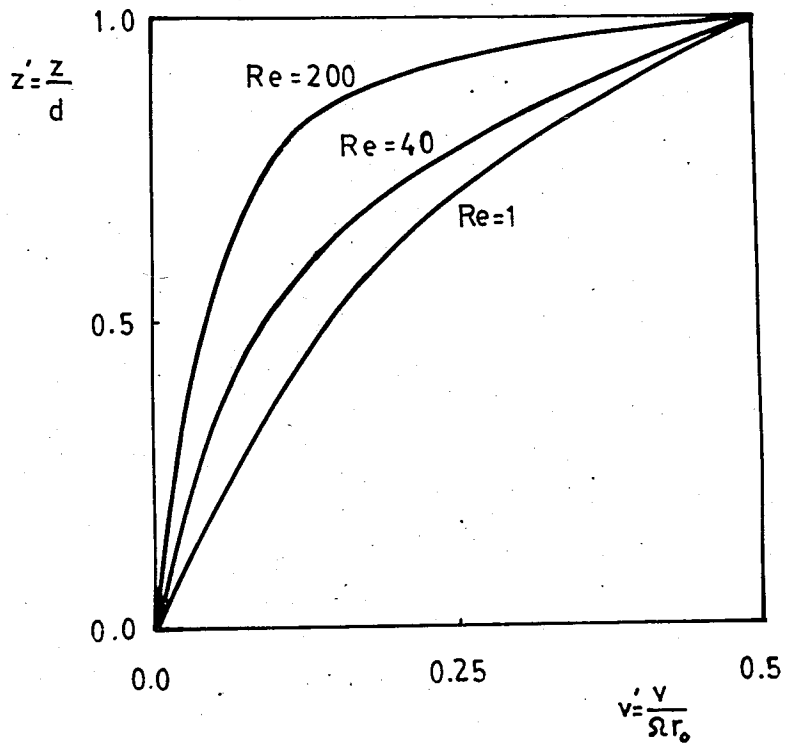


Figure IV.10- Tangential Velocity Profiles at Mid-Section For Re = 1, 40,200

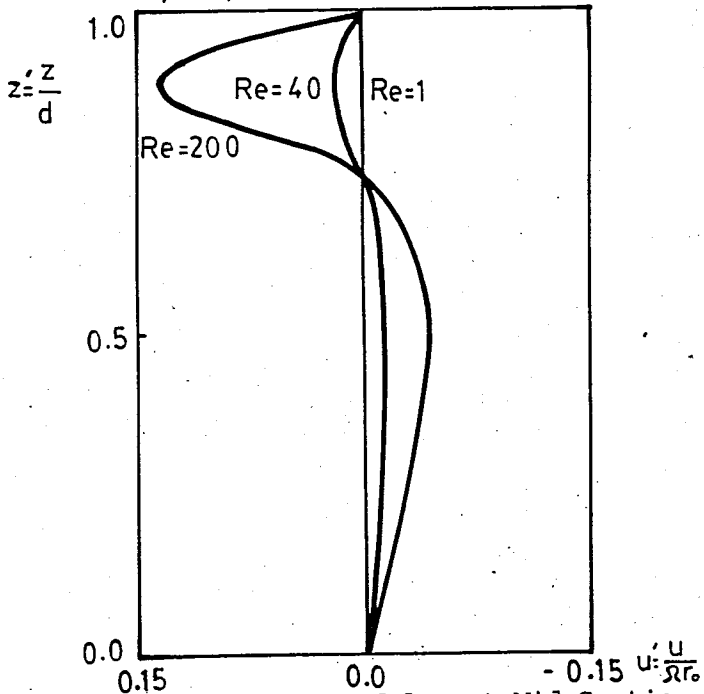


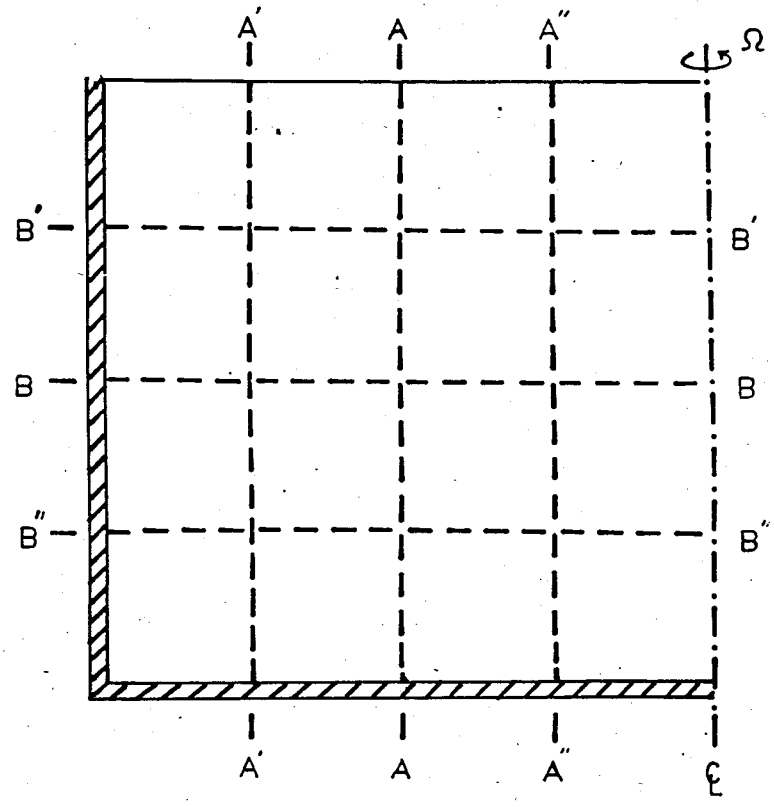
Figure IV.11- Radial Velocity Profiles at Mid-Section for Re = 1, 40, 200

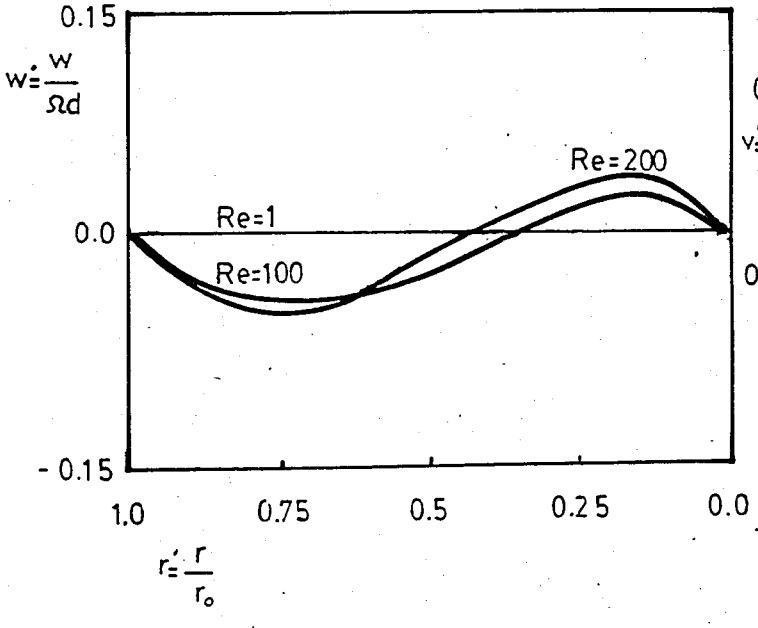
Axial and tangential velocity profiles for various Reynolds numbers are shown in Figure IV.12. Increase of the axial velocity with increasing Reynolds number can be observed from Figure IV.12 (a). Furthermore it appears that the secondary flows start to develop around Re = 10. Tangential velocity profiles at section B-B is shown in Fig. IV.12 (b). They preserve similar profiles but decrease in magnitude with increasing Reynolds number, as in Figures IV.1 (b) and 10.

Radial, axial and tangential velocity profiles at several cross sections are shown in Figure IV.13. It is noted that radial velocities and gradients increase as one moves away from the axis of rotation. Axial and tangential velocity profiles are also similar but increase in magnitude towards to the rotating disk. The peak tangential velocity occurs nearer to the outer casing. Pressure distributions on the rotating disk and the stationary shroud are shown in Figure IV.14.

Due to the lack of appropriate experimental data present results are compared with the experimental results obtained by BAYLEY, F.U., MORRIS, W.P., OWEN, J.M., and TURNER, A.B. (9) at very high Reynolds numbers, where the flow is turbulent (Fig IV-15). Tangential velocity profiles obtained experimentally for $Re = 1.7 \times 10^6$ and $Re = 3.4 \times 10^6$ are shown together with the profiles calculated by the present model for Re = 100 and 200. The profiles obtained both experimentally and numerically show similar trends, and they also show an agreement as far as the magnitudes are considered. They get steeper as Reynolds number increases as stated earlier in this chapter. This comparison shows that the tangential velocity profiles obtained by the present model are in agreement with the experimental results obtained for the turbulent case.

In Figure IV.16 the streamlines at Re = 100 are shown for the flow between a rotating disk and a stationary shroud. As it is seen from the figure, the flow shows a single-cell structure. Figure IV.17 gives a qualitative illustration of flow between a rotating disk and a stationary shroud.





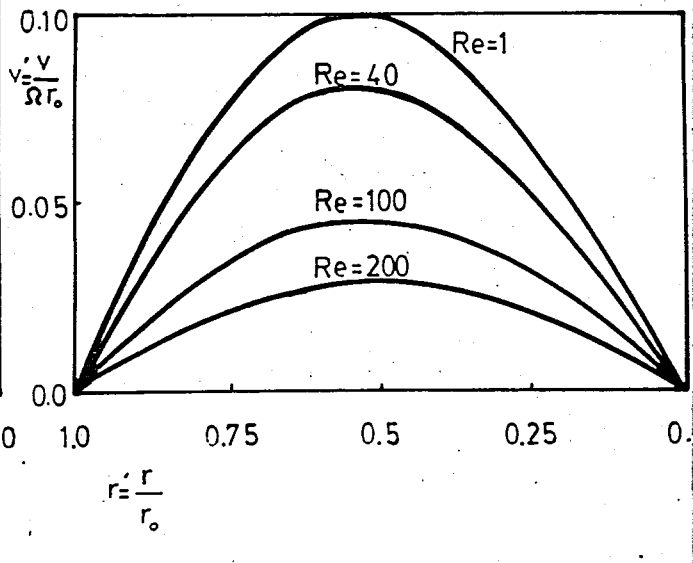


Figure IV.12- Velocity Profiles at Section B-B for Re = 1, 40, 100, 200 (a) Axial, (b) Tangential

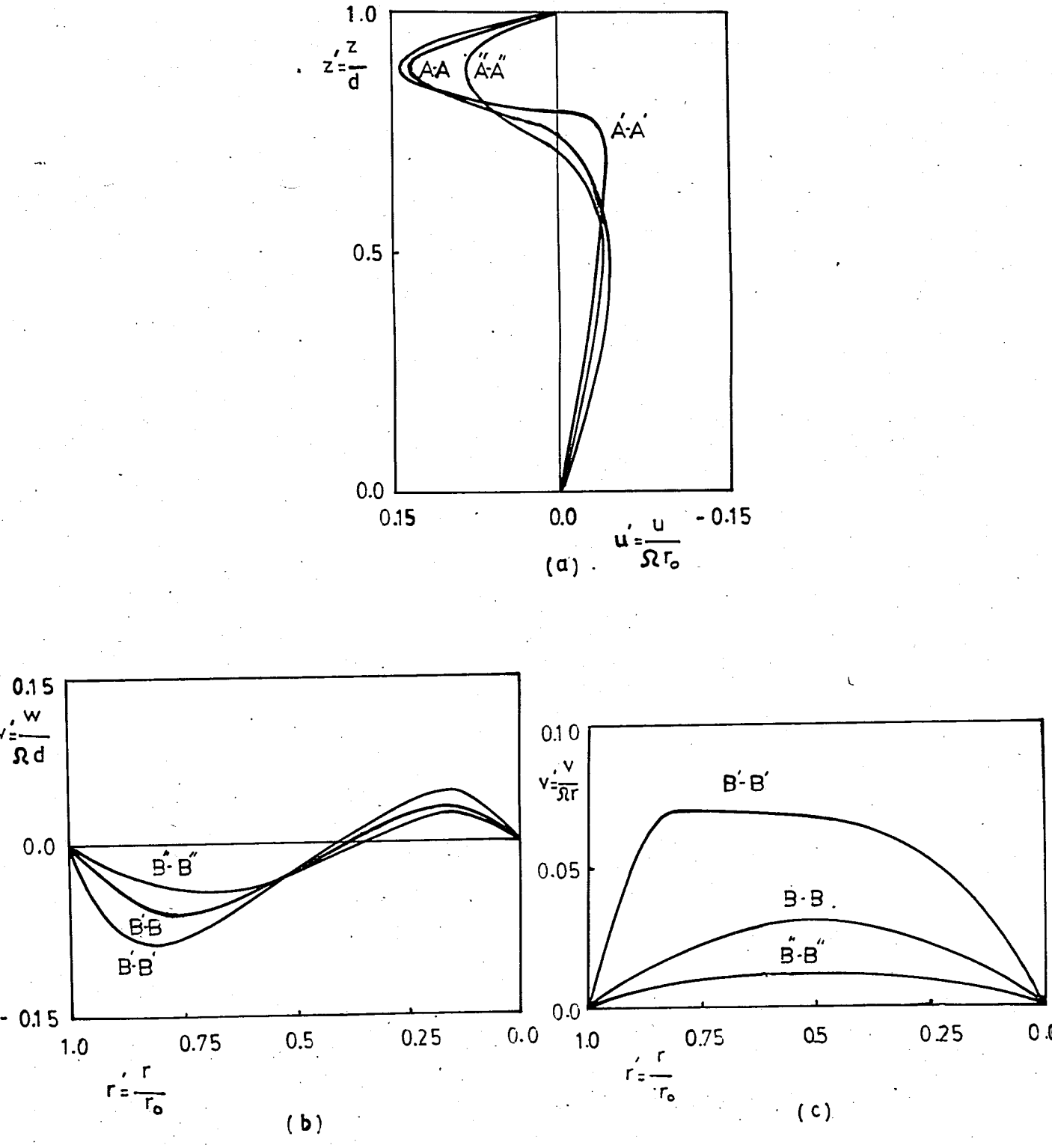


Figure IV.13- Velocity Profiles at Various Sections for Re = 200 (a) Radial, (b) Axial, (c) Tangential

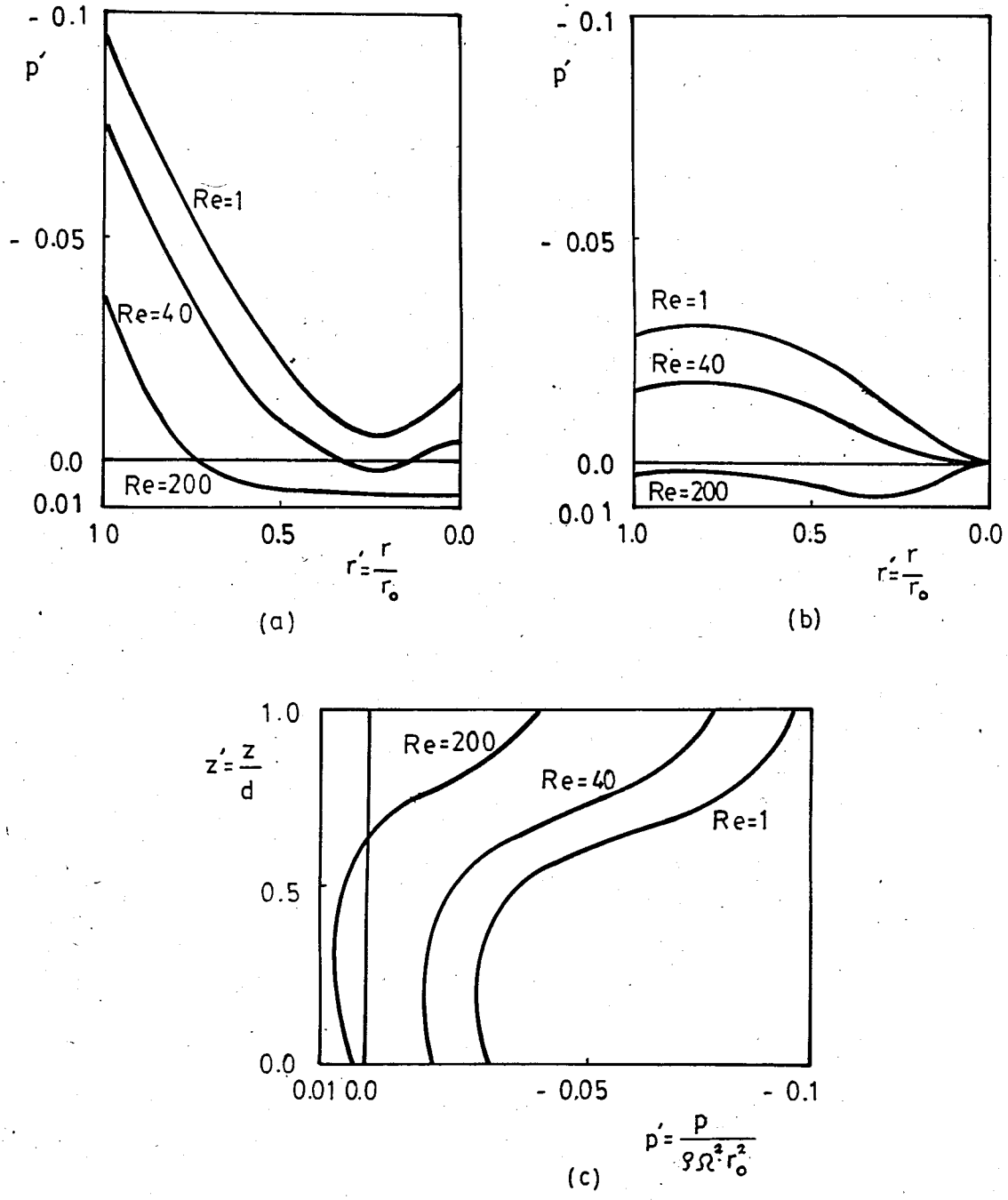


Figure IV.14- Pressure Distributions on the Rotating Disk and the Stationary Shroud for Re = 1, 40, 200 (a) Radial Distribution on the Rotating Disk, (b) Radial Distribution on the Shroud, (c) Axial Distribution on the Shroud

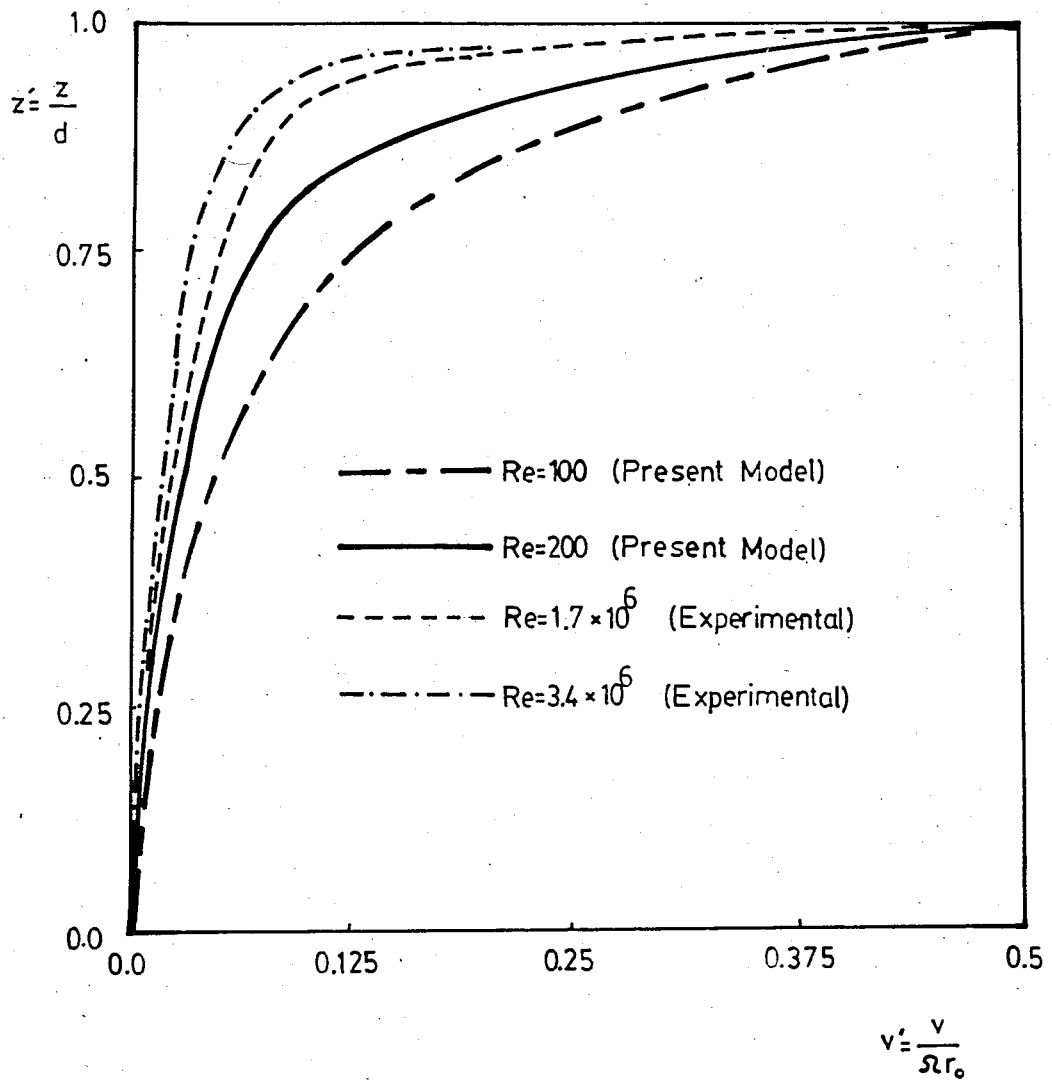


Figure IV.15- Tangential Velocity Profiles at Re = 100, 200 and Comparison With Some Experimental Results

Experiments are done for Re = 1.7010^6 , $3.4x10^6$ by BAYLEY, MORRIS, OWEN, TURNER (9).

