Durham E-Theses

Finite element analysis for the navier - stokes equations

Cheng, Ken Y-K

How to cite:

Cheng, Ken Y-K (1977) *Finite element analysis for the navier - stokes equations*, Durham theses, Durham University. Available at Durham E-Theses Online: http://etheses.dur.ac.uk/9133/

Use policy

FINITE ELEMENT ANALYSIS

FOR THE NAVIER - STOKES EQUATIONS

A thesis submitted for the degree of

Master of Science

in the

University of Durham

by

Ken Y-K Cheng

June 1977
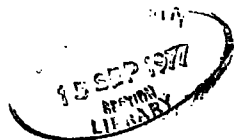
ACKNOWLEDGEMENTS

This work was carried out under the supervision of Mr.J.M.Wilson. Without his constant advice and support the work could not have started. Without his constant guidance and assistance as well as his personal encouragement it could not have reached this stage. I am extremely grateful to my supervisor.

I wish to express my thanks to members of staff of the Computer Unit Department, University of Durham, for their contribution to the various problems relevant to the development of the computer programmes.

My gratitude must also go to the staff of Durham University Library.

ABSTRACT

The finite element method was employed to solve two-dimensional, unsteady, incompressible, viscous fluid flow problems. A practical computation procedure is presented. A complete finite element computer program has been developed. The numerical technique is based upon a general formulation for the Navier-Stokes equations making use of a combined variational principle finite element approach. Solution to the system of algebraic equations is approached by the Gaussian elimination scheme. The time-dependent Navier-Stokes equations are expressed in terms of a stream function equation and a transport equation. A variational functional of the stream function and a pseudo-variational functional of the vorticity of the respective boundary value problem is presented. The pressure distribution and velocity profile are determined from stream function. Two numerical examples are presented and compared with present papers. Some new ideas about the numerical method, obtained through numerical experiments, are presented and discussed.

CONTENTS

List of Figures and Tables   Page

Chapter 1   INTRODUCTION

The Navier-Stokes equations governing the fluid flow problems, are known to have applications to a large class of engineering problems. Exact solutions of such a viscous fluid flow problem are not currently available. The necessity of providing reasonable estimates for complicated flow phenomena leaves research engineers very little choice. The numerical approach seems to be one of the very few acceptable tools.

Even the numerical methods face difficulties arising from the non-linearity and complexity of the boundary involved. The present high-speed, large storage digital computers have now made it possible to solve the Navier-Stokes equations. Numerical solution of the Navier-Stokes equations utilising modern high-speed computers have been developed by a number of investigators.

A finite-difference approach presented by Fromm and Harlow (57) has had considerable success in solving problems. Lee and Fung (79) used a method which combines the conformal mapping and finite-difference technique to analysis viscous flow problems. Mills (91) employed the finite-difference scheme to solve viscous flow through a pipe orifice at low Reynolds numbers. Rimon (105) also used this scheme to get solutions of the incompressible time-dependent viscous fluid flow past a thin oblate spheroid. Dennis and Chang (35) employed it to study problems of steady flow past a circular cylinder. Using this approach, Greenspan (51) was doing numerical studies of steady, viscous, incompressible flow in a channel with a step. He also presented in a second paper (52) some useful equations to determine wall vorticity at some special solid wall surface. Lin, Pepper and Lee (83) employed the finite-difference techniques to analyze separated flows around a circular cylinder. Macagno and Hung (87) made a study of a captive annular eddy using the finite-difference method. Roscoe (108) has been using a new finite-difference approach to study the three-dimensional Navier-Stokes equations. Carlson
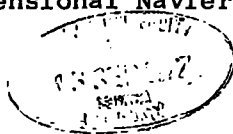
and Hornbeck (24) analysed the laminar entrance flow of an incompressible

viscous fluid in a square duct using the finite-difference procedures.

From these numerous studies, it seems to show that the applications

of the finite-difference method have been limited due to complexities

in the developed computational procedures.   It seems to require large

amounts of computer time and storage.   Another important disadvantage of

the finite-difference methods is the fact that these methods rely mostly

on meshes of very regular and symmetric patterns.   Great computational

difficulties are encountered if the geometric configuration of the fluid

flow is complicated and cannot be readily transformed into a mesh of

rectangular pattern (25,62,63,77).

These difficulties can be overcome with the finite-element method.

The finite-element has, in general, certain advantages over the finite-

difference approach.   These are the ease with which irregular geometries,

non-uniform meshes and imposition of appropriate boundary conditions can

be applied (4,16,21,37,62,96,97,100,117).

The finite-element method, developed initially for structural and

solid mechanics, has been applied to some fluid flow problems.   Structural

and non-structural elements may often be identical in shape and, further,

be represented by similar mathematical expressions.   The major difference

between the elasticity and fluid flow problems lies in the boundary

conditions to be satisfied (14,21,22,100).

Oden (95) has presented a theoretical finite element analogue for the

Navier-Stokes equations, but without a practical numerical method.   Olson

(100) presented a numerical procedure to investigate steady incompressible

flow problems using stream function formulation.   Some useful practical

techniques can be learned from his paper. Yamada, Yokouchi and Ohtsubo

(129) used the pressure-velocity · formulation to analyse steady flow

problems.   Tong (124) presented results for steady flow using this method

with pressure and velocities   as dependent variables.   Skiba employed

a   variational approach and rectangular elements to obtain results for

steady convection flow in a rectangular cavity. R,T-S Cheng (25) suggested

a versatile and widely applicable quasi-variational formulation to solve

the time-dependent Navier-Stokes equations.   Bratanow,Ecer and Kobiske

(16,17) studied unsteady incompressible flow problems using a perturbation

technique for treatment of the nonlinearities in the variational formulation

of the vorticity transport equation, and employing higher-order finite

elements for a consistent solution of the governing equations and in

describing the boundary conditions.   Baker (4,5,6) used a Galerkin method

and triangular elements for unsteady flow.

Atkinson, Brocklebank,Card and Smith (2) studied creeping flow around

a sphere, flow through a converging conical section, and developing flow

in a circular pipe using the stream function formulation.   They employed

three-node triangular elements with stream function and its first two

derivatives specified at each node.   This kind of formulation requires

less computer storage than velocity-pressure formulations, since there is

only a single equation to be solved.   However, the necessity for first

order continuous ($C^{(1)}$) elements would tend to make extension to three

dimensional work difficult.   Tong and Fung (124) used the stream function

formulation as well to investigate slow-viscous flow in a capillary in

the presence of moving particles suspended in the flow.   Their work has

direct application to the biomedical problem of determining the influence

of red blood cells on the flow in capillary blood vessels.   Taylor and

Hood employed the pressure-velocity formulation to study the problem of shear

induced fluid flow past a cavity.   Because the same interpolation functions

were used for both pressure and volocity for this problem, the accuracy of

the solution is open to question.   They have recently presented a

formulation using higher order shape functions for velocities than

pressure (117).   Tay and Davis (116) used variational principle to

study the problem of convection heat transfer between parallel planes.
Bratanow and Ecer (20) employed a variational approach to analyse the
three-dimensional unsteady flow around oscillating wings, and to study
unsteady aerodynamics (17).

Using quadratic polynomials shape functions for velocity and linear
polynomials shape functions for pressure, Kawahara and Yoshimura (71)
solved steady flow problems by the Newton-Raphson method and perturbation
method, and analysed unsteady flow problems by the perturbation method.
Laskaris (77) developed a numerical procedure to study two-dimensional
compressible and incompressible, steady state, viscous fluid flow and
heat transfer problems. The numerical scheme he presented is based on
a general formulation for the system of hydrodynamic equations, taking
into full account nonlinear convective terms, viscous terms, and heat
conduction terms, and using the method of weighted residuals applied
over discrete, distorted rectangular elements of the fluid flow regions.
Leonard (80) employed the Galerkin's method to solve perturbed compressible
flow problems. Issacs (69) used a transformation similar to that used
for quadrilateral isoparametric elements to derive a curved cubic
triangular element which has as nodal parameters the value of the function
and its two derivatives, and employed this kind of element to study
potential flow problems. He compared the triangular element with a
standard isoparametric element, and concluded that this kind of triangular
element will give similar accuracy at a significantly lower cost. Brebbia
and Smith (110) employed linear interpolation functions, a lumped mass
system and a simple Euler time integration scheme to analyse the two-
dimensional, unsteady, incompressible, viscous Navier-Stokes equations.
The results are not only extremely accurate to describe the natural
physical phenomena of the problem of vortex street development behind
a rectangular obstruction but also highly economic in computer time.

There are still many other good papers concerning numerical treatments of the Navier-Stokes equations. Some of them are chosen and will be presented in the references.

In the present work, the finite element method was employed to solve two-dimensional, unsteady, incompressible, viscous fluid flow problems. A practical computation procedure is presented. A complete finite element computer program has been developed. The formulation can be modified to cover a number of different situations. The same computer program can be used with only minor modification to solve other similar problems. The numerical technique is based upon a general formulation for the Navier-Stokes equations making use of a combined variational principle-finite element approach, applied over discrete finite elements of the fluid flow domain where the unknown fluid variables are expressed continuously in terms of interpolation functions and unknown parameters. Solution to the system of algebraic equations is approached by the Gaussian elimination scheme. It is believed that this numerical procedure is also suitable for a general three-dimensional problem. The time-dependent Navier-Stokes equations are expressed in terms of a stream function equation and a vorticity transport equation. A variational functional of the stream function and a pseudovariational functional of the vorticity of the respective boundary value problem will be presented. The pressure distribution and velocity profiles are determined from stream function.

As in the conventional procedure of time-dependent fluid flow, analysis was often carried out by the incremental method, assuming that the values calculated in the preceding step keep constant during the subsequent small time increments. This commonly used idea is also followed here. To circumvent the nonlinearity in the Navier-Stokes equations, the unsteady flow problem is assumed to be linear in the stream

function and vorticity at each time step. Steady-state solutions are
achieved by allowing the time-dependent solutions to converge.

To demonstrate the effectiveness of this numerical scheme, two
numerical expamples are presented and compared with present papers.
The numerical procedure used seems to be failry stable, and flow trends
seem to be well represented. Some new ideas about the numerical method,
obtained through numerous numerical experiments, are presented and
discussed. Although their validity for all kinds of numerical schemes
has not been ascertained yet, it is hoped to bring these observations to
people's attention.

Chapter 2   <u>VARIATIONAL FORMULATION OF NAVIER-STOKES EQUATIONS</u>

2.1   <u>Principles of Variational Calculus</u>

In this section, some fundamental principles of variational calculus, which will be used in the subsequent analysis are presented.

Variational calculus is concerned primarily with theory of maxima and minima, but the functions to be minimised or maximised are functionals. The variational calculus, in general, has alwasy been closely associated with realistic problems of continuum mechanics. Usually the functionals whose extreme values are sought are expressions of some form of system energy. For example, in fluid mechanics, for an incompressible, inviscid flow, the kinetic energy is a minimum. And another example is the principle of minimum total potential energy for elastic continua.(45,63,127)

Let us consider a simple functional expressed as

$$\phi = \int_{X_1}^{X_2} F(X, \psi, \psi_X, \psi_{XX})\, dX \tag{2-1}$$

where

F is an arbitrary function of one independent variable, X.

$$\psi = \psi(X)$$
$$\psi_X = \frac{\partial \psi}{\partial X}$$
$$\psi_{XX} = \frac{\partial^2 \psi}{\partial X^2}$$

Now the variation of the functional is defined in a manner similar to the calculus definition of a total differential

$$\delta F = \frac{\partial F}{\partial \psi} \delta \psi + \frac{\partial F}{\partial \psi_X} \delta \psi_X + \frac{\partial F}{\partial \psi_{XX}} \delta \psi_{XX} \tag{2-2}$$

It is obvious that there is an analogy between finding the minimum or maximum of a function via ordinary calculus and finding the minimum or maximum of a functional via variational calculus.(37,63,127)

So extending the concept of ordinary calculus, the following equation is obtained:

$$\delta\phi = \int_{X_1}^{X_2} \left(\frac{\partial F}{\partial \psi}\delta\psi + \frac{\partial F}{\partial \psi_x}\delta\psi_x + \frac{\partial F}{\partial \psi_{xx}}\delta\psi_{xx}\right)dx = 0 \qquad (2\text{-}3)$$

Extending the principles of ordinary calculus again, it can be learned that the first variation is also a commutative operator with both differentiation and integration if the integration limits are not to be varied. (37,63,127)

So the following equations may be written.

$$\delta\left(\int F dx\right) = \int (\delta F)dx \qquad (2\text{-}4)$$

$$\delta\left(\frac{d\psi}{dx}\right) = \frac{d}{dx}(\delta\psi) \qquad (2\text{-}5)$$

And equation (2-3) becomes

$$\delta\phi = \int_{X_1}^{X_2}\left(\frac{\partial F}{\partial \psi}\delta\psi + \frac{\partial F}{\partial \psi_x}\frac{\partial}{\partial x}(\delta x) + \cdots\right)dx = 0 \qquad (2\text{-}6)$$

Integrating each item by parts, the following equation is obtained.

$$\delta\phi = \int_{X_1}^{X_2}\left(\frac{\partial F}{\partial \psi} - \frac{d}{dx}\left(\frac{\partial F}{\partial \psi_x}\right) + \frac{d^2}{dx^2}\left(\frac{\partial F}{\partial \psi_{xx}}\right)\right)\delta\psi\, dx$$

$$+ \left(\frac{\partial F}{\partial \psi_x} - \frac{d}{dx}\left(\frac{\partial F}{\partial \psi_{xx}}\right)\right)\delta\psi\Bigg|_{X_1}^{X_2}$$

$$+ \left[\left(\frac{\partial F}{\partial \psi_{xx}}\right)\delta\psi_x\right]_{X_1}^{X_2}$$

$$= 0 \qquad (2\text{-}7)$$

Because $\delta\psi$ and $\delta\psi_x$ are arbitrary admissible variations, the integrand and remaining terms of equation (.2-7) must vanish. Thus the necessary conditions for $\psi(x)$ to minimise $\emptyset$ (x) are as follows:

$$\frac{\partial F}{\partial \psi} - \frac{d}{dx}\left(\frac{\partial F}{\partial \psi_x}\right) + \frac{d^2}{dx^2}\left(\frac{\partial F}{\partial \psi_{xx}}\right) = 0 \qquad (2\text{-}8)$$

$$\left[\frac{\partial F}{\partial \psi_x} - \frac{d}{dx}\left(\frac{\partial F}{\partial \psi_{xx}} \right) \right] \delta\psi \left. \right|_{X_1}^{X_2} = 0 \qquad (2\text{-}9)$$

$$\left[\left(\frac{\partial F}{\partial \psi_{xx}} \right) \delta\psi_x \right]_{X_1}^{X_2} = 0 \qquad (2\text{-}10)$$

Equation (2-8) is the governing differential equation for the problem and is called the Euler-Lagrange equation, or just the Euler equation. The other two conditions give the necessary boundary conditions. From equation (2 - 9) the following equation may be written. (37,63,127)

$$\left[\frac{\partial F}{\partial \psi_x} - \frac{d}{dx}\left(\frac{\partial F}{\partial \psi_{xx}} \right) \right]_{X_1}^{X_2} = 0 \qquad (2\text{-}11)$$

or

$$\psi(X_1) = 0$$

$$\psi(X_2) = 0 \qquad (2\text{-}12)$$

and from equation (2 - 10) either

$$\frac{\partial F}{\partial \psi_{xx}} \left. \right|_{X_1}^{X_2} = 0 \qquad (2\text{-}13)$$

or

$$\frac{\partial \psi}{\partial X} \left. \right|_{X_1}^{X_2} = 0 \qquad (2\text{-}14)$$

Equation (2 - 11) and (2 - 13) are called natural boundary conditions. If they are satisfied, they are called free boundary conditions. Equations (2- 12) and (2 - 14) are called geometric boundary condtions or forced boundary conditions. It may be mentioned here that the Euler-Lagrange equation expresses only a necessary and not a sufficient condition for a minimum. So the solution of an Euler- Lagrange equation may not yield a function that minimises a given functional. (See later Section 2.3 Variational Formulation).

One of the principal advantages of the finite element method
employing a suitable, valid variational principle is. that only the
geometric boundary conditions need to be specified.   The natural boundary
conditions are automatically incorporated in the formulation.   That is
why all the boundary conditions : have only been enforced on rigid
boundaries in this work and why the 'natural' boundary conditions are
always left for the program.to approximate when a suitable variational
principle-finite element method is employed to deal with test problems.
Through numerical experiments it has been found that when a combined
variational principle-finite element method is employed, the 'natural' boundary
 conditions had better   not be specified again, otherwise the results
may be in error. (21,37,42,63,100,127,131) (see Chapter 4).

A functional of two independent variables has the form

$$\phi(\psi) = \iint_A F(x, y, \psi, \psi_x, \psi_y, \psi_{xx}, \psi_{xy}, \psi_{yy}) \, dx \, dy \qquad (2\text{-}15)$$

Proceeding in a similar way, it is not difficult to derive the Euler-
Lagrange equations and boundary conditions for the above functional.
The Euler equation for equation (2-15) is

$$\frac{\partial^2}{\partial x^2}\left(\frac{\partial F}{\partial \psi_{xx}}\right) + \frac{\partial^2}{\partial x \, \partial y}\left(\frac{\partial F}{\partial \psi_{xy}}\right) + \frac{\partial^2}{\partial y^2}\left(\frac{\partial F}{\partial \psi_{yy}}\right)$$

$$-\frac{\partial}{\partial x}\left(\frac{\partial F}{\partial \psi_x}\right) - \frac{\partial}{\partial y}\left(\frac{\partial F}{\partial \psi_y}\right) + \frac{\partial F}{\partial \psi} = 0 \qquad (2\text{-}16)$$

Similarly, Euler-Lagrange equations and boundary conditions for
other functionals may be derived.   Some more detailed.discussions and
applications will be presented in Section 2.3.

2.2. Navier-Stokes Equations

The full Navier-Stokes equations representing a balance of viscous
forces, inertia forces, and pressure forces are capable of describing

some of the most interesting phenonmena in fluid mechanics. For unsteady,

incompressible, two-dimensional, viscous fluid flow with inertia, the

Navier-Stokes equations for analysing the motion of the fluids can be

written as

(2-17)

$$\frac{\partial \vec{u}}{\partial t} + \vec{u} \cdot (\nabla \vec{u}) = \frac{1}{\rho} \vec{F} - \frac{1}{\rho} \nabla P + \nu \nabla^2 \vec{u}$$

where

$\vec{u}$ = velocity vector = $[u, v]$

t = time

∇ = differential operator = $\frac{\partial}{\partial x} + \frac{\partial}{\partial y} + \frac{\partial}{\partial z}$

ρ = density of the fluid

$\vec{F}$ = body force vector

ν = kinematic viscosity of the fluid

P = pressure

∇^2 = Laplacian operator = $\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} + \frac{\partial^2}{\partial z^2}$

u = velocity in X direction

v = velocity in Y direction

in consistent units.

The equation of continuity for incompressible fluid is

$$\nabla \cdot \vec{u} = 0$$

(2-18)

The velocity components u and v may be expressed in terms of a

stream function ψ as

$$u = \frac{\partial \psi}{\partial y}$$

(2-19)

$$v = -\frac{\partial \psi}{\partial x}$$

The vector

$$\vec{u} = [u, v]$$

may be written as

$$\vec{u} = \left[\frac{\partial \psi}{\partial y} , - \frac{\partial \psi}{\partial x} \right] \tag{2-20}$$

Using equation (2-19) , fluid rotation or vorticity is defined as the average angular velocity of any two mutually perpendicular line elements of a fluid particle.

In vector notation, the following equation can be written:

$$\vec{\omega} = \nabla \times \vec{u} \tag{2-21}$$

A well-known vector identity shows that for any function P having continuous first and second derivatives,

$$\nabla \times \nabla P = 0 \tag{2-22}$$

At the same time, the following equation may be written

$$\nabla \times \vec{F} = 0 \tag{2-23}$$

Taking the curl ($\nabla \times$) of both sides of equation (2-17) gives

$$\frac{\partial \omega}{\partial t} + (\vec{u} \cdot \nabla) \omega = \nu \nabla^2 \omega \tag{2-24}$$

From equations (2-20) and (2-21), the stream functions are related to vorticity as follows:

$$\nabla^2 \psi = - \omega \tag{2-25}$$

Now the pressure distribution is to be calculated. The pressure field can be obtained by integrating the momentum equation. But, in general, it seems that a Poisson type equation yields more accurate numerical computation results and uses less computer time than the direct methods based on the momentum equation (16,21,77). So the Poisson type equation will be derived first. Equation (2-17) can be written as

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} + v\frac{\partial u}{\partial y} = \frac{1}{\rho}\vec{F_x} - \frac{1}{\rho}\frac{\partial P}{\partial x} + \nu\left[\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}\right] \tag{2-26}$$

Differentiating this equation with respect to x gives

$$\left(\frac{\partial}{\partial x}\frac{\partial u}{\partial t}\right) + \left(\frac{\partial u}{\partial x}\frac{\partial u}{\partial x} + u\frac{\partial^2 u}{\partial x^2}\right) + \left(\frac{\partial v}{\partial x}\frac{\partial u}{\partial y} + v\frac{\partial^2 u}{\partial x\partial y}\right)$$

$$= \frac{1}{\rho}\frac{\partial F_x}{\partial x} - \frac{1}{\rho}\frac{\partial^2 P}{\partial x^2} + \nu\left(\frac{\partial^2\partial u}{\partial x^2\partial x} + \frac{\partial^2\partial u}{\partial y^2\partial x}\right) \tag{2-27}$$

Similarly, from equation (2-17)

$$\frac{\partial v}{\partial t} + u\frac{\partial v}{\partial x} + v\frac{\partial v}{\partial y} = \frac{1}{\rho}F_y - \frac{1}{\rho}\frac{\partial P}{\partial y} + \nu\left[\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2}\right] \tag{2-28}$$

Differentiating it with respect to y gives

$$\frac{\partial}{\partial y}\frac{\partial v}{\partial t} + \left(\frac{\partial u}{\partial y}\frac{\partial v}{\partial x} + u\frac{\partial^2 v}{\partial x\partial y}\right) + \left(\frac{\partial v}{\partial y}\frac{\partial v}{\partial y} + v\frac{\partial^2 v}{\partial y^2}\right)$$

$$= \frac{1}{\rho}\frac{\partial F_y}{\partial y} - \frac{1}{\rho}\frac{\partial^2 P}{\partial y^2} + \nu\left(\frac{\partial^2}{\partial x^2}\frac{\partial v}{\partial y} + \frac{\partial^2}{\partial y^2}\frac{\partial v}{\partial y}\right) \tag{2-29}$$

Combining equations (2-29) and (2-27) gives

$$\frac{\partial}{\partial t}\left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y}\right) + \left(\frac{\partial u}{\partial x}\right)^2 + u\frac{\partial^2 u}{\partial x^2} + 2\frac{\partial v}{\partial x}\frac{\partial u}{\partial y} + v\frac{\partial^2 u}{\partial x\partial y}$$

$$+ u\frac{\partial^2 v}{\partial x\partial y} + \left(\frac{\partial v}{\partial y}\right)^2 + v\frac{\partial^2 v}{\partial y^2}$$

$$= \frac{1}{\rho}\left(\frac{\partial F_x}{\partial x} + \frac{\partial F_y}{\partial y}\right) - \frac{1}{\rho}\left(\frac{\partial^2 P}{\partial x^2} + \frac{\partial^2 P}{\partial y^2}\right) + \nu\left[\frac{\partial^2}{\partial x^2}\frac{\partial u}{\partial x}\right.$$

$$\left. + \frac{\partial^2\partial u}{\partial y^2\partial x} + \frac{\partial^2}{\partial x^2}\frac{\partial v}{\partial y} + \frac{\partial^2}{\partial y^2}\frac{\partial v}{\partial y}\right] \tag{2-30}$$

the following can be written $\qquad$ (2-18)

$$\nabla \cdot \vec{u} = 0$$

From equation (2-18) it is easy to get following relations:

$$\frac{\partial^2}{\partial x^2} \frac{\partial u}{\partial x} + \frac{\partial^2}{\partial y^2} \frac{\partial u}{\partial x} + \frac{\partial^2}{\partial x^2} \frac{\partial v}{\partial y} + \frac{\partial^2}{\partial y^2} \frac{\partial v}{\partial y} = 0 \qquad (2\text{-}31)$$

$$\frac{\partial}{\partial t} \left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} \right) = 0 \qquad (2\text{-}32)$$

and

$$u \frac{\partial^2 u}{\partial x^2} + v \frac{\partial^2 u}{\partial x \partial y} + u \frac{\partial^2 v}{\partial x \partial y} + v \frac{\partial^2 v}{\partial y^2} = 0 \qquad (2\text{-}33)$$

Substituting equations (2-31), (2-32) and (2-33) into equation (2-30) gives

$$\left(\frac{\partial u}{\partial x} \right)^2 + 2 \frac{\partial v}{\partial x} \frac{\partial u}{\partial y} + \left(\frac{\partial v}{\partial y} \right)^2 = \frac{1}{\rho} \nabla \cdot \vec{F} - \frac{1}{\rho} \nabla^2 P \qquad (2\text{-}34)$$

In the absence of body forces, the following equation can be obtained

$$\nabla^2 P = -\rho \left[\left(\frac{\partial u}{\partial x} \right)^2 + \left(\frac{\partial v}{\partial y} \right)^2 + 2 \frac{\partial v}{\partial x} \frac{\partial u}{\partial y} \right] \qquad (2\text{-}35)$$

or

$$\nabla^2 P = -2\rho \left[\frac{\partial v}{\partial x} \frac{\partial u}{\partial y} - \frac{\partial u}{\partial x} \frac{\partial v}{\partial y} \right] \qquad (2\text{-}36)$$

$$= 2\rho \left[\frac{\partial^2 \psi}{\partial y^2} \frac{\partial^2 \psi}{\partial x^2} - \left(\frac{\partial^2 \psi}{\partial x \partial y} \right)^2 \right] \qquad (2\text{-}37)$$

$$= -\rho Q \qquad (2\text{-}38)$$

where

$$Q = 2\left[\frac{\partial V}{\partial x}\frac{\partial u}{\partial y} - \frac{\partial u}{\partial x}\frac{\partial v}{\partial y}\right]$$ (2-39)

$$= -2\left[\frac{\partial^2 \psi}{\partial y^2}\frac{\partial^2 \psi}{\partial x^2} - \left(\frac{\partial^2 \psi}{\partial x \partial y}\right)^2\right]$$

The analysis of unsteady, incompressible, two-dimensional, viscous fluid flow involves the simultaneous solution of equations (2-24) and (2-25). Once the ψ and ω field are known, the pressure distribution can be calculated from equation (2-37).

2.3 Variational Formulation

Basically, the finite element method represents an approximate procedure for satisfying the problem in terms of its variational formulation. If the governing differential equations were all linear the variational formulation would be straightforward. The non-linear terms in the Navier-Stokes equations seem to have precluded the existence of an associated variational principle of the classical kind. It is found that the Navier-Stokes equations cannot be derived from a classical variational principle unless one of the terms $\vec{u} \times (\nabla \times \vec{u})$ or $\vec{u} \cdot [\nabla \vec{u}]$ is zero (41). However, it has been shown that 'pseudo' principles can be obtained provided some terms are not allowed to vary when the first variation is performed. The pseudo-variational functional finite-element method has the advantage of simplicity and reduced computation. It is not a true variational method since from another point of view it can be regarded as a Galerkin method used with a particular approximattion scheme.(21,25,63,100,127)

Equation (2-25) and (2-37) are in the form of Poisson's equation, for which a direct variational formulation exists. From theorems of variational methods, by inspection, the variational functionals for equations (2-25) and (2-37) can be written as follows

$$\phi_2 = \frac{1}{2}\int_A \left[\left(\frac{\partial \psi}{\partial x}\right)^2 + \left(\frac{\partial \psi}{\partial y}\right)^2\right] dA - \int_A \omega \psi \, dA$$ (2-40)

$$\phi_3 = \frac{1}{2} \int_A \left[\left(\frac{\partial P}{\partial x} \right)^2 + \left(\frac{\partial P}{\partial y} \right)^2 \right] dA - \int_A \rho P Q \, dA \qquad (2\text{-}41)$$

where

$$Q = -2 \left[\frac{\partial^2 \psi}{\partial y^2} \frac{\partial^2 \psi}{\partial x^2} - \left(\frac{\partial^2 \psi}{\partial x \partial y} \right)^2 \right]$$

and A is the region of interest. The variational functionals ϕ_2 and ϕ_3

exist such that the Euler-Lagrangian equations of equations (2-40) and

(2-41) are simply equations (2-25) and (2-37). The functions ψ and P

which satisfy the required boundary conditions and which give the functional

integrals ϕ_2 and ϕ_3 are solutions of equations (2-25) and (2-37).

Similarly, by taking $\frac{\partial \omega}{\partial t}$ as an invariance, a variational functional ϕ_1

$$\phi_1 = \int_A \omega \, \frac{\partial \omega}{\partial t} \, dA + \int_A \left[u \omega \, \frac{\partial \omega}{\partial t} + v \omega \, \frac{\partial \omega}{\partial y} \right] dA$$
$$+ \frac{\nu}{2} \int_A \left[\left(\frac{\partial \omega}{\partial x} \right)^2 + \left(\frac{\partial \omega}{\partial y} \right)^2 \right] dA \qquad (2\text{-}42)$$

exists (16,19,25,41) such that upon taking the first variation of ϕ_1, the

vorticity transport equation, equation (2-24) will be recovered. The

function ω satisfying equation (2-24) and its boundary conditions minimise

ϕ_1. Segregating stream function and vorticity solutions according to

different instants of time reduces the problem to one of consecutively

minimising ϕ_1 and ϕ_2. This can he conveniently accomplished by the finite

element method. (21,25,63,100,127)

A disadvantage of the procedure is that it is not known whether or not

a particular pseudo-functional will yield convergence or not until it has

been tried, since a mathematical criterion for convergence is not yet

available. (21,25,93,100)

Norrie and Vries (93) suggested that if a certain functional does not

converge, one has no choice but to modify it; such a change alters the

stiffness matrix and may result in convergence.

Some experience with the method assists in choosing an appropriate functional on intuitive grounds.   Norrie and Vries postulate that: "The process will converge if the terms which dominate the physical behaviour of the system are included are those terms in the functional which are not iterated upon but are used only in the minimisation procedure". (93) One of the disadvantages of the use of stream function-vorticity formulation is that unless the velocities are entirely prescribed on all boundaries it is often impossible to establish the values of stream functions on some positions of the boundary.   This is particularly serious in multiple connected boundaries, such as are presented by flow around obstacles etc. To overcome these difficulties it is necessary to introduce additional constraints on the rate of boundary work.   (see Section 6.1, example one)

Chapter 3   FINITE ELEMENT MODEL

3.1   Introduction

The finite element method is based on the use of series expansions within subdomains or elements, into which the domain of interest is divided.   It is a general numerical technique which provides an approximate piecewise continuous representation of the unknown field variables in terms of polynomials, sometimes called interpolation functions or shape functions and model parameters (22,33,37,77,127).

The continuous region is subdivided into a finite number of elements where the nodal values and/or the partial derivatives of the dependent variables at prescribed points, nodes of elements, become the unknown parameters of the problem.   The finite-element representation of the dependent field variable must be able to provide an improved approximation to the true solution as successive subdivisions of the domain using smaller and smaller elements is attempted.

The basic steps of the solution procedure are as follows (22,33,37,127).

1.   Discretisation of the continuum.

2.   Selection of interpolation functions.

3.   Evaluation of the matrices of the elements.

4.   Assembly of the element equations.

5.   Application of the boundary conditions.

6.   Solution of the system equations.

7.   Calculation of any other unknown field variables.

3.2   Matrix Formulation

Different finite-element models were chosen for representing the variations of streamfunctions, vorticities and pressure.   Fig.3-1 shows a typical finite element.   The stream functions and vorticities were assumed to vary linearly over each finite element as
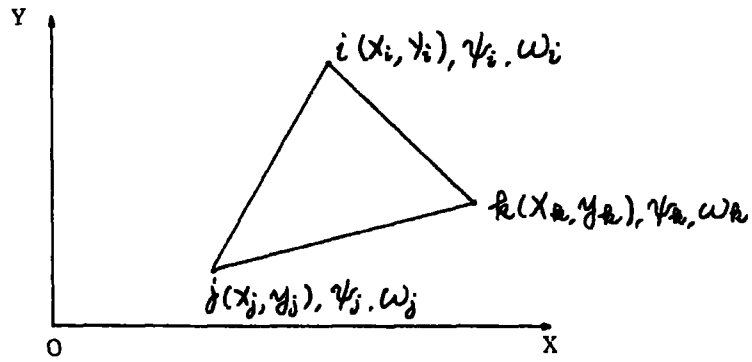
Figure 3-1 Triangular finite element

$$\psi^{(e)}(x,y,t) = [H_1^{(e)}(t), H_2^{(e)}(t), H_3^{(e)}(t)] \left\{ \begin{array}{c} T_1^{(e)}(x,y) \\ T_2^{(e)}(x,y) \\ T_3^{(e)}(x,y) \end{array} \right\} \quad (3\text{-}1)$$

$$\omega^{(e)}(x,y,t) = [H_1^{(e)}(t), H_2^{(e)}(t), H_3^{(e)}(t)] \left\{ \begin{array}{c} Q_1^{(e)}(x,y) \\ Q_2^{(e)}(x,y) \\ Q_3^{(e)}(x,y) \end{array} \right\} \quad (3\text{-}2)$$

where the T's and Q's are the trial functions and the superscript (e)
indicates the ℓth element.   At the nodal points of this element, points
i, j and k in Fig.3-1, ψ_i, ω_i, ψ_j, ω_j, and $\psi_k \omega_k$ are

$$\psi_i(t) = H_1^{(e)}(t) T_1^{(e)}(x_i, y_i) + H_2^{(e)}(t) T_2^{(e)}(x_i, y_i)$$
$$+ H_3^{(e)}(t) T_3^{(e)}(x_i, y_i)$$

$$\omega_i(t) = H_1^{(e)}(t) Q_1^{(e)}(x_i, y_i) + H_2^{(e)}(t) Q_2^{(e)}(x_i, y_i) \qquad (3\text{-}3)$$
$$+ H_3^{(e)}(t) Q_3^{(e)}(x_i, y_i)$$

$$\psi_j(t) = H_1^{(e)}(t) T_1^{(e)}(x_j, y_j) + H_2^{(e)}(t) T_2^{(e)}(x_j, y_j)$$
$$+ H_3^{(e)}(t) T_3^{(e)}(x_j, y_j)$$

$$\omega_j(t) = H_1^{(e)}(t) Q_1^{(e)}(x_j, y_j) + H_2^{(e)}(t) Q_2^{(e)}(x_j, y_j)$$
$$+ H_3^{(e)}(t) Q_3^{(e)}(x_j, y_j)$$

$$\psi_k(t) = H_1^{(e)}(t) T_1^{(e)}(x_k, y_k) + H_2^{(e)}(t) T_2^{(e)}(x_k, y_k)$$
$$+ H_3^{(e)}(t) T_3^{(e)}(x_k, y_k)$$

$$\omega_k(t) = H_1^{(e)}(t) Q_1^{(e)}(x_k, y_k) + H_2^{(e)}(t) Q_2^{(e)}(x_k, y_k)$$
$$+ H_3^{(e)}(t) Q_3^{(e)}(x_k, y_k)$$

In matrix form, equations (3-1) may be written as

$$
\begin{Bmatrix} \psi_i(t) \\ \psi_j(t) \\ \psi_k(t) \end{Bmatrix} = [C^{(e)}] \begin{Bmatrix} H_1^{(e)}(t) \\ H_2^{(e)}(t) \\ H_3^{(e)}(t) \end{Bmatrix}
$$

(3.4)

and

$$
\begin{Bmatrix} \omega_i(t) \\ \omega_j(t) \\ \omega_k(t) \end{Bmatrix} = [D^{(e)}] \begin{Bmatrix} H_1^{(e)}(t) \\ H_2^{(e)}(t) \\ H_3^{(e)}(t) \end{Bmatrix}
$$

where

$$
[C^{(e)}] = \begin{bmatrix} T_1^{(e)}(x_i, y_i) & T_2^{(e)}(x_i, y_i) & T_3^{(e)}(x_i, y_i) \\ T_1^{(e)}(x_j, y_j) & T_2^{(e)}(x_j, y_j) & T_3^{(e)}(x_j, y_j) \\ T_1^{(e)}(x_k, y_k) & T_2^{(e)}(x_k, y_k) & T_3^{(e)}(x_k, y_k) \end{bmatrix}
$$

(3-5)

and

$$
[D^{(e)}] = \begin{bmatrix} Q_1^{(e)}(x_i, y_i) & Q_2^{(e)}(x_i, y_i) & Q_3^{(e)}(x_i, y_i) \\ Q_1^{(e)}(x_j, y_j) & Q_2^{(e)}(x_j, y_j) & Q_3^{(e)}(x_j, y_j) \\ Q_1^{(e)}(x_k, y_k) & Q_2^{(e)}(x_k, y_k) & Q_3^{(e)}(x_k, y_k) \end{bmatrix}
$$

In order to express H_n's in terms of ψ_n's or ω_n's uniquely, equations (3-4) can be written as

$$
\begin{Bmatrix} H_1^{(e)}(t) \\ H_2^{(e)}(t) \\ H_3^{(e)}(t) \end{Bmatrix} = [C^{(e)}]^{-1} \begin{Bmatrix} \psi_i(t) \\ \psi_j(t) \\ \psi_k(t) \end{Bmatrix}
$$

(3-6)

and

$$
\begin{Bmatrix} H_1^{(e)}(t) \\ H_2^{(e)}(t) \\ H_3^{(e)}(t) \end{Bmatrix} = [D^{(e)}]^{-1} \begin{Bmatrix} \omega_i(t) \\ \omega_j(t) \\ \omega_k(t) \end{Bmatrix}
$$

So equations (3-1) and (3-2) give

$$\psi^{(e)}(x,y,t) = \left[T_1^{(e)}(x,y), T_2^{(e)}(x,y), T_3^{(e)}(x,y) \right] \begin{Bmatrix} H_1^{(e)}(t) \\ H_2^{(e)}(t) \\ H_3^{(e)}(t) \end{Bmatrix}$$

$$= \left[T_1^{(e)}(x,y), T_2^{(e)}(x,y), T_3^{(e)}(x,y) \right] [C^{(e)}]^{-1} \begin{Bmatrix} \psi_i(t) \\ \psi_j(t) \\ \psi_k(t) \end{Bmatrix}$$

$$= \left[N_1^{(e)}(x,y), N_2^{(e)}(x,y), N_3^{(e)}(x,y) \right] \begin{Bmatrix} \psi_i(t) \\ \psi_j(t) \\ \psi_k(t) \end{Bmatrix} \quad (3-7)$$

where

$$\left[N_1^{(e)}(x,y), N_2^{(e)}(x,y), N_3^{(e)}(x,y) \right] = \left[T_1^{(e)}(x,y), T_2^{(e)}(x,y), T_3^{(e)}(x,y) \right] [C^{(e)}]^{-1}$$

and

$$\omega^{(e)}(x,y,t) = \left[Q_1^{(e)}(x,y), Q_2^{(e)}(x,y), Q_3^{(e)}(x,y) \right] \begin{Bmatrix} H_1^{(e)}(t) \\ H_2^{(e)}(t) \\ H_3^{(e)}(t) \end{Bmatrix}$$

$$= \left[Q_1^{(e)}(x,y), Q_2^{(e)}(x,y), Q_3^{(e)}(x,y) \right] [D^{(e)}]^{-1} \begin{Bmatrix} \omega_i(t) \\ \omega_j(t) \\ \omega_k(t) \end{Bmatrix}$$

$$= \left[L_1^{(e)}(x,y), L_2^{(e)}(x,y), L_3^{(e)}(x,y) \right] \begin{Bmatrix} \omega_i(t) \\ \omega_j(t) \\ \omega_k(t) \end{Bmatrix} \quad (3-8)$$

where

$$\left[L_1^{(e)}(x,y), L_2^{(e)}(x,y), L_3^{(e)}(x,y) \right] = \left[Q_1^{(e)}(x,y), Q_2^{(e)}(x,y), Q_3^{(e)}(x,y) \right] [D^{(e)}]^{-1}$$

For a triangular element, if Tn's and Qn's are taken to be 1,x and y
then

$$N_i^{(e)}(x,y) = L_i^{(e)}(x,y) = (a_i + b_i x + c_i y) \big/ 2\Delta \quad (3-9)$$

in which

$$a_i = x_j y_k - x_k y_j$$
$$b_i = y_j - y_k$$
$$c_i = x_k - x_j$$

with the other coefficients obtained by a cyclic permutation of subscripts in order i,j,k and where

$$2\Delta = \begin{vmatrix} 1 & X_i & Y_i \\ 1 & X_j & Y_j \\ 1 & X_k & Y_k \end{vmatrix}$$

where Δ is the area of the element.