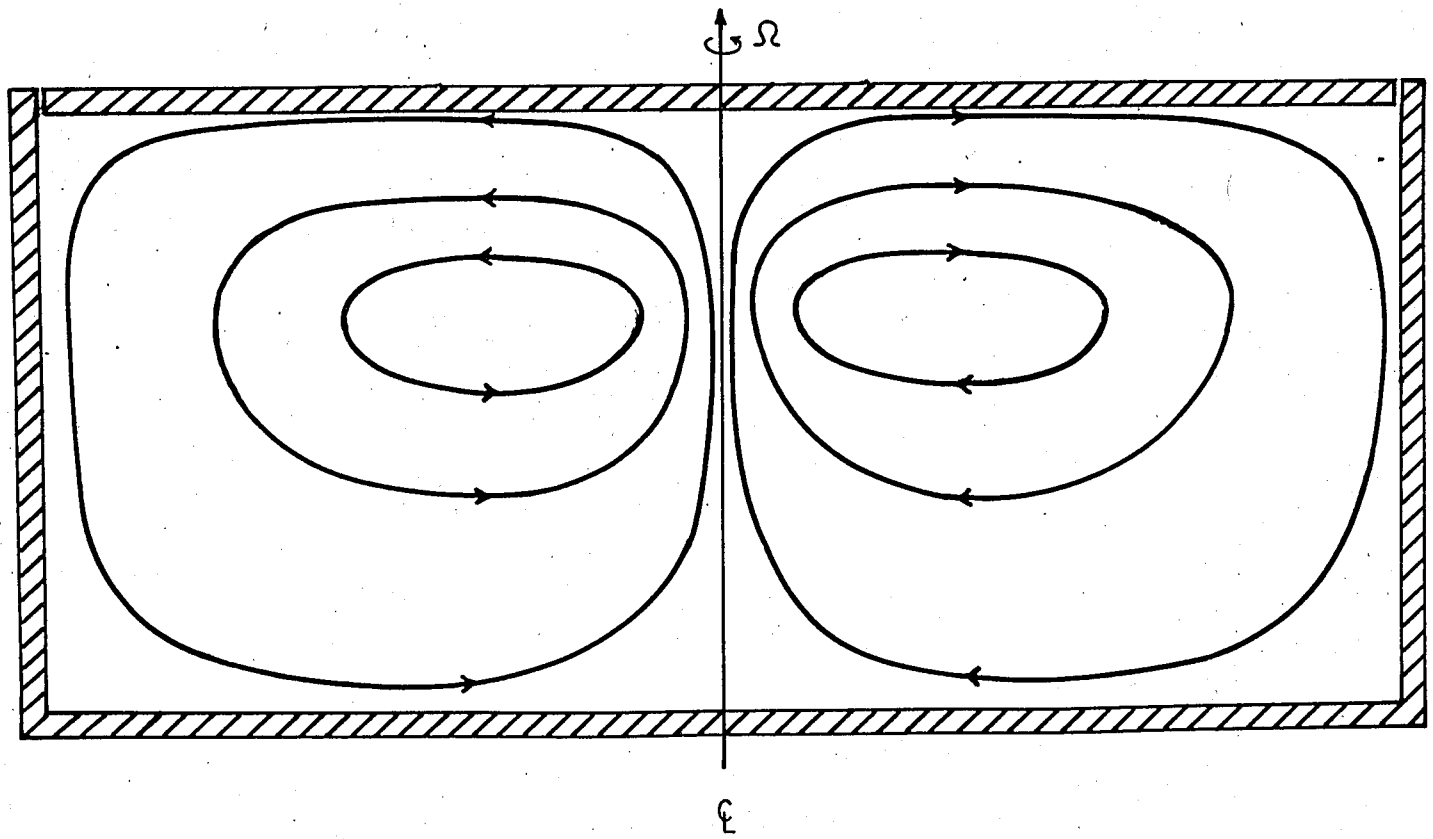


Figure IV.16- Streamlines of Flow at Re = 100 for an Enclosed Rotating Disk

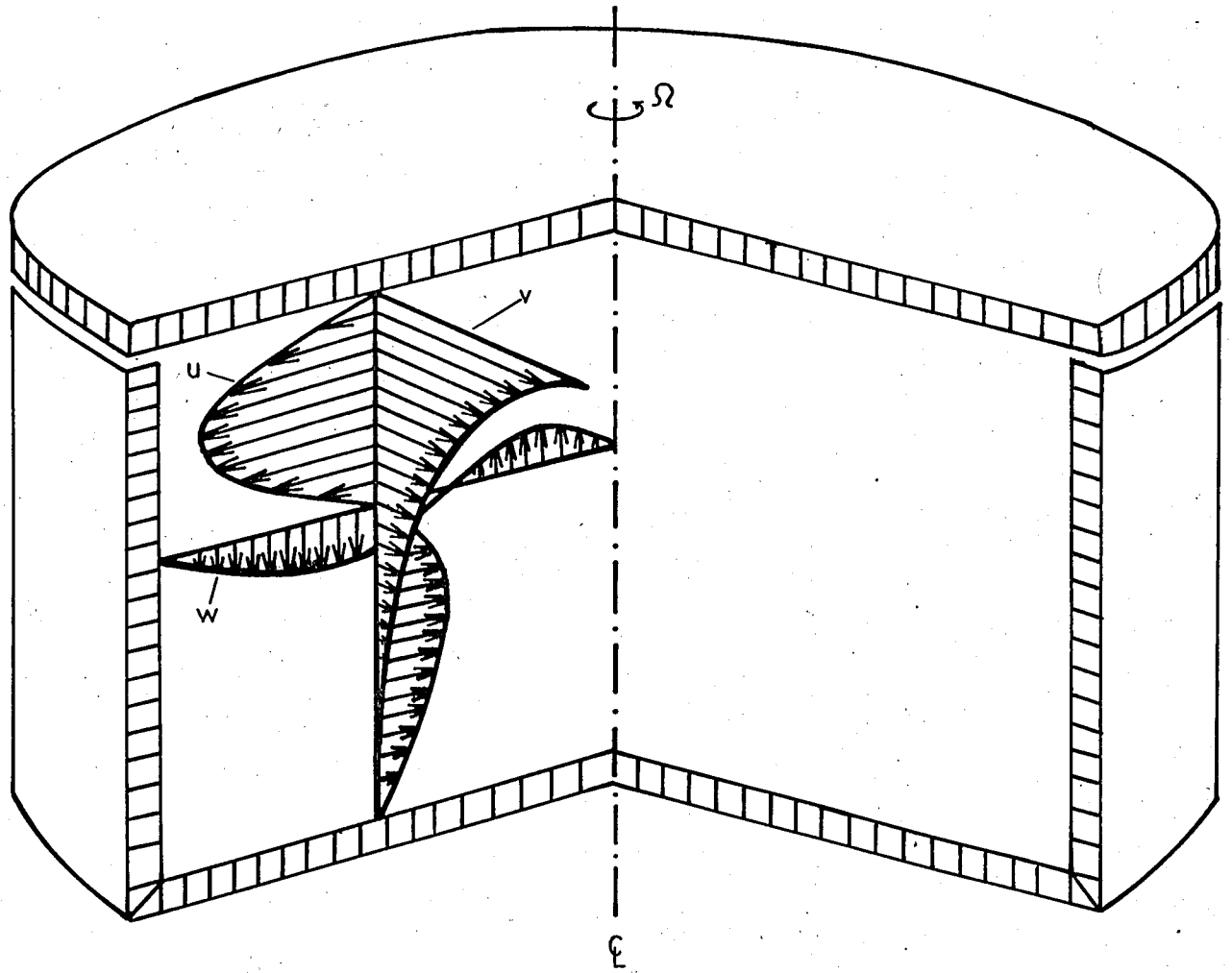


Figure IV.17- Illustration of Flow Between a Rotating Disk and a Stationary Shroud

Figures IV.18 and 19 show the effect of the parameter r_o/d. In Figure IV.18, the velocity field at Re = 60 and $r_o/d = 1$ is shown. Figure IV.19, shows the velocity field at Re = 60 and $r_o/d = 4$. As observed from the figures, the axial velocity components are negligible throughout the domain, and the resulting flow field can be represented by a two layer flow model for large values of r_o/d.

As an example for low r_o/d, the case of $r_o/d = 0.25$ is investigated, and it is observed that, since the disks are far apart, the effect of the rotating disk is not felt. Secondary flows are confined to the region near the rotating disk. It may be deduced that the prominence of secondary flows will diminish as r_o/d is reduced. To further investigate this point, a portion lying close to the rotating disk should be considered using a finer mesh structure.

Tangential and radial velocity profiles are shown for $r_o/d = 1$ and $r_o/d = 4$ at Re = 60 in Figure IV.20. As r_o/d increases the tangential velocity profile becomes flatter as shown in Figure IV.20 (a). The corresponding radial velocity profiles are shown in Figure IV.20 (b). They decrease in magnitude, and become symmetric about the axis $z' = 0.5$ as r_o/d increases.

In Figure IV.21 a comparison between the solution given by the present model and the numerical solution obtained by JAWA, M.S.(15) is presented. Jawa's numerical solution is obtained for flow between two infinite rotating disks. To approximate infinite disks with the present model, the solution for $r_o/d = 4$ is considered where Re = 80. Since the Reynolds number used by Jawa is based on the disk

Nondimensional Velocity Scale : 1 cm = 1

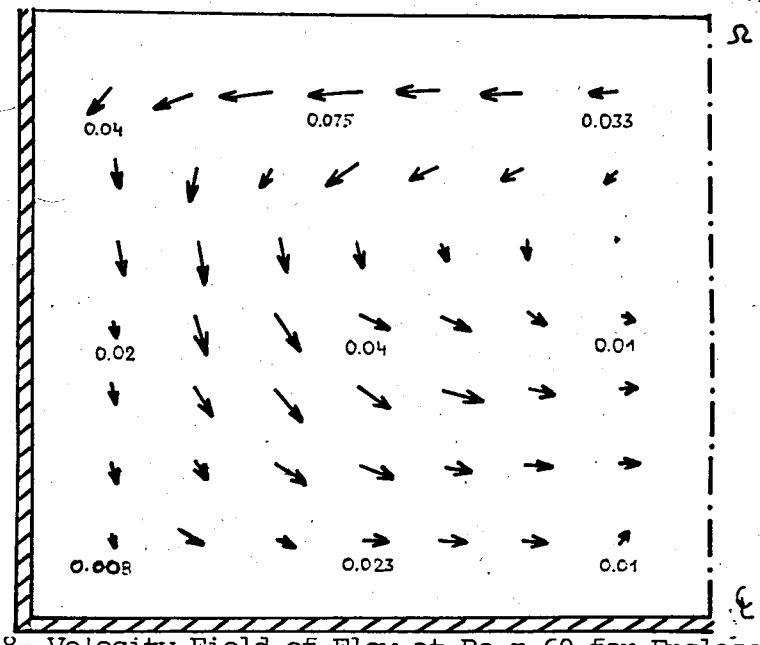


Figure IV.18- Velocity Field of Flow at Re = 60 for Enclosed Rotating Disk ($r_o/d = 1$)

Nondimensional Velocity Scale : 1 cm = 0.05

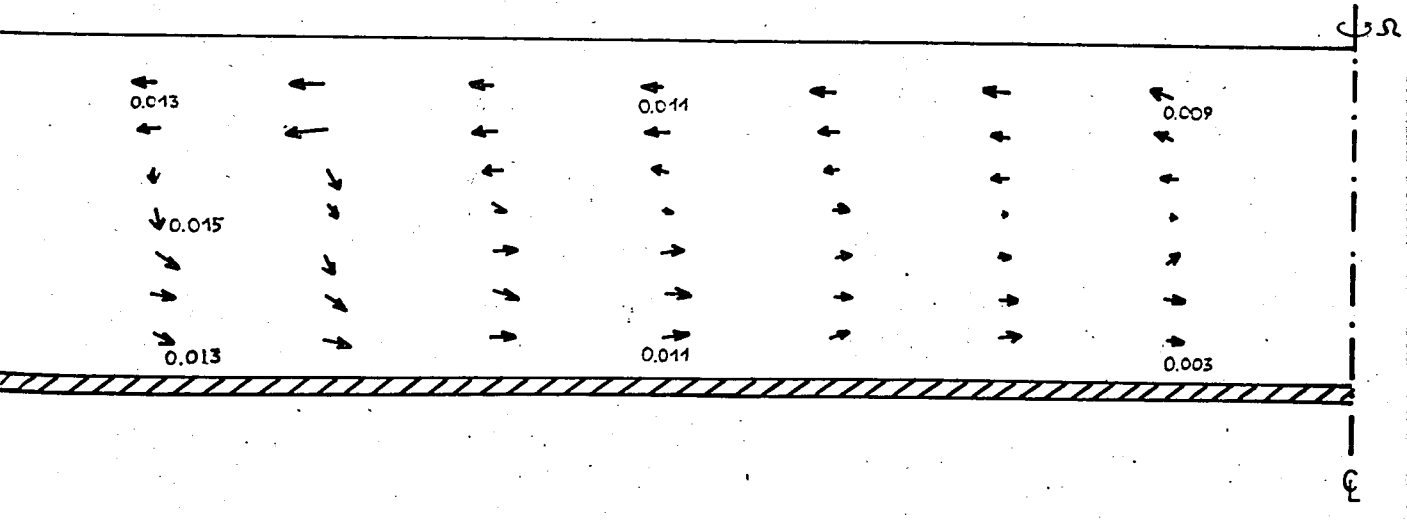


Figure IV.19- Velocity Field of Flow at Re = 60 for Enclosed Rotating Disk ($r_o/d = 4$)

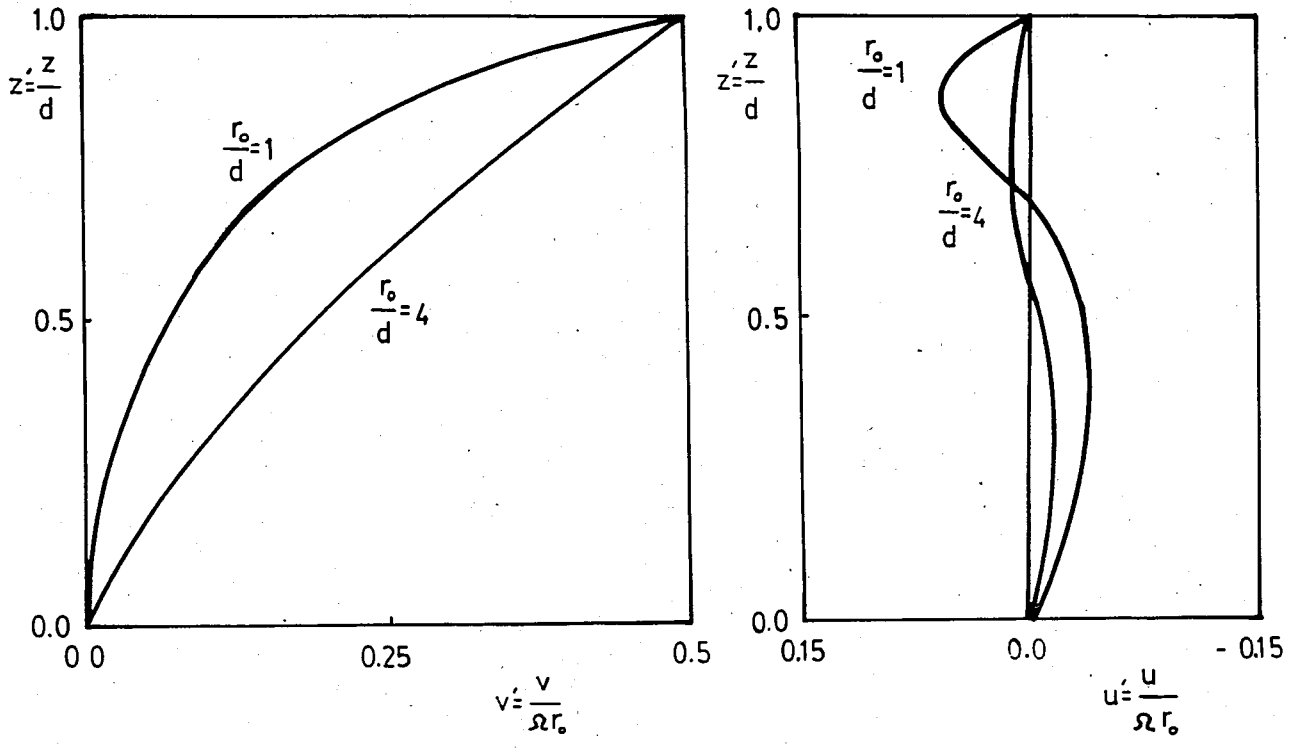


Figure IV.20- Velocity Profiles at Mid-Section for $r_o/d = 1$ and $r_o/d = 4$ at Re = 60; (a) Tangential, (b) Radial

spacing, Re = 80 which is based on the disk radius is converted and found to be equivalent to Re_d = 5. The radial velocity profile for r_o/d = 4 and Re_d = 5 at mid-section (to avoid the end effects) is plotted, together with the solutions obtained by Jawa(15) for Re_d = 5 in Figure IV.21. In Figure IV.22, the tangential velocity profile, presented by the model, for Re_d = 5 and r_o/d = 4 at mid section is compared with Jawa's(15) profiles obtained for Re_d = 5. The trends are similar.

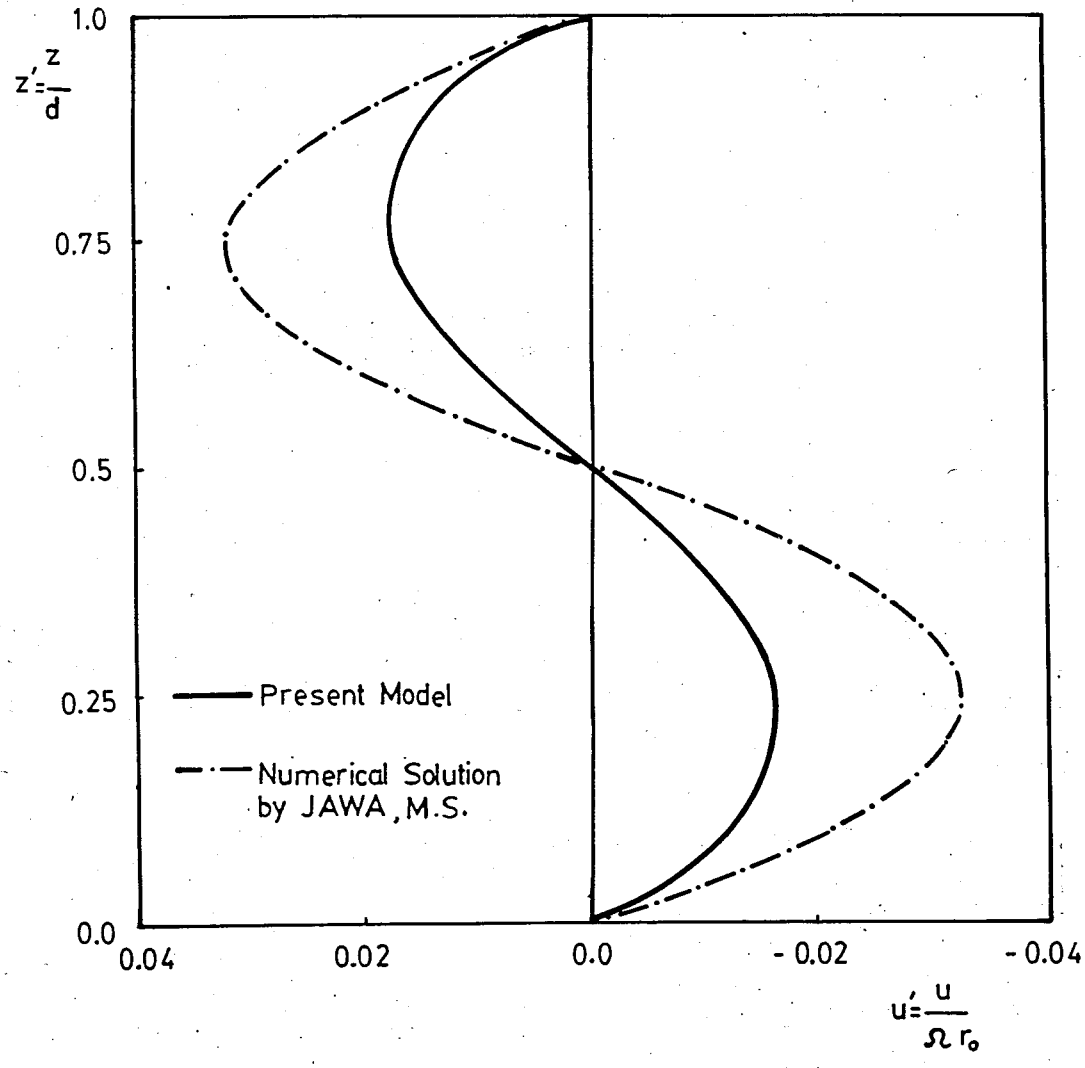


Figure IV.21- Radial Velocity Profile at r = $r_o/2$ for Re_d = 5 and r_o/d = 4 and Comparison with Jawa's(15) Numerical Solution

Jawa's solution is obtained for two infinite coaxial rotating disks at Re_d = 5.