Equations(3-7) and (3-8) may be expressed by the following matrix equations.

$$\psi^{(e)}(x,y,t) = \left\{ N^{(e)}(x,y) \right\}^T \left\{ \psi(t) \right\} \qquad (3\text{-}10)$$

$$\omega^{(e)}(x,y,t) = \left\{ N^{(e)}(x,y) \right\}^T \left\{ \omega(t) \right\}$$

$$(3\text{-}11)$$

where T denotes the transpose to the column matrix.

The gradients of $\psi^{(e)}$ and $\omega^{(e)}$ are

$$\frac{\partial \psi^{(e)}}{\partial x} = \left\{ \frac{\partial N^{(e)}}{\partial x} \right\}^T \left\{ \psi(t) \right\} \qquad (3\text{-}12)$$

$$\frac{\partial \psi^{(e)}}{\partial y} = \left\{ \frac{\partial N^{(e)}}{\partial y} \right\}^T \left\{ \psi(t) \right\}$$

$$(3\text{-}13)$$

$$\frac{\partial \omega^{(e)}}{\partial x} = \left\{ \frac{\partial N^{(e)}}{\partial x} \right\}^T \left\{ \omega(t) \right\}$$

$$(3\text{-}14)$$

$$\frac{\partial \omega^{(e)}}{\partial y} = \left\{ \frac{\partial N^{(e)}}{\partial y} \right\}^T \left\{ \omega(t) \right\}$$

$$(3\text{-}15)$$

The finite element models were employed in discretising the variational functionals in equations (2-40), (2-41) and (2-42). Substituion of equations (3-10), (3-11) and the last equations into equations (2-40) and (2-42) gives

$$\phi_1 = \int_{A^{(e)}} \left[\{N^{(e)}\}^T \{\omega\} \right] \frac{\partial \left[\{N^{(e)}\}^T \{\omega\}\right]}{\partial t} \, dA^{(e)}$$

$$+ \iint_{A^{(e)}} \left(\left[\left\{\frac{\partial N^{(e)}}{\partial y}\right\}^T \{\psi\} \right] \left[\left\{\frac{\partial N^{(e)}}{\partial x}\right\}^T \{\omega\} \right] \right.$$

$$\left. - \left[\left\{\frac{\partial N^{(e)}}{\partial x}\right\}^T \{\psi\} \right] \left[\left\{\frac{\partial N^{(e)}}{\partial y}\right\}^T \{\omega\} \right] \right) \left[\{N^{(e)}\}^T \{\omega\} \right] \right\} dA^{(e)}$$

$$+ \frac{\nu}{2} \int_{A^{(e)}} \left(\left[\left\{\frac{\partial N^{(e)}}{\partial x}\right\}^T \{\omega\} \right]^2 + \left[\left\{\frac{\partial N^{(e)}}{\partial y}\right\}^T \{\omega\} \right]^2 \right) dA^{(e)}$$

$$= \int_{A^{(e)}} \left[\{N^{(e)}\}^T \{\omega\} \right] \frac{\partial \left[\{N^{(e)}\}^T \{\omega\}\right]}{\partial t} \, dA^{(e)} \qquad (3\text{-}16)$$

$$+ \int_{A^{(e)}} \left\{ \left(\frac{\partial \psi}{\partial y} \frac{\partial \omega}{\partial x} - \frac{\partial \psi}{\partial x} \frac{\partial \omega}{\partial y} \right) \left[\{N^{(e)}\}^T \{\omega\} \right] dA^{(e)} \right.$$

$$+ \frac{\nu}{2} \int_{A^{(e)}} \left(\left[\left\{\frac{\partial N^{(e)}}{\partial x}\right\}^T \{\omega\} \right]^2 + \left[\left\{\frac{\partial N^{(e)}}{\partial y}\right\}^T \{\omega\} \right]^2 \right) dA^{(e)}$$

and

$$\phi_2 = \frac{1}{2} \int_{A^{(e)}} \left(\left[\left\{\frac{\partial N^{(e)}}{\partial x}\right\}^T \{\psi\} \right]^2 + \left[\left\{\frac{\partial N^{(e)}}{\partial y}\right\}^T \{\psi\} \right]^2 \right) dA^{(e)}$$

$$- \int_{A^{(e)}} \left(\left[\{N^{(e)}\}^T \{\omega\} \right] \left[\{N^{(e)}\}^T \{\psi\} \right] \right) dA^{(e)} \qquad (3\text{-}17)$$

The finite element solution to the problem involves picking the nodal values of ω and ψ so as to make stationary the functionals ϕ_1 and ϕ_2. To make ϕ_1 and ϕ_2 stationary with respect to the nodal values of vorticity and stream function respectively, the following conditions are required.

$$\delta \phi_1(\omega) = \sum_{i=1}^{NN} \frac{\partial \phi_1(\omega)}{\partial \omega_i} \delta \omega_i = 0 \qquad (3\text{-}18)$$

$$\delta \phi_2(\psi) = \sum_{i=1}^{NN} \frac{\partial \phi_2(\psi)}{\partial \psi_i} \delta \psi_i = 0 \qquad (3\text{-}19)$$

where NN is the total number of nodes. Since the $\delta \omega$i's and $\delta \psi$i's are independent, equations (3-18) and (3-19) can hold only if

$$\frac{\partial \phi_1 (\omega)}{\partial \omega_i} \quad = \quad 0 \tag{3-20}$$

$$\frac{\partial \phi_2 (\psi)}{\partial \psi_i} \quad = \quad 0 \tag{3-21}$$

Hence from equations (3-21) and (3-17), for a typical node i the following equation is required

$$\frac{\partial \phi_2 (\psi)}{\partial \psi_i} = \int_{A^{(e)}} \left[\left\{ \frac{\partial N^{(e)}}{\partial X} \right\}^T \{\psi\} \frac{\partial N_i}{\partial X} + \left\{ \frac{\partial N^{(e)}}{\partial Y} \right\}^T \{\psi\} \frac{\partial N_i}{\partial Y} \right] dA^{(e)}$$

$$- \int_{A^{(e)}} \left[\left\{ N^{(e)} \right\}^T \{\omega\} N_i \right] dA^{(e)} \tag{3-22}$$

$$= 0$$

Utilising equation (3-11), equation (3-22) may be written as:

$$\int_{A^{(e)}} \left[\left\{ \frac{\partial N^{(e)}}{\partial X} \right\}^T \frac{\partial N_i}{\partial X} + \left\{ \frac{\partial N^{(e)}}{\partial Y} \right\}^T \frac{\partial N_i}{\partial Y} \right] \{\psi\} dA^{(e)}$$

$$- \int_{A^{(e)}} \omega^{(e)} N_i dA^{(e)} = 0 \tag{3-23}$$

In matrix form, for the entire element the following equation is obtained

$$\left[K_\psi \right]^{(e)} \{\psi\}^{(e)} + \left\{ S_\psi \right\}^{(e)} = \{0\} \tag{3-24}$$

where

$$K_{\psi_{ij}}^{(e)} = \int_{A^{(e)}} \left[\frac{\partial N_i}{\partial X} \frac{\partial N_j}{\partial X} + \frac{\partial N_i}{\partial Y} \frac{\partial N_j}{\partial Y} \right] dA^{(e)}$$

$$S_{\psi_i}^{(e)} = - \int_{A^{(e)}} \omega^{(e)} N_i dA^{(e)}$$

Similarly, from equations (3-20) and (3-16), for a typical node i the following equation is required

$$\frac{\partial \phi(\omega)}{\partial \omega_i} = \int_{A^{(e)}} N_i N_j \frac{\partial \omega}{\partial t} dA$$

$$+ \int_{A^{(e)}} \left(\frac{\partial \psi^{(e)}}{\partial y} \frac{\partial \omega^{(e)}}{\partial x} - \frac{\partial \psi^{(e)}}{\partial x} \frac{\partial \omega^{(e)}}{\partial y} \right) N_i dA^{(e)}$$

$$+ \frac{\nu}{2} \int_{A^{(e)}} \left(2 \left[\left\{ \frac{\partial N}{\partial x} \right\}^T \{\omega\} \right] \frac{\partial N_i}{\partial x} + 2 \left[\left\{ \frac{\partial N}{\partial y} \right\}^T \{\omega\} \right] \frac{\partial N_i}{\partial y} \right) dA^{(e)}$$

$$= 0$$

In matrix form, for the entire element the following equation is obtained.

$$[K_\omega]^{(e)} \{\omega\}^{(e)} + [K_{\dot\omega}]^{(e)} \{\dot\omega\}^{(e)} + \{S_\omega\}^{(e)} = \{0\}$$

where   (3-25)

$$K_{\omega_{ij}}^{(e)} = \nu \, K_{\psi_{ij}}^{(e)}$$

$$K_{\dot\omega_{ij}}^{(e)} = \int_{A^{(e)}} N_i N_j \, dA^{(e)}$$

$$S_{\omega_i}^{(e)} = \int_{A^{(e)}} N_i \left(\frac{\partial \psi^{(e)}}{\partial y} \frac{\partial \omega^{(e)}}{\partial x} - \frac{\partial \psi^{(e)}}{\partial x} \frac{\partial \omega^{(e)}}{\partial y} \right) dA^{(e)}$$

In solving a fluid flow problem with the foregoing elements, the usual assemblage process for finite elements is followed as well. For the assembled system the foregoing equations become

$$[K_\psi]\{\psi\} + \{S_\psi\} = \{0\} \tag{3-26}$$

$$[K_\omega]\{\omega\} + \{K_{\dot\omega}\}\{\dot\omega\} + \{S_\omega\} = \{0\} \tag{3-27}$$

where

$$K_{\psi_{ij}} = \sum_{e=1}^{M} \int_{A^{(e)}} \left[\frac{\partial N_i}{\partial x} \frac{\partial N_j}{\partial x} + \frac{\partial N_i}{\partial y} \frac{\partial N_j}{\partial y} \right] dA^{(e)} \tag{3-28}$$

$$S_{\psi_i} = -\sum_{e=1}^{M} \int_{A^{(e)}} \omega^{(e)} N_i \, dA^{(e)} \tag{3-29}$$

$$K_{\omega_{ij}} = \nu \, K_{\psi_{ij}} \tag{3-30}$$

$$K_{\dot{\omega}_{ij}} = \sum_{e=1}^{M} \int_{A^{(e)}} N_i \, N_j \, dA^{(e)} \tag{3-31}$$

$$S_{\omega_i} = \sum_{e=1}^{M} \int_{A^{(e)}} N_i \left(\frac{\partial \psi^{(e)}}{\partial y} \frac{\partial \omega^{(e)}}{\partial x} - \frac{\partial \psi^{(e)}}{\partial x} \frac{\partial \omega^{(e)}}{\partial y} \right) dA^{(e)} \tag{3-32}$$

where M is the total number of elements.

3.3 Integration of the Matrix Equation

To integrate equations (2-24) and (2-25) with respect to time, the method suggested by R. T-S Cheng (25) is used. Cheng considered two solutions ψ_n and ω_n at the Nth time step and ψ_{n+1} and ω_{n+1} at a time increment Δt later. Then the governing equations may be expressed as follows:

$$\frac{\partial \omega_n}{\partial t} + \frac{\partial \psi_{n+1}}{\partial y} \frac{\partial \omega_n}{\partial x} - \frac{\partial \psi_{n+1}}{\partial x} \frac{\partial \omega_n}{\partial y} = \nu \nabla^2 \omega_{n+1} \tag{3-33}$$

and

$$\nabla^2 \psi_{n+1} = -\omega_n \tag{3-34}$$

And the assembled system equations may be written as follows:

$$[K_\psi] \{\psi\}_{n+1} + \{S_\psi\}_n = \{0\} \tag{3-35}$$

$$[K_\omega] \{\omega\}_{n+1} + [K_{\dot{\omega}}] \{\dot{\omega}\}_{n+1} + \{S_\omega\}_n = \{0\} \tag{3-36}$$

where

$$K_{\psi_{ij}} = \sum_{e=1}^{M} \int_{A^{(e)}} \left(\frac{\partial N_i}{\partial x} \frac{\partial N_j}{\partial x} + \frac{\partial N_i}{\partial y} \frac{\partial N_j}{\partial y} \right) dA^{(e)} \qquad (3\text{-}37)$$

$$S_{\psi_i} = -\sum_{e=1}^{M} \int_{A^{(e)}} \omega_n^{(e)} N_i \, dA^{(e)} \qquad (3\text{-}38)$$

$$K_{\omega_{ij}} = \nu \, K_{\psi_{ij}} \qquad (3\text{-}39)$$

$$K_{\dot{\omega}_{ij}} = \sum_{e=1}^{M} \int_{A^{(e)}} N_i N_j \, dA^{(e)} \qquad (3\text{-}40)$$

$$S_{\omega_i} = \sum_{e=1}^{M} \int_{A^{(e)}} N_i \left(\frac{\partial \psi_{n+1}^{(e)}}{\partial y} \frac{\partial \omega_n^{(e)}}{\partial x} - \frac{\partial \psi_{n+1}^{(e)}}{\partial x} \frac{\partial \omega_n^{(e)}}{\partial y} \right) dA^{(e)} \qquad (3\text{-}41)$$

The iterative solution procedure starts by assuming an initial values for ω_n (i.e., ω_0). Then the ψ_{n+1} (i.e., ψ_1) is found from equation (3-35) and used as the source function to determine the ω_{n+1} (i.e., ω_1) from equation (3-36). This process is repeated until steady state is reached. Using a two-point finite difference formula, the term $\{ \dot{\omega} \}_{n+1}$ can be written as follows (25)

$$\{ \dot{\omega} \}_{n+1} = \left\{ \frac{\partial \omega}{\partial t} \right\}_{n+1} = \left\{ \frac{\omega_{n+1} - \omega_n}{\Delta t} \right\} \qquad (3\text{-}42)$$

so that equation (3-36) becomes

$$\left[[K_\omega] + \frac{1}{\Delta t} [K_{\dot{\omega}}] \right] \{ \omega \}_{n+1} = \frac{1}{\Delta t} [K_{\dot{\omega}}] \{ \omega \}_n - \{ S_\omega \}_n \qquad (3\text{-}43)$$

which can be solved at successive time steps for the column vector of nodal values of vorticity. The coefficient $\left[[K_\omega] + \frac{1}{\Delta t} [K_{\dot{\omega}}] \right]$ is symmetric, banded, and positive definite. Numerical solutions of equations (3-35) and (3-36) were obtained by the Gaussian elimination method. Cheng reports that the iteration procedure was always stable for sufficiently small Δt. As a guidline, Δt should be chosen so that

$$\Delta t < 0.1 (\Delta \ell)^2 Re \qquad (3\text{-}44)$$

where $\Delta 1$ is the charactertistic length of an element.(25,63,127)

It has been observed that such a procedure for time dependent problems does not fulfil the requirements of the calculus of variation. This is because during variations of vorticity, the term $\frac{\partial \omega}{\partial t}$ is treated as an invariance. This principle is, therefore, referred to as pseudo-variational principle.

3.4 Evaluation of the Matrices of the Elements

Equations (3-35) and (3-41) may now be explicitly evaluated using the definition of the interpolation functions. For linear triangular elements the following euqations can be obtained.

$$N_1^{(e)}(X,Y) = \frac{a_1 + b_1 X + c_1 Y}{2\Delta}$$

$$N_2^{(e)}(X,Y) = \frac{a_2 + b_2 X + c_2 Y}{2\Delta}$$

$$N_3^{(e)}(X,Y) = \frac{a_3 + b_3 X + c_3 Y}{2\Delta}$$

(3-45)

where
$$2\Delta = \begin{vmatrix} 1 & X_1 & Y_1 \\ 1 & X_2 & Y_2 \\ 1 & X_3 & Y_3 \end{vmatrix} = 2 \times (\text{area of triangule})$$

$$a_1 = X_2 Y_3 - X_3 Y_2$$

(3-46)

$$b_1 = Y_2 - Y_3$$

$$c_1 = X_3 - X_2$$

The other coefficients are obtained by cyclically permuting the subscripts. From equations (3-45) the following equations can be obtained

$$\frac{\partial N_i}{\partial X} = \frac{b_i}{2\Delta}$$

$$\frac{\partial N_i}{\partial Y} = \frac{c_i}{2\Delta} \qquad i = 1, 2, 3.$$

(3-47)

Substitution of equations (3-48) into equation (3-37) gives:

$$K_{wij} = \sum_{e=1}^{M} \int_{A^{(e)}} \left(\frac{\partial N_i}{\partial X} \frac{\partial N_j}{\partial X} + \frac{\partial N_i}{\partial Y} \frac{\partial N_j}{\partial Y} \right) dA^{(e)}$$

$$= \sum_{e=1}^{M} \int_{A^{(e)}} \left(\frac{b_i}{2\Delta} \frac{b_j}{2\Delta} + \frac{c_i}{2\Delta} \frac{c_j}{2\Delta} \right) dA^{(e)}$$

$$= \sum_{e=1}^{M} \frac{1}{4\Delta^2} \left(b_i b_j \int_{A^{(e)}} dA^{(e)} + c_i c_j \int_{A^{(e)}} dA^{(e)} \right)$$

$$= \sum_{e=1}^{M} \frac{1}{4\Delta} \left(b_i b_j + c_i c_j \right) \tag{3-48}$$

To evaluate

$$S_{\psi_i} = -\sum_{e=1}^{M} \int_{A^{(e)}} \omega_n^{(e)} N_i \, dA^{(e)} \tag{3-38}$$

Some special considerations are required. $\omega_n^{(e)}$ can be treated as a constant within the element, and from the integration formula (22,33,61,63) the following equation may be obtained.

$$S_{\psi_i} = -\sum_{e=1}^{M} \frac{\Delta}{3} \omega_n^{(e)} \tag{3-49}$$

or the term $\omega_n^{(e)}$ may be linearly interpolated in terms of its nodal values as (25,63,127)

$$\omega_n^{(e)} = \omega_{n1}^{(e)} N_1 + \omega_{n2}^{(e)} N_2 + \omega_{n3}^{(e)} N_3 \tag{3-50}$$

in which case equation (3-38) may be written as follows:

$$S_{\psi_i} = -\sum_{e=1}^{M} \int_{A^{(e)}} \left(\omega_{n1}^{(e)} N_1 N_i + \omega_{n2}^{(e)} N_2 N_i + \omega_{n3}^{(e)} N_3 N_i \right) dA^{(e)} \tag{3-51}$$

Again, employing the integration formula, the following equation may be obtained

(3-52)

$$\{S_\psi\}_n = -\sum_{e=1}^{M} \frac{\Delta}{12} \left\{ \begin{array}{c} 2\omega_{n1}^{(e)} + \omega_{n2}^{(e)} + \omega_{n3}^{(e)} \\ \omega_{n1}^{(e)} + 2\omega_{n2}^{(e)} + \omega_{n3}^{(e)} \\ \omega_{n1}^{(e)} + \omega_{n2}^{(e)} + 2\omega_{n3}^{(e)} \end{array} \right\}$$

Similarly, from the integration formula, equation (3-40) may be written as

$$K_{\omega_{ij}} = \sum_{e=1}^{M} \int_{A^{(e)}} N_i N_j \, dA^{(e)}$$

(3-40)

$$= \sum_{e=1}^{M} \frac{\Delta}{12}$$

(3-53)

Now the equation (3-41) is to be evaluated.

$$S_{\omega_i} = \sum_{e=1}^{M} \int_{A^{(e)}} N_i \left(\frac{\partial \psi_{n+1}^{(e)}}{\partial y} \frac{\partial \omega_n^{(e)}}{\partial x} - \frac{\partial \psi_{n+1}^{(e)}}{\partial x} \frac{\partial \omega_n^{(e)}}{\partial y} \right) dA^{(e)}$$

(3-41)

Substitution of equations (3-47) into equations (3-12) - (3-15) gives

$$\frac{\partial \psi_{n+1}^{(e)}}{\partial x} = \left\{ \frac{\partial N^{(e)}}{\partial x} \right\}^T \{\psi_{n+1}\} = \frac{b_1}{2\Delta} \psi_{1(n+1)} + \frac{b_2}{2\Delta} \psi_{2(n+1)} + \frac{b_3}{2\Delta} \psi_{3(n+1)}$$

$$\frac{\partial \psi_{n+1}^{(e)}}{\partial y} = \left\{ \frac{\partial N^{(e)}}{\partial y} \right\}^T \{\psi_{n+1}\} = \frac{c_1}{2\Delta} \psi_{1(n+1)} + \frac{c_2}{2\Delta} \psi_{2(n+1)} + \frac{c_3}{2\Delta} \psi_{3(n+1)}$$

(3-54)

$$\frac{\partial \omega_n^{(e)}}{\partial x} = \left\{ \frac{\partial N^{(e)}}{\partial x} \right\}^T \{\omega_n\} = \frac{b_1}{2\Delta} \omega_n + \frac{b_2}{2\Delta} \omega_{2n} + \frac{b_3}{2\Delta} \omega_{3n}$$

$$\frac{\partial \omega_n^{(e)}}{\partial y} = \left\{ \frac{\partial N^{(e)}}{\partial y} \right\}^T \{\omega_n\} = \frac{c_1}{2\Delta} \omega_n + \frac{c_2}{2\Delta} \omega_{2n} + \frac{c_3}{2\Delta} \omega_{3n}$$

Substitution of the last equations into equation (3-41) gives

$$
\begin{aligned}
Sw_i &= \sum_{e=1}^{M} \int_{A^{(e)}} N_i \left[\left(\frac{C_1}{2\Delta} \psi_{1(n+1)} + \frac{C_2}{2\Delta} \psi_{2(n+1)} + \frac{C_2}{2\Delta} \psi_{3(n+1)} \right) \right. \\
&\qquad \left(\frac{b_1}{2\Delta} \omega_{1n} + \frac{b_2}{2\Delta} \omega_{2n} + \frac{b_3}{2\Delta} \omega_{3n} \right) \\
&\qquad - \left(\frac{b_1}{2\Delta} \psi_{1(n+1)} + \frac{b_2}{2\Delta} \psi_{2(n+1)} + \frac{b_3}{2\Delta} \psi_{3(n+1)} \right) \\
&\qquad \left. \left(\frac{C_1}{2\Delta} \omega_{1n} + \frac{C_2}{2\Delta} \omega_{2n} + \frac{C_3}{2\Delta} \omega_{3n} \right) \right] dA^{(e)} \\
&= \sum_{e=1}^{M} \frac{1}{4\Delta^2} \left[\left(C_1 \psi_{1(n+1)} + C_2 \psi_{2(n+1)} + C_3 \psi_{3(n+1)} \right) \right. \\
&\qquad \left(b_1 \omega_{1n} + b_2 \omega_{2n} + b_3 \omega_{3n} \right) \\
&\qquad - \left(b_1 \psi_{1(n+1)} + b_2 \psi_{2(n+1)} + b_3 \psi_{3(n+1)} \right) \\
&\qquad \left. \left(C_1 \omega_{1n} + C_2 \omega_{2n} + C_3 \omega_{3n} \right) \right] \int_{A^{(e)}} N_i \, dA^{(e)} \\
&= \sum_{e=1}^{M} \frac{1}{12\Delta} \left[\left(C_1 \psi_{1(n+1)} + C_2 \psi_{2(n+1)} + C_3 \psi_{3(n+1)} \right) \right. \\
&\qquad \left(b_1 \omega_{1n} + b_2 \omega_{2n} + b_3 \omega_{3n} \right) \\
&\qquad - \left(b_1 \psi_{1(n+1)} + b_2 \psi_{2(n+1)} + b_3 \psi_{3(n+1)} \right) \\
&\qquad \left. \left(C_1 \omega_{1n} + C_2 \omega_{2n} + C_3 \omega_{3n} \right) \right]
\end{aligned}
$$

(3-55)

At present, the governing equations and elements can easily be incorporated into the computer programs.

3.5 Pressure and Velocity Distributions

Now the pressure and velocity distributions are to be calculated. It may be mentioned here that serious attention must be given to the choice of interpolation functions for pressure and stream functions. To achieve the same order of approximation for stream functions and pressure, the interpolation functions for stream functions should be higher by one order than the interpolation functions for pressure. So quadratic triangular elements were used for the stream functions, and linear triangular elements for the pressure (see Fig.3.2).
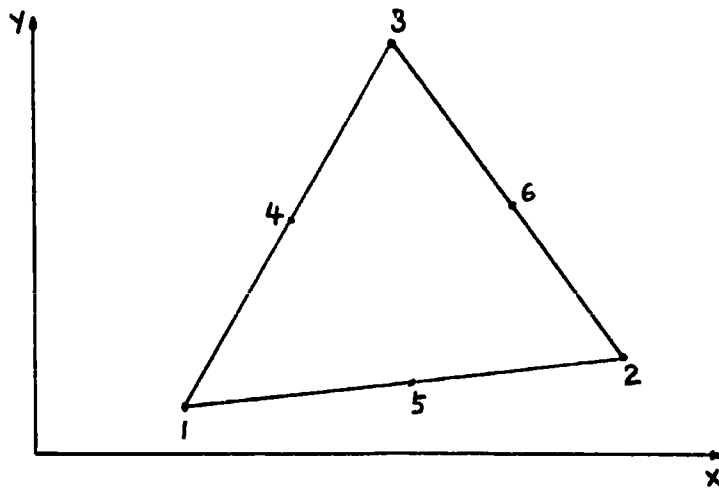
<u>Figure 3-2</u>:   Triangular element with corner nodes 1,2,3 and mid-edge nodes 4,5,6.

$$P^{(e)}(x,y,t) = \sum_{i=1}^{3} N_i^P(x,y)\, P_i(t)$$

$$\psi^{(e)}(x,y,t) = \sum_{i=1}^{6} N_i(x,y)\, \psi_i(t)$$

$$N_1 = L_1^2 - L_1(L_2+L_3)$$

$$N_2 = L_2^2 - L_2(L_3+L_1)$$

$$N_3 = L_3^2 - L_3(L_1+L_2)$$

$$N_4 = 4L_1L_3$$

$$N_5 = 4L_1L_2$$

$$N_6 = 4L_2L_3$$

$$N_i^P = L_i = \text{natural coordinates}$$

$$P^{(e)}(x,y,t) = \sum_{i=1}^{3} N_i^P(x,y) P_i(t)$$

$$= \{N^P(x,y)\}^T \{P(t)\} \tag{3-56}$$

and

$$\psi^{(e)}(x,y,t) = \sum_{i=1}^{6} N_i(x,y) \psi_i(t)$$

$$= \{N(x,y)\}^T \{\psi(t)\} \tag{3.57}$$

where

$$
\begin{aligned}
N_1 &= L_1^2 - L_1(L_2 + L_3) \\
N_2 &= L_2^2 - L_2(L_3 + L_1) \\
N_3 &= L_3^2 - L_3(L_1 + L_2) \\
N_4 &= 4 L_1 L_3 \\
N_5 &= 4 L_1 L_2 \\
N_6 &= 4 L_2 L_3 \\
N_i^P &= L_i = \text{natural coordinates}
\end{aligned} \tag{3-58}
$$

The gradients of $P^{(e)}$ are

$$\frac{\partial P^{(e)}}{\partial x} = \{\frac{\partial N^P}{\partial x}\}^T \{P(t)\}$$

$$\frac{\partial P^{(e)}}{\partial y} = \{\frac{\partial N^P}{\partial y}\}^T \{P(t)\} \tag{3-59}$$

Substitution of equations (3-59) and (3-56) into equation (2-41) gives:

$$\phi_3 = \frac{1}{2} \int_{A^{(e)}} \left(\left[\{\frac{\partial N^P}{\partial x}\}^T \{P(t)\} \right]^2 + \left[\{\frac{\partial N^P}{\partial y}\}^T \{P(t)\} \right]^2 \right) dA^{(e)}$$

$$- \int_{A^{(e)}} \rho \left(\{N^P\}^T \{P(t)\} \right) Q \, dA^{(e)} \tag{3-60}$$

Minimisation of the functional gives

$$\frac{\partial \phi_3(P)}{\partial P_i} = \int_{A^{(e)}} \left(\left[\{\frac{\partial N^P}{\partial x}\}^T \{P(t)\} \right] \frac{\partial N_i^P}{\partial x} \right.$$

$$\left. + \left[\{\frac{\partial N^P}{\partial y}\}^T \{P(t)\} \right] \frac{\partial N_i^P}{\partial y} \right) dA^{(e)} - \int_{A^{(e)}} \rho Q N_i^P dA^{(e)} \tag{3-61}$$

$$= 0$$

The last equation may be written concisely as

$$[K_P]^{(e)}\{P(t)\}^{(e)} + \{S_P\}^{(e)} = \{0\} \qquad (3\text{-}62)$$

where

$$K_{P_{ij}}^{(e)} = \int_{A^{(e)}} \left(\frac{\partial N_i^P}{\partial x} \frac{\partial N_j^P}{\partial x} + \frac{\partial N_i^P}{\partial y} \frac{\partial N_j^P}{\partial y} \right) dA^{(e)} \qquad (3\text{-}63)$$

$$S_{P_i}^{(e)} = -\int_{A^{(e)}} P N_i^P Q\, dA^{(e)} \qquad (3\text{-}64)$$

$$Q = -2\left[\frac{\partial^2 \psi}{\partial x^2} \frac{\partial^2 \psi}{\partial y^2} - \left(\frac{\partial^2 \psi}{\partial x \partial y} \right)^2 \right] \qquad (3\text{-}65)$$

Referring to equations (3-57) and (3-58), the derivatives in equation (3-65) may be written as

$$\frac{\partial^2 \psi}{\partial x^2} = \left\{ \frac{\partial^2 N}{\partial x^2} \right\}^T \{\psi\} \qquad (3\text{-}66)$$

$$= \frac{b_1}{2\Delta^2}(b_1 - b_2 - b_3)\,\psi_1$$

$$+ \frac{b_2}{2\Delta^2}(b_2 - b_3 - b_1)\,\psi_2$$

$$+ \frac{b_3}{2\Delta^2}(b_3 - b_1 - b_2)\,\psi_3$$

$$+ \frac{2b_1 b_3}{\Delta^2}\,\psi_4 + \frac{2b_1 b_2}{\Delta^2}\,\psi_5 + \frac{2b_2 b_3}{\Delta^2}\,\psi_6$$

$$\frac{\partial^2 \psi}{\partial y^2} = \left\{ \frac{\partial^2 N}{\partial y^2} \right\}^T \{\psi\}$$

$$= \frac{c_1}{2\Delta^2}(c_1 - c_2 - c_3)\,\psi_1$$

$$+ \frac{C_2}{2\Delta^2} (C_2 - C_3 - C_1) \psi_2 \qquad (3\text{-}67)$$

$$+ \frac{C_3}{2\Delta^2} (C_3 - C_1 - C_2) \psi_3$$

$$+ \frac{2C_1C_3}{\Delta^2} \psi_4 + \frac{2C_1C_2}{\Delta^2} \psi_5 + \frac{2C_2C_3}{\Delta^2} \psi_6$$

and

$$\frac{\partial^2 \psi}{\partial x \partial y} = \left\{ \frac{\partial^2 N}{\partial x \partial y} \right\}^T \{ \psi \}$$

$$= \frac{1}{4\Delta^2} (2C_1 b_1 - C_1 b_3 - C_3 b_1 - C_1 b_2 - C_2 b_1) \psi_1$$

$$+ \frac{1}{4\Delta^2} (2C_2 b_2 - C_2 b_3 - C_3 b_2 - C_1 b_2 - C_2 b_1) \psi_2$$

$$+ \frac{1}{4\Delta^2} (2C_3 b_3 - C_3 b_1 - C_1 b_3 - C_3 b_2 - C_2 b_3) \psi_3$$

$$+ \frac{1}{\Delta^2} (C_1 b_3 + C_3 b_1) \psi_4$$

$$+ \frac{1}{\Delta^2} (C_1 b_2 + C_2 b_1) \psi_5 \qquad (3\text{-}68)$$

$$+ \frac{1}{\Delta^2} (C_2 b_3 + C_3 b_2) \psi_6$$

To simplify these calculations as much as possible, the following equations are assumed.

$$\psi_4 = \frac{\psi_1 + \psi_3}{2}$$

$$\psi_5 = \frac{\psi_2 + \psi_1}{2} \qquad (3\text{-}69)$$

$$\psi_6 = \frac{\psi_2 + \psi_3}{2}$$

For a solution domain of M elements, the system equations are of the form

$$[K_P] \{P(t)\} + \{S_P\} = \{0\} \qquad (3\text{-}70)$$

which can be solved for the column vector of nodal values of pressure, $\{P\}$.

Now the velocity distribution is to be calculated. The most usual procedure of describing incompressible velocities is by using the following equations (3-71) in a direct manner. It is straightforward. At present, an alternative way is presented here. For a two-dimensional fluid flow, the velocity components u and v may be expressed in terms of a stream function ψ (x,y) as

$$u \quad = \quad \frac{\partial \psi}{\partial y}$$

$$v \quad = \quad - \frac{\partial \psi}{\partial x} \tag{3-71}$$

The variational functionals for equations (3-71) can be written as follows:

$$\phi_4(u) \quad = \quad \int_{A^{(e)}} \left(\frac{1}{2} u^2 - \frac{\partial \psi}{\partial y} u \right) dA^{(e)}$$

$$\phi_5(v) \quad = \quad \int_{A^{(e)}} \left(\frac{1}{2} v^2 + \frac{\partial \psi}{\partial x} v \right) dA^{(e)} \tag{3-72}$$

The interpolation functions for stream functions should also be higher by one order than Ni^u and Ni^v, the interpolation functions for u and v. So a quadratic interpolation function for stream function and a linear one for fluid velocity components are adopted.

$$u^{(e)} \quad = \quad \sum_{i=1}^{3} N_i^u u_i$$

$$v^{(e)} \quad = \quad \sum_{i=1}^{3} N_i^v v_i \tag{3-73}$$

$$\psi^{(e)} \quad = \quad \sum_{i=1}^{6} N_i \psi_i$$

where

$$N_i^u \quad = \quad N_i^v \quad = \quad L_i \quad = \quad \text{natural coordinates}$$

$$N_1 \quad = \quad L_1^2 - L_1(L_2 - L_3)$$

$$N_2 = L_2^2 - L_2(L_3 + L_1)$$

$$N_3 = L_3^2 - L_3(L_1 + L_2)$$

$$N_4 = 4L_1L_3$$

$$N_5 = 4L_1L_2$$

$$N_6 = 4L_2L_3$$

(3-74)

Substitution of equations (3-73) and (3-74) into equations (3-72) gives:

$$\phi_4(u) = \frac{1}{2}\int_{A^{(e)}}\left(\left[\{N^u\}^T\{u\}\right]^2 - 2\left[\left\{\frac{\partial N}{\partial y}\right\}^T\{\psi\}\right]\right.$$

$$\left.\left[\{N^u\}^T\{u\}\right]\right)dA^{(e)}$$

(3-75)

$$\phi_5(v) = \frac{1}{2}\int_{A^{(e)}}\left(\left[\{N^v\}^T\{v\}\right]^2 + 2\left[\left\{\frac{\partial N}{\partial x}\right\}^T\{\psi\}\right]\right.$$

$$\left.\left[\{N^v\}^T\{v\}\right]\right)dA^{(e)}$$

Minimization of equations (3-75) gives

$$\frac{\partial\phi_4(u)}{\partial u_i} = \int_{A^{(e)}}\{N^u\}^T\{u\}N_i^u\,dA^{(e)} - \int_{A^{(e)}}\left\{\frac{\partial N}{\partial y}\right\}^T\{\psi\}N_i^u\,dA^{(e)}$$

$$= 0$$

and

(3-76)

$$\frac{\partial\phi_5(v)}{\partial v_i} = \int_{A^{(e)}}\{N^v\}^T\{v\}N_i^v\,dA^{(e)} + \int_{A^{(e)}}\left\{\frac{\partial N}{\partial x}\right\}^T\{\psi\}N_i^v\,dA^{(e)}$$

$$= 0$$

From equations (3-76), the element equations may be written as follows:

$$[R_u]^{(e)}\{u\}^{(e)} - [Q_u]^{(e)}\{\psi\}^{(e)} = \{0\}$$

(3-77)

$$[R_v]^{(e)}\{v\}^{(e)} + [Q_v]^{(e)}\{\psi\}^{(e)} = \{0\}$$

where

$$R^{(e)}_{u_{ij}} = \int_{A^{(e)}} N_i^u N_j^u dA^{(e)} \qquad (3\text{-}78)$$

$$R^{(e)}_{v_{ij}} = \int_{A^{(e)}} N_i^v N_j^v dA^{(e)}$$

$$Q^{(e)}_{u_{ij}} = \int_{A^{(e)}} \frac{\partial N_j}{\partial y} N_i^u dA^{(e)}$$

$$Q^{(e)}_{v_{ij}} = \int_{A^{(e)}} \frac{\partial N_j}{\partial x} N_i^v dA^{(e)} \qquad (3\text{-}79)$$

The assembled system equations become

$$[R_u]\{u\} - [Q_u]\{\psi\} = \{0\} \qquad (3\text{-}80)$$

$$[R_v]\{v\} + [Q_v]\{\psi\} = \{0\}$$

which can be solved for the column vector of nodal values of velocity components u and v. Matrices $[Q_u]$ and $[Q_v]$ have been evaluated using equation (3-73), (3-74) and (3-79). Matrices $[Q_u]$ and $[Q_v]$ are given in Appendix A.

CHAPTER 4   BOUNDARY CONDITIONS AND NUMERICAL PROCEDURES

4.1 Boundary Conditions

In this section, some general ideas about boundary conditions are discussed. A fluid flow problem, governed by a system of partial differential equation, is defined only when a proper set of initial and/ or boundary conditions is given. The boundary conditions are such an important part of the definition of a problem that the patterns of two flow fields can be completely different from one another simply due to some differences in the flow boundaries, in spite of the fact that both flow obey the same system of indefinite differential equations. One cannot exaggerate the importance of the effects that boundary conditions have on the fluid flow analysis. In mixed initial- and boundary - value problems major troubles must arise if the boundary conditions are not properly handled (33,92,128).

Through theoretical considerations and numerical experiments, it is found that even though the patterns of two fluid flow fields are the same, boundary conditions may still be different. It seems that boundary conditions depend not only on fluid patterns but also on the kind of finite element formulation used or on the kind of field variables employed, and on which kind of interpolation function adopted. For example, to solve the same fluid flow problem, if velocity and pressure are used as field variables, then the boundary conditions for this formulation are different from those when stream function and vorticity are used as field variables. For another example, when a fluid flow problem is solved by using a velocity-pressure formulation, unless a high order element is used the value of second derivative $\frac{\partial^2 U}{\partial X^2}$ has to be assumed zero which normally would be valid for creeping flow only. This seems to be equivalent to a high order element being used and $\frac{\partial^2 U}{\partial X^2} = 0$ boundary condition being assumed in this interpolation function at the same time (62,132   ).

The boundary conditions for a stream function-vorticity formulation problem are of three general types; 1) the specifications of the values which stream function or vorticity must assume along the boundary; 2) the specifications of the values of the component of the gradient of ψ or ω at the normal to the boundary; 3) the provision of some algebraic relation which connects the value of ψ or ω to the values of their normal components along the boundary (16,55). In fact, it is usual for values to be specified at some parts of a boundary and for gradients to be specified at other parts (see example one, Chapter 6). When several differential equations are to be solved simultaneously, there is no need for the boundary conditions for each equation to be of the same type (55, 63).

In example one (see Section 6.1), since the flow is considered to be fully developed at the downstream end, the gradients of stream function and vorticity with respect to the flow direction should vanish at the boundary. These provide the normal boundary conditions at the downstream end. There is no need to specify the values of stream function and vorticity at this boundary. (21,25,55,63,100,110,127)

The mixed type boundary value problem, such as that appearing in example one, causes no difficulty in itself, provided a scheme can be found for specifying boundary conditions associated with first derivatives of stream function and vorticity. For example, to impose the boundary conditions for the example one, the relations $\frac{\partial \psi}{\partial X} = \frac{\partial \omega}{\partial X} = 0$ can just be incorporated into the element stiffness matrix. This will give an alternate form for the element stiffness matrix which can then be used for elements having node on downstream edge. The disadvantage is that such elements must then be incorporated into the computer programe and used as need arises. This scheme has been used in this computer program. An alternate way is to consider the boundary conditions as constraints.

The set of equations is expressed as the gross assemblage stiffness equation. The detailed procedure can be found from the papers of Bratanow (16,18) or Martin (88).

The 'natural' boundary conditions are somewhat arbitrary, since there is little agreement in the field of fluid mechanics as to what the proper ones are. The choice is therefore made on the basis of practicality (33,100). Usually, the 'rigid' boundary conditions are used. And the programe is left to seek its own approximation of the 'natural' boundary conditions (see Chapter 6). (100)

A rigid wall may be either of two types, no-slip or free-slip. The latter may be considered to represent a plane of symmetry, rather than a true wall (58,128). In the examples to be presented, no-slip boundary condition would be considered.

To satisfy the condition of no-slip at solid walls, the normal and tangential gradients of stream function must vanish at these boundaries. The tangential conditions are satisfied by setting stream function constant along these boundaries. However special attention is required to determine the boundary formulae for the normal conditions. To determine the boundary values for the wall vorticity, application of the no-slip boundary condition alone is not enough. At a point (Xo,Yo) on the wall, the vorticity may be calculated from the relation. (25,63,121)

$$\omega(X_0, Y_0) = -\frac{\partial^2 \psi}{\partial n^2}(X_0, Y_0) \tag{4-1}$$

where n is the coordinate normal to the wall. Using a Taylor series expansion, at a point (X1,Y1) along the normal direction, a small distance from the wall, the following equation may be obtained.

$$\psi(X_1, Y_1) = \psi(X_0, Y_0) + n\frac{\partial \psi}{\partial n}(X_0, Y_0) + \frac{n^2}{2}\frac{\partial^2 \psi}{\partial n^2}(X_0, Y_0) \tag{4-2}$$

Since the no-slip condition dictates that

$$\frac{\partial \psi}{\partial n}(x_0, y_0) = 0 \qquad (4\text{-}3)$$

then the vorticity on the wall may be calculated by

$$\omega(x_0, y_0) = \frac{2}{n^2}\left[\psi(x_0, y_0) - \psi(x_1, y_1)\right] \qquad (4\text{-}4)$$

Wall vorticity is then given in terms of the stream function evaluated at the wall and a small distance away from the wall. If (x_1, y_1) is not a nodal point, the stream function $\psi(x_1, y_1)$ may be obtained by interpolation between stream functions of the neighbouring nodal points of (x_1, y_1) (25,63).

There are several other ways to compute the surface vorticity, which can easily be found (4,33,110,128). Equation (4-4) is called the first-order one-sided difference formula. This formula gave numerical results in excellent agreement with the exact solution (128). It is found that the second-order one-sided difference formula sometimes led to unstable results. The detailed discussions may be found from Wu's latest paper (128). To play safe, this program employs the first-order one-sided difference formula. When using this formula, use of a finer mesh round the corners is to be encouraged. This is not only because there are big variations of values of stream function round the corners but also because it is hoped to force the effects of the corners to spread into the fluid in every direction, by using a finer mesh in regions adjacent corners (see Sec.7.6.1.).

For each system of equations there are a number of sufficient and necessary boundary conditions. For example, for a viscous flow the condition of no-slip is sufficient to determine the flow field. No other condition may be imposed on the rigid wall (92,100).

When a problem of viscous flow is treated by a numerical technique. A certain mesh is used.   At the interior mesh-points the governing equations are substituted by a matrix equation.   At each computational step, information is transmitted from each point to its neighbouring points via the numerical computation.   In this way, the boundary mesh-points influence their neighbours and transmit the effects of the boundary conditions into the flow field.   Moretti (92) maintained that if boundary conditions are not proceeded properly, the risk of over-specifying the boundary conditions themselves is faced and, in all probability these overspecified boundary conditions will not be consistent with the natural of the boundary and the limiting forms of the equations of motion (62,92).

It is not possible to isolate any portion of a fluid field and obtain the solution in only that portion.   The difficulty arises from the boundary conditions.   It is easier to deal with a fluid flow problem on a larger or entire flow field than just on a part of the whole flow field.

Some more detailed discussions about boundary conditions will be presented in example problems later.

4.2 Numerical Procedures

The present scheme for solution of the assembled system equations (3-35) and (3-43) uses an iterative method to obtain self-consistent stream function and vorticity fields.

The solution to equation (3-35) requires specification of stream function or its normal derivatives on all boundaries.   The initial vorticity is not known anywhere.   Usually an initial guess of zero vorticity is often appropriate.   Equation (3-35) is then able to be solved for the stream function.   Using the results of stream function, the wall vorticity can be obtained from equation (4-4).   And then we can use values of wall vorticities and stream functions to solve equation

(3-43) for the vorticity. The velocity boundary conditions provide derivative boundary conditions on stream function. On the boundary, stream function and the normal and tangential derivatives may all be specified. After the vorticity field has been determined, the values of vorticity can be treated as a source function, and solve the stream function for the next time instant from equation (3-35).

Once the new stream function field has been determined the wall vorticity for the new time instant can be obtained by solving equation (4-4) again. Then equation (3-43) is resolved by adjusting the boundary vorticity. This procedure is repeated until a self-consistent stream function and vorticity field is obtained. This procedure not only circumvents the nonlinearity of the governing equation but also leads to a linear algebraic system (25,63).

Employing the foregoing procedure, solutions of stream function and vorticity can be obtained for creeping flow. Once a convergent Stokes solution was determined, this solution can be used as the initial conditions of vorticity in the calculation for a solution of the governing equations at a small Reynolds number. These solutions of stream function and vorticity for a small Reynolds number are considered as the initial conditions when the numerical solution of vorticity and stream function for a bit higher new Reynolds number are solved. This procedure is repeated such that the solution at a lower Reynolds number is used as the initial conditions for the higher Reynolds number until the solutions of stream function and vorticity for a desired Reynolds number are reached (25,63).

To get an accurate result, the smaller the increment of Reynolds number is, the better.

The procedures of the numerical solution are summarised as follows:

(1)   Define suitable boundary conditions for both   These boundary conditions
         stream function and vorticity.   must be kept through this
                                                                 calculation.

(2)   Assume an initial vorticity distribution.

(3)   Compute stream function from equation (3-35).

(4)   Find the boundary vorticity values from equation (4-4)

(5)   Solve the vorticity of Stokes flow for the next time instant.

       from equation (3-43).

Convergence ?

(6)   Read a new small Reynolds number.

(7)   Treat the vorticity field of Stokes flow as a new

       initial vorticity distribution.

(8)   Compute stream function from equation (3-35).

(9)   Find the boundary vorticity values from equation (4-4).

(10) Solve the vorticity for the next time instant

       at the small Reynolds number from equation (3-43).

Convergence ?

(11) Read a bit higher Reynolds number

       Repeat the foregoing procedure until desired Reynolds

       number is reached.

       ----

(12) Compute pressure distribution.

(13) Compute velocity distribution.

(14) Solution is complete.

Chapter 5   COMPUTER WORK

5.1 Introduction

As far as actual applications of the finite element method are
concerned, it seems that computer programs play most important and
practical roles.   The main portion of this chapter is to explain how
a digital computer can solve fluid flow problems by the finite element
method.   Some descriptions and a simplified flow diagram for the finite
element work are also presented.   The analysis was programmed in
'FORTRAN IV' computer language.

The flow chart (Figure 5-1) shows how the system is used to solve
a fairly straightforward problem.

Considering the flow chart of Figure 5-1, the boundary conditions
and other data are read in at once before any of the calculation is
commenced.   If the tail end of the data is incorrect then this error
will be detected before a significant amount of computer time has been
wasted.

The input data is divided into the following main sections.

Control data: number of nodes, number of elements, and
                       maximum value of iterations.

Coordinate data: the coordinates of the nodes.

Element data: description of the topology of the element
                       interconnections.

Initial and boundary
conditions: description of the problems.

The chief purpose of the drawing scheme is to check the coordinate
data of elements and the element topology.   From a drawing of the finite
element mesh, mistakes in the data are easily recognisable.

5.2 Some Descriptions

During the early stage, when the programmes were developing, serious
problems of computer storage and the addresses for arrays were faced.
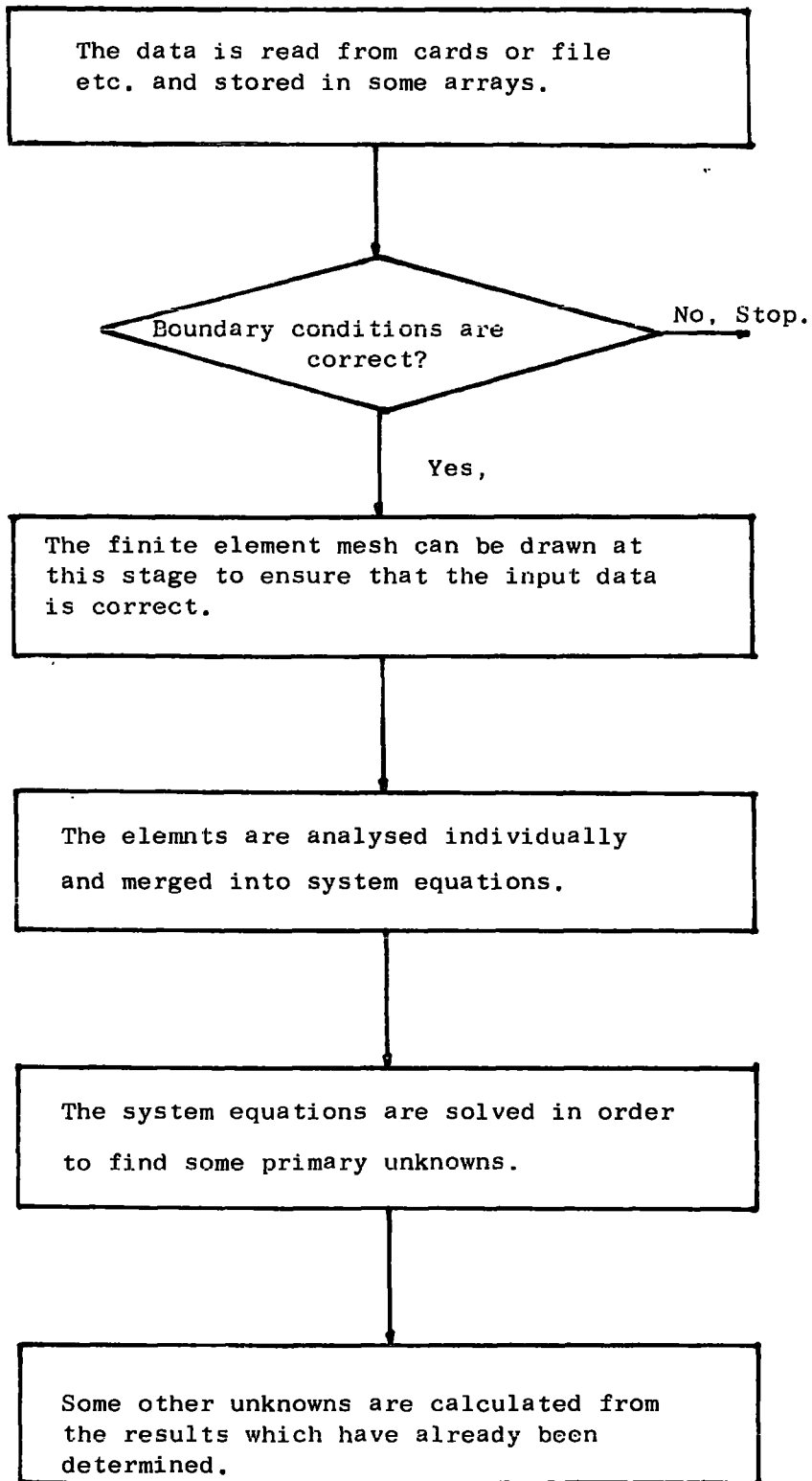To overcome these difficulties, a sub-routine called DYNMIC supplied by

```
┌─────────────────────────────────────────┐
│   The data is read from cards or file    │
│   etc. and stored in some arrays.        │
└─────────────────────────────────────────┘
                    │
                    ▼
            ╱ Boundary conditions are ╲  ──────  No, Stop.
            ╲        correct?         ╱
                    │
                  Yes,
                    ▼
┌─────────────────────────────────────────┐
│   The finite element mesh can be drawn at│
│   this stage to ensure that the input data│
│   is correct.                            │
└─────────────────────────────────────────┘
                    │
                    ▼
┌─────────────────────────────────────────┐
│   The elemnts are analysed individually  │
│   and merged into system equations.      │
└─────────────────────────────────────────┘
                    │
                    ▼
┌─────────────────────────────────────────┐
│   The system equations are solved in order│
│   to find some primary unknowns.         │
└─────────────────────────────────────────┘
                    │
                    ▼
┌─────────────────────────────────────────┐
│   Some other unknowns are calculated from│
│   the results which have already been    │
│   determined.                            │
└─────────────────────────────────────────┘
```

Figure 5.1.   Simplified Flow Chart

the computer unit of the University has been used.

This method makes use of the fact that FORTRAN allows the definition of arrays with unknown dimensions within sub-programmes. The old main program is converted into a subroutine having as arguments the name and size of each array requiring dynamic allocation. This subroutine is called MAINPR. A small main program is required to write. This small main program will read in the array dimensions and calculate the space required for each array and pass this information to the subroutine DYNMIC, the arguments of which are identical to those for MAINPR. Subroutine DYNMIC acquires space for the arrays and calls MAINPR passing the arguments given to it, having inserted the correct addresses for the arrays.

The subroutine SOLMIX is used for solving matrix equations. Within SOLMIX a subroutine SOLVE is called. The subroutine SOLVE solves this kind of equation.

$$A(I,J) \quad x \quad X(J) \quad = \quad C(I)$$

for X(J) by Gaussian elimination scheme. There are a lot of this kind of present sub-program which can be used. The subroutines SOLMIX and SOLVE used here were written by Professor J.F.Booker of Cornell University (63).

The main advantages of the finite element program are as follows:

(1) Complicated boundary conditions can be involved without difficulty.

(2) Changing the type of boundary condition requires only the change of input data, and there is no need to change the computer programmes.

(3) The convergence of the calculation can be observed by printing the values at selected points after each iteration.

The main disadvantage is that the achievement of a successful solution depends on the choise of the correct convergence parameters. For some problems, it might take a long time by trial and error before what kinds of values of parameters are the most appropriate to use are known. So it

is important to write down and keep the information about these control

factors. It may be useful when this program is used later.

5.3 Simplified Flow Diagram for the Finite-Element Programmes

A flow chart outlining the finite element procedure is presented as

follows:

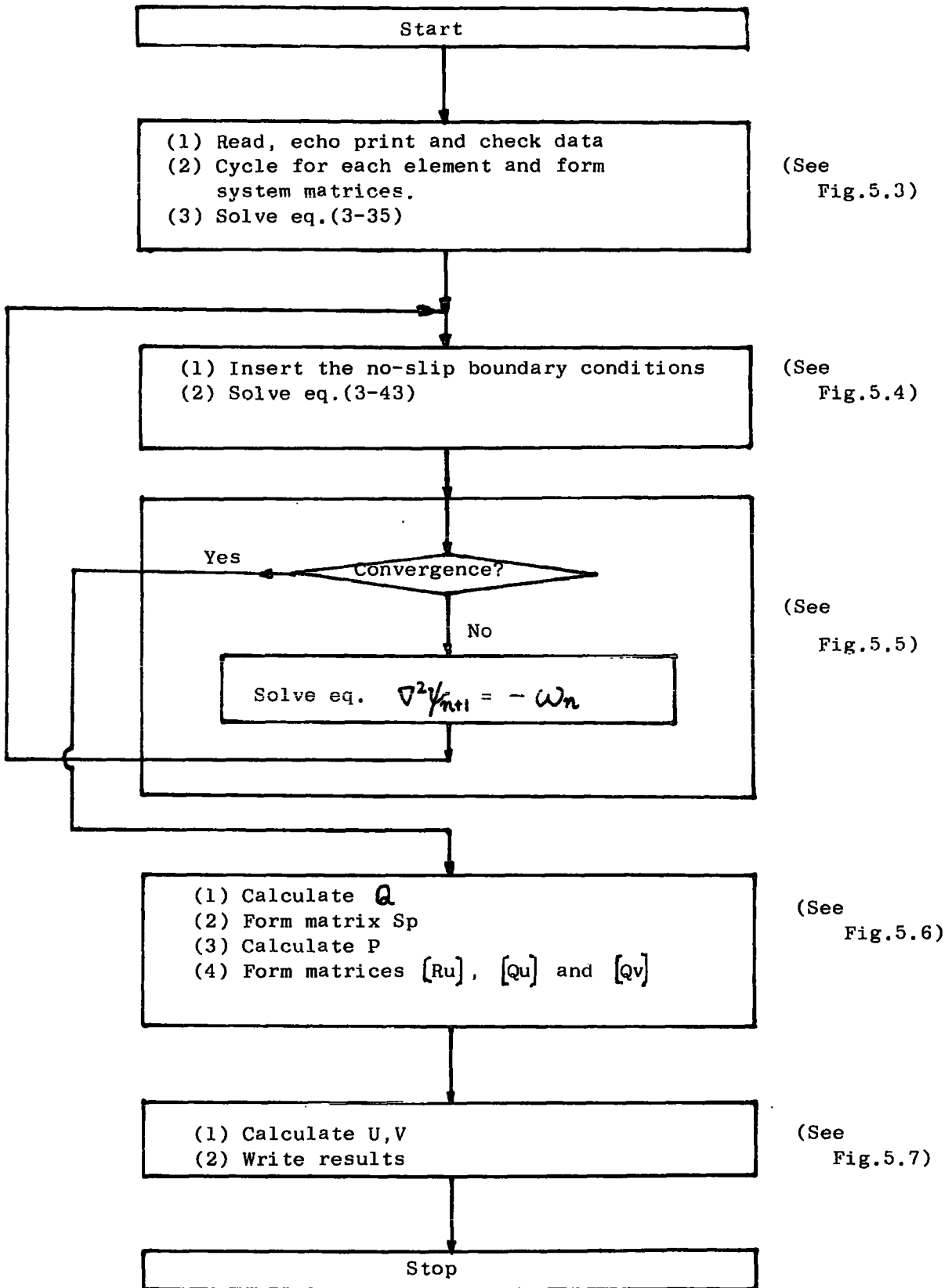Finite Element Analysis of Incompressible Unsteady Viscous Flow.

```
┌─────────────────────────────────────────────┐
│                    Start                     │
└─────────────────────────────────────────────┘
                      │
                      ▼
┌─────────────────────────────────────────────┐
│ (1) Read, echo print and check data         │        (See
│ (2) Cycle for each element and form         │        Fig.5.3)
│     system matrices.                        │
│ (3) Solve eq.(3-35)                         │
└─────────────────────────────────────────────┘
                      │
                      ▼
┌─────────────────────────────────────────────┐
│ (1) Insert the no-slip boundary conditions  │        (See
│ (2) Solve eq.(3-43)                         │        Fig.5.4)
└─────────────────────────────────────────────┘
                      │
                      ▼
┌─────────────────────────────────────────────┐
│    Yes         ◇ Convergence? ◇             │        (See
│                      │                      │
│                      │ No                   │        Fig.5.5)
│          ┌──────────────────────────┐       │
│          │ Solve eq. $\nabla^2 \psi_{n+1} = -\omega_n$ │
│          └──────────────────────────┘       │
└─────────────────────────────────────────────┘
                      │
                      ▼
┌─────────────────────────────────────────────┐
│ (1) Calculate $\mathbf{Q}$                  │        (See
│ (2) Form matrix Sp                          │        Fig.5.6)
│ (3) Calculate P                             │
│ (4) Form matrices $[Ru]$, $[Qu]$ and $[Qv]$ │
└─────────────────────────────────────────────┘
                      │
                      ▼
┌─────────────────────────────────────────────┐
│ (1) Calculate U,V                           │        (See
│ (2) Write results                           │        Fig.5.7)
└─────────────────────────────────────────────┘
                      │
                      ▼
┌─────────────────────────────────────────────┐
│                    Stop                      │
└─────────────────────────────────────────────┘
```
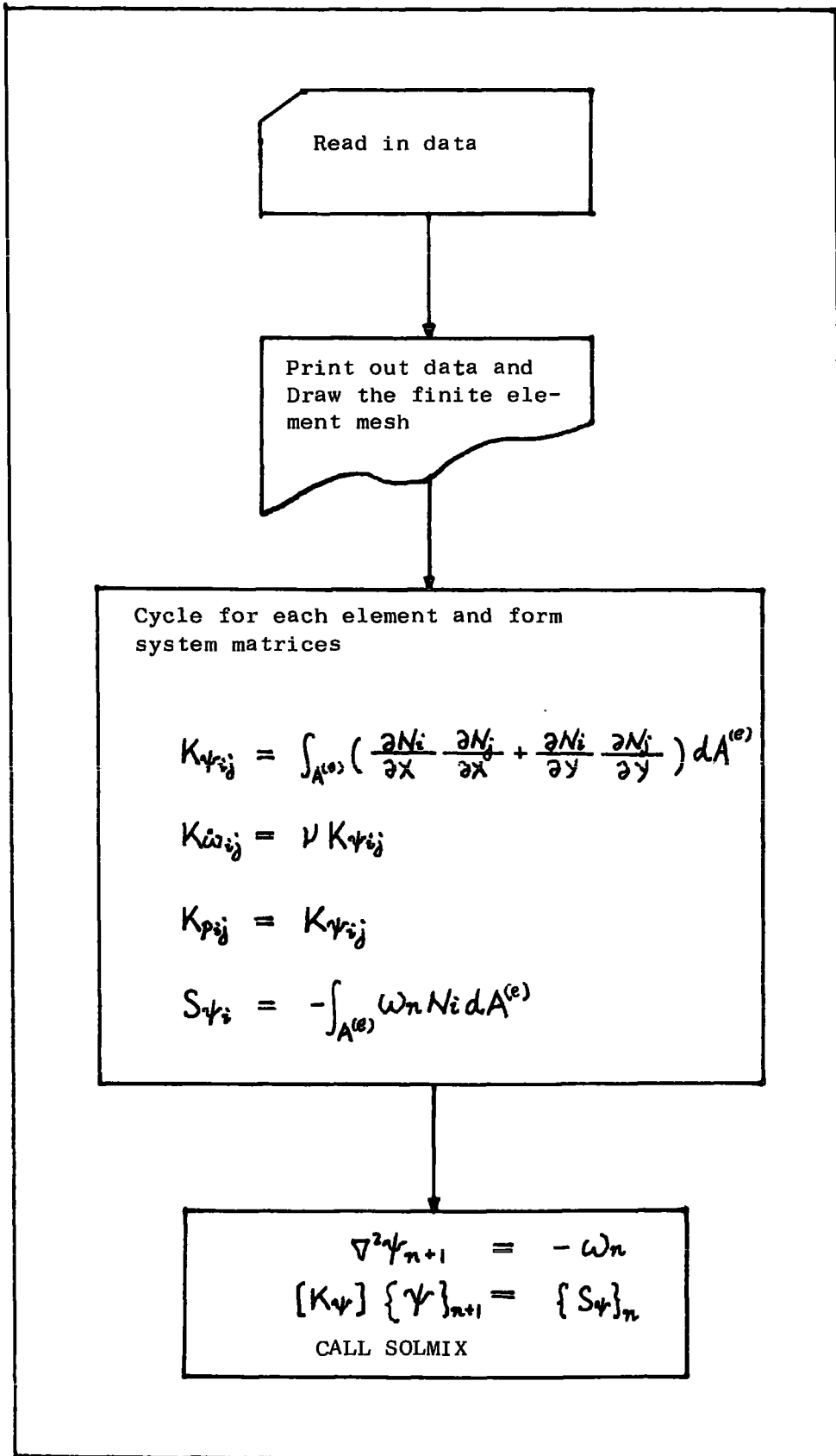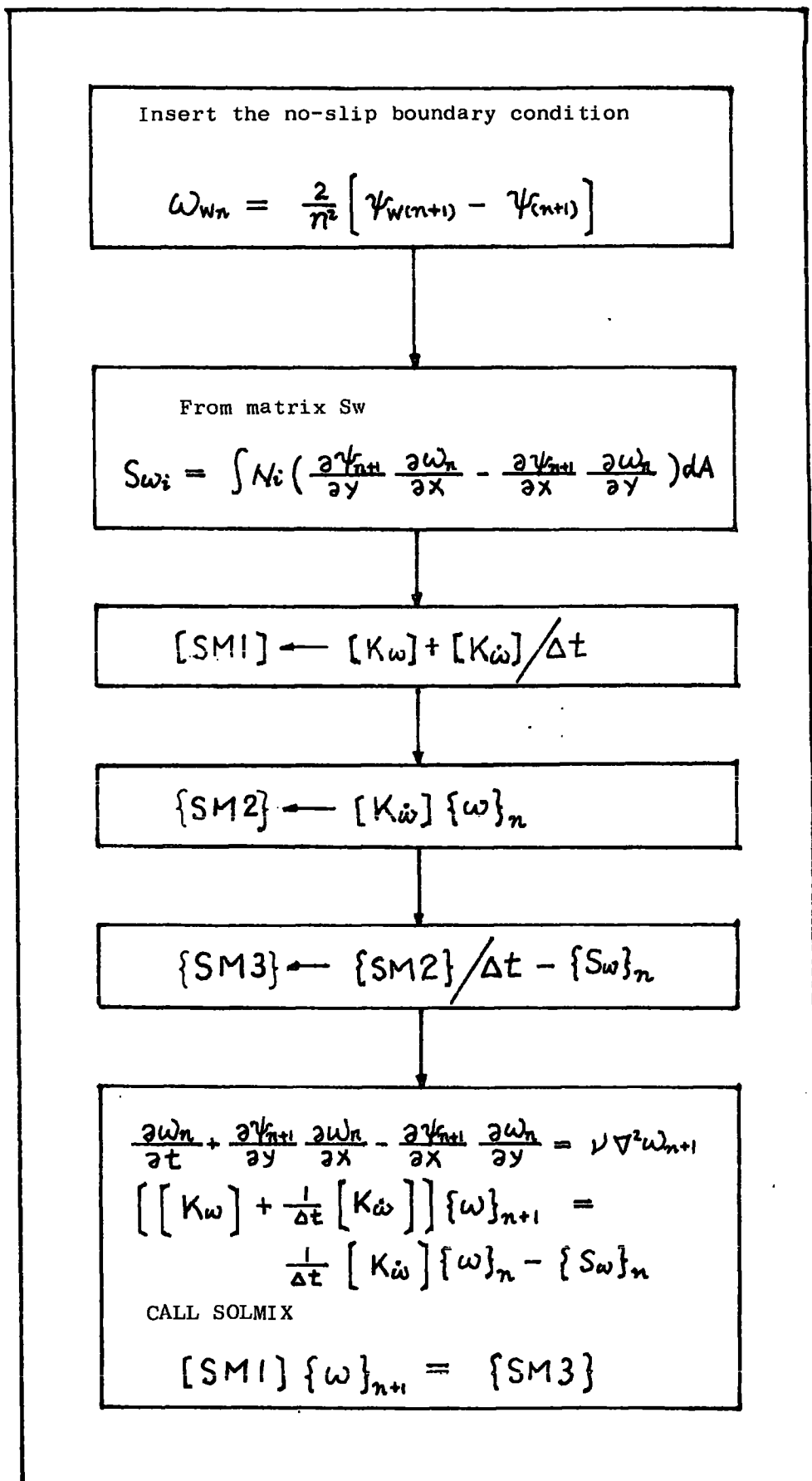
Figure 5.2

Figure 5.3

Read in data

Print out data and
Draw the finite ele-
ment mesh

Cycle for each element and form
system matrices

$$K_{\psi_{ij}} = \int_{A^{(e)}} \left(\frac{\partial N_i}{\partial x} \frac{\partial N_j}{\partial x} + \frac{\partial N_i}{\partial y} \frac{\partial N_j}{\partial y} \right) dA^{(e)}$$

$$K_{\omega_{ij}} = \nu K_{\psi_{ij}}$$

$$K_{p_{ij}} = K_{\psi_{ij}}$$

$$S_{\psi_i} = -\int_{A^{(e)}} \omega_n N_i \, dA^{(e)}$$

$$\nabla^2 \psi_{n+1} = -\omega_n$$

$$[K_\psi] \{\psi\}_{n+1} = \{S_\psi\}_n$$

CALL SOLMIX

Figure 5.4

Insert the no-slip boundary condition

$$\omega_{wn} = \frac{2}{n^2}\left[\psi_{w(n+1)} - \psi_{(n+1)}\right]$$

From matrix Sw

$$Sw_i = \int N_i\left(\frac{\partial\psi_{n+1}}{\partial y}\frac{\partial\omega_n}{\partial x} - \frac{\partial\psi_{n+1}}{\partial x}\frac{\partial\omega_n}{\partial y}\right)dA$$

$$[SM1] \longleftarrow [K\omega] + [K\dot{\omega}]/\Delta t$$

$$\{SM2\} \longleftarrow [K\dot{\omega}]\{\omega\}_n$$

$$\{SM3\} \longleftarrow \{SM2\}/\Delta t - \{Sw\}_n$$

$$\frac{\partial\omega_n}{\partial t} + \frac{\partial\psi_{n+1}}{\partial y}\frac{\partial\omega_n}{\partial x} - \frac{\partial\psi_{n+1}}{\partial x}\frac{\partial\omega_n}{\partial y} = \nu\nabla^2\omega_{n+1}$$

$$\left[\left[K\omega\right] + \frac{1}{\Delta t}\left[K\dot{\omega}\right]\right]\{\omega\}_{n+1} = \frac{1}{\Delta t}\left[K\dot{\omega}\right]\{\omega\}_n - \{Sw\}_n$$

CALL SOLMIX

$$[SM1]\{\omega\}_{n+1} = \{SM3\}$$

Figure 5.5

Figure 5.6

$$Q = -2\rho \left[\frac{\partial^2 \psi}{\partial x^2} \frac{\partial^2 \psi}{\partial y^2} - \left(\frac{\partial^2 \psi}{\partial x \partial y} \right)^2 \right]$$

Form matrix Sp

$$S_{Pi} = -\int \rho N_i Q \, dA$$

$$\nabla^2 P = -\rho Q$$
$$[K_P]\{P\} + \{S_P\} = \{0\}$$

CALL SOLMIX

Form matrices $[Ru]$, $[Qu]$ and $[Qv]$

$$Ru_{ij} = \int N_i^u N_j^v \, dA$$

$$Qu_{ji} = \int \frac{\partial N_i}{\partial y} N_j^u \, dA$$

$$Qv_{ji} = \int \frac{\partial N_i}{\partial y} N_j^v \, dA$$

Figure 5.7

Chapter 6   **TEST EXAMPLES**

6.1 Example One

6.1.1.Introduction

To test the effectiveness of this formulation, and program, and

therefore to assess solution accuracy, convergence, and stability, a

fluid flow between plane, parallel plates was chosen. The main reason

why it was selected as the first example is that this problem retains the

total non-linear character of the Navier-Stokes equations (4,6,25). For

the reason of symmetry, only one-half the problem region was considered.