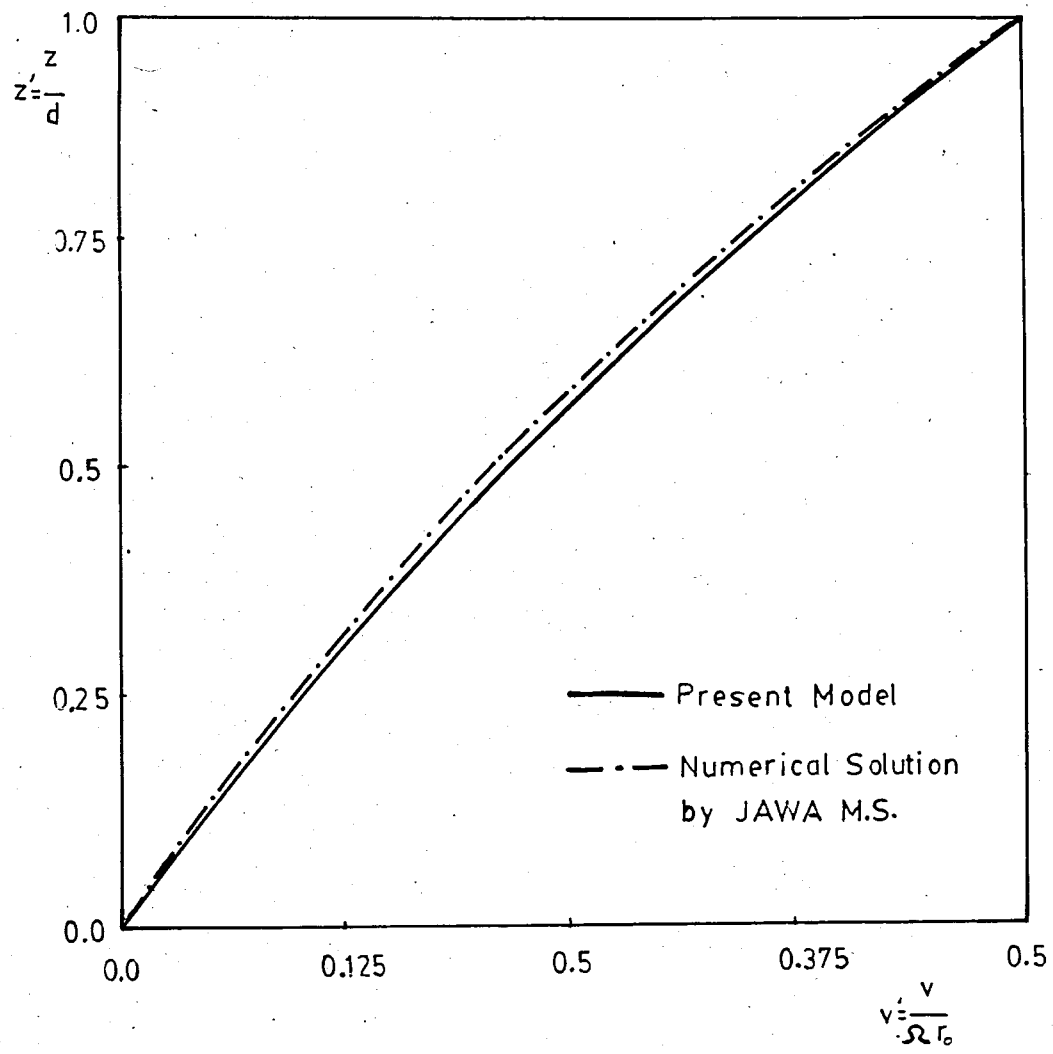


Figure IV.22- Tangential Velocity Profile at $r = r_o/2$ for $Re_d = 5$ and $r_o/d = 4$ and Comparison with Jawa's(15) Numerical solution

Jawa's solution is obtained for two infinite coaxial rotating disks at $Re_d = 5$.

C - Flow Inside a Circular Cylinder, Rotating about its Axis of Symmetry

Fig IV.23 shows the calculated velocity field at Re = 100 for flow inside a cylinder, rotating about its axis of symmetry. For the case of Stokes' Flow the fluid assumes a solid body rotation, i.e. tangential velocity is proportional to the radius. As the Reynolds number is increased, tangential velocity starts to change in the axial direction as well. It is this variation that causes secondary flows. At low Reynolds numbers, the axial variation of the tangential velocity is not significant enough to cause appreciable secondary flows. Secondary flows start to become visible at relatively high Reynolds numbers and are considerably delayed in this case as compared to the flow between a rotating disk and a stationary shroud, due to the different characters of their tangential velocity profiles. On the other hand, it became difficult to achieve convergence for Reynolds numbers, higher than 100, with the present model. Therefore the flow at Re = 100 is investigated.

Tangential velocity profile at mid-section is shown in Figure IV.24. The profile is parabolic and symmetric about the axis z' = 0.5. The radial velocity profile, shown in Figure IV.25 is also symmetric about the axis. Two different radial velocity profiles at two different sections are shown in Figure IV.25. The profiles are similar but the one nearer the outer casing has lower velocities due to the no slip condition on the shroud. Axial and tangential velocity profiles at various sections are presented in Figure IV.26. Axial and tangential velocity profiles reach their minimum values along the

axis z' = 0.5 and are symmetric about this axis. The pressure distribution over the disk and the shroud is shown in Figure IV.27. The streamlines of the flow are shown in Figure IV.28. Due to symmetry about the axis z' = 0.5, a double-cell structure is observed. A qualitative sketch of the velocity profiles is shown in Figure IV.29.

The case of flow between a rotating disk and a rotating shroud is also investigated where the shroud rotates with the half velocity of the disk. In this case, the flow charecteristics are found to be similar to that of the flow between a rotating disk and a stationary shroud.

Nondimensional Velocity Scale : 1 cm = 0.05

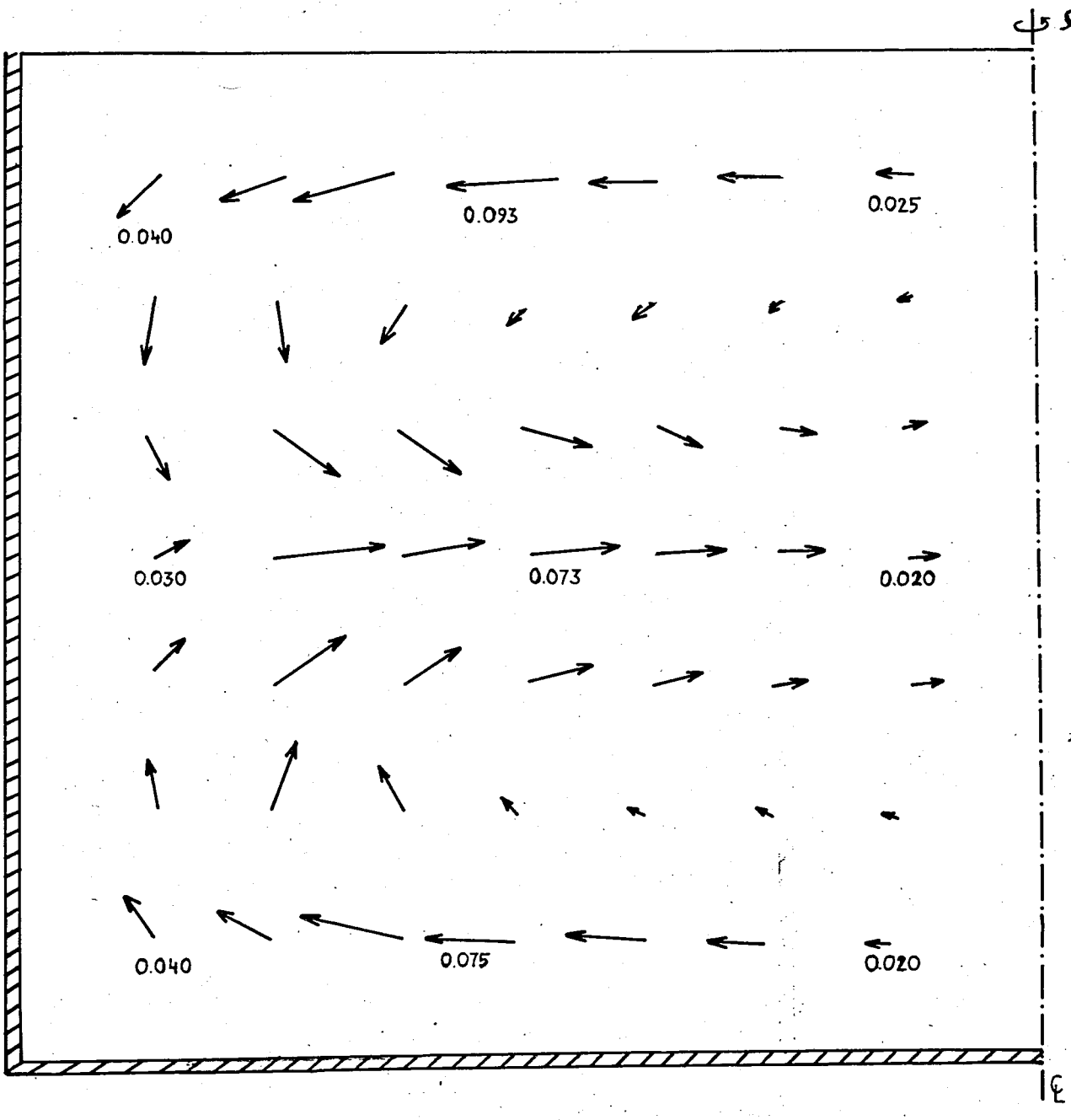


Figure IV.23- Velocity Field of Flow at Re = 100 for a Rotating Circular Cylinder ($r_o/d = 1$)

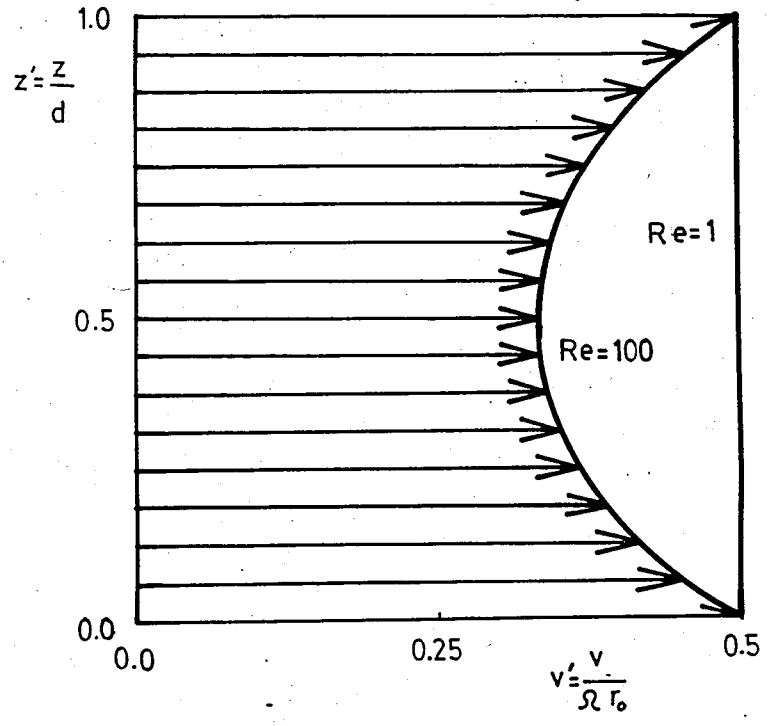


Figure IV.24- Tangential Velocity Profile at Mid-Section for Re ≙ 1,100

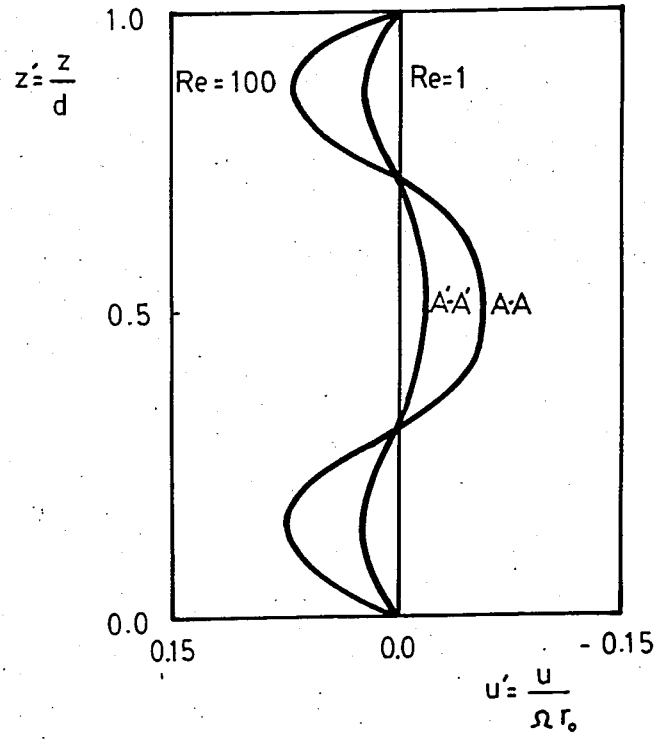
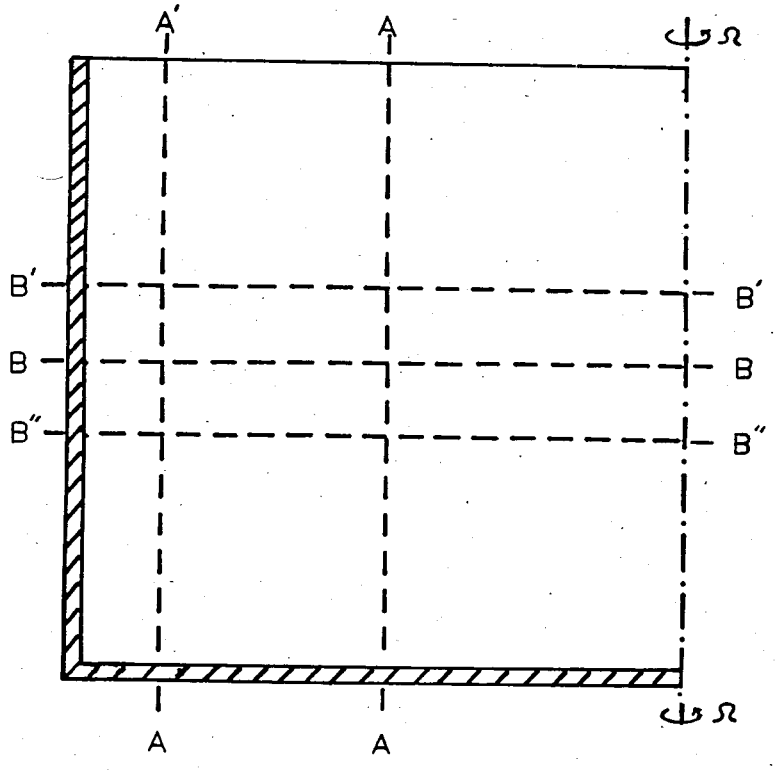
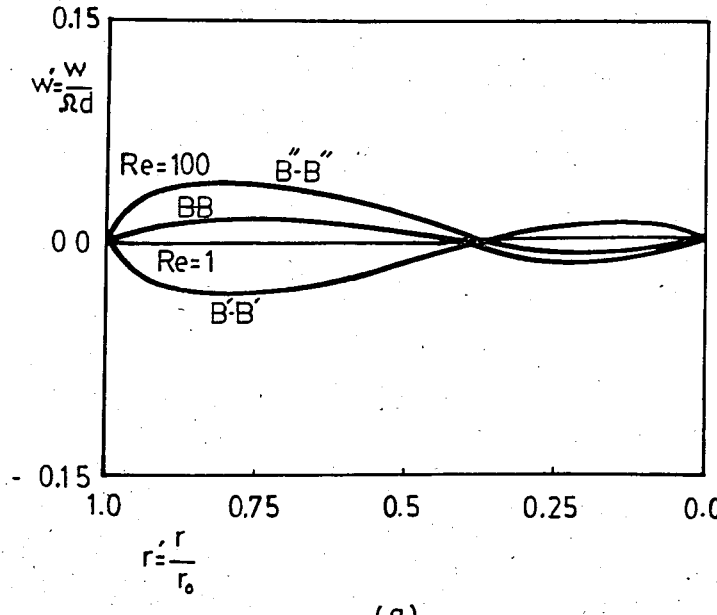


Figure IV.25- Radial Velocity Profile at Mid-Section and at Section A-A for Re = 100





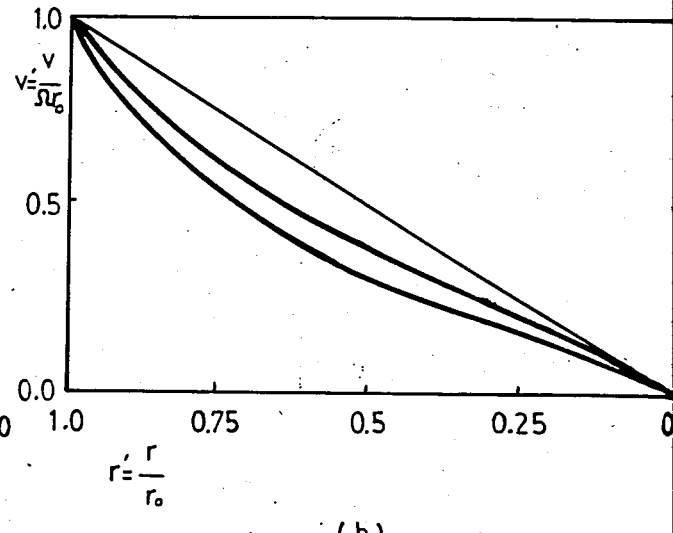


Figure IV.26- Velocity Profiles at Sections B-B, B'-B' and B"-B" for Re = 1, 100; (a) Axial, (b) Tangential

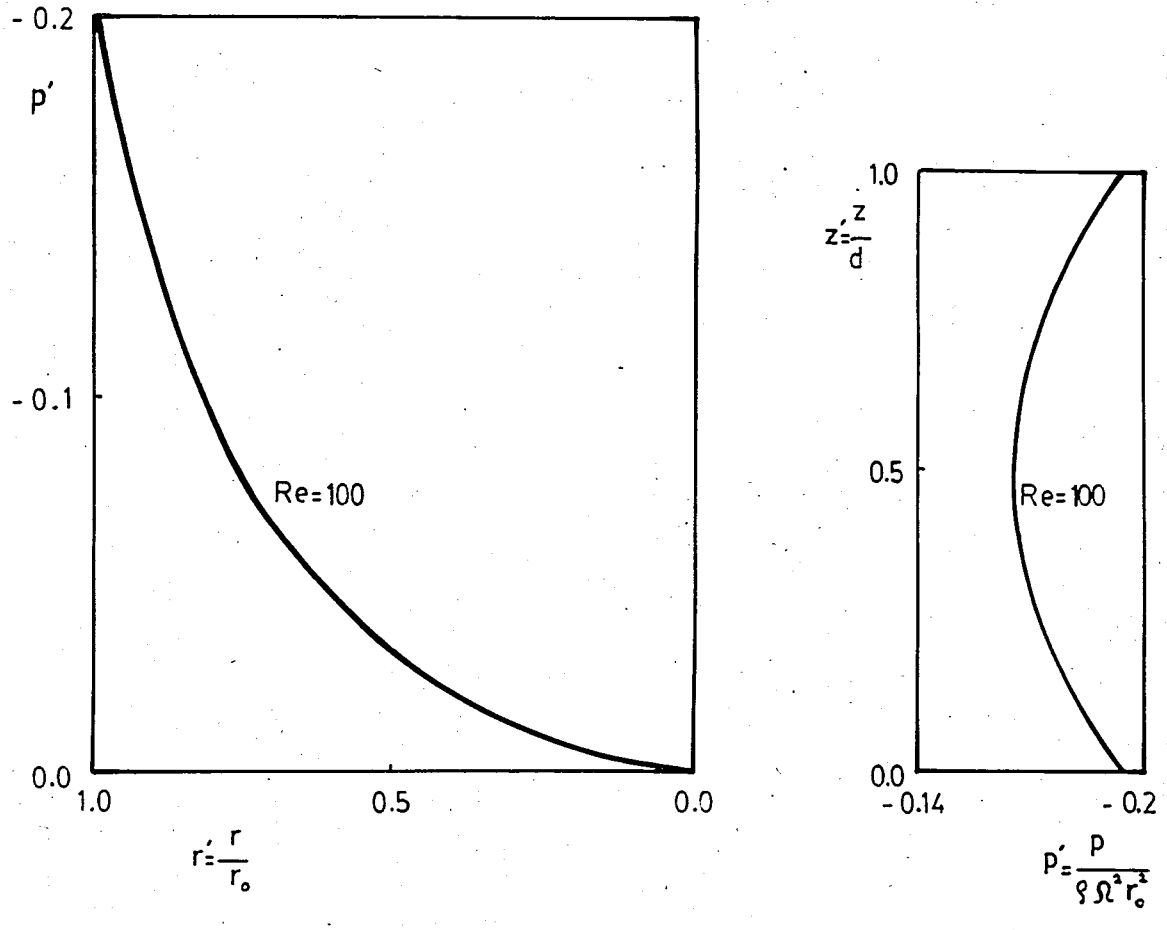


Figure IV.27- Pressure Distribution on the Rotating Cylinder for Re = 100 (a) Radial (b) Axial