The coarsest finite element mesh which was plotted by the computer is

illustrated in Figure 6-1.

6.1.2.Entry Length

The entry length plays an important role in this kind of problem.

The fact that the duct length is insufficient can lead to unstable

resulsts. So the entry length will be discussed first. The entry length

is defined as the axial position at which the centre-line velocity reaches

99% of its fully developed value (11,24,55,56). This length can be

determined by experimentation in which every parameter but the entrance

length is held fixed (14,39,55,56). Schlichting (136) has shown that the

entrance length is only a linear function of the Reynolds number for

parallel plate channels and pipes. This is true only if the shape of the

inlet velocity profile is kept the same (55). Hai (55) concluded that

the entrance length necessary for flow development is a function of channel

height, Reynolds number and shape of the entrance velocity profile for the

flow regions considered here. Laughaar (39) suggested that the entry

length for an incompressible isothermal laminar of a Newtonian fluid flow in

a circular pipe can be obtained from the following simple equation.

$$Xc = Re \times D \times K \tag{6-1}$$

where

FIG. 6 - 1

Finite Element Mesh for fluid flow between parallel plates

FINITE ELEMENT MESH
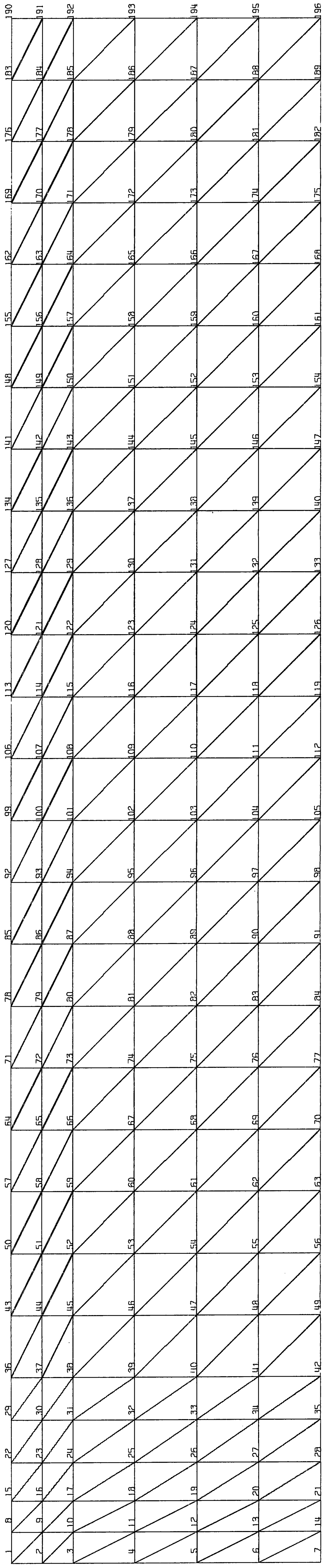
Xc   =   hydrodynamic entry length for circular pipes.

Re   =   Reynolds number   = $\dfrac{Vav \cdot D \cdot \rho}{\mu}$

ρ =   Density

μ =   dynamic viscosity

D   =   pipe diameter

K   =   constant (Laughaar suggested that K = 0.057)

After Xc is derived, the entrance length for a fluid flow between plane, parallel plates can be obtained.

6.1.3. Initial and Boundary Conditions

Now initial and boundary conditions will be discussed.

Initial Condition:

The duct is initially filled with water of density of 1.0, which is instantaneously accelerated by application of a uniform velocity of unity upstream of the duct. The equivalent vorticity initial condition is vanishing everywhere. Of course, this is by no means the only initial condition which can be used. But in this problem, it is employed to test the program.

Boundary Conditions:

Referring to what has been discussed in chapter 4, the following conditions are employed (see Fig.6-2).

(i)   To make sure that there is no mass injection cross the upper wall, the following equation should be given:

$$\psi = \text{constant} \qquad \text{along } \overline{AB} \qquad (6-2)$$

where the constant is determined from the mass flux entering the duct.

(ii)   One of the difficulties of this problem is that it is not suitable and not even possible to define values of velocity on the downstream edge. This does mean it is difficult to establish the values of stream functions along downstream edge. To overcome this difficulty, some additional constraints along this edge were introduced.
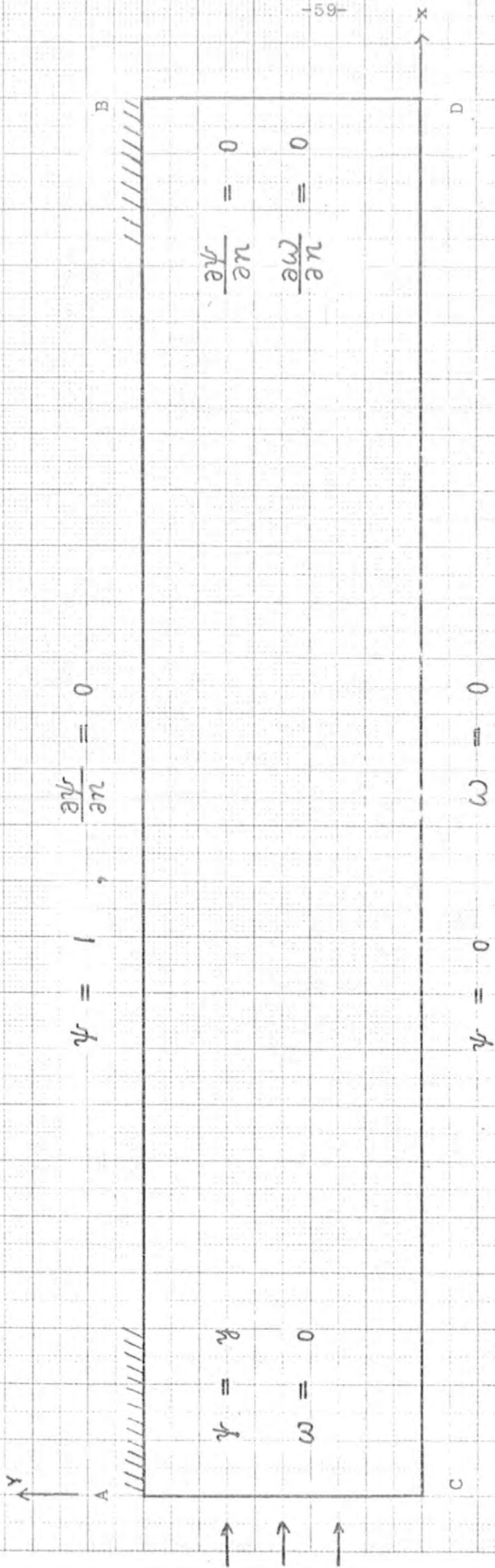
FIGURE 6-2. Boundary conditions for flow between parallel plates.

Because the flow becomes parallel along the downstream edge, the normal derivatives of both vorticity and stream function must be vanishing to enforce the flow parallel at the exit. So,

$$\frac{\partial \psi}{\partial n} = 0 \qquad \text{along } \overline{BD} \qquad (6\text{-}3)$$

$$\frac{\partial \omega}{\partial n} = 0$$

(iii) From symmetry, in the centre-line of the duct, the vorticity and streamfunction may be defined as follows:

$$\psi = 0$$

$$\omega = 0 \qquad \text{along } \overline{CD} \qquad (6\text{-}4)$$

(iv) For uniform flow at the duct entrance, the following equations can be given

$$\psi = KY$$

$$\omega = 0 \qquad \text{along } \overline{AC} \qquad (6\text{-}5)$$

where K is constant.

(v) The no-slip condition is most important (see Chapter 4). The formula used to calculate wall vorticity is as follows:

$$\omega_w(X_0, Y_0) = -\frac{2}{n^2}\left[\psi(X_0, Y_0) - \psi(X_1, Y_1)\right] \qquad (4\text{-}4)$$

6.1.4. Iteration Technique

To obtain a starting solution, Stokes' flow was assumed. And stream function and vorticity fields were calculated by giving Re=0. After the Stokes' flow solution was obtained, the iteration process was used to calculate the flow at successively larger Reynolds numbers. The solutions of stream function and vorticity at a lower Reynolds number are used as the initial conditions for the solutions at the next higher Reynolds number .

It was found that the numerical scheme was stable if a sufficient small Δt was chosen.

6.1.5. Discussion and Conclusion

At the entry section where a velocity discontinuity is occurred at

point A (see Figure 6-3(a)), the singularity introduces a considerable disturbance to the solution.   Serious numerical errors may be encountered in the calculation unless sufficiently small elements are used.

In fact, because the boundary conditions are contradictory at the point A, the approximate solution will not be able to satisfy such boundary conditions exactly.

In actual computation, two types of entrance conditions at the discontinuous point have been tested.

Case 1.   U decreases to zero from point E to point A as a parabola function.   The velocity profile for this kind of entrance condition was shown in Figure 6-3(b).

Case 2.   U decreases to zero from point E to point A as a linear function.   The velocity profile for the entrance condition was presented in Figure 6-3(c).

The phenomena shown in Figures 6-3(b) and 6-3(c) seem to agree well with Tong and Fung's (124) results (see Figure 6-4).

The input data used is presented in Appendix I.   Its main results for streamfunction and vorticity from this computer program are given in Table 6-1.   From this table, it is found that the results seem to be along with those of Baker and Gosman (4).   The streamline contours are presented in Fig.6-5.   Its main contour program is given in Appendix C.   Results for velocity and pressure are shown in Figures 6-6, 6-7 and 6-8.   From Figure 6-6 and 6-8, it is found that the velocity and pressure results from this program seem to be along with those of Goldstein (49) and Yamada (129) as well.   Some streamline contours from this program for slightly higher Reynolds numbers are presented in Appendix D.   The streamline contours of Baker (4) for Reynolds number of 200 are shown in Appendix E.   It is seen that the contours compare reasonably well thus indicating that the present program seems to be accurately representing the phenomenon.

TABLE 6-1

Steamfunction and wall vorticity distribution

X/L	Streamfunction ψ			Wall vorticity ω_w		
	F.E.M.* (Baker)	F.D.M.* (Gosman)	This ** prog.	F.E.M.*	F.D.M.*	This** prog.
0.047	0.980	0.976	0.967	4.04	5.05	6.67
0.095	0.982	0.980	0.981	3.55	4.44	3.54
0.156	0.983	0.982	0.982	3.57	4.14	3.43
0.228	0.983	0.983	0.984	3.58	3.71	3.10
0.379	0.983	0.984	0.984	3.60	3.28	3.08
0.521	0.984	0.985	0.984	3.30	3.12	3.08
1.000	0.983	0.985	0.984	3.51	3.06	3.15

(i)

$\vdash\!\!-\!\!-\!\!-\!\!-\!\!-\!\!-$ L $\longrightarrow\!\!\vdash$

$\vdash\!\!-\!\!-$ X $\longrightarrow\!\!\vdash$
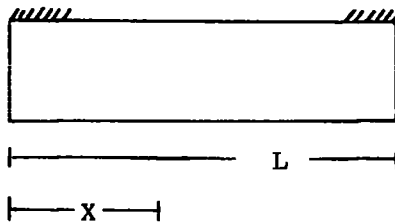
(2)   *   :   Re   = 2 00

       **   :   Re   = 0.002

(a)

(b)

(c)
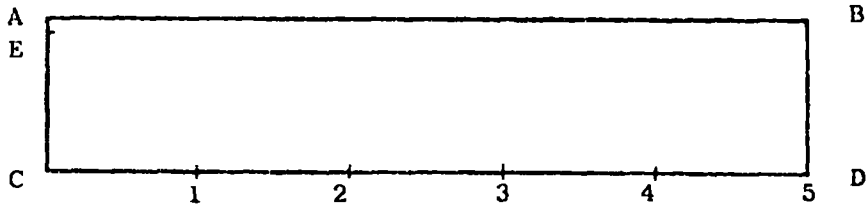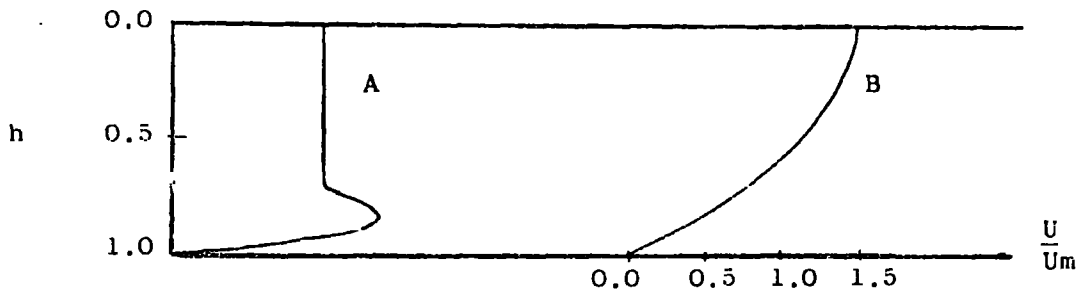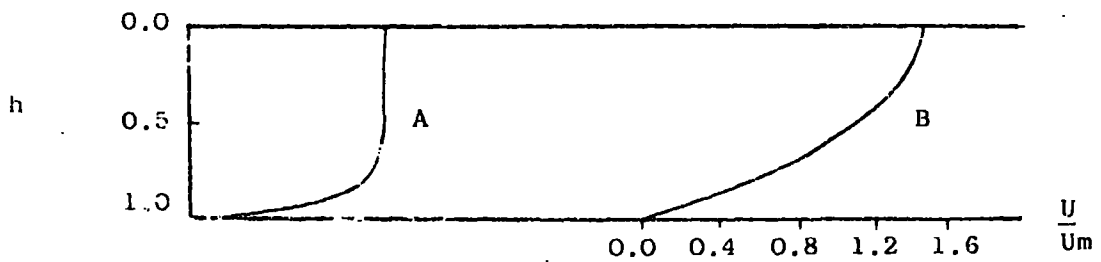
Figure 6-3 Velocity distribution for flow between parallel plates.

A: Velocity distribution at the entrance.
B: Velocity distribution for fully developed flow.
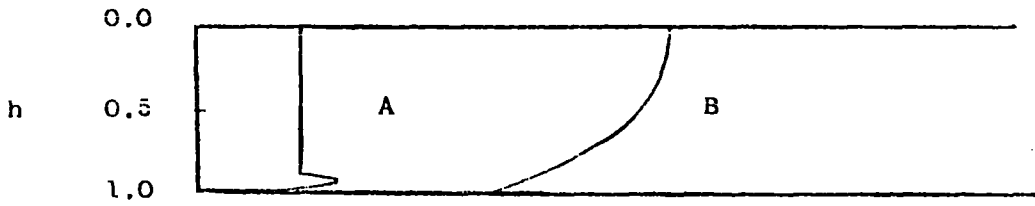
(a)   Case 1
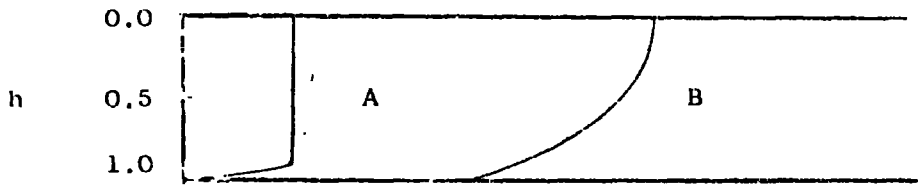
(b)   Case 2

Figure 6-4   Tong & Fung's Results (124)

A: Velocity distribution at the entrance.
B: Velocity distribution for fully developed flow.
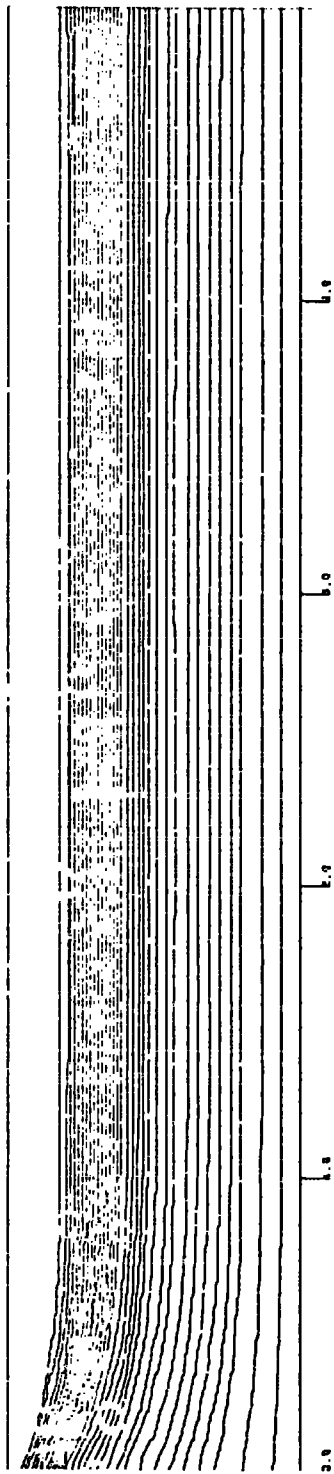
Figure 6-5

Streamline contours in flow between parallel plates.
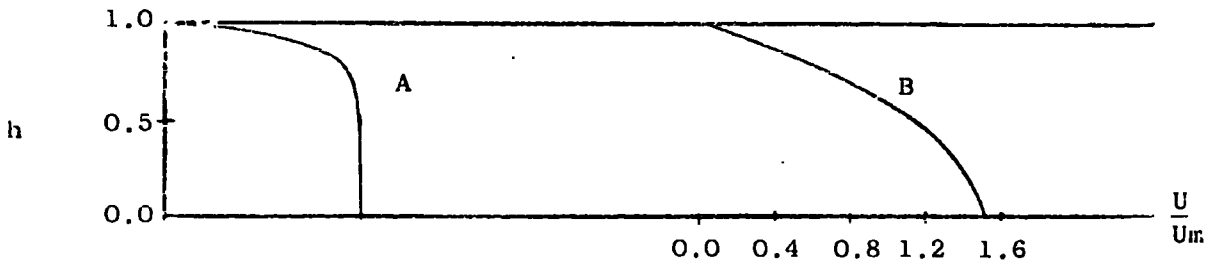
(a) Results from this program
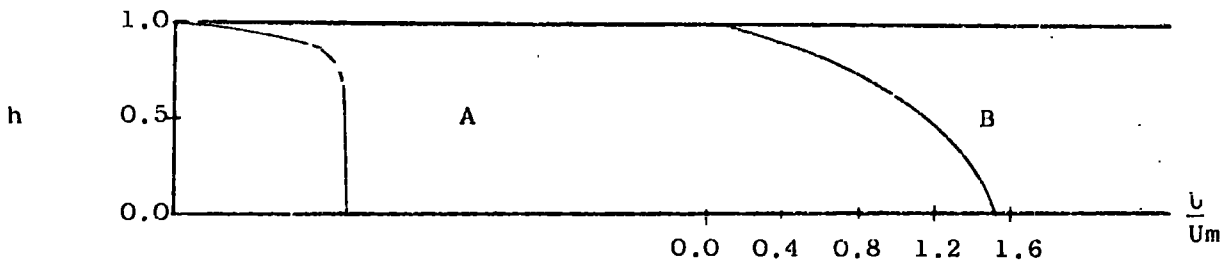
(b) Results of Goldstein

A: Velocity distribution at the entrance.
B: Velocity distribution for fully developed flow.

Figure 6-b

Velocity distribution for flow between parallel plates

FIGURE: 6-7   Velocity distribution for flow between parallel plates

U: the axial component of velocity

V: the radial component of velocity

(a)

(b)

(c)
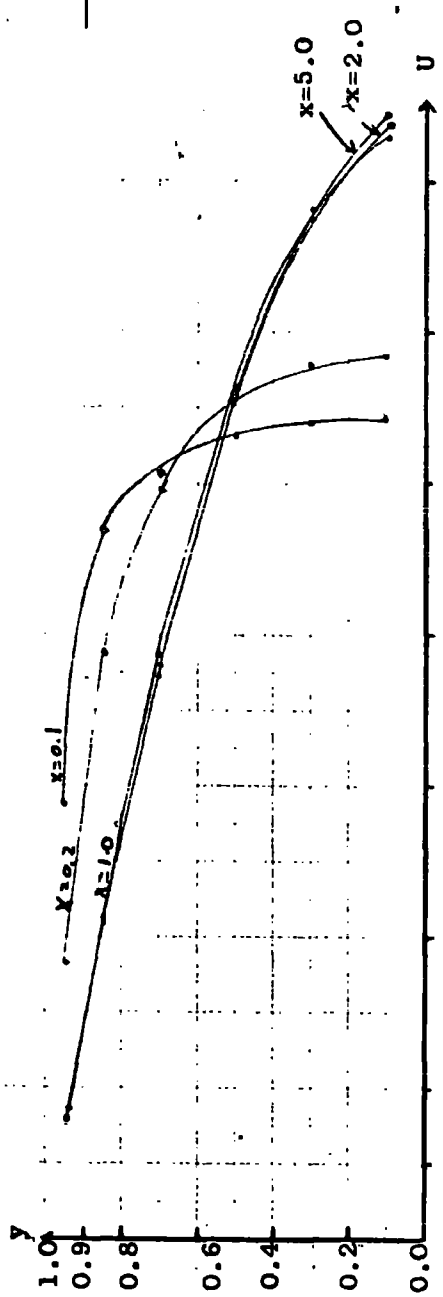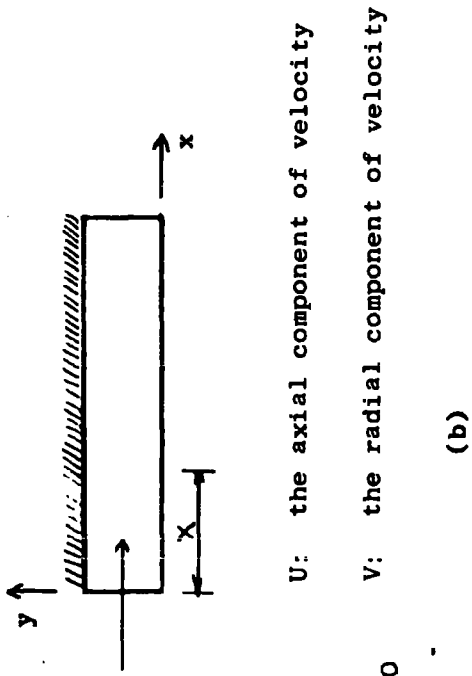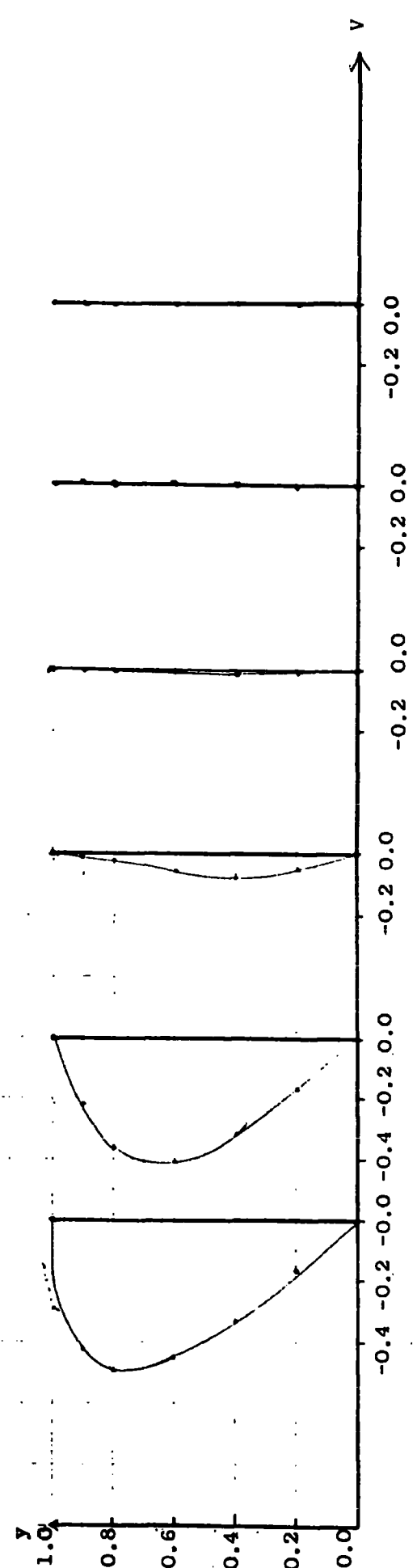
Figure 6.8

Pressure distribution for flow between parallel plates

(a) Results from this program

(b) Results given in reference (129)
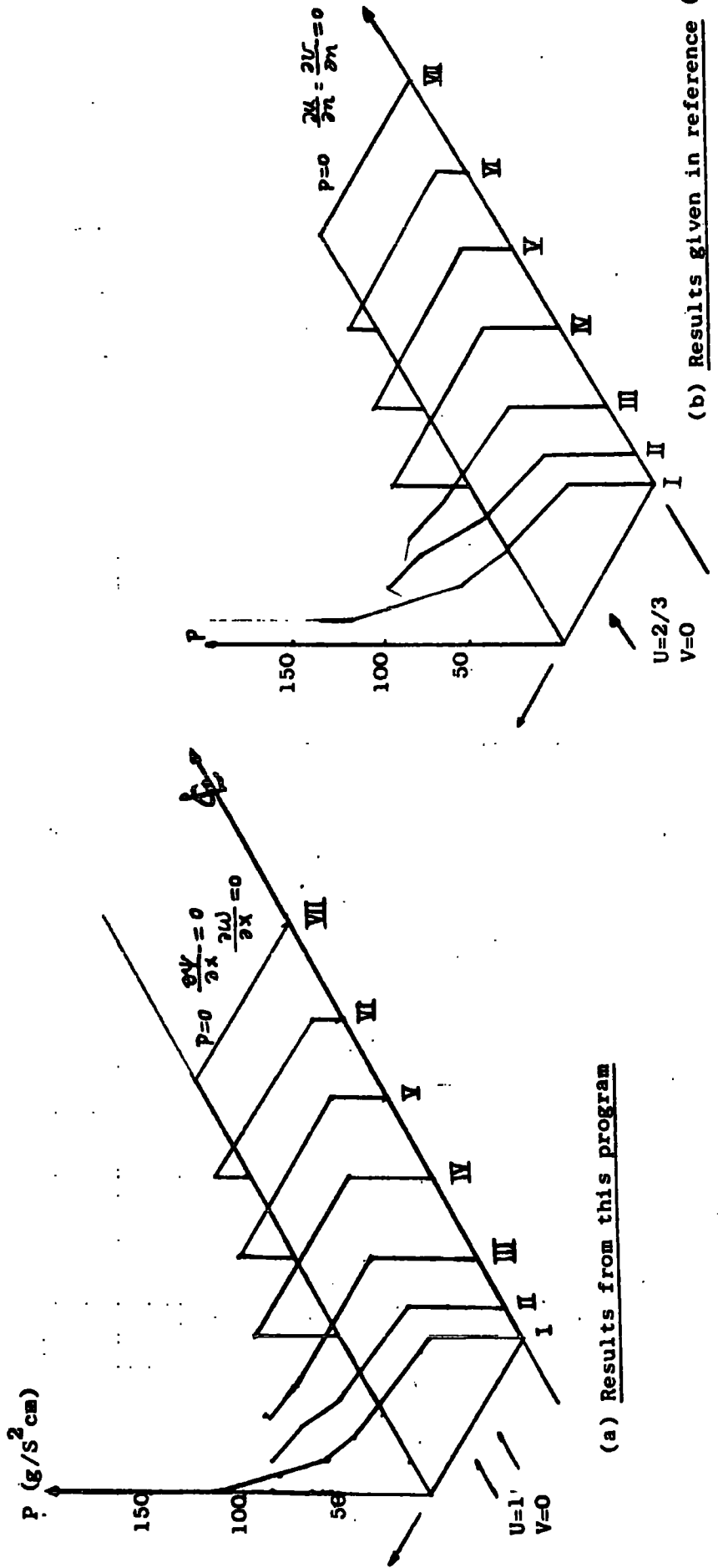
6.2 Example Two

6.2.1.Introduction

The next application was to the internal fluid flow in a channel of arbitrary cross section. The geometry of an example flow passage was shown in Fig.6-9. This type of geometry could provide a good test and demonstration for this solution procedure because the constriction causes rapid changes in stream function and vorticity near the constriction region. The coarsest finite element discretization evaluated were illustrated in Figure 6-10.

6.2.2.Boundary Conditions and Iteration Technique

It has been shown that the entrance length is reduced as the shape of the entrance velocity profile approaches that of the fully developed profile. And this entrance length is reduced to zero when the entrance velocity profile is identical with the fully developed profile at which point the flow is fully developed at this entrance to the channel (55). So Poiseuille type flow was used at the upstream edge in view of the fact that larger values of X_1 (see Figure 6-9) would imply a rapid growth of computer time required. That is

$$\psi = \frac{3}{2}\left(y - \frac{1}{3}y^3\right) \qquad \text{on} \quad \overline{AF} \qquad (6\text{-}6)$$

At the downstream edge $(\overline{EG})$ Poiseuille type flow was also assumed. It is worth emphasizing here that the values of X_3 need be provided large enough. Extending the concept of entrance length discussed in example one to this example, it is known that at higher Reynolds number, after passing the constriction, the fluid flow would have to travel a much longer distance X_3 before it was returned to a Poiseuille type flow. If the provision of X_3 was not adequately specified in the finite element mesh, the numerical procedure may become unstable. Even when the calculations are convergent, wrong results or results which are not expected, for example, a result from a different boundary condition, may still be obtained.

From the governing equation $\nabla^2\psi = -\omega$ and equation (6-6), the
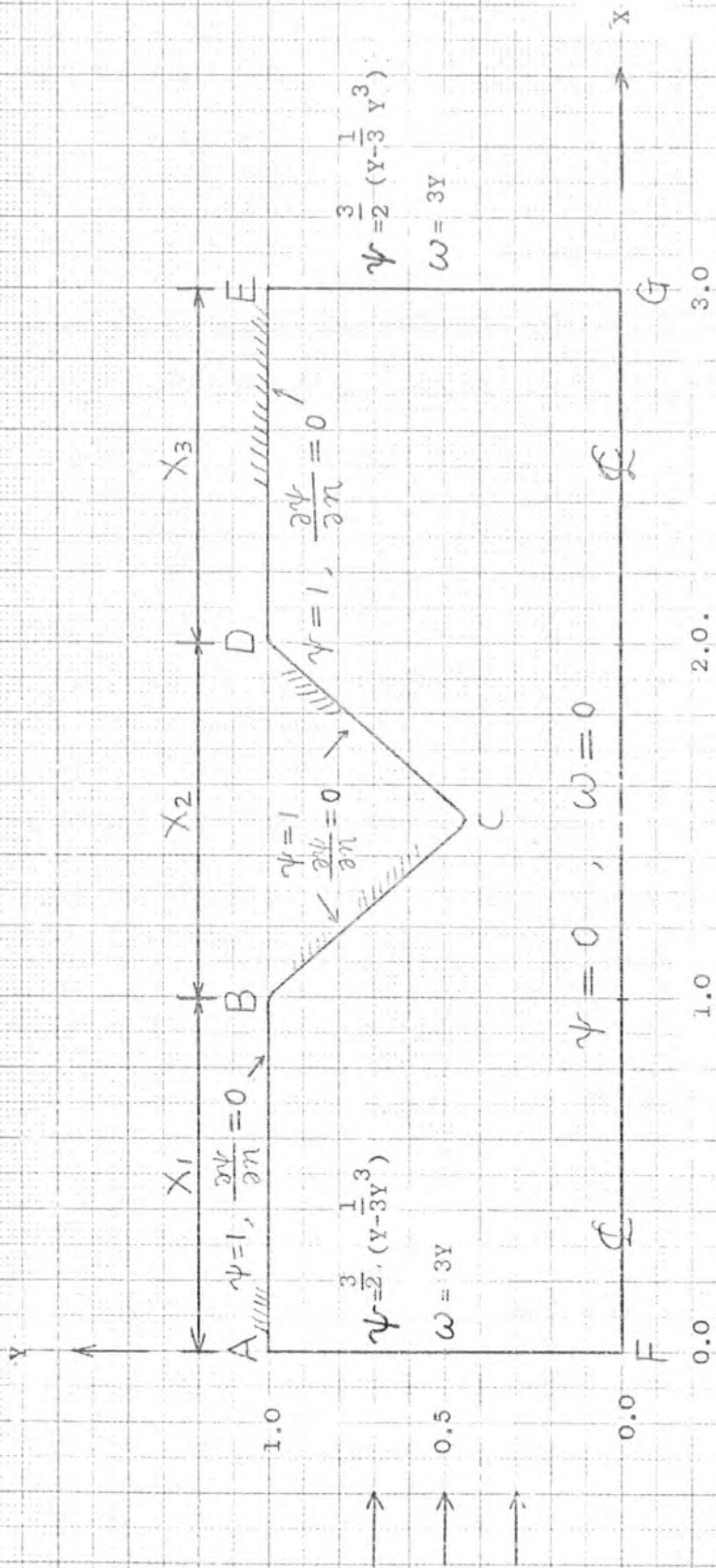
FIGURE 6 - 9

The Geometry and boundary conditions for internal fluid flow in a constricted channel.