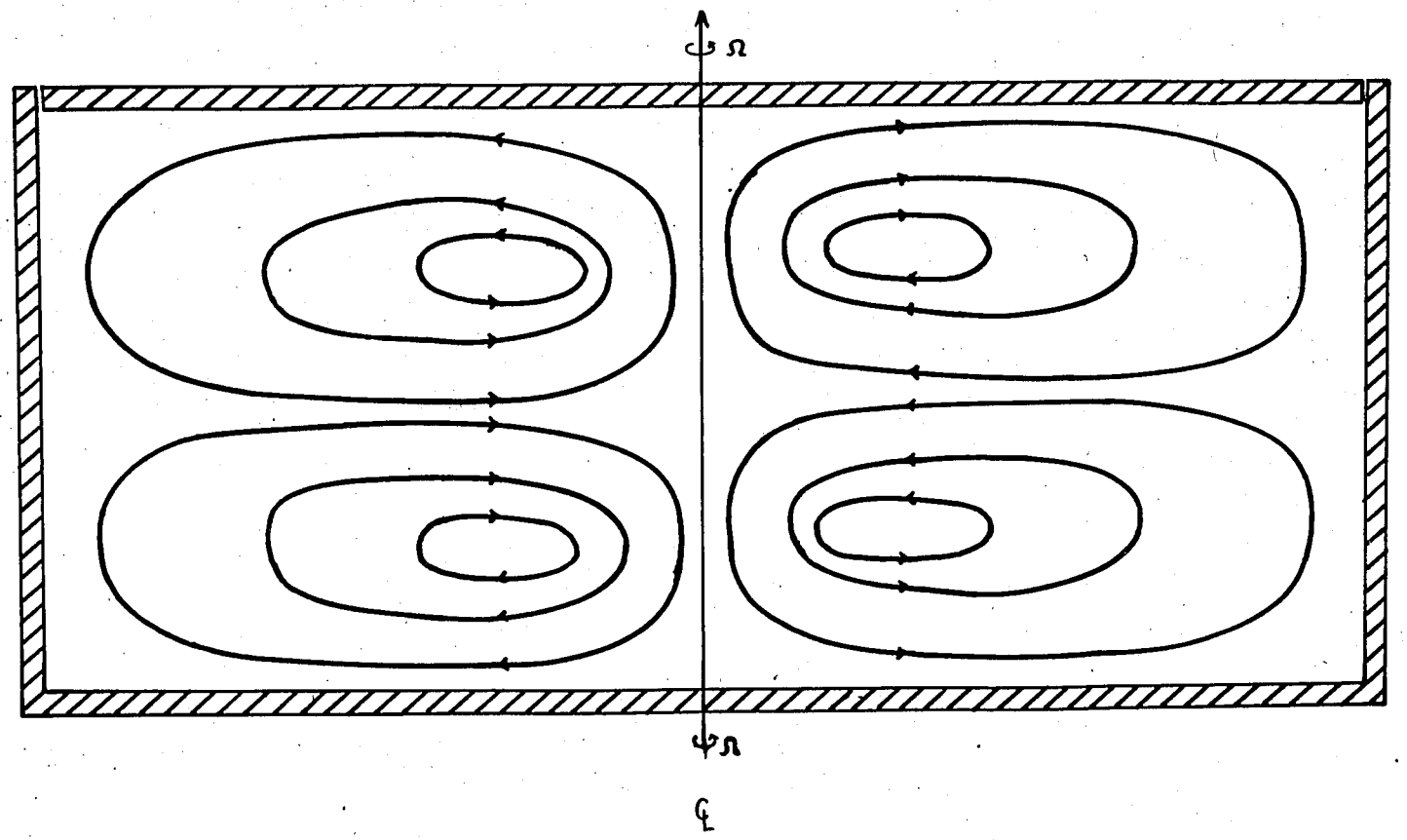


Figure IV.28- Streamlines of Flow at Re = 100 for Rotating Cylinder

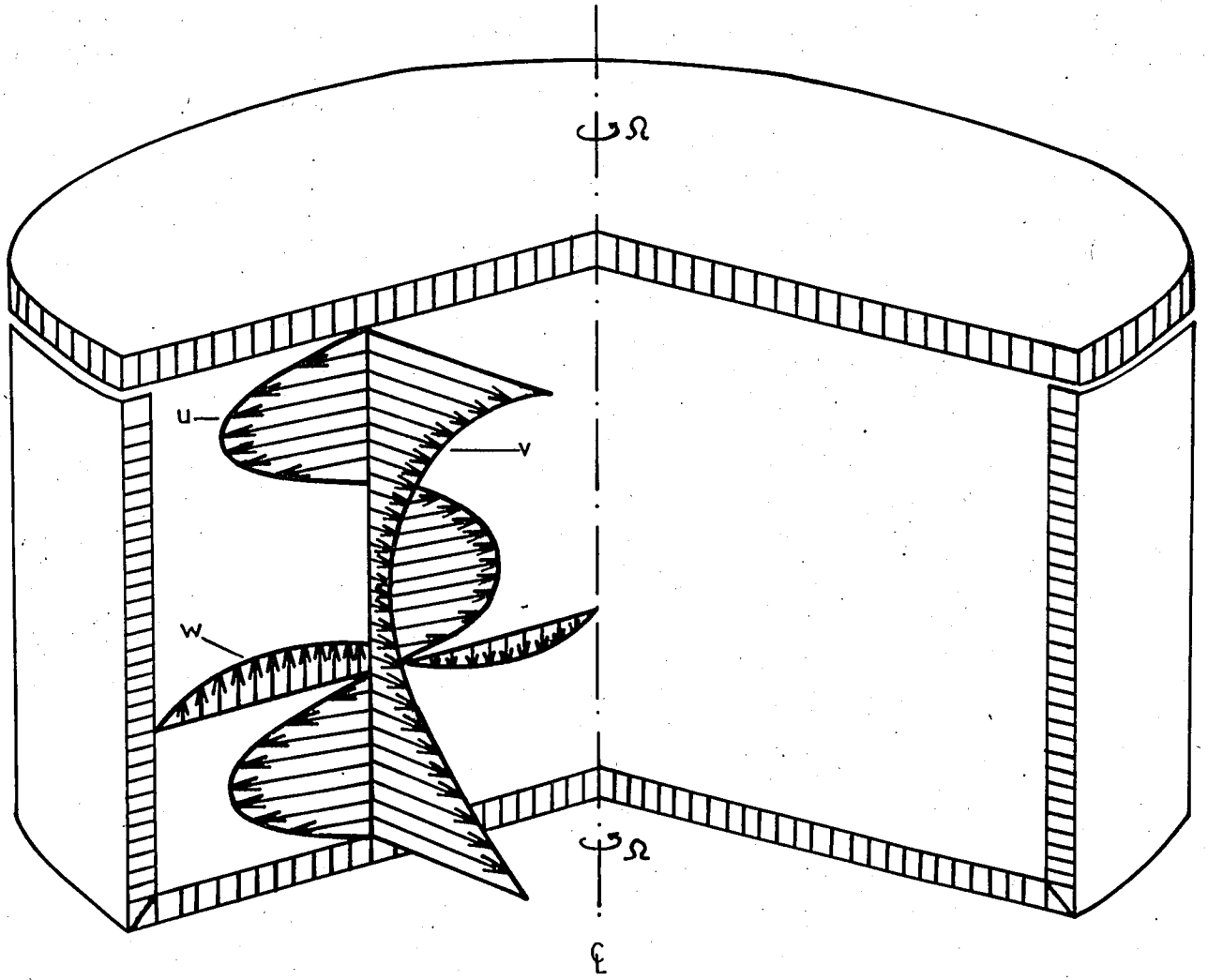


Figure IV.29- Illustration of Flow inside of a Rotating Circular Cylinder

V

CONCLUSIONS AND RECOMMENDATIONS

1- The flow between a rotating disk and a stationary shroud, and the flow between a rotating disk and a rotating shroud have been studied by using the Finite Element Method.

2- For the flow between a rotating disk and a stationary shroud, as Reynolds number is increased secondary flows become more significant and velocity gradients increase.

3- For the flow inside of a rotating circular cylinder, secondary flows are not observed for Re < 100.

4- The flow between a rotating disk and a stationary shroud exhibits a single-cell structure, and the flow inside of a rotating circular cylinder exhibits a double-cell structure.

5- For the flow between a rotating disk and a stationary shroud, it has been shown that for increasing r_o/d axial velocity vanishes except the ends. As the distance between the discs is increased , the velocity components decrease except near the rotating disk.

6- The use of the boundary layer theory is recommended for the calculation of high Reynolds number flows.

VI

REFERENCES

1- KARMAN, T.VON: "Uber Laminare und Turbulente Reibung", ZAMM, Vol.1, p.233, 1921.

2- NGUYEN, N.D., RIBAULT, J.P., and FLORENT P. "Multiple Solutions for Flow Between Coaxial Disks" J.Fluid Mech. Vol.68, part 2, pp.369-388, 1975.

3- ROBERTS, S.M., SHIPMAN, J.S.,:"Computation of the Flow Between a Rotating and a Stationary Disk" J.Fluid Mech. Vol.73. part 1, pp 53-63, 1976.

4- CEBECI, T.:"Turbulent Boundary Layer on a Rotating Disk", AIAA, J.Vol.13, p.829, 1975.

5- SCHULTZ-GRUNOW,F.: Der Reibungswiderstand Rotierender Scheibenin Gehaussen", ZAMM, Vol.15, p.191, 1935.

6- SOO,S.L.: "Laminar Flow Over an Enclosed Rotating Disk", Trans. ASME, Vol.80, p.287. 1958.

7- CANOVER,R.A.: "Laminar Flow Between a Rotating Disk and a Parallel Stationary Wall With Radial Inflow"Jour. of Basic Engineering, Vol.90, p.325, 1978.

8- MITCHELL,J.W. and D.E. METZGER: "Heat Transfer From a Shrouded Rotating Disk to a Single Fluid Stream", Journal of Heat Transfer, Vol.87, p.485, 1965.

9- BAYLEY, F.J., MORRIS,W.D., OWEN,J.M., and TURNER,A.B.: "Boundary-Layer Prediction Methods Applied to Cooling Problems in the Gas Turbine", Ministery of Aviation Supply Aeronautical Research Council Research Papers, London, 1971.

10- DAILY, J.W. and NECE, R.J. : "Chamber Dimension Effects on Induced Flow and Frictional Resistance of Enclosed Rotating Disks", Journal of Basic Engineering, Vol.82, p.217, 1960.

11- HAYAMI,H. and SENOO,Y.: "An Analysis on the Flow in a Casing Induced by a Rotating Disk Using Four-Layer Flow Model", Transactions of the ASME Journal of Fluids Engineering, Vol.98, Page 192, June 1976.

12- MELLOR,G.L., CHAPPLE,P.J. and STOKES,V.K.: "On the Flow Between a Rotating and Stationary Disk", J.Fluid Mech. Vol.31, p.95, 1968.

13- HUEBNER,K.H.: "The Finite Element Method for Engineers", John Wiley and Sons, 1975.

14- SCHLIGHTING,H.: "Boundary Layer Theory", Mc Graw Hill, 1968.

15- JAWA,M.S.: "A Numerical Study of Axially Symmetric Flow Between Two Rotating Infinite Porous Disks", Journal of Applied Mechanics, pg.683, September 1971.

APPENDICES

APPENDIX A

NATURAL COORDINATES AND INTERPOLATION FUNCTIONS

The basic purpose of a natural coordinate system is to describe the location of a point inside an element in terms of the coordinates of the nodes associated to that element. The natural coordinates L_i are functions of the global cartesian coordinates in which element is defined. For a triangular element three natural coordinates, denoted by L_1, L_2, L_3 exist.

A convenient set of coordinates L_1, L_2, L_3 for a triangle can be defined by the following linear relationships between these and the cartesian system:

$$r = L_1 r_1 + L_2 r_2 + L_3 r_3$$

$$z = L_1 z_1 + L_2 r_2 + L_3 z_3 \qquad (11)$$

$$1 = L_1 + L_2 + L_3$$

To every set of L_1, L_2, L_3 (which are not independent but are related by the third equation) corresponds a unique set of Cartesian coordinates. A linear relationship between the new and Cartesian coordinates implies that the contours of L_1 equally spaced straight lines parallell to side 2-3, on which $L_1 = 0$ and at point 1, $L_1 = 1$, $L_2 = 0$ etc. (See Fig.A.1).

$$L_i = \frac{1}{2A}\left(a_i + b_i r + c_i z\right) \qquad (i = 1,2,3) \qquad (12)$$

where $a_i = r_j z_k - r_k z_j$; $b_i = z_j - z_k$; $c_i = r_k - r_j$ (12a)

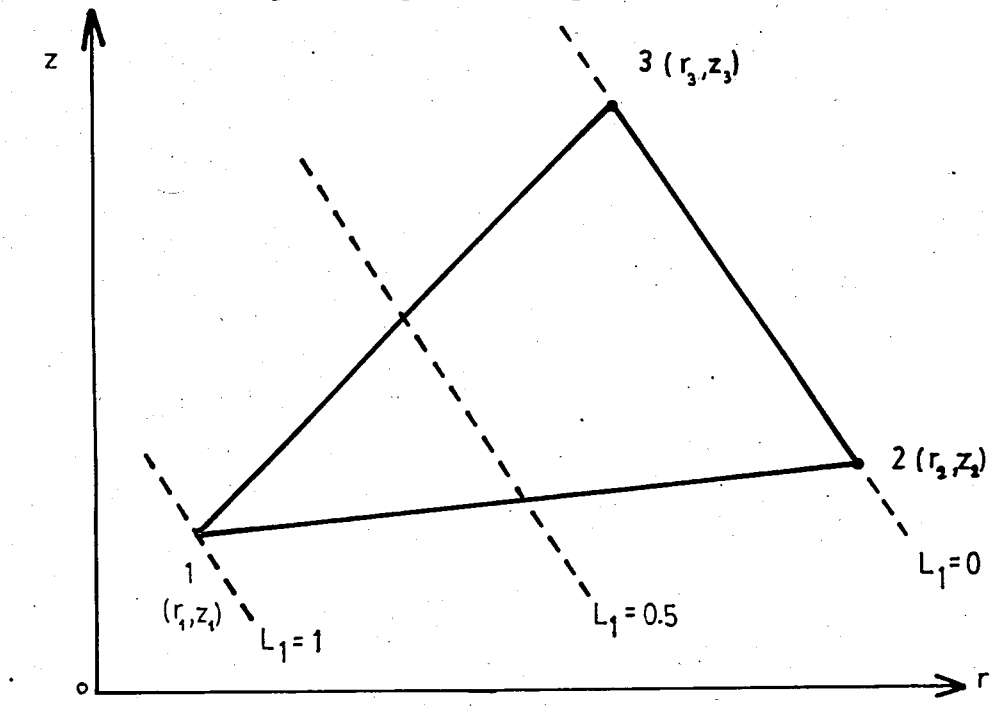


Figure A.1- Natural Coordinates for a Triangular Element

and

$$A = \frac{1}{2}\det\begin{vmatrix} 1 & r_1 & z_1 \\ 1 & r_2 & z_2 \\ 1 & r_3 & z_3 \end{vmatrix} = \text{Area of the triangle} \qquad (12b)$$

Quadratic interpolation functions for a triangular finite element can be expressed in terms of the natural coordinates as follows :

$$\begin{aligned} N_1 &= L_1(2L_1-1) \\ N_2 &= L_2(2L_2-1) \\ N_3 &= L_3(2L_3-1) \\ N_4 &= 4L_1L_3 \\ N_5 &= 4L_1L_2 \\ N_6 &= 4L_2L_3 \end{aligned} \qquad (13)$$

More information about the natural coordinates and interpolation functions can be found in Ref. 13.

APPENDIX B

DERIVATION OF THE ELEMENT EQUATIONS

Element equations are derived by using Galerkin's method. Galerkin's method is a branch of weighted residual methods, which is a technique for obtaining approximate solutions to linear and non-linear partial differential equations.

Applying the method of weighted residuals involves basically two steps. The first step is to assume the general functional behaviour of the dependent field variable to approximately satisfy the given differential equation and the boundary conditions. The error resulting from substitution of this approximation into the original differential equation and the boundary conditions is called a residual. This residual is required to vanish over the entire solution domain. The second step is to solve the equations resulting from the first step.

Consider the differential equation :

$$L(\phi) - f = 0 \qquad (14)$$

in the domain D bounded by the surface Σ. The function f is a known function of the independent variables and proper boundary conditions are described on . Let's approximate the dependent variable ϕ by $\tilde{\phi}$ as follows:

$$\phi \cong \tilde{\phi} = \sum_{i=1}^{m} N_i C_i \tag{15}$$

where N_i are the assumed functions and the C_i are the unknown functions of the independent variables. The N_i are usually chosen to satisfy the global boundary conditions.

Substituting (14) into (15).

$$L(\tilde{\phi}) - f = R \tag{16}$$

where R is the residual or error resulting from the approximation. The method of weighted residuals seeks to determine the m unknowns C_i by minimizing the error R over the solution domain. This is accomplished by forming a weighted average of the error and then forcing this weighted average to vanish over the solution domain. The weighing functions (W_i) are chosen as the interpolation functions (N_i) in Galerkin's method. Thus, Galerkin's method requires that

$$\int_D \left[L(\tilde{\phi}) - f \right] N_i \, dD = 0, \quad i = 1,2,\ldots,m \tag{17}$$

consider the r-directional momentum equation (Eq.2a). When Galerkin Method is applied, it becomes:

$$\int_{D(e)} N_i \left(u \frac{\partial u}{\partial r} - \frac{v^2}{r} + w\frac{\partial u}{\partial z}\right) dz = \int_{D(e)} N_i \left(- \frac{\partial p}{\partial r} + \frac{1}{Re} \left\{\frac{\partial^2 u}{\partial r^2} + \frac{\partial}{\partial r}\left(\frac{u}{r}\right) + \left(\frac{r_o}{d}\right)^2 \frac{\partial^2 v}{\partial z^2}\right\}\right) dV$$

$dV = r\, d\theta\, dz\, dr$

Integrating over θ,

$$\int_{D(e)} N_i \left(u \frac{\partial u}{\partial r} - \frac{v^2}{r} + w \frac{\partial u}{\partial z}\right) r\, dr\, dz = \int_{D(e)} N_i \left(- \frac{\partial p}{\partial r} + \frac{1}{Re} \left\{ \frac{\partial^2 u}{\partial r^2} + \frac{\partial}{\partial r}\left(\frac{u}{r}\right) + \left(\frac{r_o}{d}\right)^2 \frac{\partial^2 u}{\partial z^2} \right\} r\, dr\, dz \right.$$

To introduce the natural boundary conditions, it is convenient to integrate by parts the highest ordered derivatives in the viscous part.

$$I_1 = \int_{D(e)} N_i r \frac{\partial^2 u}{\partial r^2}\, dr\, dz = \int_{S(e)} N_i r \frac{\partial u}{\partial r}\, dz - \int_{D(e)} \frac{\partial u}{\partial r}\left(\frac{\partial N_i}{\partial r} r + N_i\right) dr\, dz$$

$$I_2 = \int_{D(e)} \left(\frac{r_o}{d}\right)^2 N_i r \frac{\partial^2 u}{\partial z^2}\, dr\, dz = \int_{S(e)} \left(\frac{r_o}{d}\right)^2 N_i r \frac{\partial u}{\partial z}\, dr - \int_{D(e)} \left(\frac{r_o}{d}\right)^2 \frac{\partial u}{\partial z} \frac{\partial N_i}{\partial z} r\, dr\, dz$$

$$I_1 + I_2 = \int_{S(e)} N_i r \left(\frac{\partial u}{\partial r}\right) dz + \int_{S(e)} \left(\frac{r_o}{d}\right) N_i r \frac{\partial u}{\partial z}\, dr -$$

$$\int_{D(e)} \left(\frac{\partial N_i}{\partial r} r + N_i\right) \left(\frac{\partial u}{\partial r}\right) + \left(\frac{r_o}{d}\right)^2 \frac{\partial u}{\partial z} \frac{\partial N_i}{\partial z} r\, dr\, dz$$

Back substituting these integrals and inserting the aproximations for velocity and pressure (Eqs. 8 and 10) into the Eq.13.

$$\int_{D(e)} N_i \left(u^* \frac{\partial N_j}{\partial r} u_j - \frac{V^* N_j V_j}{r} + w^* \frac{\partial N_j}{\partial z} u_j\right) r\, dr\, dz =$$

$$= \int_{D(e)} r N_i \left(-\frac{\partial H_k}{\partial r} p_k\right)$$

$$+ \frac{1}{Re}\left\{-\left(\frac{\partial N_i}{\partial r} r + N_i\right) + \left(\frac{\partial N_j}{\partial r} uj\right) + \frac{\partial N_j}{\partial r} N_i V_j - \frac{N_i N_j}{r}\right.$$

$$\left. -\left(\frac{r_o}{d}\right)^2 \frac{\partial N_j}{\partial z} \frac{\partial N_i}{\partial z} r \right\} dr\, dz + \int_{S(e)} N_i r \frac{\partial u}{\partial r} dz + \int_{S(e)} \left(\frac{r_o}{d}\right)^2 N_i r \frac{\partial v}{\partial z} dr$$

Rearranging,

$$\int_{D(e)} \left\{ u_j \left[N_i \frac{\partial N_j}{\partial r} u^* r + N_i \frac{\partial N_j}{\partial z} w^* r + \frac{1}{Re}\left(\frac{\partial N_i}{\partial r} \frac{\partial N_j}{\partial r} r + \frac{N_i N_j}{r} + \right.\right.\right.$$

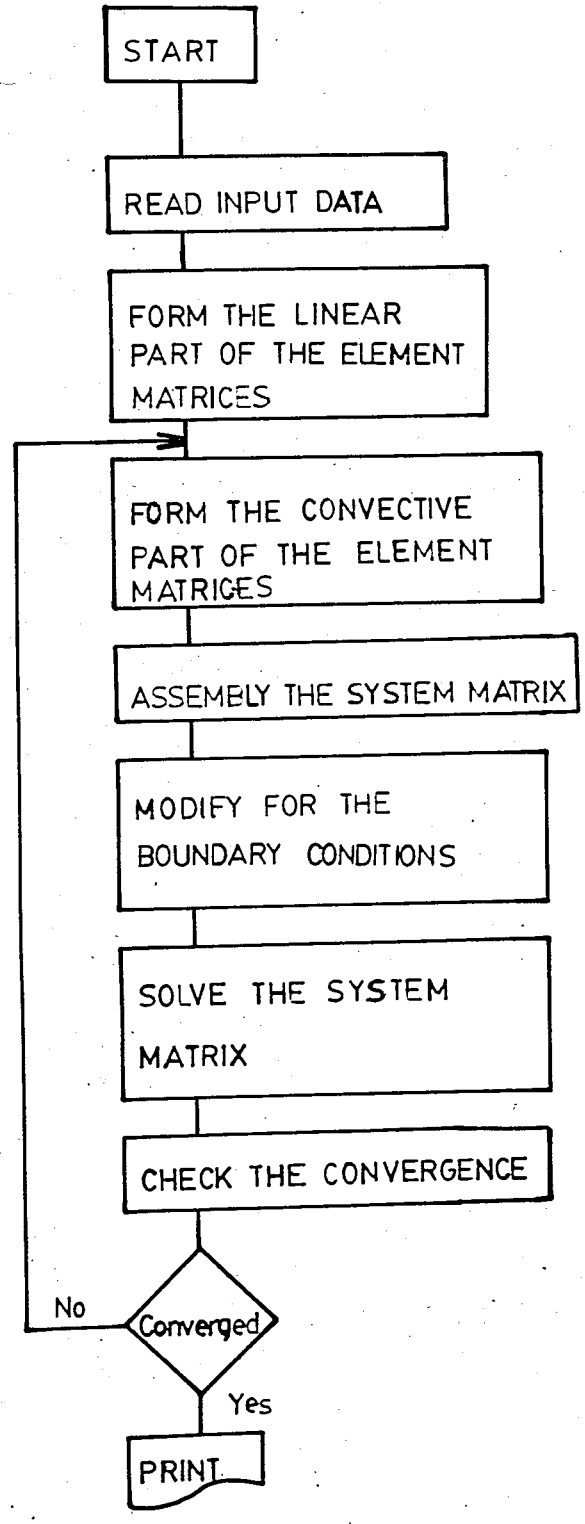
$$\left.\left.\left. \left(\frac{r_o}{d}\right)^2 \frac{\partial N_j}{\partial z} \frac{\partial N_i}{\partial z} r\right)\right] - v_j \left[N_i N_j V^*\right] + P_k \left[N_i \frac{\partial H_k}{\partial r} r\right] \right\} dr\, dz$$

$$= \int_{S(e)} N_i r \frac{\partial u}{\partial r} dz + \int_{S(e)} \left(\frac{r_o}{d}\right)^2 N_i r \frac{\partial u}{\partial z} dr \qquad (18)$$

θ and z directional components of the element equations are obtained similarly. The continuity equation is weighted by linear interpolation functions.

APPENDIX C

I. COMPUTER PROGRAM FLOWCHART



2. COMPUTER PROGRAM LOGIC

The computer program is consist of a main program and six subroutines.

SUBROUTINE INPUT : Reads some parameters, necessary constants and the boundary conditions as input data. A uniform mesh structure can be automatically generated by this subroutine by inputing the number of meshes in r- and z- directions. If a non-uniform mesh structure is desired, the coordinates of nodes and number of nodes belonging to each element must be given as input. The mesh used in this work is shown in Fig. A.4.

SUBROUTINE SMATR : This subroutine generates the code numbers and modifies them according to the given boundary conditions. Using these new code-numbers it assembles the system matrix, by calling the subroutines EMATR, PRCON and CONVEC which form the element equations, for each element. Then the system matrix is modified according to the boundary conditions.

a) Assembly of the System Matrix

At each corner node, there are four unknowns u,v,w and p whereas at each side node there are three unknowns u,v, and w. In the global system, a spesific number is assigned to each of these variables, which are called code-numbers.

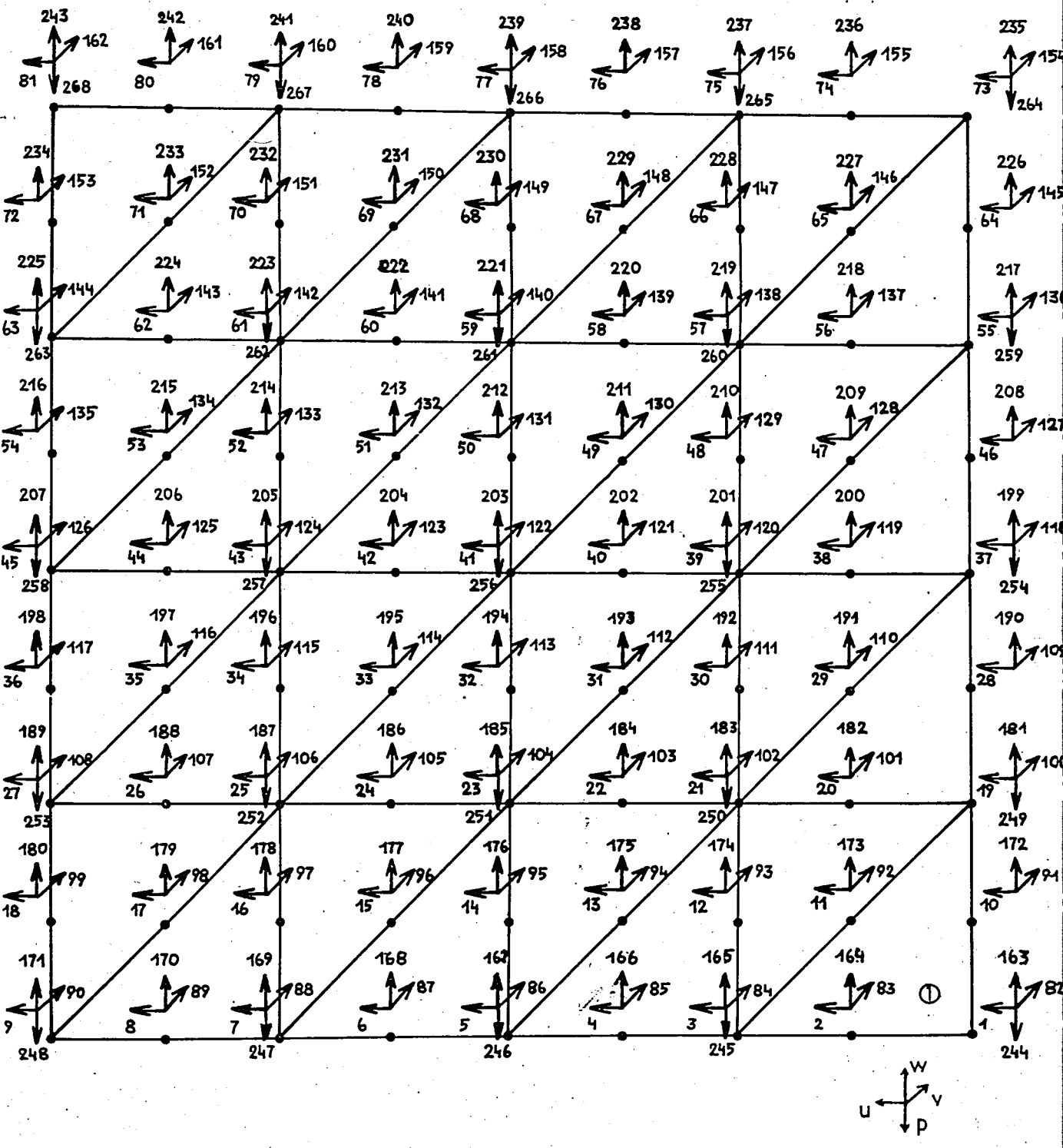


Figure A.2- Code Numbers

Since each element has 21 unknowns, local node numbers are I = 1,2,3, ... 21. The global node numbers for the element are recovered from the parameter Codenumber (Element, I). By means of this parameter, each of the local node numbers are related to system code numbers. Forexample, 1st element in Fig. A.2 assembles into the system matrix as in Fig. A.3.

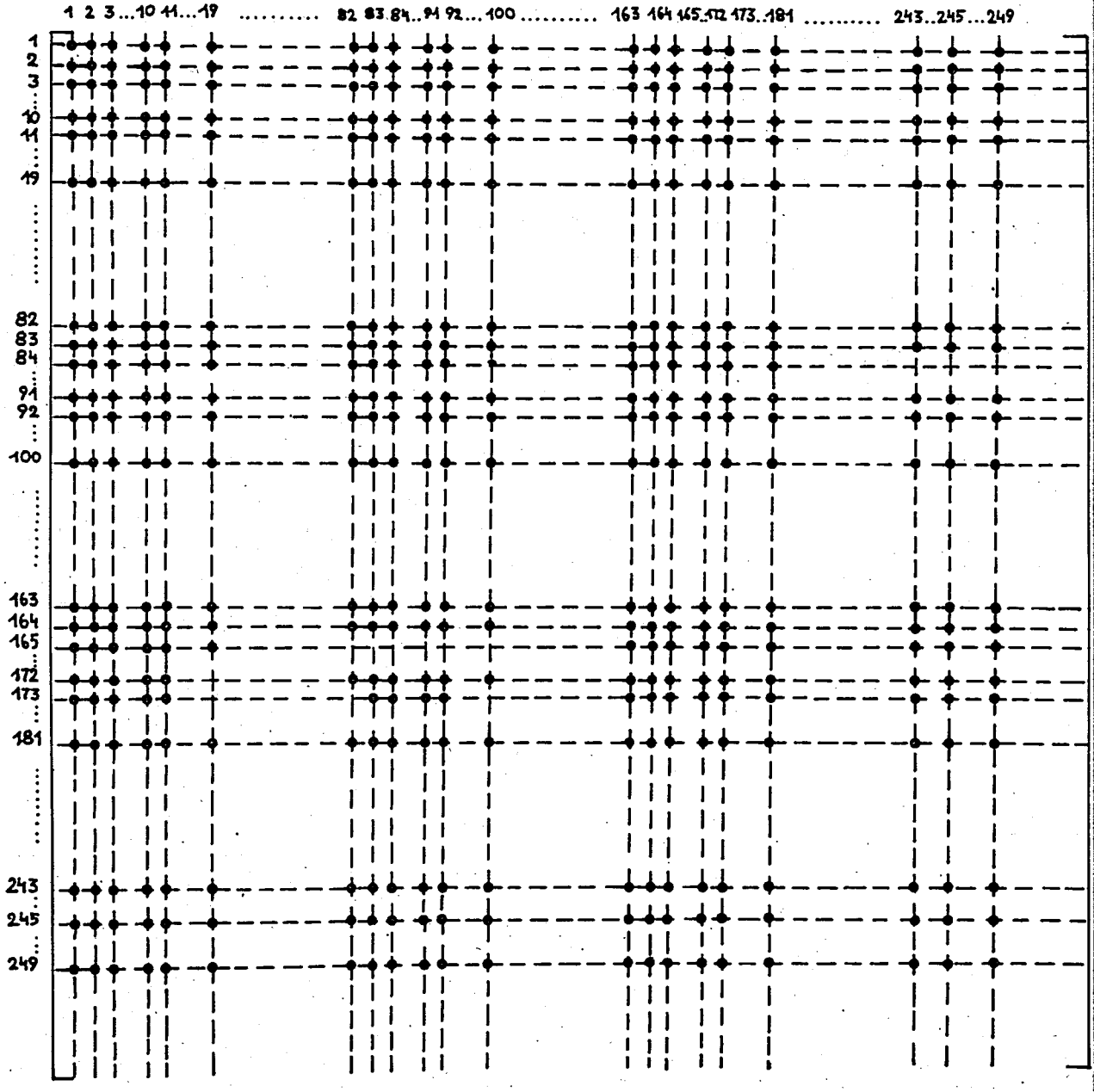


Figure A.3- Assembling Into System Matrix

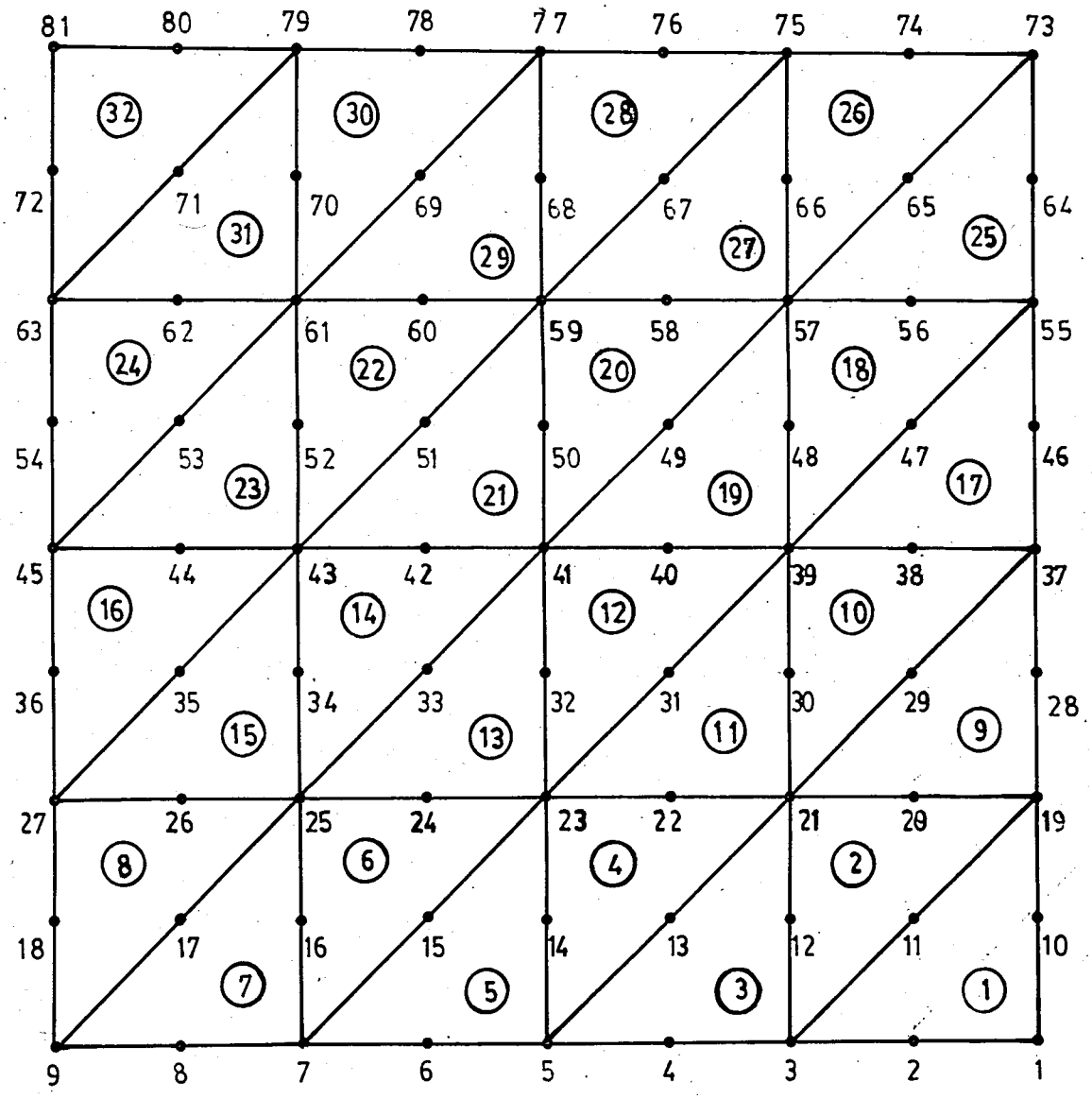


Figure A.4- The Mesh Used

b) Modification of the Boundary Conditions

For Neumann's type of boundary conditions, R_i integrals must be evaluated and added to the right-hand side of the system equations as forcing functions.

The procedure to modify for the Dirichlet's type of boundary conditions is as follows If i is the subscript of a precribed nodal variable, then the ith row and ith column of the system matrix are set equal to zero and k_{ii} is set to unity. The term R_i of the right-hand side vector is replaced by the unknown value of x_i. Each of the n-1 remaining terms of the right-hand side vector (R) is modified by substracting from it the value of the prescribed nodal variable multiplied by the appropriate column term from the original system matrix. This procedure is repeated for each prescribed x_i until all of them have been included.

To illustrate the procedure, one can consider the following simple example with only four system equations:

The final equations will have the form :

$$\begin{bmatrix} k_{11} & k_{12} & k_{13} & k_{14} \\ k_{21} & k_{22} & k_{23} & k_{24} \\ k_{31} & k_{32} & k_{33} & k_{34} \\ k_{41} & k_{42} & k_{43} & k_{44} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} R_1 \\ R_2 \\ R_3 \\ R_4 \end{bmatrix}$$

Suppose for this hyphotetical system, nodal variables x_1 and x_3 are specified as

$$x_1 = \beta_1 \qquad x_3 = \beta_3$$

When these boundary conditions are inserted according to the procedure described above, the equations become:

$$\begin{bmatrix} k_{22} & k_{24} \\ k_{42} & k_{44} \end{bmatrix} \begin{bmatrix} x_2 \\ x_4 \end{bmatrix} = \begin{bmatrix} R_2 - k_{21}\beta_1 - k_{23}\beta_3 \\ R_4 - k_{41}\beta_1 - k_{43}\beta_3 \end{bmatrix}.$$

Since there is no need to solve for x_1 and x_3, their equations are deleted from the system matrix and a 4x4 matrix is reduced to a 2x2 matrix.

Discarding the prescribed nodes from the system matrix also saves valuable computer core. To achieve this type of modification, the code-number array, relating the local numbers of unknowns to the global numbers is modified to give a new code-number array. For the prescribed values two code numbers are set to zero and the remaining numbers form the new code number array. Since prescribed nodes are not solved for, the equations are modified according to the above procedure.

If there are also Neumann's type of boundary conditions prescribed on a surface, the surface integrals R_i^1, R_i^2, R_i^3 (Chapter III) must be evaluated for the elements staying along this surface. The nodes on this surface must then be solved together with the interior nodes. For the elements which are not on this boundary the surface integrals mentioned above vanish automatically.

SUBROUTINE EMATR AND SUBROUTINE PRCON : These subroutines form the linear part of the element equations. To evaluate integrals contained in the element equations, the functions numbered from 1 to 22 are called by subroutine EMATR.

SUBROUTINE CONVEC : This subroutine forms the convective part of the element equations.

SUBROUTINE LEQTIF : Solves the system matrix.

SUBROUTINE ITER : Checks the convergence by comparing the last velocity field with the previous one. If the given convergence criterion is not satisfied, it replaces the new velocity field with the previous one. If convergence is achieved, it prints out the results and stops the program.

In the allowed computer time, convergence may not be achieved, In this case the computer program must be run in sequences. After the last iteration of the first sequence, the non-converged velocity and pressure values are written on a disk, and the next sequence starts by reading these values from disk.

Definition of the variable names used in the program.

PARAMETERS

NNN, NTE - Total number of elements
NTN - Total number of nodes
NNM, NVB - Total number of boundary conditions
N,NNI,NTU - Total number of unknowns
NTT - Total number of variables
NPB - Total number of pressure boundary conditions
NNIP - Total number of unknown pressures

VARIABLES

ERR - Acceptable relative error for the converged solution

ELM - Element matrix
FAC - Aspect ratio
IBEGIN - A control parameter. When given as "O" the velocity field is initialized. Otherwise the values are read from the disk
IDISK - Number of iterations required for each sequence
ITR - Iteration number
JNCODE - Code numbers
NOD - The associated node numbers for each element
NODCOR - The associated numbers of the corner nodes for each element
NER - Number of elements required in r-direction if automatic mesh generation will be used
NEZ - Number of elements required in z-direction if automatic mesh generation will be used
NEL - Element number
NTCN - Total number of corner nodes
NTSN - Total number of side nodes
NNT - Total number of nodes
NET - Total number of elements
NCODE - Modified code numbers
OM - Relaxation constant
P - Nodal values of the pressure
RNOD - r-coordinates of the nodes
ZNOD - z-coordinates of the nodes
RHS - Right-hand side matrix
ST,S - System matrix
STE - Element matrix
UVWP - Right-hand side matrix
U,V,W - Nodal values of the velocity components
UP, VP,WP - Nodal values of the velocity components from the previous iteration
US,VS,WS - Nodal values of velocity components from the previous iteration for each element
VAL,VBND - Prescribed values at the boundary nodes

Input Description :

1- Aspect ratio
Format: (F 12.10)

2- The first number given in this set controls the automatic mesh generation if it is given as "1" the mesh is generated automatically. If it is given as "2" the node numbers of the elements and nodal coordinates must be given in the data. The second number controls the print out of the element matrices. When given as "1" the element matrices of each element is printed on the output. Format: (2 15).

3- A small value to shift from the axis of rotation to avoid singularity.
Format: (F 12.10)

4- Some control parameters are given in this group. The first one defines the number of iterations to be carried out in each sequence. The second is given as zero for the first sequence and as "1" for the other sequence. The third one is the number of variables on which the convergence test is to be applied fourth are is the relaxation constant. Whereareas the fifth is the relative error tolerance.
Format: (3 15, 2 F10.5).

5- Reynolds number and number of elements required in r and z directions if automatic generation is preferred.
Format: (F 10.5, 2 15)

6- Number of nodes at which a boundary condition is prescribed and the prescribed values at these nodes.
Format: (8 (13, F 7.5)).

On the output, nodal values of radial, tangential and axial components of the velocity and the pressures are listed.
Format: (12 F10.5).