FINITE ELEMENT MESH

FINITE ELEMENT MESH
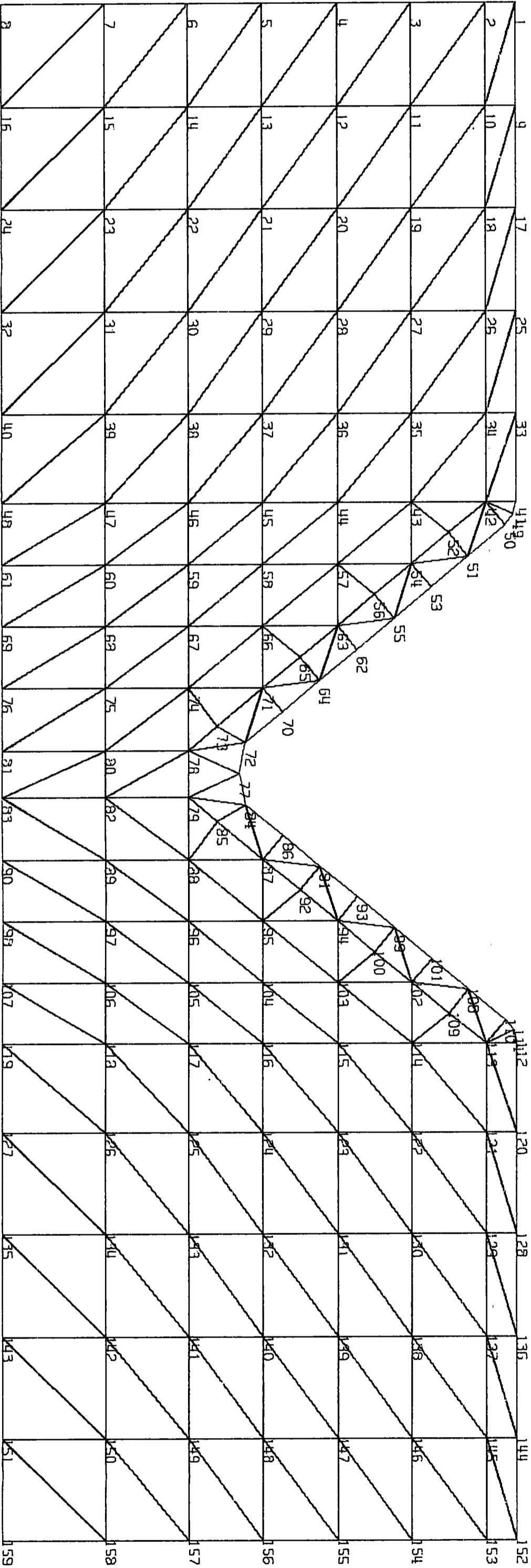
FIG. 6 - 10
Finite Element Mesh for fluid flow in a constricted channel
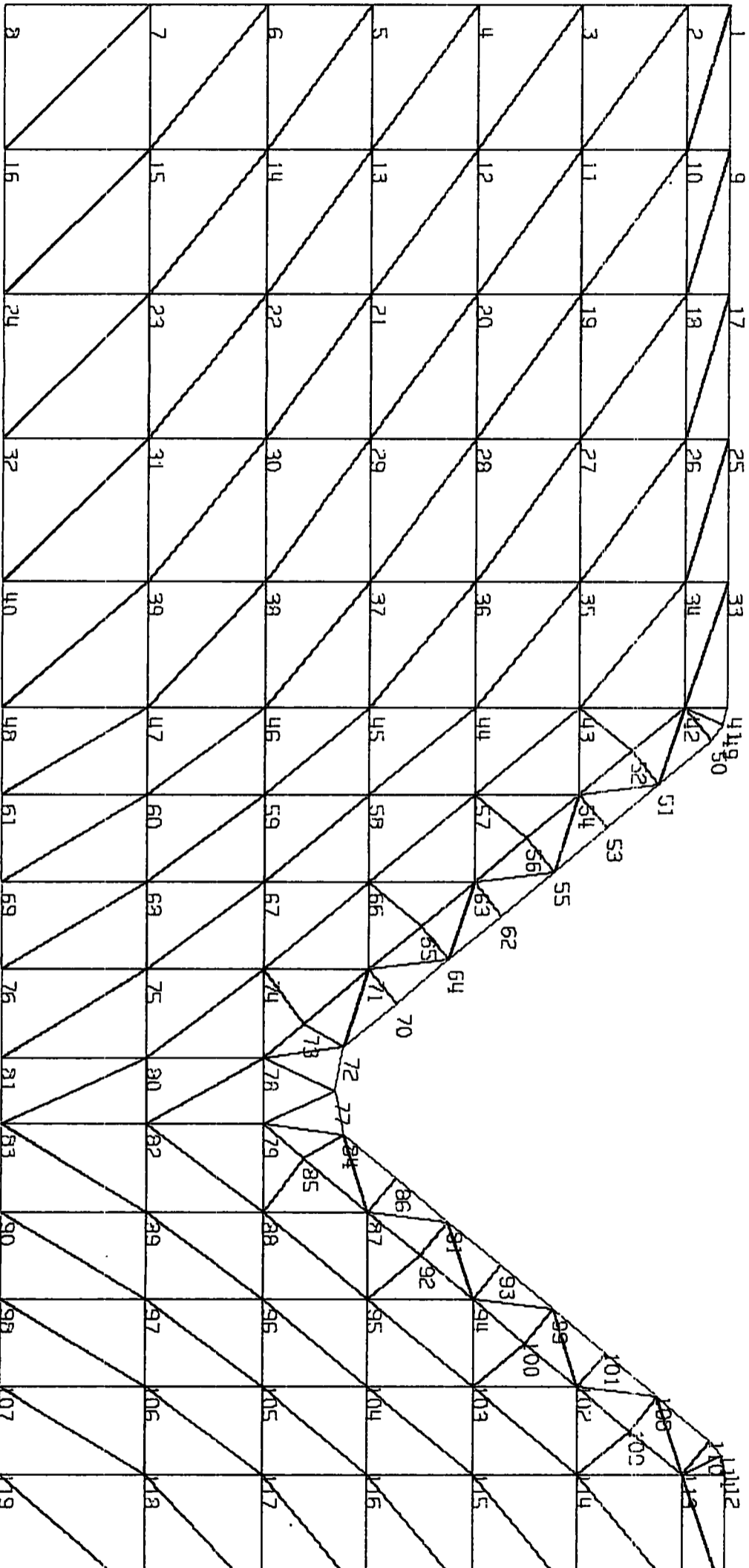
following boundary condition should be specified

$$\omega = 3Y \qquad \text{on } \overline{AF} \text{ and } \overline{EG} \qquad (6\text{-}7)$$

On the solid wall, the no-slip condition was applied as well. And the first-order one-sided difference formula (equation (4-4)) was employed to calculate wall vorticities. It is worth mentioning here that this wall vorticity formula is mainly not suitable for fields containing corners with high curvatures. Thom and Apelt in their useful book "Field Computations in Engineering and Physics" (121) suggested some helpful schemes to deal with this difficulty. Greenspan (51,52) also suggested some other formulae to solve this problem. However, no matter what kind of scheme is being used, it should be stressed that it is most important to use a finer mesh in the constriction regions. If the elements used are small enough, then the first-order one-sided difference formula can still be employed. The key point is that the effects of the corners must be forced to spread into the fluid field in every direction. It is worth emphasizing here that the finite element method is simply a man-made method. It is important to help the method by letting "him" work as close to the phenomenon which occurs physically as possible.

Referring to chapter 4 again, in which some ideas about boundary conditions have been presented, the following boundary conditions may be specified (see Figure 6-9).

$$\psi = 0 \qquad \text{on } \overline{FG} \qquad (6\text{-}8)$$

$$\psi = 1 \qquad \text{on } \overline{ABCDE} \qquad (6\text{-}9)$$

$$\omega = 0 \qquad \text{on } \overline{FG} \qquad (6\text{-}10)$$

At time t=o, values of vorticities are assumed to be zero everywhere. And these values are considered as the initial conditions when the numerical solutions of stream function and vorticity for creeping flow are solved. After the convergent creeping flow solution was obtained, this solution was used for the initial conditions of stream function and vorticity in the iteration process to calculate the flow at a small Reynolds number.

This process was carried out such that the solution at a low Reynolds number was used as the initial conditions for the solution at a higher Reynolds number (see Chapter 4). It has been found that the iteration procedure was always stable for sufficiently small Δt. Through numerical experiments, it is found that a high curvature at point C (see Figure 6-9) has profound adverse effects on both numerical stability and accuracy. Even with the same mesh size, the high-curvature body requires a time step that is an order of magnitude smaller than that for a low-curvature body in order to achieve stability (105). This is due mainly to the very steep gradients of the vorticity in the field created by the high curvature. The same gradients also cause severe problems with respect to loss of accuracy (105). In order to avoid large errors, and accelerate the speed of convergence, it seems that two kinds of time steps, Δt_1 and a smaller one Δt_2, may be used in low-curvature regions and high-curvature ones respectively. (In this example, this kind of scheme has not been employed yet.)

6.2.3. Conclusion

The input data used is presented in Appendix F. Its main results for stream function and vorticity from this program are given in Appendix G. The stream line contours are presented in Figure 6-11. The contours seem to be along with those of Lee and Fung (79) (see Figure 6-12). Results for wall vorticity are shown in Figure 6-13. From this figure, it is found that in the Stokes limit the flow patterns are symmetric before and after the constriction, no separation occurred at this limit. This result seems to agree well with both the solutions of Cheng (25) and those by Lee and Fung (79). The slight discrepancies among the results for the vorticity on the wall of Stokes flow from this program and those of Cheng (25) and Lee and Fung (79) are probably due to the differences in the geometries of constrictions and the coarseness of these meshes being used. The velocity results are presented in Figure 6-14. They seem to be in
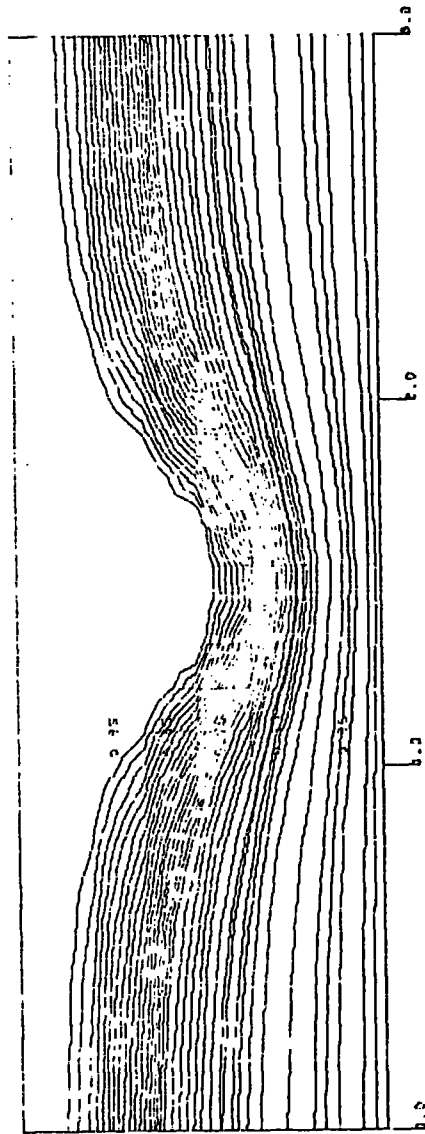
STREAM LINES

FIG. 6-.1

Streamline Contours
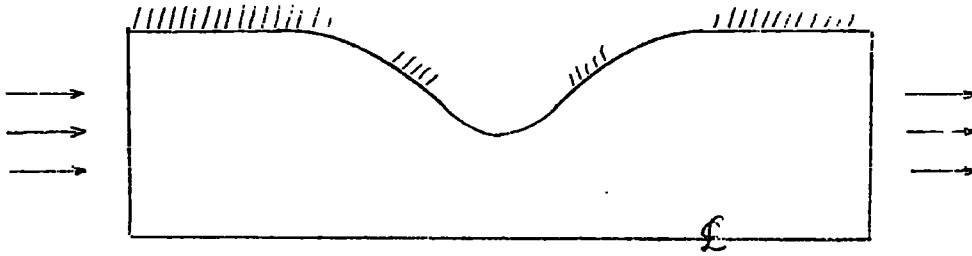
(a)   The geometry used by
         Lee & Fung
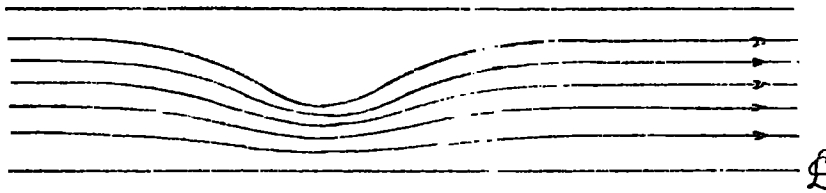
(b)   Streamlines Contours

FIG.6-12   Results of Lee & Fung
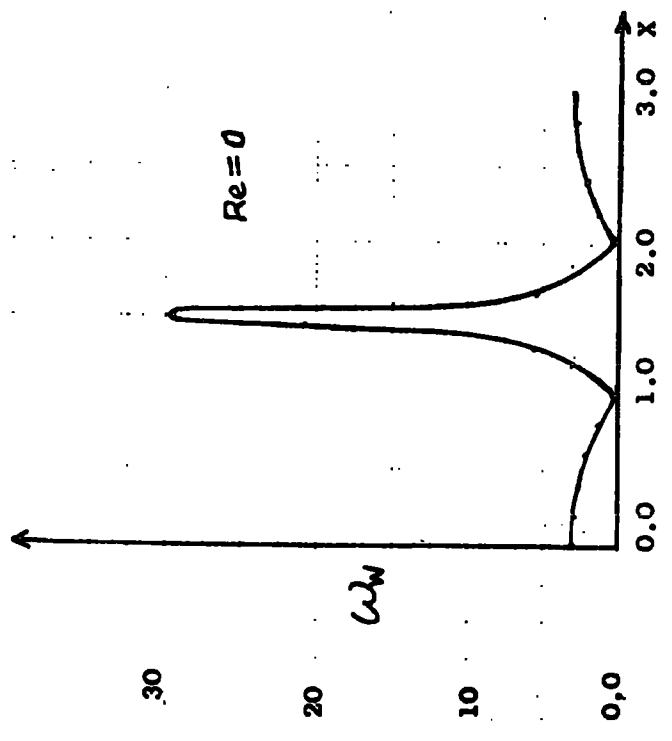
(a) Results from this program
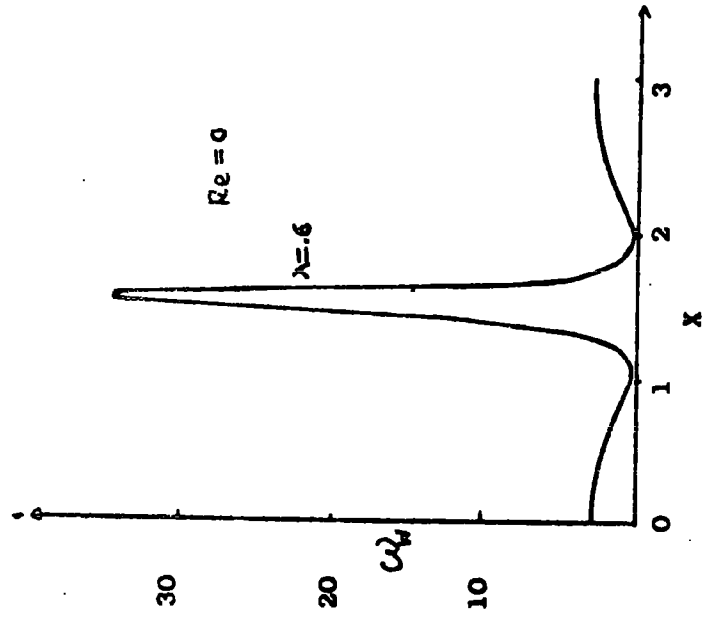
(b) By R T-S Cheng
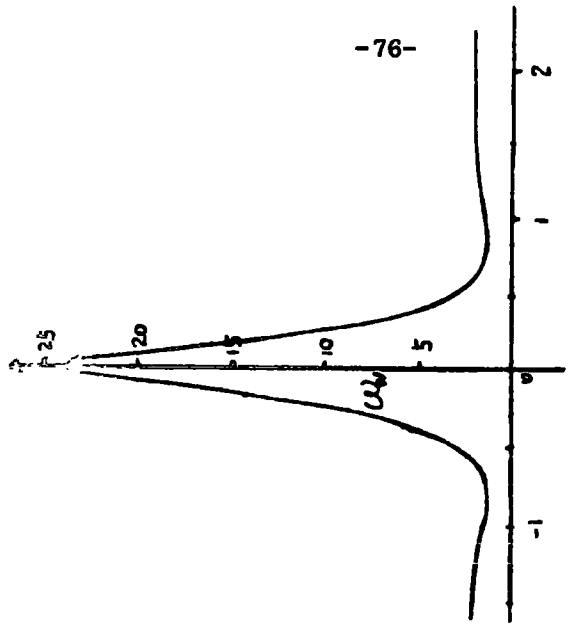
(c) By Lee and Fung
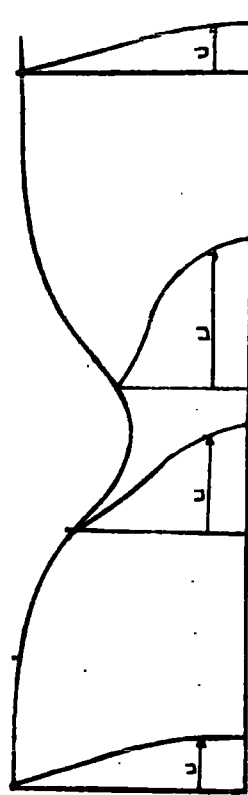
FIG. 6-13

Distribution of the vorticity on the wall

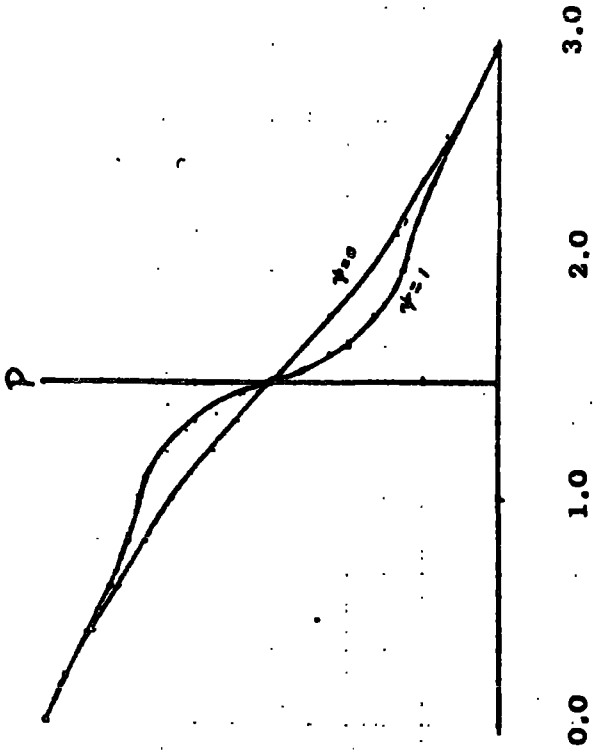VEL PROFILE
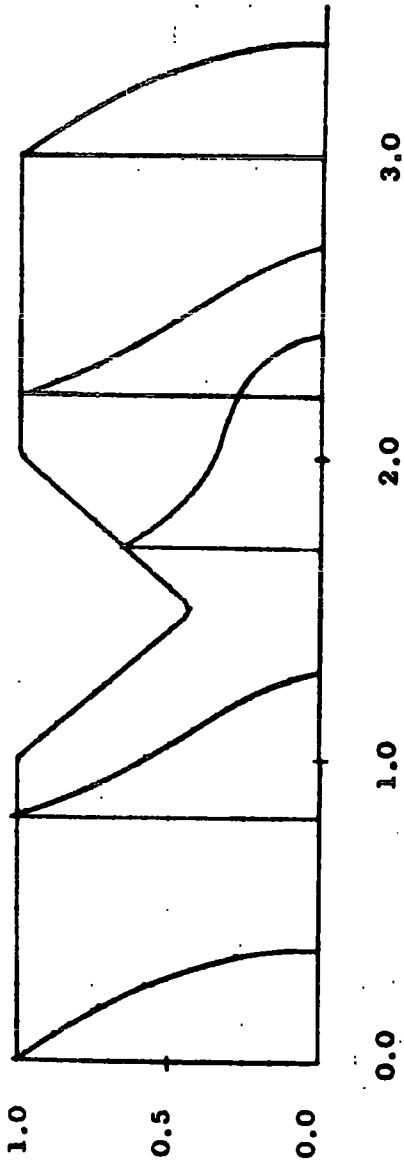
Results from this program
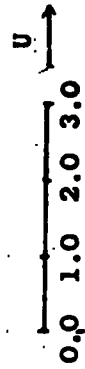
(a)

Results of Lee & Fung

(b)

FIG. 6-14

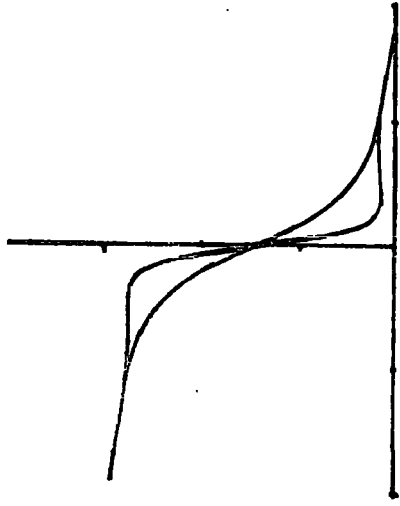Velocity distribution in a
constricted channel

(a)

(b)
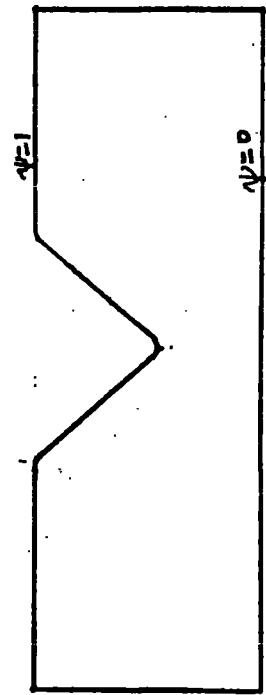
(c) Results from this program
(FEM)
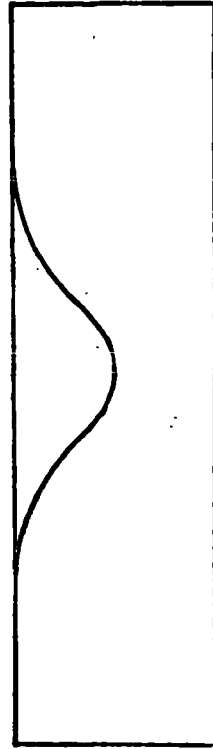
(d) Results of Lee & Fung (FDM)

FIG. 6-15 Pressure distribution.

good agreement with the solution of Cheng (25). The pressure results are given in Figure 6-15. The slight discrepancies between the solutions for pressure from this program and those of Lee and Fung (79) are also probably due to the differences in the geometries of constrictions and the coarseness of the finite element grid.

The results seem to compare reasonably well thus indicating that this program seems to be accurately representing the physical phenomenon.

Chapter 7   <u>DISCUSSION</u>

Although the numerical schemes to solve the Navier-Stokes equations have

been discussed, some realistic difficulties can still occur when the schemes

are carried out.   In this chapter, some problems will be discussed.   And

some observations about the numerical procedure, obtained through numerous

numerical experiments, will be presented and discussed.   These observations

may be useful in the finite element analysis.

7.1 <u>Convergence problems</u>

The problem of convergence is one of most important problems in a

numerical analysis.   When the foregoing process is applied to the Navier-

Stokes equations, divergence can take place in parts of the field.   Emphasis

has already been put on the need for care with the method of determining

the boundary values of vorticity, but even if these are known and fixed the

field may still diverge if the mesh size is too large (33,42,100,121).

To prevent divergence, Thom (121) suggested that at each point the value

of ω may be adjusted from the old value ω_n to the newly calculated value

ω_{n+1}.   In other words, ω its full movement may be given.   Thom claims

that the movement should be restricted by combining ω_n and ω_{n+1} in the

proportion $r{:}1$ where r is positive.   However, if r is too large, it is

obvious that the rate of advance would be very slow.

In the calculations for wall vorticity, sometimes, it happens that

to repeat this operation many times will result in an unstable oscillation

of the field.   Thom and Apelt (121) suggested that new boundary values

$$\omega_w = (\omega n)_w + K\left[(\omega n{+}1)_w - (\omega n)w\right]$$

may be used instead of original ones, where K is less than unity.   The

best value of K can only be estimated by trial and experience.   As a start

they suggested K = 0.5.   In order words the boundary values are only

moved about one-half of the amount indicated by equation (4-4).

Lee and Fung (79) also employed a similar manipulation.   They combined

the conformal mapping and finite-difference techniques to investigate

problems of viscous flow in a locally constricted tube. They found that the calculation was straightforward for Re $\leq$ 15. But when Re $\geq$ 20, the numerical procedure failed to converge. To improve the matter, they used an under-relaxation factor. They claim that the values of the vorticity ω were given one half their theoretical changes in each iteration. In this way they got the results for Re=25.

Although incorporation of the under-relaxation technique seems to be able to accelerate the speed of convergence, and improve the possibility of convergence, there is still no, to the best knowledge of the author, apparent theoretical justification for such a manipulation of the relaxation factor.

Some comparisons between the results of using a relaxation factor and not using this kind of factor have been made, and presented in appendices. Note that when TURF1=1 (see Appendix D) the relaxation factors were never used, and when TURF1 = 0.1, the relaxation factors used were 0.1.

7.2 Storage problems

In many finite element problems the amount of core store required is too great for the computer being used and it is necessary to use backing store. Sometimes a peripheral such as a magnetic disc on a magnetic tape deck can be used automatically within a program. The non-linear matrix, because of its size, may be stored out of core, the high-speed disc being the next best place.

There are lots of ways to improve storage problems such as: employing the techniques of the front solution, substructuring, overlaying, equivalence, or dynamic allocation, etc. Some important methods of them will be briefly discussed here.

7.2.1 Front Solution (33,37,67,134)

Theoretically the front solution is quite simple. There is a large linear set of simultaneous equations with the n stream functions $\{\psi\}$ of the fluid flow field as unknowns. When all the information relating to

a particular variable is complete then that variable may be eliminated since an expression for it in terms of the other variables in the problem can be obtained. Some unknowns can be eliminated before the complete "stiffness matrix" is formed and therefore the whole of the "stiffness matrix" is never needed in core at one time. Irons (67) has developed a good front solution program for finite element analysis to solve symmetric positive-definite equations. Hood (134) has also presented another front solution program which may be used for the solution of unsymmetric matrix equations. Using these schemes, core requirements and computer time may be considerably reduced.

7.2.2. Banded Solution (33,37,116)

Using the fact that the stiffness matrix is square and symmetric and all non-zero terms are concentrated in a narrow band either side of the leading diagonal, great economies of storage are possible by storing only the band.

To solve equations (3-35) and (3-43) directly by using, say, the Gauss-Jordan elimination procedure would be very inefficient in terms of computer time and storage, since that does not take advantage of the banded nature of $[K_\psi]$ and $\left[\, [Kw] + [K\dot{w}] / \Delta t \,\right]$.

It appears that the most efficient procedure is to store those elements of $[K_\psi]$ and $\left[\, [Kw] + [K\dot{w}] / \Delta t \,\right]$ those are within the band rowwise in two vectors respectively, say $[A]$ and $[B]$, and employ a modified Gaussian elimination scheme with back substitution which takes advantage of the banded nature of $[K_\psi]$ and $\left[\, [Kw] + [K\dot{w}] / \Delta t \,\right]$. In this procedure, Gaussian elimination and back substitution need only be carried out up to the lower and upper edges respectively of the bands. Thus the zeros of $[K_\psi]$ and $\left[[Kw] + [K\dot{w}] / \Delta t \right]$ outside the band are not operated upon and are actually not stored in $[K_\psi]$ and $\left[\, [Kw] + [K\dot{w}] / \Delta t \,\right]$. With this method of solution, it will be necessary to know the width of the bands and the location of the diagonal elements within the bands at every

row of $[K_\psi]$ and $\left[\, [Kw] \;+\; [K\dot{w}] \;/\; \Delta t \,\right]$.

7.3 Computer time problems (6,14,25)

There are also a lot of methods to save computer time. But if the computer program has been prepared, then the following suggestion may be a helpful way to reduce the computation time used.

It is found that the initial flow configuration does not change the steady-state solution. Regardless of the initial conditions employed, the procedure does indeed converge to the same solution. Therefore, provided that there is no interest in the transient solution to the problem, the analysis can be started at a good initial guess with attendant saving in computer time. The solution obtained with a coarse mesh can be utilized as the guess for a finer mesh.

7.4 Boundary conditions

It has been shown that all boundary conditions require detailed attention. But sometimes, even if the significance of boundary conditions has been noticed, it is still not impossible to face the problems of how to specify suitable boundary conditions for a given problem. A practical way, but not a good way from points of view of computer storage, to overcome this kind of problem is presented here.

It is pointed out that the difficulty in boundary conditions, sometimes can be by-passed if the region to be considered is changed. Usually, a bigger region or the whole region of the problem can be used instead of a smaller or half the region. For example, to study the problem of vortex street development behind some obstructions in channel of finite width, the boundary conditions in the centre line are not quite obvious. So in this case, the best way to deal with this kind of problem is just to use the total region of the flow field instead of the half symmetric one. (see Figure 7-1).(110)

7.5 Finite-element mesh

A likely distribution of contours of field variables of interest is best be predicted in advance. Then the mesh is arranged to be as similar to

$\psi = 1, \omega = 0.$

$\psi = 0, \quad \frac{\partial \psi}{\partial n} = 0$

$\psi = 0, \quad \frac{\partial \psi}{\partial n} = 0$

$\psi = 0$

$\omega = 0$

$\frac{\partial \psi}{\partial n} = 0$

$\frac{\partial \omega}{\partial n} = 0$

$\psi = -1, \omega = 0.$

1.0

0.0

-1.0
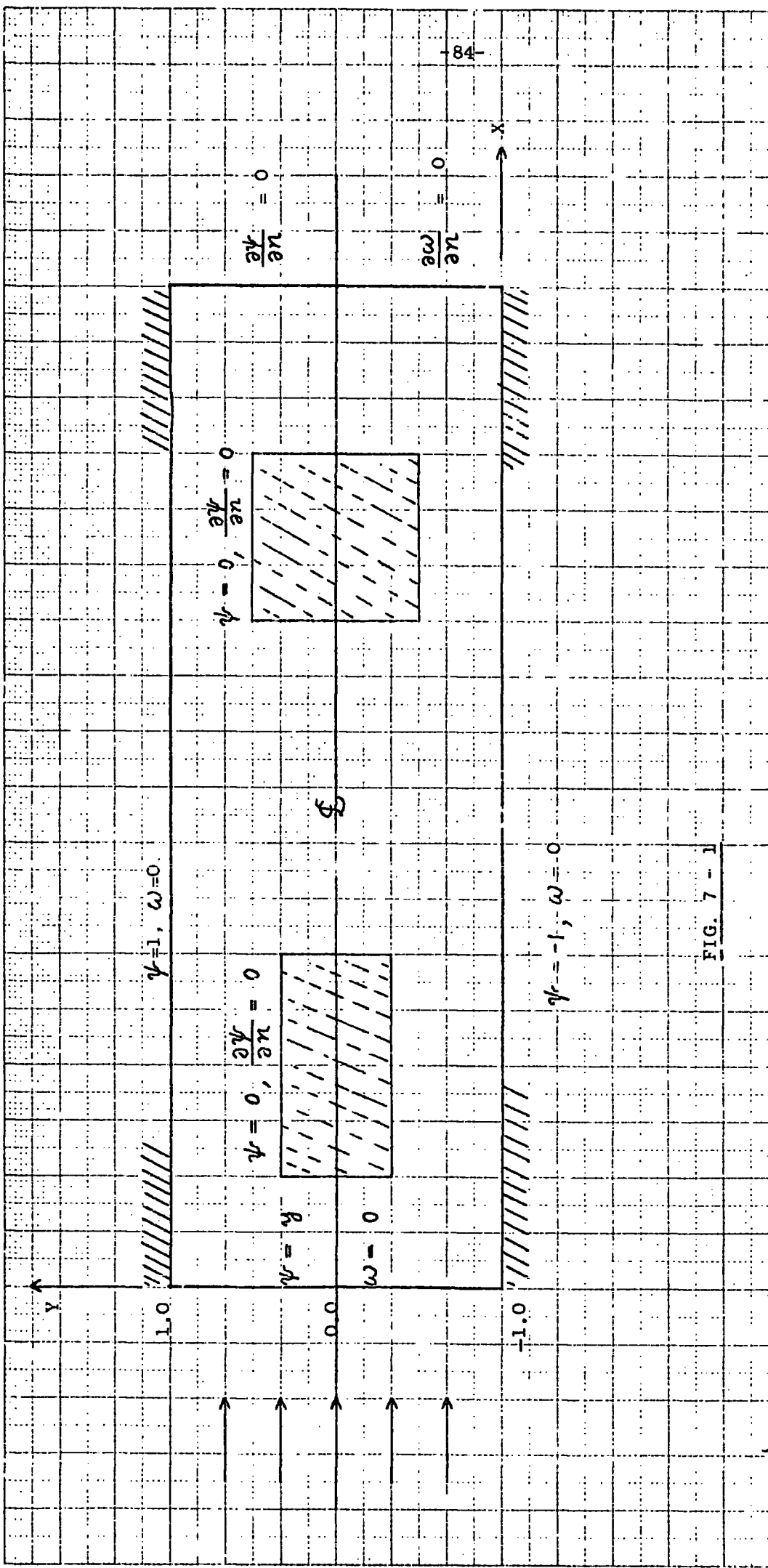
FIG. 7 - 1

Boundary conditions for flow in a channel of finite width

with some obstructions.

the predicted distribution as possible. For example, a finer mesh should be arranged for the regions where the variations of streamfunctions or vorticities are more pronounced.

If there is no idea at all about the distribution of contours of field variables, an alternative way suggested here is that a coarse mesh can first be made to analyse the problem. It is possible to get some ideas of values of results from this analysis based on the coarse mesh. And then, these values may be used as a guide to arrange a better mesh. Bearing in mind that an appropriate mesh used can save computer storage, and computation time, but a bad mesh can even make the calculation unstable.

7.6 Some Obeservations

Through numerous numerical experiments, some observations were made. They may be useful in the future applications of the finite element method. Although their validity for all kinds numerical schemes has not been ascertained yet, it is hoped to bring these observations to people's attention.

7.6.1.The transmission phenomena of a mesh line

To save computer time and storage, it is best to reduce the number of elements to as few as possible. However to improve and guarantee the accuracy and stability of a calculation, it is hoped to increase the number of elements used to as many as possible. This situation is like a famous Chinese proverb which says "It is difficult to make a horse fat without giving it enough food". Usually, if the number of elements is reduced too much, then inevitably some of the accuracy will be lost, and convergent results can not even be reached. However, even if this is the case, a coarse mesh is often forced to be employed in a complex region, even though a finer grid should have been used, in view of the limitation imposed by the computer. In this case, the best way to do this is to improve the "quality" of the mesh being used. It is found that mesh lines seem to have an ability of giving effect to the calculation by

transmitting or spreading newly calculated values into nodes of neighbouring elements. It is known that any numerical method is simply an instrument to help people do what they want to do, and analyse what they wish to analyse. If the instrument is hoped to work properly, then it should be used in a proper manner. If the finite element scheme, the instructment, is helped by using a good mesh employing the nature of a mesh line, it is not impossible, while adopting a coarse gridwork in a calculation, to get a satisfactory and accurate and stable result. It is found through numerical experiments that in a region on which a high curvature is found, more mesh lines should be used there. The higher a curvature is, the more mesh lines should be used. And it is better to do the mesh symmetrically unless the effect of the mesh on some directions is hoped to intensify through the calculation. To illustrate the idea, an example is presented.

Figure 7-2 shows an obstacle in a fluid flow or a hole on a plate. The curvatures of points B and D are the same, and so the same number of mesh lines is suggested. The curvature at point A is the highest, then the numer of mesh lines used should be more than those at points B,C and D. Note that all the mesh lines are symmetrical. Furthermore, not only to save storage but also to reduce computer time, in a region where begger unsymmetric variations of results are expected, the number of mesh lines there should be increased unsymmetrically. It is significant for a user of a finite element program to learn the effect of a shape function on the choice of a mesh for a calculation and to have an idea about the actual physical phenomena described by the problem. For example, from the foregoing discussions, it is obvious to conclude that the optimal mesh for a viscous fluid flow depends also on Reynolds number. See Figure 7-3. A mesh for case A should be different from that for case B. If the same mesh is used to analyse the two cases, then there will be a lot of computer time and storage wasted in analysing for case B. Bearing in mind that to help the procedure predict the actual physical phenomena, a suitable mesh
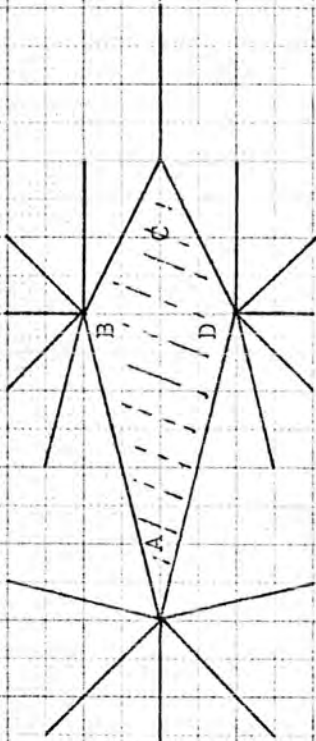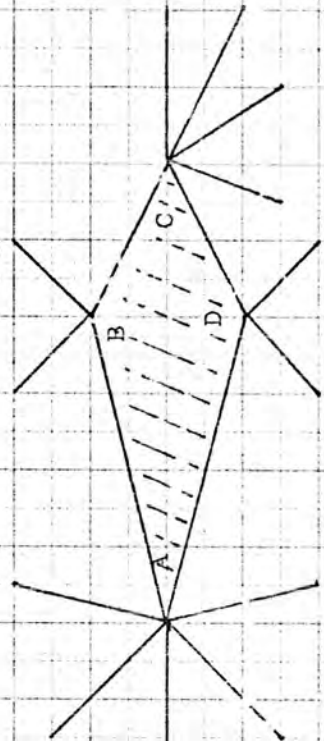
(a) Proper
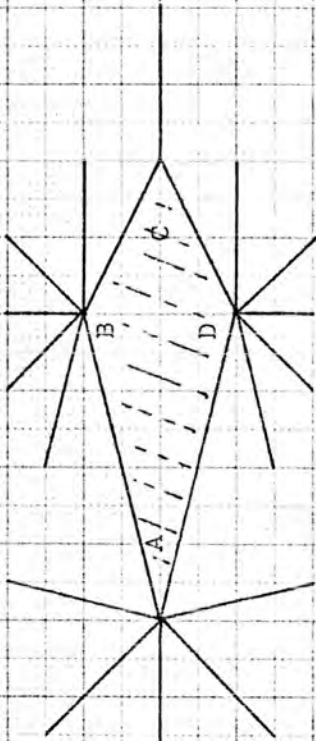
(2) Improper
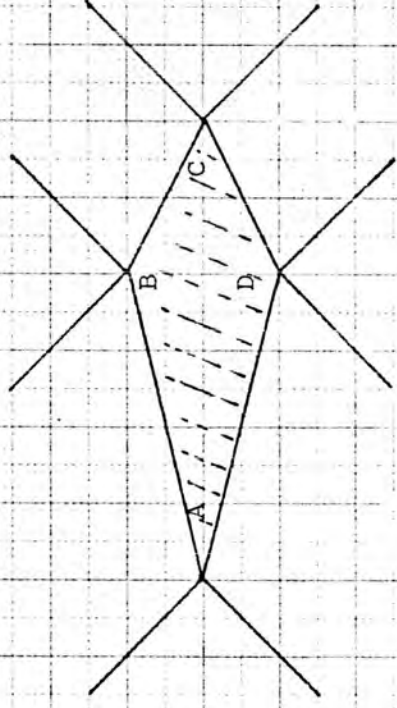
(3) Improper

(4) Improper

FIG. 7 - 2   Mesh lines for a hole on a plate or an obstacle in a fluid flow.

(1)   Case A, Re = 10

(2)   Case B, Re = 60.0

FIG. 7 - 3

Flow round a cylinder

density for each node must be used to enable the procedure to force and spread the effect of governing equations in the regions of interest.

7.6.2.Maximum stable time step Δt max.

Through numerical experiments it is found that the maximum stable time step Δt max also depends on Reynolds number, mesh size, surface curvatures of bodies, etc.

For the example one, if the input data shown in Appendix I is used, the values for $\Delta tmax$ are

Re	=	0.002	,	Δt max	=	0.00001
Re	=	1.	,	Δt max	=	0.0001
Re	=	2.	,	Δt max	=	0.0001
Re	=	3.	,	Δt max	=	0.0002
Re	=	4.	,	Δt max	=	0.0002
Re	=	5.	,	Δt max	=	0.0004
Re	=	6.	,	Δt max	=	0.0004

It may be mentioned here that it is difficult to determine exactly when Δt max is reached. Thus, all values must be considered as approximate. The parameters which affect the zones of convergence would also include maximum stable time step, Renolds number, mesh size,etc.
To accelerate the speed of convergence and to make the results stable, these .. parameters should be tried.

7.7. General discussions

It has been found that the finite element solution algorithm is capable of predicting some natural physical phenomena without resort to special devices. The finite element method is able to define the nodal points and elements arbitrarily to permit flexibility and easy accommodation of the complex boundary. Employing the knowledge of fluid dynamics concerning the anticipated solution distribution, a smaller element in regions containing larger spatial derivatives of the dependent variables can be used. Some of the other major advantages of the finite-element method

over finite difference ones are that different shapes may be represented

automatically, various boundary conditions may be satisified in a straight

forward manner, different element sizes may be used to get maximum

efficiency. A finer mesh can be used to gain details of the fluid flow

field exactly where desired. Generally significantly fewer equations are

required to provide a given accuracy. (16,62,77,100).

As far as computational efforts are concerned, a computer program for

the finite element computation seems to be more complicated than its

counterpart for the finite difference method. However, the complication

stems from the intrinsic generality of the finite element. The generality

of the finite element method usually leads to the computer program to be

applicable to a class of similar problems (19,25,33).

However, just as a Chinese saying has it "There is nothing in the world

which is perfect!" The finite element method also suffers from some

disadvantages. It is known that error analysis is very important in

numerical methods. However, up to now, there does not seem to be a method

which can be used to calculate the truncation error incurred by using a

particular kind of element shape. With the finite difference method, on

the other hand, the truncation error involved in any finite difference

formula can be analysed using the calculus of finite difference. However,

it is expected that the truncation error incurred will be comparable with

that of a finite difference mesh of the same size, so it is possible to

get some ideas about the order of approximation of a particular element

shape. When a higher order of approximation to the unknown function is

sought, the situation may become more complicated with the finite element

method. With the finite difference method, increasing the order of

approximation presents no real difficulty (25,116).

It can be shown that the finite element approach converges to the

exact solution as the number of elements is increased. Solution convergence

with finer finite element mesh is very significant for numerical solution

of non-linear equations like the Navier-Stokes equations. An insufficient

number of points on the solid boundary, at which no-slip boundary conditions are specified may cause unsatisfactory results. The no-slip condition determines the amount of vorticity created at the solid surface. The highest values of the wall vorticity and the vorticity gradient, which govern the spreading of vorticity in the fluid flow field are found on the solid wall (20,25,79,121).

The coefficient matrix $[[Kw] + [K\dot{w}] / \Delta t]$ is symmetric, banded, and positive definite. To keep the bandwidth of the coefficient matrix to a minimum, the nodal points should be ordered in such a way that the difference in nodal point numbers for any element be a minimum (25,33).

The three-node triangular element seemed to give quite accurate results. If the sizes of the elements are small enough, the approximation of the unknown function with the element is adequate. Owing to the simpler formulation and the ability to cater for arbitrary boundary shape, the three-node triangular element seems to be adequate for most purposes (63,116).

Although convergence is expected for higher Reynolds number, such a study was discontinued, in view of the fact that for higher Reynolds number fluid flow field, the channel between parallel plates or a constricting internal passage must be elongated and finer mesh must be used to get stable results, thus necessitating that the number of mesh points be increased so mush as to be impracticable for this computer.

Chapter 8   CONCLUSIONS

A general numerical procedure for the analysis of two-dimensional, time-dependent,incompressible,viscous fluid flow is presented.   A finite-element computer program is developed.

Using a combined variational principle-finite element method, difficulties arising from the nonlinearity of time-dependent Navier-Stokes equations have been remedied.   The numerical results obtained by the method have revealed very similar properties to known solutions of similar problems.   In the Stokes limit, the flow patterns are symmetric before and after the constriction, no separation occurred at the limit. The high-curvature body requires a time step that is an order of magnitude smaller than that for a low-curvature body in order to achieve stability. In order to avoid large errors, an extremely fine mesh must be used in the regions of large gradients of the vorticity.

The accuracy of the finite element scheme depends basically on the number of nodal points in the finite element mesh, the time step, and order of the numerical integration procedure.   Stable results can be obtained for a sufficiently small time step.   The simple time integration scheme was found to be sufficiently accurate for present tests.   To maintain the accuracy of the calculation, the number of iterations required increases slightly with Reynolds number.

Finally, some important points would be stressed as follows:

(1)   The maximum stable time step Δt max also depends on Reynolds number, grid sizes and the shape of a body.

(2)   Even if the finite element formulation used is the same, the zones of convergence for different problems may not be identical.   The zones of convergence also depend on the nature of the flow problem, Reynolds number, mesh sizes, time step, the way of constructing a mesh, and the shapes of obstacles, etc.

(3)   The boundary conditions at the body surface play a decisive part in the solution procedure.

(4) The topology and number of elements of a mesh also depend on the natural boundary conditions used. Many elements would be required to reasonably approximate such natural boundary conditions as occurred in test problem one.

(5) The fact that the numerical procedure will be convergent if Δt used is small enough means that the inertia terms play a stabilizing role in the scheme.

Using the finite element method, the region of low pressure of a body in flow, which accounts for most of the drag force may numerically be calculated. When the velocity of the fluid increases, a symmetric eddies can be produced behind the body which are alternatively shed. For low Reynolds number cases, this phenomenon can be predicted and the shedding frequencies can be found by employing the finite element method (135). These problems are interesting in the design of offshore structures.

It appears that the finite element method may be powerful to predict the natural physical phenomena of a fluid flow field. The next major fields for study would include stability and convergence problems as well as further research into applications of the method to the design of offshore structures.

REFERENCES

1.   Adey,R. and Brebbia, C.A.,
        "Finite Element Solution for Effluent. Dispersion"
        Numerical Methods in Fluid Dynamics, C.Brebbia and J.J.Connor
        (Eds.), Pentech Press, 1974.

2.   Atkinson, B., Brocklebank, M.P., Card, C.C.H. and Smith,J.M.
        "Low Reynolds Number Developing Flows",
        Am.Inst.Chem.Eng.J.,Vol.15, July 1969.

3.   Atkinson,B.,Card,C.C.H.and Irons,B.M.,
        "Application of the Finite Element Method to Creeping Flow Problems",
        Trans.Inst.Chem.Eng.,Vol.48.,1970.

4.   Baker,A.J.,
        "Finite Element Solution Algorithm for Viscous Incompressible
        Fluid Dynamics",
        Int.J'.Numer.Methods Eng.,Vol.6,No.1,1973.

5.   Baker,A.J.,
        A.Highly Stable Explicit Integration Technique for Computational
        Continuum Mechanics",
        Numerical Methods in Fluid Dynamics C.Brebbia and J.J.Conner (Eds.)
        Pentech Press,1974.

6.   Baker,A.J.,
        "Finite Element Solution Algorithm for Incompressible Fluid Dynamics"
        Proc.Int.Symp.on Finite Element Methods in Flow Problems, Swansea,
        1974.

7.   Baker,A.J.,
        "Predictions in Environmental Hydrodynamics.   Using the Finite
        Element Method, I.Theoretical Development,"
        J.A.I.A.A. Vol.13,No.1,1975.

8.   Bazeley,G.P.,Cheung,Y.K.,Irons,B.M., and Zienkiewicz,O.C.,
        "Triangular Elements in Plate Bending-Conforming and Non-
        Conforming Solutions",
        Proceedings of Conference on Matrix Methods and Structural
        Mechanics,AFFDL-TR-66-80.Oct.26-28,1965, Wright-Patterson Air Force
        Base,Ohio,p.547-576.

9.   Batchelor,G.K.,
        An Introduction to Fluid Dynamics,
        Cambridge University Press,1967.

10. Bateman,H.,
        "Notes on a Differential Equation which Occurs in Two-Dimensional
        Motion of a Compressible Fluid and Its Associated Variational
        Principles",
        Proceedings of the Royal Society,London,Series A.Vol.125,1929,
        p.598-618.

11. Beavers,G.S.,Sparrow,E.M., and Magnuson,R.A.,
        "Experiments on Hydrodynamically Developing Flow in Rectangular
        Ducts of Arbitrary Aspect Ratio",
        International Journal of Heat and Mass Transfer,Vol.13,1970,p.689-702

12. Bird,R.B.,
      "New Variational Principle for Incompressible Non-Newtonian Flow"
      Phys.Fluids,3,p.539-541,1960.

13. Bogner,F.K.,Fox,R.L., and Schmit,L.A.,
      "The Generation of Interelement Compatible Stiffness and Mass
      Matrices by the Use of Interpolation Formulae",
      Proceedings of Conference on Matrix Methods and Structural
      Mechanics, Oct.26-28.1965, Wright-Patterson Air Force Base,Ohio.

14. Bowley,W.W.,and Prince,J.F.,
      "Finite Element Analysis of General Fluid"
      A.I.A.A. Paper No.71-602,1971.

15. Bramlette,T.T.,
      "Plane Poiseville Flow of a Rarefied Gas Based on the Finite Element
      Method",
      Phys.Fluids,Vol.14,No.2,1971.

16. Bratanow,T.,Ecer,A., and Kobiski,M.,
      "Finite Element Analysis of Unsteady Incompressible Flow around
      an Oscillating Obstacle of Arbitrary Shape",
      J.A.I.A.A.,Vol.11,No.11,Nov.1973, p.1471-1477.

17. Bratanow,T., and Ecer,A.,
      "On the Application of the Finite Element Method in Unsteady
      Aerodynamics",
      J.A.I,A.A.,Vol.12,No.4,April 1974,p.503-510.

18. Bratanow,T., and Ecer,A.,
      "Analysis of Moving Body Problems in Aerodynamics",
      Proc.Int..Symp.on Finite Element Methods in Flow Problems,Swansea,1974.

19. Bratanow,T., and Ecer,A.,
      "Suitability of the Finite Element Method for Analysis of Unsteady
      Flow around Oscillating Airfoils",
      International Conference on Numerical Methods in Fluid Dynamics,
      University of Southampton, Southampton,England,Sept.26-28,1973.

20. Bratanow,T., and Ecer,A.,
      "Analysis of Three-Dimensional Unsteady Viscous Flow around
      Oscillating Wings",
      J.A.I.A.A.,Vol.12,No.11,Nov.1974.

21. Brebbia,C.A.,
      "Some Applications of Finite Elements for Flow Problems",
      Int.Conf.Variational Meth.Engng.,Southampton,Univ.C.A.Brebbia
      and H.Tottenham (Eds.)Vol.I p.5/1-5/26,1973.

22. Brebbia,C.A., and Connor,J.J.,
      Fundamentals of Finite Element Techniques for Structural Engineers,
      Butterworths,1973.

23. Burggraf, O.R.,
      "Analytic and Numerical Studies of the Structure of SteadySeparated
      Flows",
      J.Fluid Mech.,24, p.113-151,1966.

24. Carlson,G.A., and Hornbeck,R.W.,
   "A Numerical Solution for Laminar Entrance Flow in a Square Duct",
   J.of Applied Mechanics,March 1973 , p.25-30,Transation of the ASME.

25. Cheng,R.T.,
   "Numerical Solution of the Navier-Stokes Equations by the Finite
   Element Method",
   Phys.Fluids,Vol.15,No.12,1972.

26. Cheng,R.T.,
   "Numerical Investigation of Lake Circulation Around Islands by the
   Finite Element Method",
   Int.J.Numer.Methods Engng.,Vol.5, No.1,1972.

27. Cheng,S., and Hoff,N.J.,
   "Bending of Thin Circular Rings"
   Int.J.Solids Structures,Vol.11,p.139-152,1975.

28. Cheng,S.I.,
   "Accuracy of Difference Formulation of Navier-Stokes Equations"
   The Physics of Fluids,Suppl.II,Vol.12,No.12,Dec.1969,p.34.

29. Cheng,S.I., and Rimon,Y.,
   "Numerical Solution of a Uniform Flow over a Sphere at Intermediate
   Reynolds Numbers",
   The Physics of Fluid,Vol.12,No.5,May 1969,p.949.

30. Chorin,A.J.,
   "Numerical Study of Slightly Viscous Flow",
   J.of Fluid Mechanics,Vol.57,Part 4,March 1973, p.785-796.

31. Chorlton,F.,
   Textbook of Fluid Dynamics
   D.Van Nostrand Company Ltd.,1967.

32. Collins,M. and Schowalter,W.R.,
   Physics of Fluids 5, p.1122-1124, 1962.

33. Connor,J.J. and Brebbia,C.A.
   Finite Element Techniques for Fluid Flow,
   Butterworth & Co.Ltd.,1976.

34. Davis,R.T.
   "Numerical Solution of the Navier-Stokes Equations for Symmetric
   Laminar Incompressible Flow Past a Parabola",
   J.of Fluid Mechanics,Vol.51,Part 3,p.417-433,1972.

35. Dennis,S.C.R. and Chang,G.Z.
   "Numerical Solutions for Steady Flow Past a Circular Cylinder at
   Reynolds Numbers Up to 100",
   J.of Fluid Mechanics,Vol.42,Part 3,1970.

36. Dennis,S.C.R. and Walsh,J.D.
   "Numerical Solutions for Steady Symmetric Viscous Flow Past a
   Parabolic Cylinder in a Uniform Stream",
   J.Fluid Mechanics,Vol.50,Part 4,p.801-814,1971.

37. Desai,C.S., and Abel,J.F.
          Introduction to the Finite Element Method:A Numerical Method for
          Engineering Analysis,
          Van Nostrand-Reinhold,New York,1972.

38. Doctors,L.J.
          "An Application of the Finite Element Technique to Boundary Value
          Problems of Potential Flow,"
          Int.J.Numer Methods Engng.,Vol.2,No.2.1970.

39. Fargie,D. and Martin,B.W.
          "Developing Laminar Flow in a Pipe of Circular Cross-Section",
          Proc.Roy.Soc.Lond.A.321.p.461-476.1971.

40. Fenner,R.T.
          "Finite Element Methods for Engineers"
          The Macmillan Press Limited,1975.

41. Finlayson,B.A.
          "Exsistence of Variational Principles for the Navier-Stokes Equations",
          The Physics of Fluids,Vol.15,No.6,June 1972.

42. Finlayson,B.A.
          "Weighted Residual Methods and their Relation to Finite Element
          Methods in Flow Problems",
          Proc.Int.Symp.on Finite Element Methods in Flow Problems,Swansea,1974.

43. Finlayson,B.A. and Scriven,L.E.
          "The Method of Weighted Residuals - a Review",
          Applied Mech.Rev.Vol.19,No.9,p.735-748,1966.

44. Finlayson,B.A. and Scriven,L.E.
          "On the Search for Variational Principles",
          Int.J.Heat Mass Transfer,10,p.799-821.1967.

45. Farray,M.J.
          Variational Calculus in Science and Engineering,
          New York, McGraw-Hill Book Co., 1968.

46. Fromm,J.E.
          "Practical Investigation of Convective Difference Approximations
          of Reduced Dispersion",
          The Physics of Fluids,Suppl.II,Vol.12,No.12,Dec.1969,p.3.

47. Gelder,D.
          "Solution of the Compressible Flow Equations",
          Int.J.Numer.Methods Eng.,Vol.3,No.1,1971.

48. Ghia,U. and Davis,R.T.
          "Navier-Stokes Solutions for Flow Past a Class of Two-Dimensional
          Semi-Infinite Bodies",
          J.A.I.A.A.,Vol.12,No.12,Dec.1974.p.1658-1665.

49. Goldstein,S.
          "Modern Developments in Fluid Dynamics",
          Oxford.

50. Goldstein,R.J. and Kreid,D.K.
    "Measurement of Laminar Flow Development in a Square Duct
    Using a Laser-Doppler Flow-meter",
    Journal of Applied Mechanics,Vol.34, Trans.ASME,Vol.89,1967,p.813-818

51. Greenspan,D.
    "Numerical Studies of Steady, Viscous, Incompressible Flow in a
    Channel with a Step",
    Journal of Engineering Mathematics,Vol.3No.1 Jan. 1969.p.21-28.

52. Greenspan,D.
    "Numerical Studies of Viscous Incompressible Flow Through an Orifice
    for Arbitrary Reynolds Number",
    Int.J.Numer.Methods.Eng.Vol.6,p.486-496,1973.

53. Guran,R.J.,Ujihara,B.H.and Welch,P.W.
    "Hydroelastic Analysis of Axisymmetric Systems by a Finite Element
    Method",
    Proceedings of 2nd Conference on Matrix Methods in Structural
    Mechanics,Wright-Patterson Air Force Base,Dayton,Ohio,1968.

54. Gurtin,M.E.
    "Variational Principles for Linear Initial Value Problems",
    Quart.Appl.Maths,22,p.252-256,1964.

55. Hai,S.M.
    "Pressure Development in the Entrance Region and Fully Developed
    Region of Generalised Channel Turbulent Flows",
    J.of Applied Mechanics,March 1976,p.13-19.Trans of the ASME.

56. Han.L.S.
    "Hydrodynamics Entrance Lengths for Incompressible Flow in
    Rectangular Ducts",
    J.of Applied Mechanics,Vol.27.Trans.ASME,Vol.82,Series,E.1960.
    p.403-409.

57. Harlow,F.H. and Fromm,J.E.
    "Numerical Solution of the Problem of Vortex Street Development",
    The Physics of Fluids,Vol.6,No.7,July 1963.

58. Harlow,F.H. and Welch,J.E.
    "Numerical Calculations of Time Dependent Viscous Incompressible
    Flow of Fluid with Free Surface",
    The Physics of Fluids,Vol.8,No.12,1965,p.2182-2189.

59. Hildebrand,F.B.
    Introduction to Numerical Analysis,
    McGraw-Hill,New York,1956.

60. Hirsch,C.H. and Warzee,G.
    "A Finite Element Method for through Flow-Calculations in Turbo-
    machines",
    J.of Fluids Engineering,Sept.1976,p.403-421,Trans.of the ASME.

61. Holand,I.and Bell,K.eds.
    Finite Element Methods in Stress Analysis,
    Tapir,Trondheim,Norway.

62. Hood,P.
        PhD.Thesis,University of Wales,Swansea,U.K.1974.

63. Huebner,K.H.
        The Finite Element Method for Engineers,
        John Wiley and Sons,Inc.,1975.

64. Hussey,M.J.L.,Thatcher,R.W. and Bernal,M.J.M.
        "On the Construction and Use of Finite Elements",
        J.Inst.Maths.Applics.6, p.263-282,1970.

65. Ikegawa,M. and Washizer,K.
        "Finite Element Method Applied to Analysis Flow over a Spillway Crest",
        Int.,J.Numer.Methods.Eng.,Vol.6,No.2,1973.

66. Irons,B.M.,
        "Economical Computer Techniques for Numerically Integrated Finite
        Elements"
        Int.J.Numer.Methods Eng.,Vol.1,p.201-203,1969.

67. Irons,B.M.,
        "A Frontal Solution Program for Finite Elements Analysis",
        Int.J.Numer.Methods Eng.,Vol.2,No.1,1970.

68. Irons,B.M.,de Oliveria,E.A. and Zienkiewicz,O.C.
        Comments on the paper: Theoretical Foundations of the Finite Element
        Methods,
        Int.J.Solids Struct.6,p.695-697.

69. Isaacs,L.T.
        "A Curved Cubic Triangular Finite Element for Potential Flow Problems"
        Int.J.Numer.Methods Eng.Vol.7,p.337-344,1973.

70. Jain,P.C. and Rao,K.S.
        "Numerical Solution of Unsteady Viscous Incompressible Flow Past
        a Circular Cylinder",
        Physics of Fluids Supplement II,p.57-64,1969.

71. Kawahara,M. and Yoshimara,N.
        "Steady and Unsteady Finite Element Analysis of Incompressible
        Viscous Fluid",
        Int.J.Numer.Methods Eng.Vol.10,p.437-456,1976.

72. Kinney,R.B. and Paolino,M.A.
        "Flow Transient Near the Leading Edge of a Semi-Infinite Flat
        Plate Moving Through a Viscous Fluid",
        Journal of Applied Mechanics,Vol.42,No.4,Dec.1974,p.919-924.

73. Kitchens,C.W.
        "Calculation of Low Reynolds Number Flow Past a Square Protuberance",
        J.AIAA, July 1974, p.1005-1007,Technical Notes.

74. Kuwahara,K. and Imai,I.
        "Steady, Viscous Flow within a Circular Boundary",
        The Physics of Fluids Supplement II,p.94-104,1969.

75. Lamb,H.
        Hydrodynamics.
        Dover,N.Y.,1945.

76. Langhaar,H.L.
    "Steady Flow in the Transition Length of a Straight Tube",
    Journal of Applied Mechanics,Vol.9,Trans.ASME,Vol.64,1942,
    p.55-58.

77. Laskaris,T.E.
    "Finite Element Analysis of Compressible and Incompressible
    Viscous Flow and Heat Transfer Problems"
    The Physics of Fluids,Vol.18,No.12,Dec.1975. p.1639-1648.

78. Lee,C.H.
    "Finite Element Method for Transient Linear Viscous Flow Problems"
    Proc.Int.Conf.Numerical Methods in Fluid Dynamics,Southampton,1973.

79. Lee,J.S. and Fung,Y.C.
    "Flow in Locally Constricted Tubes at Low Reynolds Number",
    Journal of Applied Mechanics,Trans.of the ASME,March 1970,p.9-16.

80. Leonard,J.W.
    "Finite Element Analysis of Perturbed Compressible Flow",
    Int.J.Numer.Methods Eng.,Vol.4,No.1,1972.

81. Lew,H.S. and Fung,Y.C.
    "On the Low Reynolds Number Entry Flow Into a Circular Cylindrical
    Tube",
    Journal of Biomechanics,Vol.2,1969,p.105-119.

82. Lew,H.S. and Fung,Y.C.
    "Arbitrary Reynolds Numbers",
    Journal of Biomechanics,Vol.3,1970,p.23-38.

83. Lin,C.L.,Pepper,D.W. and Lee,S.C.
    "Numerical Methods for Separated Flow Solutions around a Circular
    Cylinder",
    J.AIAA, Vol.14,No.7,1976,p.900-907.

84. Loitsyanskii L.G.
    Mechanics of Liquids and Gases,
    Pergamon Press,1966.

85. Lundgren,T.S.Sparrow,E.M. and Starr,J.B.
    "Pressure Drop due to the Entrance Regions in Ducts of Arbitrary
    Cross Sections",
    Journal of Basic Engineering,Trans.ASME,Series D,Vol.86,1964,p.620-
    626.

86. Lyness,J.F.,Owen,D.R.J. and Zienkiewicz,O.C.
    "Finite Element Analysis of the Steady Flow of Non-Newtonian Fluids
    through Parallel Sided Conducts",
    Int.Symp.on Finite Element Methods in Flow Problems,Swansea,1974.

87. Macagno,E.O. and Hung,T.K.
    "Computational and Experimental Study of a Captive Annular Eddy",
    Journal of Fluid Mechanics,Vol.28,No.1,Jan.1967,p.43-64.

88. Martin,H.C.
    "Finite Element Analysis of Fluid Flows",
    Proceedings of Second Conference on Matrix Methods in Structural
    Mechanics,AFFDL-TR68-150,Wright-Patterson Air Force Base,Dayton,
    Ohio,October 1968.

89. Martin,H.C. and Carey,G.F.
        Introduction to Finite Element Analysis,
        McGraw-Hill,1973.

90. Meissner,V.
        "A Mixed Finite Element Model for Use in Potential Flow Problems",
        Int.J.Numer.Methods Eng.,Vol.6,No.4,1973.

91. Mills,R.D.
        "Numerical Solutions of Viscous Flow through a Pipe Orifice at
        Low Reynolds Numbers",
        Journal of Mechanical Engineering Sciences,Vol.10,1968,p.133-140.

92. Moretti,G.
        "Importance of Boundary Conditions in the Numerical Treatment of
        Hyperbolic Equations,"
        Physics of Fluids,Vol.12,No.12,Dec.1969,p.II.13-20.

93. Norrie,D.H. and de Vries,G.
        "Application of the Pseudo-Functional Finite Element Method to
        Non-Linear Problems,"
        Proc.Int.Symp.on Finite Element Methods in Flow Problems,Swansea,
        1974.

94. Norrie,D.H. and de Vries,G.
        The Finite Element Method,Fundametals and Applications,
        Academic Press,1973.

95. Oden,J.T.
        "A General Theory of Finite Elements",
        Int.J.Numer.Methods Eng.Vol.1,p.247-259,1969.

96. Oden,J.T.
        "A Finite Element Analog of Navier-Stokes Equations",
        Proc.ASCE,96,EM4, p.529-534,1970.

97. Oden,J.T.
        Finite Elements of Nonlinear Continua,
        McGraw-Hill,N.Y.,1972.

98. Oden,J.T. and Somogyi,D.
        "Finite Element Applications in Fluid Dynamics",
        J.Eng.Mech.Div.Proc.ASCE,95,EM4,p.821-826,1969.

99. Oliveira,E.R.A.
        "Convergence of Finite Element Solutions in Viscous Flow Problems"
        Int.J.Numer,Methods Eng.,Vol.9,p.739-763,1975.

100. Olson,M.D.
        "Variational-Finite Element Methods for Two-Dimensional and
        Axisymmetric Navier-Stokes Equations",
        Proc.Int.Symp.on Finite Element Methods in Flow Problems,
        Swansea,1974.

101. Palit,K. and Fenner,R.T.
        "Finite Element Analysis of Slow Non-Newtonian Channel Flow",
        A.I.Ch.E.J.,18,p.628-633,1972.

102. Payne,R.B.
        "Calculations of Unsteady Viscous Flow Past a Circular Cylinder",
        Journal of Fluid Mechanics,Vol.4,Part 1,May 1958,p.81-86.

103. Pearson,C.E.
        "A Computional Method for Viscous Flow Problems",
        Journal of Fluid Mechanics,p.611,Vol.21,1965.

104. Pian,T.H.H. and Tong,P.
        "Basis of Finite-Element Methods for Solid Continua",
        Int.J.Numer.Methods Eng.,Vol.1 1969,p.3-28.

105. Rimon,Y.
        "Numerical Solution of the Incompressible Time-Dependent Viscous
        Flow past a Thin Oblate Spheroid",
        The Physics of Fluids Supplement II,1969,p.65-75.

106. Roache,P.J. and Mueller,T.J.
        "Numerical Solutions of Laminar Separated Flows",
        J.AIAA,Vol.8,No.3,March 1970,p.530-538.

107. Rouse,H.
        Advanced Mechanics of Fluids,
        John Wiley & Sons,Inc.,1959.

108. Roscoe,D.F.
        "The Solution of the Three-Dimensional Navier-Stokes Equations
        Using a New Finite Difference Approach",
        Int.J.Numer.Methods Eng.,Vol.10,p.1299-1308,1976.

109. Schmall,R.A. and Kinney,R.B.
        "Numerical Study of Unsteady Viscous Flow Past a Lifting Plate",
        J.AIAA,Vol.12,Nov.1974,p.1573-1576.

110. Smith,S.L. and Brebbia,C.A.
        "Finite-Element Solution of Navier-Stokes Equations for Transient
        Two-Dimentional Incompressible Flow",
        Journal of Computational Physics,Vol.17,p.235-245,1975.

111. Sparrow,E.M.,Hixon,C.W. and Shavit,G.
        "Experiments on Laminar Flow Development in Rectangular Ducts",
        Journal of Basic Engineering,Trans.ASME,Series D.Vol.89,No.1,
        March 1967, p.116-124.

112. Sparrow,E.M.,Lin,S.H. and Lundgren,T.S.
        "Flow Development in the Hydrodynamic Entrance Region of Tubes
        and Ducts",
        Physics of Fluids,vol.7.1964,p.338-347.

113. Strang,G. and Fix,G.J.
        "An Analysis of the Finite Element Method",
        Prentice-Hall,Inc.,Englewood Cliffs,N.J.,1973.

114. Takaisi,Y.
        "Numerical Studies of a Viscous Liquid past a Circular Cylinder",
        The Physics of Fluids,Supplement V,Vol.12,No.12,Dec.1969,p.86.

115. Takami,H. and Keller,H.B.
        "Steady Two-Dimensional Viscous Flow of an Incompressible Fluid
        past a Circular Cylinder",
        The Physics of Fluids,Supplement II,Vol.12,No.12,Dec.1969,p.51-56

116. Tay,A.O. and de Vahl Davis,G.
     "Application of Finite Element Method to Convection Heat
     Transfer Between Parallel Plantes",
     Int.J.Heat Mass Transfer,Vol.14,No.8,1971.

117. Taylor,C. and Hood,P.
     "A Numerical Solution of the Navier-Stokes Equations Using the
     Finite Element Technique",
     Computers and Fluids,Vol.1,p.73-100,1973.

118. Thompson,E.C. and Hague,M.I.
     "A High-Order Finite Element for Completely Incompressible
     Creeping Flow",
     Int.J.Numer.Methods Eng.,Vol.6,No.3,1973.

119. Thompson,J.F.,Shanks,S.P., and Wu,J.C.
     "Numerical Solution of Three-Dimensional Navier-Stokes Equations
     Showing Trailing Tip Vortices",
     J.AIAA,Vol.12,June 1974,p.787-794.

120. Thom,A.
     "The Flow Past Circular Cylinders at Low Speeds",
     Proc.Royal Society,London,Series A,Vol.126,1933,p.651-669.

121. Thom,A and Apelt,C.F.
     Field Computations in Engineering and Physics
     Van Nostrand,Princeton,N.J.,1961.

122. Thoman,D.C. and Szewczyk,A.A.
     "Time Dependent Viscous Flow over a Circular Cylinder",
     The Physics of Fluids,Supplement V,Vol.12,No.12,p.76-85,1969.

123. Tong,P.
     "The Finite Element Method for Fluid Flow"
     Recent Advances in Matrix Method of Structural Analysis and
     Design,R.Gallagher et al (Eds.) 1971.

124. Tong,P. and Fung,Y.C.
     "Slow Particulate Viscous Flow in Channels and Tubes - Applications
     to Biomechanics"
     Trans.ASME,Series E,Journal of Applied Mechanics,Vol.38,Dec.1971.

125. Vallentine,H.R.
     Applied Hydrodynamics,
     Butterworths,London,1959.

126. Wiginton,C.L. and Dalton,C.
     "Incompressible Laminar Flow in the Entrance Region of a
     Rectangular Duct",
     Journal of Applied Mechanics,Vol.37,Trans.ASME,Vol.92,1970,p.854-856.

127. Wilson,J.M.
     Lectures on Finite Element Methods,Dept.of Engineering Science,
     University of Durham,England,1976.

128. Wu,J.C.
     "Numerical Boundary Conditions for Viscous Flow Problems",
     J.AIAA,Vol.14,No.8,p.1042-1049,1976.

129. Yamada,Y.Yokouchi,K.Ito,Y.and Ohtsubo,T.
      "Finite Element Analysis of Steady Fluid and Metal Flow",
      Proc.Int.Symp.on Finite Element Methods in Flow Problems,
      Swansea,1974.

130. Zlamal,M.
      "The Finite Element Method in Domains with Curved Boundaries",
      Int.J.Numer.Methods Eng.Vol.5,p.367-373,1973.

131. Ziekiewicz,O.C.
      "The Finite Element Method in Engineering Science,
      McGraw-Hill, London,1971.

132. Ziekiewicz,O.C. and Cheung,Y.k.
      "Finite Elements in the Solution of Field Problems",
      The Engineer, 220,p.507-510,1965.