3. LISTINGS

MAIN PROGRAM

```
      PARAMETER NNN=32,NTN=81,NNH=97,N=171
      COMMON R1,R2,R3,KP(6,6),UNM(6,6),STE(21,21),RNOD(NTN),ZNOD(NTN)
     1NOD(NNN,6),NPT(NNH),VAL(NNH),ST(N,N),RHS(N,1),NODS(6),NCORS(3),
     2NCOR(NNN,3),NODCOR(NNN,3),FAC,RE,IPRINT,UN,RO
      DIMENSION WKAREA(N)
      CALL INPUT
   10 CONTINUE
      CALL SMATR
      CALL LEQTIF(ST,1,N,N,RHS,0,WKAREA,IER)
      CALL ITER
      GO TO 10
      END
```

```
      SUBROUTINE INPUT
      PARAMETER NTE=32,NTN=81,NVB=97,NNI=171
      COMMON R1,R2,R3,KP(6,6),UNM(6,6),ELM(21,21),RNOD(NTN),ZNOD(NTN),
     1NOD(NTE,6),NVELS(NVB),VBND(NVB),S(NNI,NNI),UVWP(NNI),NODS(6),
     2NCORS(3),NCOR(NTE,3),NODCOR(NTE,3),FAC,RE,IPRINT,UN,RO
      COMMON/IN/OM,ERR,IBEGIN,ITR,IDISK,NOC,NER,NEZ
      REAL KP
      ITR=1
      READ(5,149)FAC
      WRITE(6,149)FAC
      READ(5,55)NOP,IPRINT
      WRITE(6,55)NOP,IPRINT
      READ(5,149)EPS
      WRITE(6,149)EPS
      READ(5,150)IDISK,IBEGIN,NOC,OM,ERR
      WRITE(6,150)IDISK,IBEGIN,NOC,OM,ERR
  150 FORMAT(3I5,2F10.5)
  149 FORMAT(F12.10)
      IF(NOP.EQ.2)GO TO 9111
      READ(5,5661)RE,NER,NEZ
 5661 FORMAT(F10.5,2I5)
      ZMAX=NER*2
      RMAX=NEZ*2
      NNMB=(2*NEZ+1)*(2*NER)+1
      NNM=1
      DO 7766 I=1,2*NER+1
      DO 7766 J=1,2*NEZ+1
      RNOD(NNM)=J-1
      ZNOD(NNM)=I-1
      WRITE(6,1)NNM,RNOD(NNM),ZNOD(NNM)
 7766 NNM=NNM+1
      NNT=NNM-1
      NET=NER*NEZ*2
      NTVB=4*NER+2
      NTPB=NER+NEZ+1
      NTCN=(NER+1)*(NEZ+1)
      NTSN=NNT-NTCN
      WRITE(6,1212)RE,NER,NEZ,NNT,NET,NTVB,NTPB,NTCN,NTSN
 1212 FORMAT(F10.5,8I5)
      READ(5,624)(NVELS(I),VBND(I),I=1,NVB)
      WRITE(6,624)(NVELS(I),VBND(I),I=1,NVB)
  624 FORMAT(8(I3,F7.5))
      DO 5555 I=1,NER
      M=(I-1)*(2+4*NEZ)+1
      DO 6666 NEL=2*NEZ*(I-1)+1,2*NEZ*I,2
      NOD(NEL,1)=M
      NOD(NEL,2)=4*NEZ+2+M
      NOD(NEL,3)=2+M
      NOD(NEL,4)=1+M
      NOD(NEL,5)=2*NEZ+1+M
      NOD(NEL,6)=2*NEZ+2+M
      NOD(NEL+1,1)=M+2
      NOD(NEL+1,2)=4*NEZ+2+M
```

```
      NOD(NEL+1,3)=4*NEZ+4+M
      NOD(NEL+1,4)=2*NEZ+3+M
      NOD(NEL+1,5)=2*NEZ+2+M
      NOD(NEL+1,6)=4*NEZ+3+M
 6666 M=M+2
      M=(I-1)*(NEZ+1)+1
      DO 6565 NEL=2*NEZ*(I-1)+1,2*NEZ*I,2
      NODCOR(NEL,1)=M
      NODCOR(NEL,2)=NEZ+1+M
      NODCOR(NEL,3)=M+1
      NODCOR(NEL+1,1)=M+1
      NODCOR(NEL+1,2)=NEZ+1+M
      NODCOR(NEL+1,3)=NEZ+2+M
 6565 M=M+1
 5555 CONTINUE
      WRITE(6,6)((NOD(I,J),J=1,6),I=1,NET)
    6 FORMAT(6I5)
      GO TO 9112
 9111 NOP2=2
      I=0
      READ(5,1010)RE
      WRITE(6,1010)RE
 1010 FORMAT(2F10.5,I5)
      READ(5,11)NNT,NET,NTVB,NTPB,NTCN,NTSN
      WRITE(6,11)NNT,NET,NTVB,NTPB,NTCN,NTSN
   11 FORMAT(6I4)
   10 READ(5,1)NNM,RC,ZC
      WRITE(6,1)NNM,RC,ZC
    1 FORMAT(I4,2F10.5)
      RNOD(NNM)=RC
      ZNOD(NNM)=ZC
      I=I+1
      IF(I.LT.NNT)GO TO 10
 9112 CONTINUE
      DO 1011 NNM=1,NNT
      RNOD(NNM)=RNOD(NNM)/ZMAX
      IF(RNOD(NNM).EQ.0.)RNOD(NNM)=EPS
      ZNOD(NNM)=ZNOD(NNM)/ZMAX
 1011 WRITE(6,1)NNM,RNOD(NNM),ZNOD(NNM)
      IF(NOP.EQ.1)GO TO 9113
      I=0
   20 READ(5,2)NEL,NODS,NCORS
      WRITE(6,2)NEL,NODS,NCORS
    2 FORMAT(I4,6I4,3I4)
      DO 25 J=1,6
   25 NOD(NEL,J)=NODS(J)
      DO 26 J=1,3
   26 NCOR(NEL,J)=NCORS(J)
      I=I+1
      IF(I.LT.NET)GO TO 20
   55 FORMAT(2I5)
 9113 CONTINUE
      UN=1./RE
      RO=1.
      RETURN
      DEBUG SUBCHK
      END
```

```
      SUBROUTINE SMATR
C
C       .....THE ELEMENT k MATRICES OF ALL ELEMENTS ARE ASSEMBLED
C            BY NODES, THE PRESCRIBED BOUNDARY CONDITIONS ARE
C            INSERTED,AND THE FINAL SYSTEM MATRIX EQUATION IS
C            OBTAINED.....
      PARAMETER NNN=32,NTN=81,NNM=97,N=171,NTT=268

      COMMON R1,R2,R3,KP(6,6),UNM(6,6),STE(21,21),RNOD(NTN),ZNOD(NTN
     1NOD(NNN,6),NPT(NNM),VAL(NNM),ST(N,N),RHS(N,1),NODS(6),NCORS(3)
     2NCOR(NNN,3),NODCOR(NNN,3),FAC,RE,IPRINT
      COMMON/CODE/NCODE(NNN,21),JDEF(NTT),JNCODE(NNN,21)
      DIMENSION Q(21),F(21)
      DO 290 I=1,N
      DO 280 J=1,N
  280 ST(I,J)=0.0
  290 RHS(I,1)=0.0
      L=0
      K=0
      DO 370 NODE=1,NTT
      L=L+1
      DO 310 I=1,NNM
      IF(NODE-NPT(I))310,370,310
  310 CONTINUE
      K=K+1
      JDEF(L)=K
  370 CONTINUE
      DO 360 MM=1,NNM
      DO 281 K=1,3
      DO 281 I=1,6
  281 JNCODE(MM,(K-1)*6+I)=NOD(MM,I)+(K-1)*NTN
      DO 282 I=1,3
  282 JNCODE(MM,I+18)=NODCOR(MM,I)+3*NTN
      WRITE(6,111)(MM,JNCODE(MM,J),J=1,21)
  111 FORMAT(22I5)
      CALL EMATR(MM)
      DO 320 I=1,21
      JN=JNCODE(MM,I)
      Q(I)=0.0
      DO 330 J=1,NNM
      IF(JN.NE.NPT(J)) GO TO 330
      Q(I)=VAL(J)
  330 CONTINUE
  320 NCODE(MM,I)=JDEF(JN)
      DO 340 I=1,21
      F(I)=0.0
      DO 340 J=1,21
  340 F(I)=F(I)+STE(I,J)*Q(J)
      DO 350 I=1,21
      IF(NCODE(MM,I).EQ.0) GO TO 350
      K=NCODE(MM,I)
      RHS(K,1)=RHS(K,1)-F(I)
  350 CONTINUE
      DO 360 I=1,21
      IF(NCODE(MM,I).EQ.0) GO TO 360
      K=NCODE(MM,I)
      DO 360 J=1,21
      IF(NCODE(MM,J).EQ.0) GO TO 360
      L=NCODE(MM,J)
```

```
60          ST(K,L)=ST(K,L)+STE(I,J)
61      360 CONTINUE
62          WRITE(6,10)
63          WRITE(6,111)(I,(NCODE(I,J),J=1,21),I=1,NNN)
64       10 FORMAT(//,1X,'THE ELEMENTS AND THEIR',
65         $'NEW CODE NUMBERS',/)
66          RETURN
67          DEBUG SUBCHK
68          END
```

```
      SUBROUTINE EMATR(NEL)
      PARAMETER NTE=32,NTN=81,NVB=97,NNI=171
      COMMON R1,R2,R3,KP(6,6),UNM(6,6),ELM(21,21),RNOD(NTN),ZNOD(NTN),
     1NOD(NTE,6),NVELS(NVB),VBND(NVB),S(NNI,NNI),UVWP(NNI),NODS(6),
     2NCORS(3),NCOR(NTE,3),NODCOR(NTE,3),FAC,RE,IPRINT,UN,RO
      REAL KP
      N1=NOD(NEL,1)
      N2=NOD(NEL,2)
      N3=NOD(NEL,3)
      B1=ZNOD(N2)-ZNOD(N3)
      B2=ZNOD(N3)-ZNOD(N1)
      B3=ZNOD(N1)-ZNOD(N2)
      C1=RNOD(N3)-RNOD(N2)
      C2=RNOD(N1)-RNOD(N3)
      C3=RNOD(N2)-RNOD(N1)
      R1=RNOD(N1)
      R2=RNOD(N2)
      R3=RNOD(N3)
      A=0.5*(RNOD(N1)*(ZNOD(N2)-ZNOD(N3))+RNOD(N2)*
     1(ZNOD(N3)-ZNOD(N1))+RNOD(N3)*(ZNOD(N1)-ZNOD(N2)))
      A=ABS(A)
      KP(1,1)=UN*((B1*B1+FAC*FAC*C1*C1)/(20*A))*(3*R1+R2+R3)
      KP(1,2)=-UN*((B1*B2+FAC*FAC*C1*C2)/(60*A))*(2*R1+2*R2+R3)
      KP(1,3)=-UN*((B1*B3+FAC*FAC*C1*C3)/(60*A))*(2*R1+R2+2*R3)
      KP(1,4)=UN*((B1/(60*A))*(R1*(3*B1+14*B3)+R2*(-B1+3*B3)+R3*(-2*B1+
     13*B3))+(FAC*FAC*C1/(60*A))*(R1*(3*C1+14*C3)+R2*(-C1+3*C3)+
     2R3*(-2*C1+3*C3)))
      KP(1,5)=UN*((B1/(60*A))*(R1*(3*B1+14*B2)+R2*(-2*B1+3*B2)+R3*(-B1+
     13*B2))+(FAC*FAC*C1/(60*A))*(R1*(3*C1+14*C2)+R2*(-2*C1+3*C2)+
     2R3*(-C1+3*C2)))
      KP(1,6)=UN*((B1/(60*A))*(R1*3*(B2+B3)-R2*(B2+2*B3)-R3*(2*B2+B3))+
     1(FAC*FAC*C1/(60*A))*(R1*3*(C2+C3)-R2*(C2+2*C3)-R3*(2*C2+C3)))
      KP(2,2)=UN*((B2*B2+FAC*FAC*C2*C2)/(20*A))*(R1+3*R2+R3)
      KP(2,3)=-UN*((B2*B3+FAC*FAC*C2*C3)/(60*A))*(R1+2*R2+2*R3)
      KP(2,4)=UN*((B2/(60*A))*(-R1*(B1+2*B3)+R2*3*(B1+B3)-R3*(2*B1+B3))+
     5(FAC*FAC*C2/(60*A))*(-R1*(C1+2*C3)+R2*3*(C1+C3)-R3*(2*C1+C3)))
      KP(2,5)=UN*((B2/(60*A))*(R1*(3*B1-2*B2)+R2*(14*B1+3*B2)+
     1R3*(3*B1-B2))+(FAC*FAC*C2/(60*A))*(R1*(3*C1-2*C2)+R2*(14*C1+3*C2)+
     2R3*(3*C1-C2)))
      KP(2,6)=UN*((B2/(60*A))*(R1*(-B2+3*B3)+R2*(3*B2+14*B3)+
     1R3*(-2*B2+3*B3))+(FAC*FAC*C2/(60*A))*(R1*(-C2+3*C3)+
     2R2*(3*C2+14*C3)+R3*(-2*C2+3*C3)))
      KP(3,3)=UN*((B3*B3+FAC*FAC*C3*C3)/(20*A))*(R1+R2+3*R3)
      KP(3,4)=UN*((B3/(60*A))*(R1*(3*B1-2*B3)+R2*(3*B1-B3)+
     1R3*(14*B1+3*B3))+(FAC*FAC*C3/(60*A))*(R1*(3*C1-2*C3)+R2*(3*C1-C3)+
     2R3*(14*C1+3*C3)))
      KP(3,5)=UN*((B3/(60*A))*(-R1*(B1+2*B2)-R2*(2*B1+B2)+3*R3*(B1+B2))+
     1(FAC*FAC*C3/(60*A))*(-R1*(C1+2*C2)-R2*(2*C1+C2)+3*R3*(C1+C2)))
      KP(3,6)=UN*((B3/(60*A))*(R1*(3*B2-B3)+R2*(3*B2-2*B3)+
     1R3*(14*B2+3*B3))+(FAC*FAC*C3/(60*A))*(R1*(3*C2-C3)+R2*(3*C2-2*C3)+
     2R3*(14*C2+3*C3)))
      KP(4,4)=UN*(2/(15*A))*(R1*(B1*B1+2*B1*B3+3*B3*B3+FAC*FAC*(C1*C1+
     12*C1*C3+3*C3*C3))+R2*(B1*B1+B1*B3+B3*B3+FAC*FAC*(C1*C1+C1*C3+C3*C
     2))+R3*(3*B1*B1+2*B1*B3+B3*B3+FAC*FAC*(3*C1*C1+2*C1*C3+C3*C3)))
      KP(4,5)=UN*(1/(15*A))*(R1*(B1*B1+2*B1*B2+2*B1*B3+4*B2*B3+FAC*FAC*
     1(C1*C1+2*C1*C2+2*C1*C3+6*C2*C3))+R2*(2*B1*B1+B1*B2+2*B1*B3+2*B2*B
     2+FAC*FAC*(2*C1*C1+C1*C2+2*C1*C3+2*C2*C3))+R3*(2*B1*B1+2*B1*B2+B1*
     3B3+2*B2*B3+FAC*FAC*(2*C1*C1+2*C1*C2+C1*C3+2*C2*C3)))
      KP(4,6)=UN*(1/(15*A))*(R1*(2*B1*B2+B1*B3+2*B2*B3+2*B3*B3+FAC*FAC*
```

```
     1(2*C1*C2+C1*C3+2*C2*C3+2*C3*C3))+R2*(2*B1*B2+2*B1*B3+B2*B3+2*B3*B3
     2+FAC*FAC*(2*C1*C2+2*C1*C3+C2*C3+2*C3*C3))+R3*(6*B1*B2+2*B1*B3+2*B2
     3*B3+B3*B3+FAC*FAC*(6*C1*C2+2*C1*C3+2*C2*C3+C3*C3)))
      KP(5,5)=UN*(2/(15*A))*(R1*(B1*B1+2*B1*B2+3*B2*B2+FAC*FAC*(C1*C1+
     12*C1*C2+3*C2*C2))+R2*(3*B1*B1+2*B1*B2+B2*B2+FAC*FAC*(3*C1*C1+2*C1*
     3C2+C2*C2))+R3*(B1*B1+B1*B2+B2*B2+FAC*FAC*(C1*C1+C1*C2+C2*C2)))
      KP(5,6)=UN*(1/(15*A))*(R1*(B1*B2+2*B1*B3+2*B2*B2+2*B2*B3+FAC*FAC*
     1(C1*C2+2*C1*C3+2*C2*C2+2*C2*C3))+R2*(2*B1*B2+6*B1*B3+B2*B2+2*B2*B3
     3+FAC*FAC*(2*C1*C2+6*C1*C3+C2*C2+2*C2*C3))+R3*(2*B1*B2+2*B1*B3+2*B2
     4*B2+B2*B3+FAC*FAC*(2*C1*C2+2*C1*C3+2*C2*C2+C2*C3)))
      KP(6,6)=UN*(2/(15*A))*(R1*(B2*B2+B2*B3+B3*B3+FAC*FAC*(C2*C2+C2*C3+
     1C3*C3))+R2*(B2*B2+2*B2*B3+3*B3*B3+FAC*FAC*(C2*C2+2*C2*C3+3*C3*C3))
     2+R3*(3*B2*B2+2*B2*B3+B3*B3+FAC*FAC*(3*C2*C2+2*C2*C3+C3*C3)))
      IF(F22(.33,.33,.33).EQ.0.)GO TO 111
      IF(F22(0.,.5,.5).EQ.0.)GO TO 111
      IF(F22(.5,0.,.5).EQ.0.)GO TO 111
      IF(F22(.5,.5,0.).EQ.0.)GO TO 111
      IF(F22(1.,0.,0.).EQ.0.)GO TO 111
      IF(F22(0.,1.,0.).EQ.0.)GO TO 111
      IF(F22(0.,0.,1.).EQ.0.)GO TO 111
      GO TO 112
  111 WRITE(6,108)
  108 FORMAT('SINGULARITY')
      STOP
  112 UNM(1,1)=UN*(27*F1(.33,.33,.33)+8*(F1(.5,.5,0)+F1(0.,.5,.5)+F1(.5,0
     1.,.5))+3.*(F1(1.,0.,0.)+F1(0.,1.,0.)+F1(0.,0.,1.)))/60.
      UNM(1,2)=UN*(27*F2(.33,.33,.33)+8*(F2(.5,.5,0)+F2(0.,.5,.5)+F2(.5,0
     1.,.5))+3.*(F2(1.,0.,0.)+F2(0.,1.,0.)+F2(0.,0.,1.)))/60.
      UNM(1,3)=UN*(27*F3(.33,.33,.33)+8*(F3(.5,.5,0)+F3(0.,.5,.5)+F3(.5,0
     1.,.5))+3.*(F3(1.,0.,0.)+F3(0.,1.,0.)+F3(0.,0.,1.)))/60.
      UNM(1,4)=UN*(27*F4(.33,.33,.33)+8*(F4(.5,.5,0)+F4(0.,.5,.5)+F4(.5,0
     1.,.5))+3.*(F4(1.,0.,0.)+F4(0.,1.,0.)+F4(0.,0.,1.)))/60.
      UNM(1,5)=UN*(27*F5(.33,.33,.33)+8*(F5(.5,.5,0)+F5(0.,.5,.5)+F5(.5,0
     1.,.5))+3.*(F5(1.,0.,0.)+F5(0.,1.,0.)+F5(0.,0.,1.)))/60.
      UNM(1,6)=UN*(27*F6(.33,.33,.33)+8*(F6(.5,.5,0)+F6(0.,.5,.5)+F6(.5,0
     1.,.5))+3.*(F6(1.,0.,0.)+F6(0.,1.,0.)+F6(0.,0.,1.)))/60.
      UNM(2,2)=UN*(27*F7(.33,.33,.33)+8*(F7(.5,.5,0)+F7(0.,.5,.5)+F7(.5,0
     1.,.5))+3.*(F7(1.,0.,0.)+F7(0.,1.,0.)+F7(0.,0.,1.)))/60.
      UNM(2,3)=UN*(27*F8(.33,.33,.33)+8*(F8(.5,.5,0)+F8(0.,.5,.5)+F8(.5,0
     1.,.5))+3.*(F8(1.,0.,0.)+F8(0.,1.,0.)+F8(0.,0.,1.)))/60.
      UNM(2,4)=UN*(27*F9(.33,.33,.33)+8*(F9(.5,.5,0)+F9(0.,.5,.5)+F9(.5,0
     1.,.5))+3.*(F9(1.,0.,0.)+F9(0.,1.,0.)+F9(0.,0.,1.)))/60.
      UNM(2,5)=UN*(27*F10(.33,.33,.33)+8*(F10(.5,.5,0)+F10(0.,.5,.5)+F10(
     1.5,0.,.5))+3.*(F10(1.,0.,0.)+F10(0.,1.,0.)+F10(0.,0.,1.)))/60.
      UNM(2,6)=UN*(27*F11(.33,.33,.33)+8*(F11(.5,.5,0)+F11(0.,.5,.5)+F11(
     1.5,0.,.5))+3.*(F11(1.,0.,0.)+F11(0.,1.,0.)+F11(0.,0.,1.)))/60.
      UNM(3,3)=UN*(27*F12(.33,.33,.33)+8*(F12(.5,.5,0)+F12(0.,.5,.5)+F12(
     1.5,0.,.5))+3.*(F12(1.,0.,0.)+F12(0.,1.,0.)+F12(0.,0.,1.)))/60.
      UNM(3,4)=UN*(27*F13(.33,.33,.33)+8*(F13(.5,.5,0)+F13(0.,.5,.5)+F13(
     1.5,0.,.5))+3.*(F13(1.,0.,0.)+F13(0.,1.,0.)+F13(0.,0.,1.)))/60.
      UNM(3,5)=UN*(27*F14(.33,.33,.33)+8*(F14(.5,.5,0)+F14(0.,.5,.5)+F14(
     1.5,0.,.5))+3.*(F14(1.,0.,0.)+F14(0.,1.,0.)+F14(0.,0.,1.)))/60.
      UNM(3,6)=UN*(27*F15(.33,.33,.33)+8*(F15(.5,.5,0)+F15(0.,.5,.5)+F15(
     1.5,0.,.5))+3.*(F15(1.,0.,0.)+F15(0.,1.,0.)+F15(0.,0.,1.)))/60.
      UNM(4,4)=UN*(27*F16(.33,.33,.33)+8*(F16(.5,.5,0)+F16(0.,.5,.5)+F16(
     1.5,0.,.5))+3.*(F16(1.,0.,0.)+F16(0.,1.,0.)+F16(0.,0.,1.)))/60.