133. Ziekiewicz,O.C. and Godbole,P.N.
      "Flow of Plastic and Viscoplastic Solids with Special Reference
      to Extrusion and Forming Processes",
      Int.J.Numer.Methods Eng.,Vol.8, p.3-16,1974.

134. Hood,P.
      "Frontal Solution Program for Unsymmetric Matrices",
      Int.J.Numer.Methods Eng.Vol.10,p.379-400, 1976.

135. Brebbia,C.A.
      "Fluid Structure Interaction Problems", in Vibrations of
      Engineering Structures, Computational Mechanics Ltd.,
      Southampton, U.K.,1975.

136. Schlichting,H.
      Boundary Layer Theory, 6th ed.,
      McGraw-Hill New York,1968.

$$
\left[Q_u\right]^{(e)} = \frac{1}{24}
\begin{bmatrix}
2(c1-c2-c3) & -(c1+c2+c3) & -(c1+c2+c3) & 4(2c3+c1) & 4(2c2+c1) & 4(c3+c2) \\
-(c1+c2+c3) & 2(c2-c3-c1) & -(c1+c2+c3) & 4(c1+c3) & 4(c2+2c1) & 4(2c3+c2) \\
-(c1+c2+c3) & -(c1+c2+c3) & 2(c3-c1-c2) & 4(c3+2c1) & 4(c2+c1) & 4(c3+2c2)
\end{bmatrix}
$$

$$
\left[Q_v\right]^{(e)} = \frac{1}{24}
\begin{bmatrix}
2(b1-b2-b3) & -(b1+b2+b3) & -(b1+b2+b3) & 4(2b3+b1) & 4(2b2+b1) & 4(b3+b2) \\
-(b1+b2+b3) & 2(b2-b3-b1) & -(b1+b2+b3) & 4(b1+b3) & 4(b2+2b1) & 4(2b3+b2) \\
-b1+b2+b3 & 2(b3-b1-b2) & -(b1+b2+b3) & 4(b3+2b1) & 4(b2+b1) & 4(b3+2b2)
\end{bmatrix}
$$

APPENDIX A

```
      COMMON M,NL,I,IL
      REAL*4 NCNC
      EQUIVALENCE (R1,M2,M3,M4,M5,M6,M7,M8,M9,M10,M11,
     1 M12,M13,M14,M15,M16,M17,M18,M19,M20,M21,M22,
     2 M23,M24,M25,M26,M27,M28,M29,M30,M31,M32,M33,M34,M35,M36,M37,M38,
     3 M39,M40,M41,M42,M43,M44,M45,M46,M47,M48,M49,M50,M51,M52,M53,
     4 M54,M55,M56,M)
      EQUIVALENCE (IA1,IA2,IA3,IA4,IA5,IA6,IA7,IA8,IA9,IA10,IA)
      EQUIVALENCE (IB1,IB2,IB3,IB)
      EQUIVALENCE (IC1,IC2,IC3,IC)
      EQUIVALENCE (IC1,IC2,IC3,IC4,IC5,IC6,IC7,IC8,IC9,IC10,IC11,IC)
      EQUIVALENCE (AL,NL1,NL2,NL3,NL4)
C
      READ(5,5002) M,NL,I,IL
      WRITE(6,5002) M,NL,I,IL
5002  FORMAT(4I5)
C
      IA=NL+IL
      IB=L+M
      IC=N+M
      IC=NL+IL+IL
      IF=NL+6
      M1=M
      IA1=IA
      IB1=IB
      IC1=IC
      IC1=D
      NL1=NL
C
      CALL DYNNIC(
     1 XC,M1,YC,M2,NCNC,IA1,NIS,M3,VIS,M4,N2S,M5,N3S,M6,A3S,M7,V3S,M8,
     2 N4S,M9,V4S,M10,A5S,M11,VT,IA1,N6S,M12,N7S,M13,VC,M14,VK,M15,
     3 YECL,IA3,N2F,IA4,N3F,IA5,N5LF,IA6,FKSTF,IC1,N9S,M33,N5M,
     4 N2F,FKVTTF,IC2,M5,M1,M47,AEKVT,IC1,N8F,M22,ASSTF,IA9,
     5 VTN1,N16,VTF,M19,VTF,M17,V7S,FKVTC,M12,ASSI,IA9,VISI,IE,AMK,IAC,
     6 XECL,IA2,ASST,IB3,KVK,M17,V7S,FKVTC,M12,ASSI,IA9,VISI,IE,AMK,IAC,
     7 XECL,IA2,ASST,IB3,N8F,M13,ASSTAT,M32,VISI,IE,AMK,IAC,
     8 ASSTN1,M25,VTT,IA10,UVV,M30,VVV,M31,AESVT,IA8,AESVT,M16,
     9 AEKPT,IC10,AEKUT,ID11,VPS,M54,P1S,M55,PS,M56)
C
      STOP
      END
```

```
C     FINITE-ELEMENT ANALYSIS OF INCOMPRESSIBLE,UNSTEADY
C
C     VISCOUS FLOW
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
      SUBROUTINE MAINPF(
     1 XC,M1,YC,M2,NCNC,IA1,NIS,M3,VIS,M4,N2S,M5,N3S,M6,A3S,M7,V3S,M8,
     2 N4S,M9,V4S,M10,N5S,M11,VT,IA1,N6S,M12,A7S,M13,ACV,I05,IA4,V1S,
     3 YECL,IA3,N2F,M4,N5F,IA5,CC,IA6,N5LE,M14,FKSTF,IC1,ASST,IA7,
     4 N2F,FKVTTF,IC3,N6F,IA7,AEKVT,IC1,ASST,IE1,AEKVT,IC1,ACV,I05,
     5 VTN1,N18,VTF,M19,VTF,M1,M54,I04,SM2,M21,DIFF,M22,ASSTF,IA9,
     6 ABU,IO5,AUI,IO6,ACV,IO7,R,IO8,Z,IO9,Z,M26,C,M27,SM5,M28,SS,M37
     7 ,XECL,IA2,ASSTA,IB3,KKK,M17,V7S,M14,ASSTAT,M32,VISI,M14,AM,IAC,
     8 NF1,NL1,NE2,NL2,NL3,NL2,NL3,FKVTF,IC2,ASST,IB2,FSVT,IA9,ASVT,IA9,
     9 ASSTN1,M25,VTT,IA10,UVU,M30,VVV,M31,AESPP,M24,PF,M25,SKEFF,IA3,
     1 RP,M35,NPM,M40,NL,M41,NUM,M43,AV,M42,NUA,M43,MUB,M49,NSUB,M50,NPS,M51,
     2 NUA,M46,ASUA,M47,AMUA,M42,AMUA,N54,P1S,M55,PS,M56)
     3 AEKPT,IC10,AEKUT,ID11,VPS,M54,P1S,M55,PS,M56)
C
      COMMON M,NL,I,IL
C
      INTEGER PTC,FK,RTC1,TFST,TFS1
C
      DATA STCP/'STCP'/
      DATA SLF1/'SLF1'/
      DATA SLF2/'SLF2'/
      DATA VTFI/'VTFI'/
      DATA FK/' FK'/
      DATA PTC1/'RTC1'/
      DATA TFS1/'TFS1'/
C
      REAL*4 NCNC
      DIMENSION XC(M),YC(M)
      DIMENSION NCNC(NL,IL)
      DIMENSION N1S(M),V1S(M),A2S(M),V2S(M)
      DIMENSION N3S(M),V3S(M),N4S(M),V4S(M)
      DIMENSION N5S(M),VT(L,M),N6S(M),N7S(M),V7S(M)
      DIMENSION NCNC(M),A5S(M)
      DIMENSION NF1(NL),NE2(NL),NE3(NL)
      DIMENSION XCL(3),YCL(3)
      DIMENSION XECL(NL,IL),YECL(NL,IL)
      DIMENSION AA(3),PU(3),CC(3)
      DIMENSION ZAF(NL,IL),FEE(NL,IL),CCE(NL,IL)
      DIMENSION DEIS(NL)
```

```
      DIMENSION ASST(L,M),ASSTAT(M),ASSTAT(L,M)
      DIMENSION BP(M),PFI(M),AFM(M),NI(M)
      DIMENSION NUP(M),NVM(M),NVM(M)
      DIMENSION EKVTT(3,3),
     1          EKVTTE(AL,IL,IL),
     2          N(3),NI(ML,IL)
      DIMENSION NUA(M),ASOA(M),APJA(M),PUP(M)
      DIMENSION NSLB(M),NPUR(M),C1(M),V21(M)
      DIMENSION EKUT(3,3),AEKUT(M,3),VPS(M),PIS(M),PS(M)
      DIMENSION         AEKVTT(M,P),AEKMT(M,M),DEKST(M,M),AEKVT(M,M),AEKVTT(M,M)
      DIMENSION         ESVT(AL,IL)
      DIMENSION AESVT(M),KKF(M),VTNL(I),VTN(M)
      DIMENSION SML(M,M),ASS(M),NSM(M)
      DIMENSION SM2(M),SM3(M),VKEEP(M)
      DIMENSION DIFF(M),VISI(AL,IL)
      DIMENSION ASSTEA(AL,IL)
      DIMENSION ASSTK(M),EKKEP(M),AWN(M,M),WN(3,3),VKT(V)
      DIMENSION VTT(AL,IL)
      DIMENSION VISXX(3),VISYY(3),VISXY(3)
      DIMENSION GGG(2),ESPF(3)
      DIMENSION AESPF(M),PF(M)
      DIMENSION RU(3,3),QVI3,6),QV(3,6)
      DIMENSION ARU(3,M),AQU(M,M),AQV(M,M)
      DIMENSION B(M,M),Z(M),C(M)
      DIMENSION SMC(M),SM6(M)
      DIMENSION LCC(M),SLEL(7,26),VVV(M)
      DIMENSION XC4(2),YC4(2)
C
C
C     READ THE NUMBER OF SYSTEM NODES AND ELEMENTS
C
      REAC(5,1) NN,NE
      WRITE(8,1) NN,NE
C
C     INITIALIZE PARAMETERS
C
      DO 5 I=1,NN
      N1S(I)=0
      V1S(I)=0.0
      N2S(I)=0
      V2S(I)=0.0
      N3S(I)=0
      V3S(I)=0.0
      N4S(I)=0
      V4S(I)=0.0
      N5S(I)=0
      N6S(I)=0
      N7S(I)=0
      N8S(I)=0
      V7S(I)=0.0
      NCAC(I)=0.0
      AESVT(I)=0.0
      ASSTAT(I)=0.0
      SP2(I)=0.0
      AESPP(I)=0.0
      PP(I)=0.0
      PS(I)=0.0
      C(I)=0.0
      SMS(I)=0.0
      GMC(I)=0.0
C
      DO 3 J=1,NN
      AKU(I,J)=0.0
      AQU(I,J)=0.0
      AQV(I,J)=0.0
      K(I,J)=0.0
      URU(I,J)=0.0
      VVV(I)=0.0
    3 CONTINUE
C
      DO 4 I=1,NN
      DO 4 J=1,NN
      AEKST(I,J)=0.0
      AEKPT(I,J)=0.0
      AEKVT(I,J)=0.0
      AEKVT(I,J)=0.0
      AEKVTT(I,J)=C.0
    4 CONTINUE
C
      DO 2 I=1,L
      DO 2 J=1,M
      ASST(I,J)=0.0
      VT(I,J)=0.0
    2 CONTINUE
C
      DO 5 K=1,NE
      NF1(K)=0
      NE2(K)=0
      NE3(K)=0
    5 CONTINUE
C
      ICCINT=1
      NCCLNT=1
C
C     READ NODE NUMBERS AND NODAL COORDINATES
C
      DO 7 I=1,NN
      READ(5,8) J,XC(I),YC(I)
    7 CONTINUE
C
C     READ SYSTEM TOPOLOGY (ELEMENT NO. AND NODE NUMBERS
C     IN COUNTER-CLOCKWISE FASHION)
C
      DO 9 I=1,NE
      READ(5,10) J,NCCF(I,1),NCCF(I,2),NCCF(I,3)
    9 CONTINUE
C
C     READ KINEMATIC VISCOSITY, AND DENSITY
C
      READ(5,11) CNIU,DEN
C
C     READ BOUNDARY CONDITIONS AND INITIAL CONDITIONS
C
C     READ BOUNDARY CONDITION 1 (NI1=NODE NUMBER, V1=STEADY FUNCTION)
```

```
      N1S(I)=N11
      V1S(N1S(I))=V1
      IF (YC(N1S(I)).LE.SLF1) GO TO 1222
      V1=1.5*(YC(N1S(I))-YC(N1S(I)))**2/3.0)
      V1S(N1S(I))=V1
1222  IF (WORD.AG.SLF2) GO TO 12
      V1=YC(N1S(I))
      V1S(N1S(I))=V1
12    CONTINUE
      I=AA+1
15    N1T=I-1
C
C     READ BOUNDARY CONDITION 2 (N22=NODE NUMBER,
C     V2=THE DERIVATIVE OF THE STREAM FUNCTION WITH RESPECT TO X)
C
      DO 17 I=1,AA
      READ(5,18) WORD,N22,V2
      IF (WORD.EQ.STOP) GO TO 518
      N2S(I)=N22
      V2S(N2S(I))=V2
17    CONTINUE
      I=AA+1
518   N2T=I-1
C
C     READ BOUNDARY CONDITION 3 (N33=NODE NUMBER,
C     V3=THE DERIVATIVE OF THE STREAM FUNCTION WITH RESPECT TO Y)
C
      DO 20 I=1,NN
      READ(5,21) WORD,N33,V3
      IF (WORD.EQ.STOP) GO TO 521
      N3S(I)=N33
      V3S(N3S(I))=V3
20    CONTINUE
      I=AA+1
521   N3T=I-1
C
C     READ BOUNDARY CONDITION 4
C
      DO 23 I=1,NN
      READ(5,24) WORD,N44,V4
      IF (WORD.EQ.STOP) GO TO 524
      N4S(I)=N44
      V4S(N4S(I))=V4
      IF (WORD.NE.SLF1) GO TO 25
      V4=1.5*(1.0-YC(N4S(I))**2)
      V4S(N4S(I))=V4
23    CONTINUE
      I=AA+1
524   N4T=I-1
C
C     READ BOUNDARY CONDITION 5
C
      DO 28 I=1,NN
      READ(5,29) WORD,N55,V5
      IF (WORD.EQ.STOP) GO TO 30
      N5S(I)=N55
      N5S(I)=N5S(I)
      I=AA+1
30    N5T=I-1
C
C     READ INITIAL CONDITIONS
C
      DO 32 I=1,NN
      READ(5,33) WORD,N66,V6
      IF (WORD.EQ.STOP) GO TO 34
      N6S(I)=N66
      V6=3.0*YC(N6S(I))
      VT(ICOUNT,N6S(I))=V6
32    CONTINUE
      I=AA+1
34    N6T=I-1
C
C     READ BOUNDARY CONDITION 7
C
      DO 39 I=1,NN
      READ(5,40) WORD,N77,V7
      IF (WORD.EQ.STOP) GO TO 41
      N7S(I)=N77
      V7S(N7S(I))=V7
39    CONTINUE
      I=AA+1
41    N7T=I-1
40    FORMAT(6X,A4,I10,F15.7)
C
      DO 45 I=1,NN
      READ(5,46) J,N8
      N8=1
      NCMC(I)=N8
45    CONTINUE
C
      DO 48 I=1,NN
      READ(5,53) WORD,N59,V9
      IF (WORD.EQ.STOP) GO TO 47
      DO 48 J=1,L
      ICCLAT=J
      N9S(I)=N9S
      II=AST+I
      II=1
      N9S(I)=N99
      VT(ICOUNT,N9S(I))=V9
      VREF(N9S(I))=VT(ICOUNT,N9S(I))
      IF (WORD.NE.V1FL) GO TO 48
      VT(ICOUNT,N9S(I))=3.0*YC(N9S(I))
      VREF(N9S(I))=VT(ICOUNT,N9S(I))
48    CONTINUE
      I=AA+1
47    N9T=I-1
      NSST=N9T+AST
      NSST=N9T
      JJ=0
      DO 7234 IJ=1,NN
      II=IJ
      I=1
      DO 7235 I=1,NSS
      IF (II.EQ.N9S(I)) GO TO 7234
```

```
7234  CONTINUE
      ASMT=JJ
      WRITE(8,7237) AA,ASST,ASMT
7237  FORMAT(' ',3I6)
C
      JCCLRT=1
      DO 7240 I=1,AA
      READ(5,33) WCRC,MPP,VPR
      IF (WORD.EQ.STCP) GO TO 7241
      MP(I)=MPP
      MPIMP(I)=VPR
7240  CONTINUE
7241  MPRT=I-1
      JJ=0
      DO 7246 IJ=1,AA
      IT=IJ
      I=1
      DO 7247 I=1,MPPT
      IF (II.EQ.MP(I)) GO TO 7246
7247  CONTINUE
      JJ=JJ+1
      NMA(JJ)=II
7246  CONTINUE
      NMMT=JJ
      WRITE(8,7237) AA,AMPT,MEMT
C
      DO 7242 I=1,AN
      READ(5,33) WCRC,RUU,VU
      IF (WORD.EQ.STCP) GO TO 7243
      NU(I)=NUU
      UU(NU(I))=VU
      NSLA(I)=NUU
7242  CONTINUE
7243  NULLT=I-1
      JJ=0
      DO 7947 IJ=1,AA
      IT=IJ
      I=1
      DO 7248 I=1,NULT
      IF (II.EQ.NU(I)) GO TO 7947
7947  CONTINUE
      JJ=JJ+1
      NMA(JJ)=II
7248  CONTINUE
      NUMT=JJ
      WRITE(8,7237) NN,NULLT,NUMT
C
      DO 7244 I=1,AN
      READ(5,33) WCRC,NVV,VV
      IF (WCRD.EQ.STCP) GO TO 7245
      NV(I)=NVV
      VVV(NV(I))=VV
7244  CONTINUE
7245  NVVT=I-1
      JJ=0
      DO 7249 IJ=1,AA
```

```
      DO 7250 I=1,AVV
      IF (II.EQ.NV(II)) GO TO 7249
7250  CONTINUE
      JJ=JJ+1
      NVA(JJ)=II
7249  CONTINUE
      NVMT=JJ
      WRITE(8,7237) AA,AVVT,AVMT
      DO 8000 I=1,AN
      READ(5,33) WCRC,IUA,UAC
      IF (WORD.EQ.STCP) GO TO 8001
      NUA(I)=IUA
8000  CONTINUE
8001  NSLAT=I-1
      NSUAR=NSUAT+NUIT
      JJ=0
      DO 8002 IJ=1,NN
      IT=IJ
      I=1
      DO 8003 I=1,NSUAR
      IF (II.EQ.NSLA(I)) GO TO 8002
8003  CONTINUE
      JJ=JJ+1
      NMUA(JJ)=II
8002  CONTINUE
      NMUAR=JJ
      WRITE(8,7237)NN,NSUAR,NMUAR
C
      DO 8004 I=1,NN
      READ(5,33) WCRC,IUB,UBC
      IF (WCPD.EQ.STCP) GO TO 8005
      NUB(I)=IUB
      II=NUUT+I
      NSUB(II)=IUB
      UUU(NUF(I))=UBC
8004  CONTINUE
8005  NSUBT=I-1
      NSUBR=NSUBT+NUUT
      JJ=0
      DO 8006 IJ=1,NN
      IT=IJ
      I=1
      DO 8007 I=1,NSUBR
      IF (II.EQ.NSLF(I)) GO TO 8006
8007  CONTINUE
      JJ=JJ+1
      NMLF(JJ)=II
8006  CONTINUE
      NMUBR=JJ
      WRITE(8,7237) NN,NSUBB,NMUBR
C
      READ(5,1051) TIME
      READ(5,1051) TUBF1
      READ(5,1055) TCPFF,TEST
      READ(5,1055) PL,KTC
1055  FORMAT(' ',F15.9,A4)
      READ(5,1051) EPSL
      READ(5,1051) FCST
```

```
      WRITE(8,1051) EPSL
      WRITE(6,1051) EPSL
      WRITE(8,1051) PCST
      WRITE(8,21) CREU,DEN
      DC 900 I=1,NN
      KKA(I)=1
900   CONTINUE
      CALL PLTXMX(60.0)
7238  SCAVE=10.0
      XF=20.0/5.0
      XF=C.5/2.54
      YF=XF
C
      CALL PLTOFS(0.0,XF,0.0,YF,1.0,0.0,0.5)
      CALL PSYMB(4.0,6.5,-0.15,'FINITE  ELEMENT   MESH
     1',0.0,23)
      DC 2070 I=1,NE
      CALL PENUPS(XC(NODE(I,1)),YC(NODE(I,1)))
      CALL PENONS(XC(NODE(I,2)),YC(NODE(I,2)))
      CALL PENONS(XC(NODE(I,3)),YC(NODE(I,3)))
      CALL PENONS(XC(NODE(I,1)),YC(NODE(I,1)))
2070  CONTINUE
C
      DO 8074 I=1,NN
8074  CALL PENMBR(XC(I),YC(I),-0.1,KKA(I),0.0,'I3*',1.0)
      CALL PLTEND
C
C
C
C     CYCLE FOR EACH ELEMENT AND FORM SYSTEM MATRICES
C
      DO 1000 K=1,NE
      N1=NODE(K,1)
      N2=NODE(K,2)
      N3=NODE(K,3)
C
      NE1(K)=NODE(K,1)
      NE2(K)=NODE(K,2)
      NE3(K)=NODE(K,3)
C
      XCL(1)=XC(NODE(K,1))
      XCL(2)=XC(NODE(K,2))
      XCL(3)=XC(NODE(K,3))
C
      XECL(K,1)=XCL(1)
      XECL(K,2)=XCL(2)
      XECL(K,3)=XCL(3)
C
      YCL(1)=YC(NODE(K,1))
      YCL(2)=YC(NODE(K,2))
      YCL(3)=YC(NODE(K,3))
C
      YECL(K,1)=YCL(1)
      YECL(K,2)=YCL(2)
      YECL(K,3)=YCL(3)
C
      4A(1)=XCL(2)*YCL(3)-XCL(3)*YCL(2)
```

```
C     FETRI  ALL LATE (PUT THE  FAO REST NEAD IN
C
      WRITE(6,50) NM,NE
      WRITE(6,52)
      DO 54 I=1,MN
      IF (TEST.NE.TESI) GC TO 53
      YC(I)=YC(I)*C.5
53    WRITE(6,55) I,XC(I),YC(I)
54    CONTINUE
C
      WRITE(6,56)
      DC 58 I=1,NE
      WRITE(6,60) I,NCDE(I,1),NCDE(I,2),NCDE(I,3)
58    CONTINUE
C
      WRITE(6,62) CNEU,DEN
      WRITE(6,64)
      DO 65 I=1,N1T
      SKEE(N1S(II))=VIS(N1S(II))
      WRITE(6,66) I,N1S(II),VIS(N1S(II))
65    CONTINUE
C
      WRITE(6,68)
      DO 69 I=1,N2T
      WRITE(6,79) I,N2S(II),V2S(N2S(II))
69    CONTINUE
C
      WRITE(6,78)
      DO 79 I=1,N4T
      WRITE(6,80) I,N4S(II),V4S(N4S(II))
84    CONTINUE
C
      WRITE(6,83)
      DC 84 I=1,N5T
      WRITE(6,85) I,N5S(II),VT(ICCUNT,N5S(II))
84    CONTINUE
C
      WRITE(6,82)
      DC 89 I=1,N6T
      WRITE(6,90) I,N6S(II),VT(ICCUNT,N6S(II))
89    CONTINUE
C
      WRITE(6,93)
      DO 94 I=1,N7T
      WRITE(6,85) I,N7S(II),V7S(N7S(II))
94    CONTINUE
C
      DO 97 I=1,N9T
      WRITE(6,96) I,N9S(II),VT(ICCUNT,N9S(II))
97    CONTINUE
      WRITE(6,98)
      DO 99 I=1,NN
      WRITE(8,100) I,NCMO(I)
99    CONTINUE
      WRITE(6,1052) TIME
      WRITE(6,1052) TIME
1052  FORMAT(' TIME  =',F20.15,5X,'SECCAD')
      WRITE(6,1553) TURF1,TURF2
```

```
C     FORM MATRIX S   (EQ.  3-52)
C
      ASST(ICOUNT,1)=(DEL/12.0)*(2.0*VT(ICOUNT,NODE(K,1))
     1 +VT(ICOUNT,NODE(K,2))+VT(ICOUNT,NODE(K,3)))
      ASST(ICOUNT,2)=(DEL/12.0)*(VT(ICOUNT,NODE(K,1))
     1 +2.0*VT(ICOUNT,NODE(K,2))+VT(ICOUNT,NODE(K,3)))
      ASST(ICOUNT,2)=(DEL/12.0)*(VT(ICOUNT,NODE(K,1))
     1 +VT(ICOUNT,NODE(K,2))+2.0*VT(ICOUNT,NODE(K,3)))
C
C     FORM MATRIX K    (EQ.  3-53)
C
      DO 125 I=1,3
      DO 125 J=1,3
      FKVTT(I,J)=DEL/12.0
      EKVTT(K,I,J)=EKVTT(I,J)
125   CONTINUE
C
C     ASSEMBLY SYSTEM EQUATIONS
C
      N(1)=N1
      N(2)=N2
      N(3)=N3
C
      NEE(K,1)=N1
      NEE(K,2)=N2
      NEE(K,3)=N3
C
      DO 130 IL=1,3
      I=N(IL)
      ASSTAT(I,)=ASSTAT(I)+ASST(ICOUNT,IL)
      DO 130 JL=1,3
      J=K(JL)
      AEKVTT(I,J)=AEKVTT(I,J)+EKVTT(IL,JL)
      AFKST(I,J)=AEKST(I,J)+FKST(IL,JL)
      AEKPT(I,J)=AEKPT(I,J)+EKPT(IL,JL)
      AEKVT(I,J)=AEKVT(I,J)+EKLT(IL,JL)
      AFKVT(I,J)=AEKVT(I,J)+EPVT(IL,JL)
130   CONTINUE
C
1000  CONTINUE
      CALL SOLMIX(FA,AEKVT,PIS,PS,...,KKA,AA,KKA,P,Z,C)
C
C     EQ.  3-35    TC BE SOLVED
C
      CALL SOLMIX(KK,AFKST,VIS,ASSTAT,HI,VIS,VET,HAS,A,Z,C)
C
2000  CONTINUE
      DO 3491 I=1,NA
C
3491  AWM(I,J)=J,0
C
      DO 3477 K=1,NE
      N(1)=NEE(K,1)
      N(2)=NEE(K,2)
      N(3)=NEE(K,3)
      DO 3478 IL=1,3
```

```
      ZAE(K,1)=AA(1)
      ZAE(K,2)=AA(2)
      ZAE(K,3)=AA(3)
C
      BB(1)=YCL(2)-YCL(3)
      BB(2)=YCL(3)-YCL(1)
      BB(3)=YCL(1)-YCL(2)
C
      BBE(K,1)=BB(1)
      BBE(K,2)=BB(2)
      BBE(K,3)=BB(3)
C
      CC(1)=XCL(3)-XCL(2)
      CC(2)=XCL(1)-XCL(3)
      CC(3)=XCL(2)-XCL(1)
C
      CCE(K,1)=CC(1)
      CCE(K,2)=CC(2)
      CCE(K,3)=CC(3)
C
      DEL=ABS(0.5*(XCL(1)*(YCL(2)-YCL(3))+
     1 XCL(2)*(YCL(3)-YCL(1))+
     2 XCL(3)*(YCL(1)-YCL(2))))
      DELE(K)=DEL
C
C     FORM MATRIX K    (EQ.  3-48)
C     FORM MATRIX K    (EQ.  3-42)
C     FORM MATRIX K    (EQ.  3-78)
C
      DO 101 I=1,3
      DO 102 J=1,3
      EKST(I,J)=(BB(I)*BB(J)+CC(I)*CC(J))/(4.0*DEL)
      EKLT(I,J)=BB(I)*BB(J)+CC(I)*CC(J))/(4.0*DEL)
      EKPT(I,J)=EKST(I,J)
      DO 8023 II=1,3
      JKK=NODE(K,II)
      DO 8024 IKK=1,NN
      KKK=ASS(IKK)
      KKK=ASS(IKK)
      IF (KKST.NE.KK) GO TO 8024
      EKPT(I,J)=EKST(I,J)+FKST(J)/(4.0*DEL)
      EKST(I,J)=CC(I)*CC(J)/(4.0*DEL)
8024  CONTINUE
8023  CONTINUE
      JKK=NODE(K,J)
      DO 8026 IKK=1,NA
      KKPT=NODE(K,KK)
      IF (IKPT.AG.JKK) GO TO 8026
      EKPT(I,J)=CC(I)*CC(J)/(4.0*DEL)
      EKST(I,J)=BB(I)*BB(J)/(4.0*DEL)
8026  CONTINUE
8027  CONTINUE
      EKVT(I,J)=CNEJ*EKST(I,J)
      EKSTE(K,I,J)=EKST(I,J)
      EKVTE(K,I,J)=EKVT(I,J)
```

```
J=N(IL)
    WN(1,1)=DELE(K)*2.0
    WN(1,2)=DELF(K)
    WN(1,3)=DULE(K)
    WN(2,1)=DELF(K)
    WN(2,2)=DELE(K)*2.0
    WN(2,3)=DELE(K)
    WN(3,1)=DULE(K)
    WN(3,2)=DELE(K)
    WN(3,3)=DELE(K)*2.0
3478 AWN(IL,JL)=AWN(IL,JL)+WN(IL,JL)/12.0
3477 CONTINUE
3492 VKT(N9S(JJ))=VKEEP(N9S(JJ))
C
C  INSERT THE NC-SLIP BOUNDARY CONDITION  (EQ.  4-1-4)
C
C     RC=ABS(XC(2)-XC(1))
      DO 3592 I=1,NST
3592 VT(ICOUNT,N9S(I))=VKEEP(N9S(I))
      DO 145 I=1,NST
     VT(ICOUNT,N9S(I))=(2.0/(RC**2))*(VAS(N9S(I))-
    1                    VIS(N9S(I)+1))
145 CONTINUE
    WRITE(8,1055)RC,ETC
C
    IF (ETC.NE.ETCB) GO TO 1283
    VT(ICOUNT,77)=(2.0/(0.079,2,2.2) +
   1 (VIS(77)-(VIS(78)+VIS(79))/2.0)
    VT(ICOUNT,49)=(2.0/(0.064*2))*(VIS(49)-VIS(42))
    VT(ICOUNT,71)=(2.0/(0.04*2))*(VIS(71)-VIS(12))
    VT(ICOUNT,72)=(2.0/(0.06*2))*(VIS(72)-VIS(72))
    VT(ICOUNT,54)=(2.0/(0.04*2))*(VIS(54)-VIS(55))
    VT(ICOUNT,50)=(2.0/(RC**2))*(VIS(50)-VIS(42))
    VT(ICOUNT,110)=(2.0/(RC**2))*(VIS(110)-VIS(112))
    DO 148 I=1,NST
    VT(ICOUNT,N9S(I))=VT(ICOUNT-1,N9S(I))+
   1(VT(ICOUNT,N9S(I))-VT(ICOUNT-1,N9S(I)))*TURF1
148 CONTINUE
1283 CONTINUE
C
    IF (ICOUNT.LE.2) GO TO 1285
    DO 1282 I=1,NA
    VT(ICOUNT,I)=VT(ICOUNT,I)*TURF2
1282 CONTINUE
1285 CONTINUE
C
    CALL SOLMIX(AN,AWN,VKT,ASSTAT,NSST,NSS,NSVT,NSN,8,Z,C)
    DO 3479 I=1,NA
C3479 VT(ICOUNT,I)=VKT(I)
C
    DO 4110 I=1,NST
4110 VT(ICOUNT,N9S(I))=VKEEP(N9S(I))
C
C   FORM MATRIX S  (EQ.  3-41 ) WITHOUT REGARDING BOUNDARY CONDITIONS
C
    DO 150 J=1,NE
```

```
    2   CCE(J,2)*VIS(NODE(J,3)))+
    3   (PRE(J,1)*VT(ICOUNT,NODE(J,1))+
    4   PRE(J,2)*VT(ICOUNT,NODE(J,2))+
    5   PRE(J,3)*VT(ICOUNT,NODE(J,3)))-
    6   (PRE(J,1)*VIS(NODE(J,1))+
    7   PRE(J,2)*VIS(NODE(J,2))+
    8   PRE(J,3)*VIS(NODE(J,3))+
    9   (CCE(J,1)*VIS(NODE(J,1))+
    1   CCE(J,2)*VT(ICOUNT,NODE(J,2))+
        CCE(J,3)*VT(ICOUNT,NODE(J,3)))))
150 CONTINUE
C
C   MODIFY MATRIX S  (EQ.  3-41 ) TO INCLUDE BOUNDARY CONDITIONS
C
    DO 165 J=1,NE
    DO 167 I=1,3
    KK=ACOE(J,I)
    DO 164 I2=1,2
    KKI2=N9S(I2)
    IF (KKI2.NE.KK) GO TO 164
    ESVT(J,I)=(1.0/(12.0*DELE(J))*(
   1   (CCE(J,1)*VIS(NODE(J,1))+
   2   CCE(J,2)*VIS(NODE(J,2))+
   3   CCE(J,3)*VIS(NODE(J,3)))+
   4   (PRE(J,1)*VT(ICOUNT,NODE(J,1))+
   5   PRE(J,2)*VT(ICOUNT,NODE(J,2))+
   6   PRE(J,3)*VT(ICOUNT,NODE(J,3)))*
        PRE(J,3)*VT(ICOUNT,NODE(J,3)))))
164 CONTINUE
167 CONTINUE
165 CONTINUE
C
    DO 175 J=1,NE
    DO 174 I=1,3
    KK=ACOE(J,I)
    DO 178 I4=1,NST
    KKI4=N9S(I4)
    IF (KKI4.NE.KK) GO TO 178
    ESVT(J,I)=1.0/(2.0)*(VAS(N9S(I4)))*(
   1   (PRE(J,1)*VT(ICOUNT,NODE(J,1))+
   2   PRE(J,2)*VT(ICOUNT,NODE(J,2))+
   3   PRE(J,3)*VT(ICOUNT,NODE(J,3)))+
   4   (1.0/(12.0*DELF(J))*(
   5   (PRE(J,1)*VIS(NODE(J,1))+
   6   PRE(J,2)*VIS(NODE(J,2))+
   7   PRE(J,3)*VIS(NODE(J,3)))+
   8   (CCE(J,1)*VT(ICOUNT,NODE(J,1))+
   9   CCE(J,2)*VT(ICOUNT,NODE(J,2))+
        CCE(J,3)*VT(ICOUNT,NODE(J,3)))))
178 CONTINUE
176 CONTINUE
175 CONTINUE
C
    DO 185 J=1,NE
    DO 186 I=1,3
    KK=ACOE(J,I)
    DO 187 I7=1,NST
    KKI7=N9S(I7)
```

```
215  CONTINUE
C
     CALL SOLMIX(NN,SM1,VTN,SM3,ASST,NSS,NSMT,NSV,R,Z,C)
C
C    ARE THE VALUES OF VORTICITIES SATISFACTORY ?
C
     DO 5220 I=1,NN
     DIFF(I)=ABS(VTN(I)-VT(ICCUNT,I))
     DIFF(I)=ABS((VTN(I)-VT(ICCUNT,I))/VT(ICCUNT,I))
     IF (DIFF(I).GT.EPSL) GO TO 222
5220 CONTINUE
     DO 5225 I=1,NN
5225 VT(ICCUNT,I)=VTN(I)
     GO TO 270
222  CONTINUE
C
     ICCUNT=ICCUNT+1
8025 FORMAT(' NCOUNT =',I3)
C
     DO 226 I=1,NN
226  ASST(I)=0.0
C
     DO 235 K=1,NE
     ASST(ICCUNT,1)=((DELE(K)/12.0)*2.0*VTN(NODE(K,1))
    1 +VTN(NODE(K,2))+VTN(NODE(K,3)))
     ASST(ICCUNT,2)=(DELE(K)/12.0)*(VTN(NODE(K,1))
    1 +2.0*VTN(NODE(K,2))+VTN(NODE(K,3)))
     ASST(ICCUNT,3)=(DELE(K)/12.0)*(VTN(NODE(K,1))
    1 +VTN(NODE(K,2))+2.0*VTN(NODE(K,3)))
     DO 236 IL=1,3
     I=NEE(K,IL)
     ASST(I)=ASST(I)+ASST(ICCUNT,IL)
236  CONTINUE
235  CONTINUE
C
     IF (ICCUNT.GT.100) GO TO 265
C
     CALL SOLMIX(NN,AFKST,VIS,ASSTAT,NIT,MIS,NBT,NCS,R,Z,C)
C
     DO 262 I=1,NN
262  VT(ICCUNT,I)=VTN(I)
C
     DO 267 I=1,NST
     VT(ICCUNT,NCS(I))=VKFEP(NGS(I))
267  WRITE(8,266)ICCUNT
     NIC=ICCUNT+3
     GO TO 2500
265  WRITE(8,264) ICCUNT
     ICCUNT=ICCUNT-1
270  CONTINUE
     DO 8031 I=1,NN
8031 VPS(I)=VIS(I)
C
C    CALCULATE THE PRESSURE DISTRIBUTION
C
     DO 300 K=1,NL
     DELESQ=DELE(K)**2
     APEK1=CEE(K,1)
```

```
     2    PRE(J,2)=VT(ICCUNT,NODE(J,2))+
     3    PRE(J,3)=VT(ICCUNT,NODE(J,3)))
187  CONTINUE
186  CONTINUE
185  CONTINUE
C
     IF (N2T.EQ.0) GO TO 9111
     IF (N4T.EQ.0) GO TO 9111
     DO 9112 I=1,N2T
     DO 9114 J=1,N4T
     IF (N2S(I).NE.N4S(J)) GO TO 9114
     DO 9112 J1=1,NE
     DO 9112 I2=1,3
     KK=NODE(J1,I1)
     IF (N2S(I).NE.KK) GO TO 9115
     ESVT(J1,I1)=(1.0/6.0)*(V4S(N4S(J))*
    1    (PRE(J1,1)*VT(ICCUNT,NODE(J1,1))+
    2     PRE(J1,2)*VT(ICCUNT,NODE(J1,2))+
    3     PRE(J1,3)*VT(ICCUNT,NODE(J1,3)))
9115 CONTINUE
9114 CONTINUE
9112 CONTINUE
9111 CONTINUE
C
     DO 188 I=1,NN
168  AES(I)=0.0
     DO 190 J=1,NN
     DO 191 IL=1,3
     I=NEE(K,IL)
     AESVT(I)=AESVT(I)+ESVT(K,IL)
191  CONTINUE
190  CONTINUE
C
     CUETI=TIME*ICCUNT
     WRITE(8,1053) CUETI
1053 FORMAT(' ****************************************',
    1    /' TIME   =',F10.2,5X,'SECOND'.
    2 ///,' ****************************************')
C
     DO 195 I=1,NN
     DO 195 J=1,NN
     SM1(I,J)=AEKVT(I,J)+AEKVTT(I,J)/TIME
195  CONTINUE
C
     DO 200 I=1,NN
     KKA(I)=I
     VTN(I)=VT(ICCUNT,I)
200  CONTINUE
C
     CALL SOLMIX(NN,AEKVTT,VTN,SM2,NN,KKA,C,KKN,R,Z,C)
C
     DO 205 I=1,NN
     SM3(I)=SM2(I)/TIME-AESVT(I)
205  CONTINUE
C
C EQ.  3-43   IS READY TO BE SOLVED
```

```
      CCEK1=CCE(K,1)
      CCEK2=CCE(K,2)
      CCEK3=CCE(K,3)
C
      VISK1=VIS(NCCE(K,1))
      VISK2=VIS(NCCE(K,2))
      VISK3=VIS(NCCE(K,3))
C
      VISXX(1)=(BPEK1/(2.0*DELESQ)*
     1 (BPEK1-BPEK2-BPEK3)*VISK1
     2 +(BPEK2/(2.0*DELESQ))*
     3 (BPEK2-BPEK1-BPEK3)*VISK2
     4 +(BPEK3/(2.0*DELESQ))*
     5 (BPEK3-BPEK1-BPEK2)*VISK3
     6 +(2.0*BPEK1*BPEK2/(DELESQ))*(VISK1+VISK2-VISK3)
     7 +(2.0*BPEK1*BPEK3/(DELESQ)*BPEK2/(DELESQ))*
     1 ((VISK1+VISK2/2.0)
     2 +(2.0*BPEK2*BPEK3/(DELESQ))*
     4 ((VISK2+VISK3)/2.0)
     5 +(2.0*BPEK3*BPEK1/2.0))
C
      VISYY(1)=(CCEK1/(2.0*DELESQ)*
     1 (CCEK1-CCEK2-CCEK3)*VISK1
     3 +(CCEK2/(2.0*DELESQ))*
     4 (CCEK2-CCEK1-CCEK3)*VISK2
     6 +(CCEK3/(2.0*DELESQ))*
     7 (CCEK3-CCEK1-CCEK2)*VISK3
     9 +(2.0*CCEK1*CCEK2/(DELESQ))*
     1 ((VISK1+VISK2)/2.0)
     2 +(2.0*CCEK1*CCEK3/(DELESQ))*
     4 ((VISK1+VISK3)/2.0)
     5 +(2.0*CCEK2*CCEK3/(DELESQ))*
     7 ((VISK2+VISK3)/2.0))
C
      VISXY(1)=((2.0*CCEK1*BPEK1-CCEK1*BPEK3-
     1 CCEK3*BPEK1-CCEK1*BPEK2-
     2 CCEK2*BPEK1)/(4.0*DELESQ))*VISK1
     4 +((2.0*CCEK2*BPEK2-CCEK2*BPEK3-
     5 CCEK3*BPEK2-CCEK1*BPEK2-
     6 CCEK2*BPEK1)/(4.0*DELESQ)*VISK2-
     8 +((2.0*CCEK3*BPEK3-CCEK3*BPEK2-
     9 CCEK1*BPEK3-CCEK3*BPEK1-
     1 CCEK2*BPEK3)/(4.0*DELESQ))*VISK3
     2 +(((CCEK1+BPEK2+CCEK2*BBEK1)/
     3 (DELESQ))*((VISK1+VISK3)/2.0)
     5 +(((CCEK1*BPEK2+CCEK2*BPEK1)/
     7 +VISK2)/2.0)+(((CCEK2*BPEK3+CCEK3
     8 *BPEK2)/(DELESQ))*((VISK2+VISK3)/2.0)
C
      CCC(1)=-2.0*(VISXX(1)*VISYY(1)-VISXY(1)**2)
      ESP=CEN*DELE(K)*CCC(1)/3.0
C
      DO 320 IL=1,3
      I=NEE(K,IL)
      AESPP(I)=AESPP(I)+ESP
  320 CONTINUE
  300 CONTINUE
```

```
      CALL SLCMA.....
C
C     CALCULATE FLOW VELOCITIES
C
      DO 8064 I=1,NN
      IF(-(YC(I+1)-YC(I)).LT.0.0) GO TO 8060
      IF(-(YC(I+1)-YC(I)).EQ.0.0) GO TO 8060
      U1(I)=(VIS(I+1)-VIS(I))/(YC(I+1)-YC(I))
      WRITE(7,8064) I,U1(I)
 8064 FORMAT(' I =',I3,10X,'VEL(U) =',F10.3)
 8060 CONTINUE
      AAA=AN-IK
      DO 8062 I=1,AAA
      IF(+(XC(IK+1)-XC(I)).GG.0.01 GO TO 8062
      V2I(I)=(VIS(IK+2)-VIS(I))/(-(XC(IK+1)-XC(I)))
      WRITE(7,8067) I,V2I(I)
 8067 FORMAT(' I =',I3,10X,'VEL(V) =',F10.3)
 8062 CONTINUE
C
      DO 335 I=1,AA
      DO 335 J=1,NA
      ACU(I,J)=0.0
      AGV(I,J)=0.0
      APU(I,J)=0.0
  335 CONTINUE
C
      DO 333 K=1,NE
      DO 334 J=1,3
      DO 334 I=1,3
      AU(J,I)=DELE(K)/12.0
  334 CONTINUE
C
      DO 338 IL=1,3
      I=NEE(K,IL)
      DO 338 JL=1,3
      J=NEE(K,JL)
      APU(I,J)=APU(I,J)+PU(IL,JL)
  338 CONTINUE
  333 CONTINUE
C
      DO 8030 I=1,NST
      VPS(NSS(I))=FIS(NSS(I))
 8030 CONTINUE
      CALL VELSTS(EEL,CCC,GU,GV,NCCE,AGU,AGV,NE,NCNG,VIS,SMS,SM6,
     1 AA,VISI)
      CALL SCLMIX(NA,APU,UIL,SM5,PU(T,NU,PURT,MIN,P,7,C)
      CALL SCLMIX(NA,APU,VVV,SM6,AVVT,AV,AVVT,NVV,F,4,C)
      ILOOP=1
      DO 360 I=1,NA
      PRI(I)=FCST-((UUU(I))**2+(VVV(I))**2)*DEN*0.5
  360 CONTINUE
      DO 1061 I=1,AA
```

```
      WRITE(8,1063) IC,PI,IC,VCT,I,LCU(I),VV(I)
1061  CONTINUE
1062  FORMAT(' IC CUNT =',I2,5X,'VORTICITY =',F10.5,5X,
     1        'STREAM FUNCTION =',F10.5,2X,'PFI =',F10.5)
1063  FORMAT(' I =',I3,5X,'PRESSURE =',F10.5,5X,'VEL(U) =',F10.5,
     1   15X,'VEL(V) =',F10.5)
C
      REY=1.C*2.0/CNEL
      WRITE(7,8015) REY
8015  FORMAT(' REYNOLDS NUMBER =',F10.5)
C
      DO 1071 I=1,NN
1071  WRITE(7,1C72) ICCUNT,I,VIS(I)
1072  FORMAT(' ICCUNT =',I3,7X,'I =',I3,4X,'PFI =',F10.5)
      DO 1673 I=1,NN
1673  WRITE(7,1674) I,VT(ICCUNT,I),PR(I)
1674  FORMAT(' I =',I3,4X,'VORTICITY =',F20.12,7X,'PRESSURE =',F20.12)
      DO 1676 I=1,NN
1676  WRITE(8,1677) I,LLC(I),VV(I)
1677  FORMAT(' I =',I3,'VFL(U) =',F20.12,5X,'VEL(V) =',F20.12)
C
C
1     FORMAT(2I10)
8     FORMAT(I10,2F16.8)
10    FORMAT(4I10)
11    FORMAT(2F15.5)
14    FORMAT(6X,A4,I10,F15.7)
18    FORMAT(5X,A4,IC,F15.7)
21    FORMAT(6X,A4,I10,F15.7)
24    FORMAT(6X,A4,I10,F15.7)
20    FORMAT(6X,A4,I10,F15.7)
25    FORMAT(6X,A4,I10,F15.7)
46    FORMAT(2I10)
50    FORMAT(5X,'NO. OF NODES =',I5,//)
     1     5X,'NC. OF ELEMENTS =',I2,//)
52    FORMAT(7X,'SUMMARY OF NODAL COORDINATES',//,
     1     7X,'I',10X,'X(I)',20X,'Y(I)',//)
55    FORMAT(5X,I3,2(7X,F16.8))
56    FORMAT(5X,'LISTING OF SYSTEM TOPOLOGY',//,5X,
     1     'ELEMENT NUMBER',23X,'NODE NUMBERS',//)
60    FORMAT(5X,I3,24X,3(5X,I3))
62    FORMAT(5X,'KINEMATIC VISCOSITY =',F10.5,//,
     1     5X,'DENSITY =',F10.5)
69    FORMAT(5X,'I =',I5,10X,'M1S =',I5,10X,'V1S =',F15.7)
70    FORMAT(5X,'I =',I5,10X,'N2S =',I5,10X,'V2S =',F15.7)
75    FORMAT(5X,'I =',I5,10X,'N3S =',I5,10X,'V3S =',F15.7)
80    FORMAT(5X,'I =',I5,10X,'N4S =',I5,10X,'V4S =',F15.7)
85    FORMAT(5X,'I =',I5,10X,'N5S =',I5,10X,'VT =',F15.7)
90    FORMAT(5X,'I =',I5,10X,'N6S =',I5,10X,'VT =',F15.7)
86    FORMAT(1H0)
63    FORMAT(1H0)
72    FORMAT(1H0)
64    FORMAT(1H0)
93    FORMAT(1H0)
95    FORMAT(5X,'I =',I5,10X,'N7S =',I5,10X,'V7S =',F15.7)
96    FORMAT(5X,'I =',I5,10X,'N8S =',I5,10X,'VT =',F15.7)
```

```
135   FORMAT(1H0)
138   FORMAT(5X,'KN =',I3,'RE =',I3,'KIT =',I3,'RET =',I3)
139   FORMAT(5X,'I =',I3,'J =',I2,'AEKST(I,J) =',F20.5)
140   FORMAT(5X,'I =',I3,'J =',I2,'VIS =',F20.5,'ASSTA =',F20.5,
     1     5X,'KIS =',I3,'HDS =',I3,'Z =',F20.5,'IC =',F20.5)
     2     5X,'ICCLNT =',I3)
147   FORMAT(5X,'PE =',F10.8)
146   FORMAT(5X,'VT =',F20.5,'I =',I3,'VSS =',F20.5)
152   FORMAT(5X,'J =',I2,'I =',I3,'FSVT =',F20.5)
156   FORMAT(5X,'J =',I2,'I =',I3,'AESVT =',F20.5)
160   FORMAT(5X,'J =',I2,'I =',I3,'AESVT =',F20.5)
173   FORMAT(5X,'I =',I3,'J =',I2,'FSVT =',F20.5)
570   FORMAT(5X,'J =',I2,'NAS =',I3)
133   FORMAT(5X,'I =',I2,'ACSVT =',F20.5)
193   FORMAT(5X,'I =',I2,'FCVT =',F20.5)
198   FORMAT(5X,'I =',I2,'J =',I2,'FFKT =',F20.5)
      5X,'I/KVTT =',F20.5,'SM1 =',F20.5,'SM2 =',F20.5)
201   FORMAT(5X,'I =',I3,'KKN =',I2,'VTAL =',F20.5)
206   FORMAT(5X,'I =',I2,'FSVT =',F20.5,'SM2 =',F20.5)
     1     5X,'SA2 =',F20.5)
208   FORMAT(5X,'I =',I2,'VTH1 =',F20.5,'SV2 =',F20.5)
     1     5X,'KKA =',I3,'Z =',F20.5,'C =',F20.5)
210   FORMAT(5X,'P =',F20.5,'AFKVTT =',F20.5)
     1     5X,'J =',F20.5)
217   FORMAT(5X,'I =',I3,'VTN =',F20.5)
209   FORMAT(5X,'I =',I3,'J =',I2,'VTN =',F20.5)
220   FORMAT(5X,'I =',I3,'J =',I2,'VTN =',F20.5)
238   FORMAT(5X,'I =',ICCUNT,I3,'ASSTUJ =',F20.5)
246   FORMAT(5X,'VIS =',F20.5,'ASSTUJ =',F20.5)
     1     5X,'KS =',I2,'J =',I3,'Z =',F20.5)
249   FORMAT(5X,'I =',I2,'I =',I3,'ARKVT =',F20.5)
256   FORMAT(5X,'KKA =',I3,'AQV =',F20.5)
272   FORMAT(' ',I2,'I =',I3,'CLNT =',I3)
322   FORMAT(5X,'K =',I3,'I =',I3,'FSSIR =',F20.5)
327   FORMAT(5X,'I =',I3,'J =',I2,'SPR =',F20.5)
328   FORMAT(5X,'I =',I3,'DF =',F2.12)
336   FORMAT(5X,'I =',I2,'AQU =',F20.5)
341   FORMAT(5X,'I =',I3,'ARKVT =',F20.5)
     1     5X,'ACU =',F20.5,'AQV =',F20.5)
346   FORMAT(5X,'I =',I2,'VIS =',F20.5,'VT5 =',F20.5)
347   FORMAT(5X,'KKA =',I3,'J =',I2,'FSS =',F20.5)
     1     5X,'H =',F20.5)
351   FORMAT(5X,'I =',I3,'VIS =',F20.5,'SVC =',F20.5)
     1     5X,'KKA =',I3,'Z =',F20.5,'C =',F20.5)
353   FORMAT(5X,'I =',I3,'AQV =',F20.5)
361   FORMAT(5X,'I =',I3,'SMS =',F20.5,'GUU =',I3)
     1     5X,'SME =',F20.5)
362   FORMAT(5X,'I =',I3,'J =',I2,'SMS =',F20.5,'V =',F20.5)
363   FORMAT(5X,'I =',I3,'J =',I2,'ARU =',F20.5)
C
      J1=1
      J2=1
      J3=1
      J4=1
      J5=1
```

```fortran
      SUBROUTINE SOLMIX(N,A,X,Y,NX,LISTX,NY,LISTY,B,Z,C)
      DIMENSION A(N,N),X(N),Y(N),LISTX(N),LISTY(N)
      DIMENSION B(NY,NY),Z(NY),C(NY)
C
      IF (N.EQ.(NX+NY)) GO TO 100
      WRITE(6,111) N,NX,NY
  111 FORMAT(10X,'N = ',I2,10X,'NX = ',I2,10X,'NY = ',I2)
      STOP
  100 CONTINUE
C
      IF (NX.NE.0) GO TO 120
      CALL SOLVE(N,A,Y,X)
      RETURN
  120 CONTINUE
C
      IF (NY.NE.0) GO TO 140
      DO 130 I=1,N
      Y(I)=0.0
      DO 130 J=1,N
  130 Y(I)=Y(I)+A(I,J)*X(J)
      RETURN
  140 CONTINUE
C
      DO 200 I=1,NY
      DO 200 J=1,NY
  200 B(I,J)=A(LISTY(I),LISTY(J))
      DO 205 I=1,NY
      C(I)=Y(LISTY(I))
      DO 300 J=1,NX
  300 C(I)=C(I)-A(LISTY(I),LISTX(J))*X(LISTX(J))
C
      CALL SOLVE(NY,B,C,Z)
      DO 400 I=1,NY
  400 X(LISTY(I))=Z(I)
C
      DO 800 I=1,NX
      C(I)=0.0
      DO 600 J=1,NY
  600 C(I)=C(I)+A(LISTX(I),LISTY(J))*X(LISTY(J))
      DO 700 J=1,NX
  700 C(I)=C(I)+A(LISTX(I),LISTX(J))*X(LISTX(J))
  800 Y(LISTX(I))=C(I)
      RETURN
      END
```

```fortran
      DO 8041 I=1,NX
      IF (YC(I).EQ.YC(1)) GO TO 8042
      IF (YC(I).EQ.YC(2)) GO TO 8043
      IF (YC(I).EQ.YC(3)) GO TO 8044
      IF (YC(I).EQ.YC(4)) GO TO 8045
      IF (YC(I).EQ.YC(5)) GO TO 8046
      IF (YC(I).EQ.YC(6)) GO TO 8047
      IF (YC(I).EQ.YC(7)) GO TO 8048
      GO TO 8041
 8042 SLPL(1,J1)=VIS(I)
      J1=J1+1
      GO TO 8041
 8043 SLPL(2,J2)=VIS(I)
      J2=J2+1
      GO TO 8041
 8044 SLPL(3,J3)=VIS(I)
      J3=J3+1
      GO TO 8041
 8045 SLPL(4,J4)=VIS(I)
      J4=J4+1
      GO TO 8041
 8046 SLPL(5,J5)=VIS(I)
      J5=J5+1
      GO TO 8041
 8047 SLPL(6,J6)=VIS(I)
      J6=J6+1
      GO TO 8041
 8048 SLPL(7,J7)=VIS(I)
      J7=J7+1
 8041 CONTINUE
      DO 8050 IJ=1,7
      WRITE(11,8052) (SLPL(IJ,II),II=1,28)
 8052 FORMAT( 2(E15.8,2X) )
 8050 CONTINUE
 8040 CONTINUE
      RETURN
      END
```

```fortran
C     LINEAR EQUATIONS ...
C
      SUBROUTINE SOLVE(N,A,C,X)
      DIMENSION A(N,N),X(N),C(N)
      DO 400 K=1,N
      BIG=ABS(A(K,K))
      IBIG=K
      DO 100 I=K,N
      SIZE=ABS(A(I,K))
      IF (SIZE.LT.BIG) GO TO 100
      BIG=SIZE
      IBIG=I
100   CONTINUE
C
      IF (K.EQ.IBIG) GO TO 280
      DO 200 J=K,N
      BIG=A(IBIG,J)
      A(IBIG,J)=A(K,J)
200   A(K,J)=BIG
      CBIG=C(IBIG)
      C(IBIG)=C(K)
      C(K)=CBIG
280   CONTINUE
      IF (A(K,K).EQ.0.0) GO TO 600
      DO 400 I=1,N
      IF (I.EQ.K) GO TO 400
      RATIO=A(I,K)/A(K,K)
      DO 300 J=K,N
300   A(I,J)=A(I,J)-RATIO*A(K,J)
      C(I)=C(I)-RATIO*C(K)
400   CONTINUE
C
      DO 500 K=1,N
500   X(K)=C(K)/A(K,K)
      RETURN
600   WRITE (6,666)
666   FORMAT(10X,'SINGULAR MATRIX')
      DO 700 I=1,N
700   X(I)=0.0
      RETURN
      END
```

```fortran
C
      SUBROUTINE MATMLT(A,U,T,M,N,P,NL)
      INTEGER P
      DIMENSION A(3,6),U(6,1),T(3,1)
C
      DO 1 I=1,M
      DO 1 J=1,P
1     T(I,J)=0.0
C
      DO 2 I=1,M
      DO 2 J=1,P
      DO 2 K=1,N
      T(I,J)=A(I,K)*U(K,J)+T(I,J)
C
2     CONTINUE
      RETURN
      END
C
```

```
      SUBROUTINE VELST3(PBE,CCE,QU,CV,NCDE,AQU,ACV,NE,NCMO,VIS,SM5,SM6
     1  ,NN,VISI)
C
      COMMON N,NL,L,IL
      REAL*4 NCNO
C
      DIMENSION PBE(NL,IL),CCE(NL,IL)
      DIMENSION QU(3,6),CV(3,6),VISI(NL,6),VISI(M)
      DIMENSION N(3),EQU(3,1),EQV(3,1)
      DIMENSION NCDE(NL,IL),SM5(M),SM6(M)
      DIMENSION AQU(M,M),EEQV(3),EVISI(6,1),ACV(M,M)
C
      DO 15 I=1,NN
      SM5(I)=0.0
   15 SM6(I)=0.0
C
      DO 1000 K=1,NE
C
      QU(1,1)=2.0*(CCE(K,1)-CCE(K,2)-CCE(K,3))
      QU(1,2)=-(CCE(K,1)-CCE(K,2)+CCE(K,3))
      QU(1,3)=+QU(1,2)
      QU(1,4)=4.0*(2.0*CCE(K,3)+CCE(K,1))
      QU(1,5)=4.0*(2.0*CCE(K,2)-CCE(K,1))
      QU(1,6)=4.0*(CCE(K,2)+CCE(K,2))
      QU(2,1)=0U(1,2)
      QU(2,2)=2.0*(CCE(K,2)-CCE(K,3)-CCE(K,1))
      QU(2,3)=0U(1,2)
      QU(2,4)=4.0*(CCE(K,1)+CCE(K,3))
      QU(2,5)=4.0*(CCE(K,2)+2.0*CCE(K,1))
      QU(2,6)=4.0*(2.0*CCE(K,2)+CCE(K,2))
      QU(3,1)=0U(1,2)
      QU(3,2)=0U(1,2)
      QU(3,3)=2.0*(CCE(K,3)-CCE(K,2)-CCE(K,1))
      QU(3,4)=4.0*(CCE(K,3)+2.0*CCE(K,1))
      QU(3,5)=4.0*(CCE(K,3)+CCE(K,1))
      QU(3,6)=4.0*(CCE(K,3)+2.0*CCE(K,2))
C
      CV(1,1)=2.0*(BBE(K,1)-BBE(K,2)-BBE(K,3))
      CV(1,2)=-(BBE(K,1)+BBE(K,2)-BBE(K,3))
      CV(1,3)=QV(1,2)
      CV(1,4)=4.0*(2.0*BBE(K,3)+BBE(K,1))
      CV(1,5)=4.0*(2.0*BBE(K,2)-BBE(K,1))
      CV(1,6)=4.0*(BBE(K,2)+BBE(K,2))
      QV(2,1)=QV(1,2)
      QV(2,2)=2.0*(BBE(K,2)-BBE(K,3)-BBE(K,1))
      QV(2,3)=QV(1,2)
      QV(2,4)=4.0*(BBE(K,1)+BBE(K,3))
      QV(2,5)=4.0*(BBE(K,2)+2.0*BBE(K,1))
      QV(2,6)=4.0*(2.0*BBE(K,2)+BBE(K,2))
      QV(3,1)=QV(1,2)
      QV(3,2)=QV(1,2)
      QV(3,3)=2.0*(BBE(K,3)-BBE(K,2)-BBE(K,1))
      QV(3,4)=4.0*(BBE(K,3)+2.0*BBE(K,1))
      QV(3,5)=4.0*(BBE(K,3)+BBE(K,1))
      QV(3,6)=4.0*(BBE(K,3)+2.0*BBE(K,2))
C
      DO 23 I=1,3
C
   33 CONTINUE
      VISI(K,1)=VISI(NODE(K,1))/NCRC(NODE(K,1))
      VISI(K,2)=VIS(NODE(K,2))/NCMO(NODE(K,2))
      VISI(K,3)=VIS(NODE(K,3))/NCAF(NODE(K,3))
      VISI(K,4)=(VIS(NODE(K,1))/NCRC(NCCE(K,1))
     1      +VISI(NODE(K,3))/NCM)(NODE(K,3)))/2.0
      VISI(K,5)=(VIS(NODE(K,1))/NCRC(NODE(K,1))
     1      +VIS(NODE(K,2))/NCMO(NODE(K,2)))/2.0
      VISI(K,6)=(VIS(NODE(K,2))/NCMO(NODE(K,2))
     1      +VISI(NODE(K,3))/NCMO(NODE(K,3)))/2.0
      EVISI(1,1)=VISI(K,1)
      EVISI(2,1)=VISI(K,2)
      EVISI(3,1)=VISI(K,3)
      EVISI(4,1)=VISI(K,4)
      EVISI(5,1)=VISI(K,5)
      EVISI(6,1)=VISI(K,6)
C
      CALL MATMLT(QU,EVISI,EQU,3,6,1,NL)
      CALL MATMLT(CV,EVISI,EQV,3,6,1,NL)
C
      EEQU(1)=EQU(1,1)
      EEQU(2)=EQU(2,1)
      EEQU(3)=EQU(3,1)
      EEQV(1)=EQV(1,1)
      EEQV(2)=EQV(2,1)
      EEQV(3)=EQV(3,1)
C
      N1=NCDE(K,1)
      N2=NCDE(K,2)
      N3=NCDE(K,3)
C
      N(1)=N1
      N(2)=N2
      N(3)=N3
C
      DO 10 IL=1,3
      I=N(IL)
      SM5(I)=SM5(I)+EEQU(IL)
      SM6(I)=SM6(I)+EEQV(IL)
      DO 10 JL=1,1
      J=N(JL)
      AQU(I,J)=AQU(I,J)+EQU(IL,JL)
      AQV(I,J)=AQV(I,J)+EGV(IL,JL)
C
   10 CONTINUE
 1000 CONTINUE
      DO 66 I=1,NN
   66 SM5(I)=-SM6(I)
C
      RETURN
      END
```

```
C
      CALL IGMA(0.1,-0.5)
      CALL IGTXT('STREAM LINES<E>')
      CALL IGTRAN('AMPE','WINDOW',0.0,5.0,0.0,5.0,0.0,5.0,'MOVE',0.0,1.0)
      CALL IGDRUM('CALCCMP')
C
      STOP
      END
```