      UNM(4,5)=UN*(27*F17(.33,.33,.33)+8*(F17(.5,.5,0)+F17(0.,.5,.5)+F17(
     1.5,0.,.5))+3.*(F17(1.,0.,0.)+F17(0.,1.,0.)+F17(0.,0.,1.)))/60.
      UNM(4,6)=UN*(27*F18(.33,.33,.33)+8*(F18(.5,.5,0)+F18(0.,.5,.5)+F18(
     1.5,0.,.5))+3.*(F18(1.,0.,0.)+F18(0.,1.,0.)+F18(0.,0.,1.)))/60.
```

```
      UNM(5,5)=UN*(27*F19(.33,.33,.33)+8*(F19(.5,.5,0)+F19(0,.5,.5)+F19
     1.5,0.,.5))+3.*(F19(1.,0.,0.)+F19(0.,1.,0.)+F19(0.,0.,1.)))/60.
      UNM(5,6)=UN*(27*F20(.33,.33,.33)+8*(F20(.5,.5,0)+F20(0,.5,.5)+F20
     1.5,0.,.5))+3.*(F20(1.,0.,0.)+F20(0.,1.,0.)+F20(0.,0.,1.)))/60.
      UNM(6,6)=UN*(27*F21(.33,.33,.33)+8*(F21(.5,.5,0)+F21(0,.5,.5)+F21
     1.5,0.,.5))+3.*(F21(1.,0.,0.)+F21(0.,1.,0.)+F21(0.,0.,1.)))/60.
      DO 621 J=2,6
      DO 621 I=1,J-1
      KP(J,I)=KP(I,J)
  621 UNM(J,I)=UNM(I,J)
      DO 623 I=1,21
      DO 623 J=1,21
  623 ELM(I,J)=0.0
      DO 622 K=1,2
      DO 622 I=1,6
      DO 622 J=1,6
      ELM(I+(K-1)*6,J+(K-1)*6)=KP(I,J)+A*UNM(I,J)
  622 ELM(I+12,J+12)=KP(I,J)
      IF(IPRINT.EQ.1)WRITE(6,81)((KP(I,J),UNM(I,J),J=1,6),I=1,6)
   81 FORMAT(12F10.5)
      CALL PRCON(NEL)
      CALL CONVEC(NEL)
      RETURN
      DEBUG SUBCHK
      END
```

```
      SUBROUTINE PRCON(NEL)
      PARAMETER NTE=32,NTN=81,NVB=97,NNI=171
      COMMON R1,R2,R3,KP(6,6),UNM(6,6),ELM(21,21),RNOD(NTN),ZNOD(NTN),
     1NOD(NTE,6),NVELS(NVB),VBND(NVB),S(NNI,NNI),UVWP(NNI),NODS(6),
     2NCORS(3),NCOR(NTE,3),NODCOR(NTE,3),FAC,RE,IPRINT,UN,RO
      DIMENSION A4(6,3),A3(6,3),A5(3,6),A6(3,6)
      N1=NOD(NEL,1)
      N2=NOD(NEL,2)
      N3=NOD(NEL,3)
      B1=ZNOD(N2)-ZNOD(N3)
      B2=ZNOD(N3)-ZNOD(N1)
      B3=ZNOD(N1)-ZNOD(N2)
      C1=RNOD(N3)-RNOD(N2)
      C2=RNOD(N1)-RNOD(N3)
      C3=RNOD(N2)-RNOD(N1)
      R1=RNOD(N1)
      R2=RNOD(N2)
      R3=RNOD(N3)
      A=0.5*(RNOD(N1)*(ZNOD(N2)-ZNOD(N3))+RNOD(N2)*
     1(ZNOD(N3)-ZNOD(N1))+RNOD(N3)*(ZNOD(N1)-ZNOD(N2)))
      A=ABS(A)
      A4(1,1)=B1*(2*R1-R2-R3)/(120*RO)
      A4(1,2)=B2*(2*R1-R2-R3)/(120*RO)
      A4(1,3)=B3*(2*R1-R2-R3)/(120*RO)
      A4(2,1)=B1*(-R1+2*R2-R3)/(120*RO)
      A4(2,2)=B2*(-R1+2*R2-R3)/(120*RO)
      A4(2,3)=B3*(-R1+2*R2-R3)/(120*RO)
      A4(3,1)=B1*(-R1-R2+2*R3)/(120*RO)
      A4(3,2)=B2*(-R1-R2+2*R3)/(120*RO)
      A4(3,3)=B3*(-R1-R2+2*R3)/(120*RO)
      A4(4,1)=B1*(2*R1+R2+2*R3)/(30*RO)
      A4(4,2)=B2*(2*R1+R2+2*R3)/(30*RO)
      A4(4,3)=B3*(2*R1+R2+2*R3)/(30*RO)
      A4(5,1)=B1*(2*R1+2*R2+R3)/(30*RO)
      A4(5,2)=B2*(2*R1+2*R2+R3)/(30*RO)
      A4(5,3)=B3*(2*R1+2*R2+R3)/(30*RO)
      A4(6,1)=B1*(R1+2*R2+2*R3)/(30*RO)
      A4(6,2)=B2*(R1+2*R2+2*R3)/(30*RO)
      A4(6,3)=B3*(R1+2*R2+2*R3)/(30*RO)
      A3(1,1)=C1*(2*R1-R2-R3)/(120*RO)
      A3(1,2)=C2*(2*R1-R2-R3)/(120*RO)
      A3(1,3)=C3*(2*R1-R2-R3)/(120*RO)
      A3(2,1)=C1*(-R1+2*R2-R3)/(120*RO)
      A3(2,2)=C2*(-R1+2*R2-R3)/(120*RO)
      A3(2,3)=C3*(-R1+2*R2-R3)/(120*RO)
      A3(3,1)=C1*(-R1-R2+2*R3)/(120*RO)
      A3(3,2)=C2*(-R1-R2+2*R3)/(120*RO)
      A3(3,3)=C3*(-R1-R2+2*R3)/(120*RO)
      A3(4,1)=C1*(2*R1+R2+2*R3)/(30*RO)
      A3(4,2)=C2*(2*R1+R2+2*R3)/(30*RO)
      A3(4,3)=C3*(2*R1+R2+2*R3)/(30*RO)
      A3(5,1)=C1*(2*R1+2*R2+R3)/(30*RO)
      A3(5,2)=C2*(2*R1+2*R2+R3)/(30*RO)
      A3(5,3)=C3*(2*R1+2*R2+R3)/(30*RO)
      A3(6,1)=C1*(R1+2*R2+2*R3)/(30*RO)
      A3(6,2)=C2*(R1+2*R2+2*R3)/(30*RO)
      A3(6,3)=C3*(R1+2*R2+2*R3)/(30*RO)
      DO 818 I=1,6
      DO 818 J=1,3
```

```
      818 A3(I,J)=FAC*FAC*A3(I,J)
          A5(1,1)=B1*((14*R1+3*R2+3*R3)/120.)+A/30.
          A5(1,2)=B2*(-2*R1+3*R2-R3)/120-A/60
          A5(1,3)=B3*(-2*R1-R2+3*R3)/120-A/60
          A5(1,4)=(R1*(2*B1+6*B3)+R2*(B1+2*B3)+R3*(2*B1+2*B3))/30+2.*A/15
          A5(1,5)=(R1*(2*B1+6*B2)+R2*(2*B1+2*B2)+R3*(B1+2*B2))/30+2.*A/15
          A5(1,6)=(R1*(2*B2+2*B3)+R2*(B2+2*B3)+R3*(2*B2+B3))/30+A/15
          A5(2,1)=B1*(3*R1-2*R2-R3)/120-A/60.
          A5(2,2)=B2*(3*R1+14*R2+3*R3)/120+A/30
          A5(2,3)=B3*(-R1-2*R2+3*R3)/120-A/60
          A5(2,4)=(R1*(B1+2*B3)+R2*(2*B1+2*B3)+R3*(2*B1+B3))/30+A/15
          A5(2,5)=(R1*(2*B1+2*B2)+R2*(6*B1+2*B2)+R3*(2*B1+B2))/30+2.*A/15
          A5(2,6)=(R1*(B2+2*B3)+R2*(2*B2+6*B3)+R3*(2*B2+2*B3))/30+2.*A/15
          A5(3,1)=B1*(3*R1-R2-2*R3)/120-A/60
          A5(3,2)=B2*(-R1+3*R2-2*R3)/120-A/60
          A5(3,3)=B3*((3*R1+3*R2+14*R3)/120)+A/30
          A5(3,4)=(R1*(2*B1+2*B3)+R2*(2*B1+B3)+R3*(6*B1+2*B3))/30+2.*A/15
          A5(3,5)=(R1*(B1+2*B2)+R2*(2*B1+B2)+R3*(2*B1+2*B2))/30+A/15
          A5(3,6)=(R1*(2*B2+B3)+R2*(2*B2+2*B3)+R3*(6*B2+2*B3))/30+2.*A/15
          A6(1,1)=C1*(14*R1+3*R2+3*R3)/120.
          A6(1,2)=C2*(-2*R1+3*R2-R3)/120.
          A6(1,3)=C3*(-2*R1-R2+3*R3)/120.
          A6(1,4)=(R1*(2*C1+6*C3)+R2*(C1+2*C3)+R3*(2*C1+2*C3))/30
          A6(1,5)=(R1*(2*C1+6*C2)+R2*(2*C1+2*C2)+R3*(C1+2*C2))/30
          A6(1,6)=(R1*(2*C2+2*C3)+R2*(C2+2*C3)+R3*(2*C2+C3))/30
          A6(2,1)=C1*(3*R1-2*R2-R3)/120-0.
          A6(2,2)=C2*(3*R1+14*R2+3*R3)/120.
          A6(2,3)=C3*(-R1-2*R2+3*R3)/120
          A6(2,4)=(R1*(C1+2*C3)+R2*(2*C1+2*C3)+R3*(2*C1+C3))/30
          A6(2,5)=(R1*(2*C1+2*C2)+R2*(6*C1+2*C2)+R3*(2*C1+C2))/30
          A6(2,6)=(R1*(C2+2*C3)+R2*(2*C2+6*C3)+R3*(2*C2+2*C3))/30
          A6(3,1)=C1*(3*R1-R2-2*R3)/120
          A6(3,2)=C2*(-R1+3*R2-2*R3)/120.
          A6(3,3)=C3*(3*R1+3*R2+14*R3)/120.
          A6(3,4)=(R1*(2*C1+2*C3)+R2*(2*C1+C3)+R3*(6*C1+2*C3))/30
          A6(3,5)=(R1*(C1+2*C2)+R2*(2*C1+C2)+R3*(2*C1+2*C2))/30
          A6(3,6)=(R1*(2*C2+C3)+R2*(2*C2+2*C3)+R3*(6*C2+2*C3))/30
          DO 89 I=1,6
          DO 89 J=1,3
          A4(I,J)=-A5(J,I)
       89 A3(I,J)=-A6(J,I)
          DO 88 I=1,6
          DO 88 J=1,3
          ELM(I,J+18)=A4(I,J)
          ELM(I+12,J+18)=A3(I,J)
          ELM(J+18,I)=A5(J,I)
       88 ELM(J+18,I+12)=A6(J,I)
          IF(IPRINT.EQ.1)WRITE(6,81)((A4(I,J),A3(I,J),A5(J,I),A6(J,I),
         1J=1,3),I=1,6)
       81 FORMAT(12F10.5)
          RETURN
          DEBUG SUBCHK
          END
```

```
      SUBROUTINE CONVEC(NEL)
      PARAMETER NTE=32,NTN=81,NVB=97,NNI=171,NBN=32
      COMMON R1,R2,R3,KP(6,6),UNM(6,6),ELM(21,21),RNOD(NTN),ZNOD(NTN),
     1NOD(NTE,6),NVELS(NVB),VBND(NVB),S(NNI,NNI),UVWP(NNI),NODS(6),
     2NCORS(3),NCOR(NTE,3),NODCOR(NTE,3),FAC,RE,IPRINT,UN,RO
      COMMON/IN/OM,ERR,IBEGIN,ITR,IDISC,NOC
      COMMON/CON/US(NTE,3),VS(NTE,3),WS(NTE,3)
      DIMENSION C(6,6),A2(6,6)
      IF(ITR.EQ.1)GO TO 1
      GO TO 3
    1 IF(IBEGIN.EQ.0)GO TO 2
      GO TO 5
    2 DO 4 I=1,3
      US(NEL,I)=0.0
      VS(NEL,I)=0.0
      WS(NEL,I)=0.0
      DO 423 J=1,NBN
  423 IF(NVELS(J).EQ.NOD(NEL,I))GO TO 424
      GO TO 4
  424 WRITE(6,87)J
   87 FORMAT(I5)
      US(NEL,I)=VBND(J)
      VS(NEL,I)=VBND(J+NBN)
      WS(NEL,I)=VBND(J+2*NBN)
    4 CONTINUE
      GO TO 3
    5 READ(8,331)(US(NEL,I),VS(NEL,I),WS(NEL,I),I=1,3)
  331 FORMAT(3E15.4)
    3 CONTINUE
      WRITE(6,10)(US(NEL,I),VS(NEL,I),WS(NEL,I),I=1,3)
   10 FORMAT(3F10.5)
      U1=US(NEL,1)
      U2=US(NEL,2)
      U3=US(NEL,3)
      V1=VS(NEL,1)
      V2=VS(NEL,2)
      V3=VS(NEL,3)
      W1=WS(NEL,1)
      W2=WS(NEL,2)
      W3=WS(NEL,3)
      N1=NOD(NEL,1)
      N2=NOD(NEL,2)
      N3=NOD(NEL,3)
      B1=ZNOD(N2)-ZNOD(N3)
      B2=ZNOD(N3)-ZNOD(N1)
      B3=ZNOD(N1)-ZNOD(N2)
      C1=RNOD(N3)-RNOD(N2)
      C2=RNOD(N1)-RNOD(N3)
      C3=RNOD(N2)-RNOD(N1)
      R1=RNOD(N1)
      R2=RNOD(N2)
      R3=RNOD(N3)
      A=0.5*(RNOD(N1)*(ZNOD(N2)-ZNOD(N3))+RNOD(N2)*
     1(ZNOD(N3)-ZNOD(N1))+RNOD(N3)*(ZNOD(N1)-ZNOD(N2)))
      A=ABS(A)
      A2(1,1)=-A*(5*V1+V2+V3)/210
      A2(1,2)=A*(4*V1+4*V2-V3)/1260
      A2(1,3)=A*(4*V1-V2+4*V3)/1260
      A2(1,4)=-A*(3*V1-V2-2*V3)/315
```

```
A2(1,5)=-A*(3*V1-2*V2-V3)/315
A2(1,6)=A*(V1+3*V2+3*V3)/315
A2(2,2)=-A*(V1+5*V2+V3)/210
A2(2,3)=A*(-V1+4*V2+4*V3)/1260
A2(2,4)=A*(3*V1+V2+3*V3)/315
A2(2,5)=A*(2*V1-3*V2+V3)/315
A2(2,6)=A*(V1-3*V2+2*V3)/315
A2(3,3)=-A*(V1+V2+5*V3)/210
A2(3,4)=A*(2*V1+V2-3*V3)/315
A2(3,5)=A*(3*V1+3*V2+V3)/315
A2(3,6)=A*(V1+2*V2-3*V3)/315
A2(4,4)=-(8*A/315)*(3*V1+V2+3*V3)
A2(4,5)=-(4*A/315)*(3*V1+2*V2+2*V3)
A2(4,6)=-(4*A/315)*(2*V1+2*V2+3*V3)
A2(5,5)=-(8*A/315)*(3*V1+3*V2+V3)
A2(5,6)=-(4*A/315)*(2*V1+3*V2+2*V3)
A2(6,6)=-(8*A/315)*(V1+3*V2+3*V3)
C(1,1)=(B1/840)*(R1*(34*U1+4*U2+4*U3)+R2*(4*U1+2*U2+U3)+
1R3*(4*U1+U2+2*U3))+(C1/840)*(R1*(34*W1+4*W2+4*W3)+R2*(4*W1+2*W2+
2W3)+R3*(4*W1+W2+2*W3))
C(1,2)=-(B2/2520)*(R1*(30*U1+8*U2+4*U3)+R2*(8*U1+22*U2+5*U3)+R3*
1(4*U1+5*U2-2*U3))-(C2/2520)*(R1*(30*W1+8*W2+4*W3)+R2*(8*W1+22*W2+
25*W3)+R3*(4*W1+5*W2-2*W3)
C(1,3)=-(B3/2520)*(R1*(30*U1+4*U2+8*U3)+R2*(4*U1-2*U2+5*U3)+
2R3*(8*U1+5*U2+22*U3))-(C3/2520)*(R1*(30*W1+4*W2+8*W3)+R3*(4*W1-2*
2W2+5*W3)+R3*(8*W1+5*W2+22*W3))
C(1,4)=(R1*(U1*(3*B1+36*B3)+U2*(-B1+3*B3)+U3*(-2*B1+3*B3)+W1*(3*C1
1+36*C3)+W2*(-C1+3*C3)+W3*(-2*C1+3*C3))+R2*(U1*(-B1+3*B3)+U2*(-3*B1
2-2*B3)+U3*(-3*B1-B3)+W1*(-C1+3*C3)+W2*(-3*C1-2*C3)+W3*(-3*C1-C3))+
3R3*(U1*(-2*B1+3*B3)+U2*(-3*B1-B3)+U3*(-9*B1-2*B3)+W1*(-2*C1+3*C3)+
4W2*(-3*C1-C3)+W3*(-9*C1-2*C3)))/630
C(1,5)=(R1*(U1*(3*B1+36*B2)+U2*(-2*B1+3*B2)+U3*(-B1+3*B2)+W1*(3*C1
1+36*C2)+W2*(-2*C1+3*C2)+W3*(-C1+3*C2))+R2*(U1*(-2*B1+3*B2)+U2*(-9*
2B1-2*B2)+U3*(-3*B1-B2)+W1*(-2*C1+3*C2)+W2*(-9*C1-2*C2)+W3*(-3*C1-
3C2))+R3*(U1*(-B1+3*B2)+U2*(-3*B1-B2)+U3*(-3*B1-2*B2)+W1*(-C1+3*C2)
4+W2*(-3*C1-C2)+W3*(-3*C1-2*C2)))/630
C(1,6)=(R1*(U1*3*(B2+B3)+U2*(-B2-2*B3)+U3*(-2*B2-B3)+W1*3*(C2+C3)+
1W2*(-C2-2*C3)+W3*(-2*C2-C3))+R2*(U1*(-B2-2*B3)+U2*(-3*B2-9*B3)+U3*
2(-3*B2-3*B3)+W1*(-C2-2*C3)+W2*(-3*C2-
59*C3)+W3*(-3*C2-3*C3))+R3*(U1
3*(-2*B2-B3)+U2*(-3*B2-3*B3)+U3*(-9*B2-3*B3)+W1*(-2*C2-C3)+W2*(-3*
4C2-3*C3)+W3*(-9*C2-3*C3)))/630
C(2,1)=-(B1/2520)*(R1*(22*U1+8*U2+5*U3)+R2*(8*U1+30*U2+4*U3)+R3*(5
1*U1+4*U2-2*U3))-(C1/2520)*(R1*(22*W1+8*W2+5*W3)+R2*(8*W1+30*W2+4*
2W3)+R3*(5*W1+4*W2-2*W3))
C(2,2)=(B2/840)*(R1*(2*U1+4*U2+U3)+R2*(4*U1+34*U2+4*U3)+R3*(U1+4*
1U2+2*U3))+(C2/840)*(R1*(2*W1+4*W2+W3)+R2*(4*W1+34*W2+4*W3)+R3*(W1+
24*W2+2*W3))
C(2,3)=-(B1/2520)*(R1*(-2*U1+4*U2+5*U3)+R2*(4*U1+30*U2+8*U3)+R3*(5
1*U1+8*U2+22*U3))-(C3/2520)*(R1*(-2*W1+4*W2+5*W3)+R2*(4*W1+30*W2+8*
2W3)+R3*(5*W1+8*W2+22*W3))
C(2,4)=(R1*(U1*(-3*B1-9*B3)+U2*(-B1-2*B3)+U3*(-3*B1-3*B3)+W1*(-3*
1C1-9*C3)+W2*(-C1-2*C3)+W3*(-3*C1-3*C3))+R2*
5(U1*(-B1-2*B3)+U2*(3*B1
2+3*B3)+U3*(-2*B1-B3)+W1*(-C1-2*C3)+W2*(3*C1+3*C3)+W3*(-2*C1-C3))+
3R2*(U1*(-3*B1-3*B3)+U2*(-2*B1-B3)+U3*(-9*B1-3*B3)+W1*(-3*C1-3*C3)+
4W2*(-2*C1-C3)+W3*(-9*C1-3*C3)))/630
C(2,5)=(R1*(U1*(-2*B1-9*B2)+U2*(3*B1-2*B2)+U3*(-B1-3*B2)+W1*(-2*C1
1-9*C2)+W2*(3*C1-2*C2)+W3*(-C1-3*C2))+R2*(U1*(3*B1-2*B2)+U2*(36*B1+
23*B2)+U3*(3*B1-B2)+W1*(3*C1-2*C2)+W2*(36*C1+3*C2)+W3*(3*C1-C2))+R3
```

```
3*(U1*(-B1-3*B2)+U2*(3*B1-B2)+U3*(-2*B1-3*B2)+W1*(-C1-3*C2)+W2*(3*
4C1-C2)+W3*(-2*C1-3*C2)))/630
 C(2,6)=(R1*(U1*(-3*B2-2*B3)+U2*(-B2+3*B3)+U3*(-3*B2-B3)+W1*(-3*C2-
12*C3)+W2*(-C2+3*C3)+W3*(-3*C2-C3))+R2*(U1*(-B2+3*B3)+U2*(3*B2+36*
2B3)+U3*(-2*B2+3*B3)+W1*(-C2+3*C3)+W2*(3*C2+36*C3)+W3*(-2*C2+3*C3))
3+R3*(U1*(-3*B2-B3)+U2*(-2*B2+3*B3)+U3*(-9*B2-2*B3)+W1*(-3*C2-C3)+
4W2*(-2*C2+3*C3)+W3*(-9*C2-2*C3)))/630
 C(3,1)=-(B1/2520)*(R1*(22*U1+5*U2+8*U3)+R2*(5*U1-2*U2+4*U3)+R3*(8
1*U1+4*U2+30*U3))-(C1/2520)*(R1*(22*W1+5*W2+8*W3)+R2*(5*W1-2*W2+4*W
23)+R3*(8*W1+4*W2+30*W3))
 C(3,2)=-(B2/2520)*(R1*(-2*U1+5*U2+4*U3)+R2*(5*U1+22*U2+8*U3)+R3*(4
1*U1+8*U2+30*U3))-(C2/2520)*(R1*(-2*W1+5*W2+4*W3)+R2*(5*W1+22*W2+8*
2W3)+R3*(4*W1+8*W2+30*W3))
 C(3,3)=(B3/840)*(R1*(2*U1+U2+4*U3)+R2*(U1+
52*U2+4*U3)+R3*(4*U1+4*U2+34*U3
1))+(C3/840)*(R1*(2*W1+W2+4*W3)+R2*(W1+2*W2+4*W3)+R3*(4*W1+4*W2+34*
2W3))
 C(3,4)=(R1*(U1*(-2*B1-9*B3)+U2*(-B1-3*B3)+U3*(3*B1-2*B3)+W1*(-2*C1
1-9*C3)+W2*(-C1-3*C3)+W3*(3*C1-2*C3))+R2*(U1*(-B1-3*B3)+U2*(-2*B1-3
2*B3)+U3*(3*B1-B3)+W1*(-C1-3*C3)+W2*
5(-2*C1-3*C3)+W3*(3*C1-C3))+R3*(
3U1*(3*B1-2*B3)+U2*(3*B1-B3)+U3*(36*B1+3*B3)+W1*(3*C1-2*C3)+W2*(3*C
41-C3)+W3*(36*C1-3*C3)))/630
 C(3,5)=(R1*(U1*(-3*B1-9*B2)+U2*(-3*B1-3*B2)+U3*
5(-B1-2*B2)+W1*(-3*C1-
19*C2)+W2*(-3*C1-3*C2)+W3*(-C1-2*C2))+R2*(U1*(-3*B1-3*B2)+U2*(-9*B1
2-3*B2)+U3*(-2*B1-B2)+W1*(-3*C1-3*C2)+W2*(-9*C1-3*C2)+W3*(-C1-2*C2
3)+R3*(U1*(-B1-2*B2)+U2*(-2*B1-B2)+U3*(3*B1+3*B2)+W1*(-C1-2*C2)+W2*
4(-2*C1-C2)+W3*(3*C1+3*C2)))/630
 C(3,6)=(R1*(U1*(-2*B2-3*B3)+U2*(-B2-3*B3)+
5U3*(3*B2-B3)+W1*(-2*C2-3*C
13)+W2*(-C2-3*C3)+W3*(3*C2-C3))+R2*(U1*(-3*B3-B2)+U2*(-2*B2-9*B3)+
23*(-2*B3+3*B2)+W1*(-B2-3*B3)+W2*(-2*C2-9*C3)+
5W3*(3*B2-2*B3))+R3*(U1*
3(3*B2-B3)+U2*(3*B2-2*B3)+U3*(36*B2+3*B3)+W1*(3*C2-C3)+W2*(3*C2-2*C
43)+W3*(36*C2+3*C3)))/630
 C(4,1)=B1*(R1*(27*U1+5*U2+10*U3)+R2*(5*U1+U2+U3)+R3*(10*U1+U2+3*U3
1))/630+C1*(R1*(27*W1+5*W2+10*W3)+R2*(5*W1+W2+W3)+R3*(10*W1+W2+3*W3
2))/630
 C(4,2)=B2*(R1*(-9*U1+U2-6*U3)+R2*(U1+5*U2+U3)+R3*(-6*U1+U2-9*U3))/
1630+C2*(R1*(-9*W1+W2-6*W3)+R2*(W1+5*W2+W3)+R3*(-6*W1+W2-9*W3))/630
 C(4,3)=B3*(R1*(3*U1+U2+10*U3)+R2*(U1+U2+5*U3)+R3*(10*U1+5*U2+27*U3
1))/630+C3*(R1*(3*W1+W2+10*W3)+R2*(W1+W2+5*W3)+R3*(10*W1+5*W2+27*W3
2))/630
 C(4,4)=2*(R1*(U1*(B1+12*B3)+U2*(2*B1+3*B3)+U3*(6*B1+6*B3)+W1*(6*C
1+12*C3)+W2*(2*C1+3*C3)+W3*(6*C1+6*C3))+R2*(U1*(2*B1+3*B3)+U2*(2*B
2+2*B3)+U3*(3*B1+2*B3)+W1*(2*C1+3*C3)+W2*(2*C1+2*C3)+W3*(3*C1+2*C3
3)+R3*(U1*(6*B1+6*B3)+U2*(3*B1+2*B3)+U3*(12*B1+6*B3)+W1*(6*C1+6*C3
4)+W2*(3*C1+2*C3)+W3*(12*C1+6*C3)))/315
 C(4,5)=2*(R1*(U1*(3*B1+12*B2)+U2*(2*B1+3*B2)+U3*(2*B1+6*B2)+W1*(3*
1C1+12*C2)+W2*(2*C1+3*C2)+W3*(2*C1+6*C2))+R2*(U1*(2*B1+3*B2)+U2*(3*
2B1+2*B2)+U3*(2*B1+2*B2)+W1*(2*C1+3*C2)+W2*(3*C1+2*C2)+W3*(2*C1+2*
32))+R3*(U1*(2*B1+6*B2)+U2*(2*B1+2*B2)+U3*(3*B1+6*B2)+W1*(2*C1+6*C
4)+W2*(2*C1+2*C2)+W3*(3*C1+6*C2)))/315
 C(4,6)=2*(R1*(U1*(6*B2+3*B3)+U2*(2*B2+
52*B3)+U3*(6*B2+2*B3)+W1*(6*C2+
13*C3)+W2*(2*C2+2*C3)+W3*(6*C2+2*C3))+R2*(U1*(2*B2+2*B3)+U2*(2*B2+
2*B3)+U3*(3*B2+2*B3)+W1*(2*C2+2*C3)+W2*(2*C2+3*C3)+W3*(3*C2+2*C3))+
3R3*(U1*(6*B2+2*B3)+U2*(3*B2+2*B3)+U3*(12*B2+3*B3)+W1*(6*C2+2*C3)+
42*(3*C2+2*C3)+W3*(12*C2+3*C3)))/315
```

```
      C(5,1)=B1*(R1*(27*U1+10*U2+5*U3)+R2*(10*U1+3*U2+U3)+R3*(5*U1+U2+U3
     1))/630+C1*(R1*(27*W1+10*W2+5*W3)+R2*(10*W1+3*W2+W3)+R3*(5*W1+W2+W3
     2))/630
      C(5,2)=B2*(R1*(3*U1+10*U2+U3)+R2*(10*U1+27*U2+5*U3)+R3*(U1+5*U2+U3
     1))/630+C2*(R1*(3*W1+10*W2+W3)+R2*(10*W1+27*W2+5*W3)+R3*(W1+5*W2+W3
     2))/630
      C(5,3)=B3*(R1*(-9*U1-6*U2+U3)+R2*(-6*U1-9*U2+U3)+R3*(U1+U2+5*U3))/
     1630+C3*(R1*(-9*W1-6*W2+W3)+R2*(-6*W1-9*W2+W3)+R3*(W1+W2+5*W3))/630
      C(5,4)=2*(R1*(U1*(3*B1+12*B3)+U2*
     5(2*B1+6*B3)+U3*(2*B1+3*B3)+W1*(3*
     1C1+12*C3)+W2*(2*C1+6*C3)+W3*(2*C1+3*C3))+R2*
     5(U1*(2*B1+6*B3)+U2*(3*
     2B1+6*B3)+U3*(2*B1+2*B3)+W1*(2*C1+6*C3)+W2*(3*C1+6*C3)+W3*(2*C1+2*C
     33))+R3*(U1*(2*B1+3*B3)+U2*(2*B1+2*B3)+U3*(3*B1+2*B3)+W1*(2*C1+3*C3
     4)+W2*(2*C1+2*C3)+W3*(3*C1+2*C3)))/315
      C(5,5)=2*(R1*(U1*(6*B1+12*B2)+U2*(6*B1+6*B2)+U3*(2*B1+3*B2)+W1*(6*
     1C1+12*C2)+W2*(6*C1+6*C2)+W3*(2*C1+3*C2))+R2*(U1*(6*B1+6*B2)+U2*(12
     2*B1+6*B2)+U3*(3*B1+2*B2)+W1*(6*C1+6*C2)+W2*(12*C1+6*C2)+W3*(3*C1+2
     3*C2))+R3*(U1*(2*B1+3*B2)+U2*(3*B1+2*B2)+U3*(2*B1+2*B2)+W1*(2*C1+3*
     4C2)+W2*(3*C1+2*C2)+W3*(2*C1+2*C2)))/315
      C(5,6)=2*(R1*(U1*(3*B2+6*B3)+U2*(2*B2+6*B3)+U3*(2*B2+2*B3)+W1*(3*C
     12+6*C3)+W2*(2*C2+6*C3)+W3*(2*C2+2*C3))+R2*(U1*(2*B2+6*B3)+U2*(3*B2
     2+12*B3)+U3*(2*B2+3*B3)+W1*(2*C2+6*C3)+W2*(3*C2+12*C3)+W3*(2*C2+3*C
     33))+R3*(U1*(2*B2+2*B3)+U2*(2*B2+3*B3)+U3*(3*B2+2*B3)+W1*(2*C2+2*C3
     4)+W2*(2*C2+3*C3)+W3*(3*C2+2*C3)))/315
      C(6,1)=B1*(R1*(5*U1+U2+U3)+R2*(U1-9*U2-6*U3)+R3*(U1-6*U2-9*U3))/
     1630.+C1*(R1*(5*W1+W2+W3)+R2*(W1-9*W2-6*W3)+R3*(W1-6*W2-9*W3))/630.
      C(6,2)=B2*(R1*(U1+5*U2+U3)+R2*(5*U1+27*U2+10*U3)+R3*(U1+10*U2+3*U3
     1))/630.+C2*(R1*(W1+5*W2+W3)+R2*(5*W1+27*W2+10*W3)+R3*(W1+10*W2+3*W3
     23))/630.
      C(6,3)=B3*(R1*(U1+U2+5*U3)+R2*(U1+3*U2+10*U3)+R3*(5*U1+10*U2+27*U3
     1))/630.+C3*(R1*(W1+W2+5*W3)+R2*(W1+3*W2+10*W3)+R3*(5*W1+10*W2+27*W3
     23))/630.
      C(6,4)=2*(R1*(U1*(2*B1+3*B3)+U2*(2*B1+2*B3)+U3*(3*B1+2*B3)+W1*(2*C
     11+3*C3)+W2*(2*C1+2*C3)+W3*(3*C1+2*C3))+R2*(U1*(2*B1+2*B3)+U2*(6*B1
     2+3*B3)+U3*(6*B1+2*B3)+W1*(2*C1+2*C3)+W2*(6*C1+3*C3)+W3*(6*C1+2*C3
     3)+R3*(U1*(3*B1+2*B3)+U2*(6*B1+2*B3)+U3*(12*B1+3*B3)+W1*(3*C1+2*C3
     4+W2*(6*C1+2*C3)+W3*(12*C1+3*C3)))/315
      C(6,5)=2*(R1*(U1*(3*B2+2*B1)+U2*(2*B2+3*B1)+U3*(2*B2+2*B1)+W1*(3*C
     12+2*C1)+W2*(2*C2+3*C1)+W3*(2*C2+2*C1))+R2*(U1*(2*B2+3*B1)+U2*(3*B2
     2+12*B1)+U3*(2*B2+6*B1)+W1*(2*C2+3*C1)+W2*(3*C2+12*C1)+W3*(2*C2+6*C
     31))+R3*(U1*(2*B2+2*B1)+U2*(2*B2+6*B1)+U3*(3*B2+6*B1)+W1*(2*C2+2*C3
     4)+W2*(2*C2+6*C1)+W3*(3*C2+6*C1)))/315
      C(6,6)=2*(R1*(U1*(2*B3+2*B2)+U2*(3*B3+2*B2)+U3*(2*B3+3*B2)+W1*(2*C
     53
     1+2*C2)+W2*(3*C3+2*C2)+W3*(2*C3+3*C2))+R2*(U1*(3*B3+2*B2)+U2*(12*B3
     2+6*B2)+U3*(6*B3+6*B2)+W1*(3*C3+2*C2)+W2*(12*C3+6*C2)+W3*(6*C3+6*C3
     3))+R3*(U1*(2*B3+3*B2)+U2*(6*B3+6*B2)+U3*(6*B3+12*B2)+W1*(2*C3+3*C3
     4)+W2*(6*C3+6*C2)+W3*(6*C3+12*C2)))/315
      DO 9 J=2,6
      DO 9 I=1,J-1
    9 A2(J,I)=A2(I,J)
      DO 90 I=1,6
      DO 90 J=1,6
      ELM(I,J+6)=A2(I,J)
      ELM(I+6,J)=-A2(I,J)
      DO 90 K=1,3
   90 ELM(I+(K-1)*6,J+(K-1)*6)=ELM(I+(K-1)*6,J+(K-1)*6)+C(I,J)
      IF(IPRINT.EQ.1)WRITE(6,81)((A2(I,J),C(I,J),J=1,6),I=1,6)
   81 FORMAT(12F10.5)

      RETURN
      END
```

```
      SUBROUTINE ITER
      PARAMETER NTE=32,NTN=81,NVB=97,NTU=171,NNI=49,NNIP=24,NPB=9
      COMMON R1,R2,R3,KP(6,6),UNM(6,6),ELM(21,21),RNOD(NTN),ZNOD(NTN),
     1NOD(NTE,6),NVELS(NVB),VBND(NVB),S(NTU,NTU),UVWP(NTU),NODS(6),
     2NCORS(3),NCOR(NTE,3),NODCOR(NTE,3),FAC,RE,IPRINT,UN,RO
      COMMON/IN/OM,ERR,IBEGIN,ITR,IDISK,NOC,NER,NEZ
      COMMON/CON/US(NTE,3),VS(NTE,3),WS(NTE,3),PBND(NPB)
      COMMON/INP/IPR,IUS,IVS,IWS,IPS
      DIMENSION U(NNI),V(NNI),W(NNI),P(NNI),UP(NNI),VP(NNI),WP(NNI),
     1PSI(NNI),INT(NNI)
      NET=NTE
      IF(ITR.EQ.1)GO TO 1
      GO TO 3
    1 IF(IBEGIN.EQ.0)GO TO 2
      GO TO 5
    2 DO 4 I=1,NNI
      UP(I)=0.0
      VP(I)=0.0
    4 WP(I)=0.0
      GO TO 3
    5 READ(8,330)(UP(I),VP(I),WP(I),I=1,NNI)
      REWIND 8
    3 CONTINUE
      WRITE(6,10)(UP(I),VP(I),WP(I),I=1,NNI)
   10 FORMAT(3F10.5)
      DO 202 I=1,NNI
      DO 203 I=1,NNIP
      WRITE(6,2223)(U(I),I=1,NNI)
      WRITE(6,2223)(V(I),I=1,NNI)
      WRITE(6,2223)(W(I),I=1,NNI)
      WRITE(6,2223)(P(I),I=1,NNIP)
 2223 FORMAT(12F10.5)
      IU=1
      IV=1
      IW=1
      DO 433 I=1,NNI
      IF(U(I).EQ.0.)GO TO 81
      GO TO 86
   81 IF(UP(I).EQ.0.)GO TO 82
      GO TO 87
   82 IF(V(I).EQ.0.)GO TO 83
      GO TO 88
   83 IF(VP(I).EQ.0.)GO TO 84
      GO TO 89
   84 IF(W(I).EQ.0.)GO TO 85
      GO TO 90
   85 IF(WP(I).EQ.0.)GO TO 433
      GO TO 91
   86 IF(ABS((U(I)-UP(I))/U(I)).LE.ERR)IU=IU+1
      GO TO 82
   87 IF(ABS((U(I)-UP(I))/UP(I)).LE.ERR)IU=IU+1
      GO TO 82
   88 IF(ABS((V(I)-VP(I))/V(I)).LE.ERR)IV=IV+1
      GO TO 84
   89 IF(ABS((V(I)-VP(I))/VP(I)).LE.ERR)IV=IV+1
```

```
      GO TO 84
   90 IF(ABS((W(I)-WP(I))/W(I)).LE.ERR)IW=IW+1
      GO TO 433
   91 IF(ABS((W(I)-WP(I))/WP(I)).LE.ERR)IW=IW+1
  433 CONTINUE
  223 WRITE(6,2002)IU,IV,IW
 2002 FORMAT(3I5)
      IF((IU+IV+IW).GE.NOC)GO TO 224
      DO 331 I=1,NNI
      UP(I)=U(I)
      VP(I)=V(I)
  331 WP(I)=W(I)
      DO 5991 J=1,2*NEZ-1
      DO 5991 I=1,2*NER-1
 5991 INT((I-1)*(2*NEZ-1)+J)=I*(2*NEZ+1)+J+1
      WRITE(6,1101)(INT(I),I=1,NNI)
 1101 FORMAT(26I3)
      DO 225 NEL=1,NET
      DO 225 J=1,3
      M=NOD(NEL,J)
      DO 5142 IM=1,NNI
 5142 IF(M.EQ.INT(IM))GO TO 5143
      GO TO 225
 5143 US(NEL,J)=(1.+OM)*U(IM)-OM*US(NEL,J)
      VS(NEL,J)=(1.+OM)*V(IM)-OM*VS(NEL,J)
      WS(NEL,J)=(1.+OM)*W(IM)-OM*WS(NEL,J)
      WRITE(6,12)US(NEL,J),VS(NEL,J),WS(NEL,J)
   12 FORMAT(3F10.5)
  225 CONTINUE
      WRITE(6,10)(UP(I),VP(I),WP(I),I=1,NNI)
      NNT=NTN
      WRITE(6,8765)NNT,NTN
 8765 FORMAT(2I5)
      WRITE(6,1111)NTN,NNT,NER,NEZ
 1111 FORMAT(4I5)
      ITR=ITR+1
      IF(ITR.EQ.IDISK) GO TO 224
      RETURN
  224 WRITE(6,2223)(U(I),I=1,NNI)
      WRITE(6,2223)(V(I),I=1,NNI)
      WRITE(6,2223)(W(I),I=1,NNI)
      WRITE(6,2223)(P(I),I=1,NNIP)
      WRITE(6,2223)(PSI(I),I=1,NNI)
      WRITE(8,332)((US(I,J),VS(I,J),WS(I,J),J=1,3),I=1,NTE)
  332 FORMAT(3E15.4)
      WRITE(8,330)(U(I),V(I),W(I),I=1,NNI)
  330 FORMAT(6E15.4)
      REWIND 8
      STOP
      DEBUG SUBCHK
      END
```

```
1.          FUNCTION F1(X1,X2,X3)


2.          COMMON R1,R2,R3
3.          PRINT *,X1,X2,X3
4.          F1=X1*X1*(2*X1-1)**2./(R1*X1+R2*X2+R3*X3)
5.          RETURN
6.          END

1.          FUNCTION F2(X1,X2,X3)
2.          COMMON R1,R2,R3
3.          F2=X1*X2*(2*X1-1)*(2*X2-1)/(R1*X1+R2*X2+R3*X3)
4.          RETURN
5.          END

1.          FUNCTION F3(X1,X2,X3)
2.          COMMON R1,R2,R3
3.          F3=X1*X3*(2*X1-1)*(2*X3-1)/(R1*X1+R2*X2+R3*X3)
4.          RETURN
5.          END

1.          FUNCTION F4(X1,X2,X3)
2.          COMMON R1,R2,R3
3.          F4=4*X1*X1*X3*(2*X1-1)/(R1*X1+R2*X2+R3*X3)
4.          RETURN
5.          END

1.          FUNCTION F5(X1,X2,X3)
2.          COMMON R1,R2,R3
3.          F5=4*X1*X1*X2*(2*X1-1)/(R1*X1+R2*X2+R3*X3)
4.          RETURN
5.          END

1.          FUNCTION F6(X1,X2,X3)
2.          COMMON R1,R2,R3
3.          F6=4*X1*X2*X3*(2*X1-1)/(R1*X1+R2*X2+R3*X3)
4.          RETURN
5.          END

1.          FUNCTION F7(X1,X2,X3)
2.          COMMON R1,R2,R3
3.          F7=X2*X2*(2*X2-1)**2/(R1*X1+R2*X2+R3*X3)
4.          RETURN
5.          END

1.          FUNCTION F8(X1,X2,X3)
2.          COMMON R1,R2,R3
3.          F8=X2*X3*(2*X2-1)*(2*X3-1)/(R1*X1+R2*X2+R3*X3)
4.          RETURN
5.          END
```

```
1.          FUNCTION F9(X1,X2,X3)
2.          COMMON R1,R2,R3
3.          F9=4*X1*X2*X3*(2*X2-1)/(R1*X1+R2*X2+R3*X3)
4.          RETURN
5.          END
```

```
1.          FUNCTION F10(X1,X2,X3)
2.          COMMON R1,R2,R3
3.          F10=4*X1*X2*X2*(2*X2-1)/(R1*X1+R2*X2+R3*X3)
4.          RETURN
5.          END
```

```
1.          FUNCTION F11(X1,X2,X3)
2.          COMMON R1,R2,R3
3.          F11=4*X2*X2*X3*(2*X2-1)/(R1*X1+R2*X2+R3*X3)
4.          RETURN
5.          END
```

```
1.          FUNCTION F12(X1,X2,X3)
2.          COMMON R1,R2,R3
3.          F12=X3*X3*(2*X3-1)**2./(R1*X1+R2*X2+R3*X3)
4.          RETURN
5.          END
```

```
1.          FUNCTION F13(X1,X2,X3)
2.          COMMON R1,R2,R3
3.          F13=4*X1*X3*X3*(2*X3-1)/(R1*X1+R2*X2+R3*X3)
4.          RETURN
5.          END
```

```
1.          FUNCTION F14(X1,X2,X3)
2.          COMMON R1,R2,R3
3.          F14=4*X1*X2*X3*(2*X3-1)/(R1*X1+R2*X2+R3*X3)
4.          RETURN
5.          END
```

```
1.          FUNCTION F15(X1,X2,X3)
2.          COMMON R1,R2,R3
3.          F15=4*X2*X3*X3*(2*X3-1)/(R1*X1+R2*X2+R3*X3)
4.          RETURN
5.          END
```

```
1.          FUNCTION F16(X1,X2,X3)
2.          COMMON R1,R2,R3
3.          F16=16*X1*X1*X3*X3/(R1*X1+R2*X2+R3*X3)
4.          RETURN
5.          END
```

```
1.          FUNCTION F17(X1,X2,X3)
2.          COMMON R1,R2,R3
3.          F17=16*X1*X1*X2*X3/(R1*X1+R2*X2+R3*X3)
4.          RETURN
5.          END
```

```
1.          FUNCTION F18(X1,X2,X3)
2.          COMMON R1,R2,R3
3.          PRINT *,X1,X2,X3
4.          F18=16*X1*X3*X2*X3/(R1*X1+R2*X2+R3*X3)
5.          RETURN
6.          END
```

```
1.          FUNCTION F19(X1,X2,X3)
2.          COMMON R1,R2,R3
3.          F19=16*X1*X1*X2*X2/(R1*X1+R2*X2+R3*X3)
4.          RETURN
5.          END
```

```
1.          FUNCTION F20(X1,X2,X3)
2.          COMMON R1,R2,R3
3.          F20=16*X1*X2*X2*X3/(R1*X1+R2*X2+R3*X3)
4.          RETURN
5.          END
```

```
1.          FUNCTION F21(X1,X2,X3)
2.          COMMON R1,R2,R3
3.          F21=16*X2*X2*X3*X3/(R1*X1+R2*X2+R3*X3)
4.          RETURN
5.          END
```

```
1.          FUNCTION F22(X1,X2,X3)
2.          COMMON R1,R2,R3
3.          F22=R1*X1+R2*X2+R3*X3
4.          RETURN
5.          END
```