APPENDIX C

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C   THIS PROGRAM CAN ONLY BE USED FOR PRODUCING
C   CONTOUR LINES OF DATA SURFACE DEFINED ON
C   RECTANGULAR GRID.
C
C   CONTOUR ARE PLOTTED AS A SERIES OF STRAIGHT LINES
C   OVER EACH ELEMENT CONSISTENT WITH THIS NODE DATA.
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      EXTERNAL IGMA, IGDA
      LOGICAL SWCHES(5) / 4* .TRUE. , .FALSE. /
      REAL X(23), Y(7), Z(28,7), CZ(32)
C
      DATA X / 0.0, 0.1, 0.2, 0.325, 0.4625,
     &0.6, 0.8, 1.0, 1.2, 1.4, 1.5,
     &1.8, 2.0, 2.2, 2.4, 2.6, 2.8, 3.0,
     &3.2, 3.4, 3.6, 3.8, 4.0,
     &4.2, 4.4, 4.6, 4.8, 5.0 /
C
      DATA Y /0.0, 0.2, 0.4, 0.6, 0.8, 0.9, 1.0 /
C
      DATA CZ / 0.1, 0.3, 0.5, 0.7, 3.8, 0.85, 0.9, 0.95,
     $0.94,0.93,0.92,0.91,0.89,0.88,0.87,0.86,0.54,0.83,0.82,0.81,
     &0.78,0.76,0.74,0.72,0.67,0.63,0.6,0.55,0.45,0.4,0.25,0.2 /
C
      NC = 32
      IDX = 28
      IY = 7
      IX = 28
C
      DO 90 JC=1,IY
      J=IY-JC
      READ(4,70) ( Z(I,J), I = 1, IX )
70    FORMAT( 2( E15.8  2X ) )
      WRITE(6,80) ( Z(I,J), I = 1,IX )
80    FORMAT( 4( 2X, E9.2 ) )
90    CONTINUE
C
      CALL IGINIT
      CALL IGONSET( IGMA, IGDA, 6 )
C
C   IGONSET IS CALLED TO SET UP THE NAMES OF THE FUNCTIONS
C   TO DISPOSE OF PICTURE (COORDINATES,AND TO PASS IN THE
C   LOGICAL I/O UNIT UPON WHICH ERROR COMMENTS (IF ANY)
C   WILL BE WRITTEN.
C
      CALL CONTUR ( X, IX, Y, IY, Z, IDX, CZ, NC, SWCHES )
C
C   CONTUR PRODUCES COORDINATE PAIRS FOR DRAWING
C   A PICTURE WHICH IS CONTOUR LINES OF
C   THE DATA IN THE ARRAY Z.
C
      DO 110 J=1,6
      XJ=J-1
      CALL IGMA(XJ,0.0)
```

1	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
2	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
3	0.9947	0.9949	0.9953	0.9961	0.9977	1.0	1.0	1.0	1.0	1.0
4	1.0	1.0	1.0	1.0	1.0	0.9977	0.9961	0.9953	0.9949	0.9947
5	0.9416	0.9442	0.9486	0.9566	0.7703	0.9867	0.9982	1.0	1.0	1.0
6	1.0	1.0	1.0	0.9992	0.9868	0.9704	0.9566	0.9486	0.9442	0.9416
7	0.8385	0.845	0.855	0.8728	0.903	0.9406	0.9704	0.996	1.0	1.0
8	1.0	1.0	0.996	0.9706	0.9408	0.9031	0.8729	0.8551	0.845	0.8385
9	0.6945	0.7041	0.7185	0.7437	0.7864	0.8421	0.8919	0.9448	0.9902	1.0
10	1.0	0.9902	0.9451	0.8923	0.8425	0.7867	0.7439	0.7186	0.7042	0.6945
11	0.5188	0.5289	0.5441	0.5704	0.6151	0.6750	0.7314	0.7973	0.8667	0.913
12	0.9082	0.8663	0.7977	0.732	0.6755	0.6155	0.5706	0.5442	0.53	0.5188
13	0.296	0.3033	0.3143	0.3327	0.3643	0.4067	0.4462	0.4915	0.5365	0.5674
14	0.5696	0.5376	0.4923	0.4467	0.4071	0.3646	0.3329	0.3142	0.3034	0.296
15	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
16	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

END OF FILE
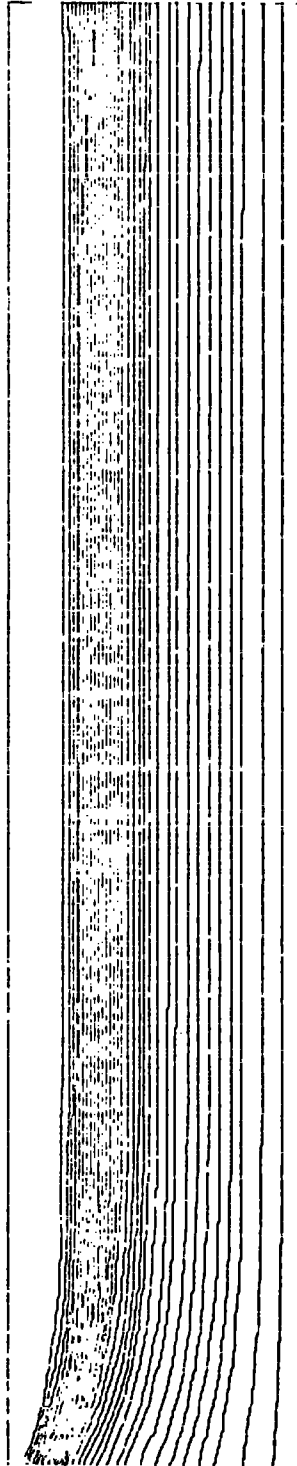
TURF1 = 1.0   Re = 1

FIG. D - 1

Streamline Contours in Flow between Parallel Plates.
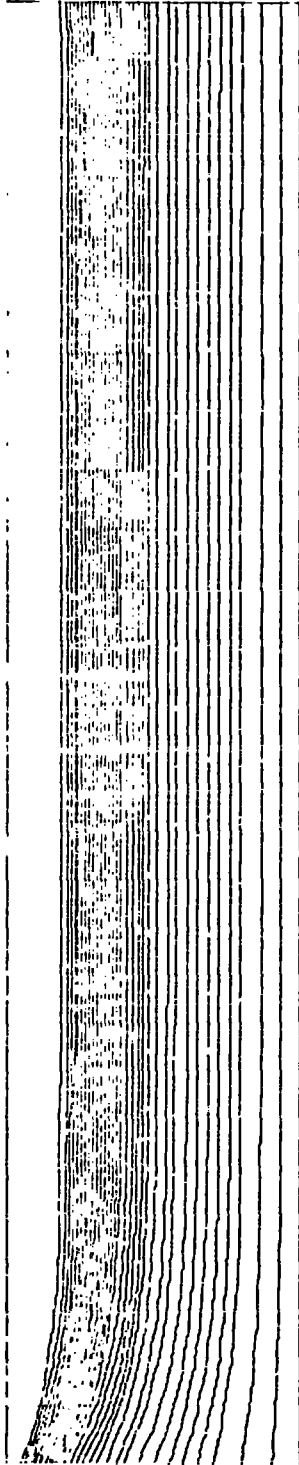
TURF1 = 0.1   Re = 1.0

FIG. D - 2

Streamline Contours in Flow between Parallel Plates
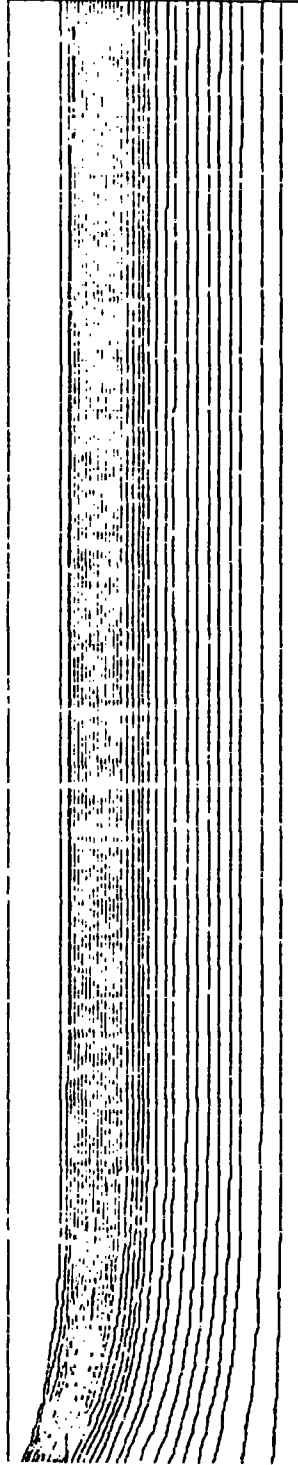
TURF1 = 1.0   Re = 2.0

FIG. D - 3

Streamline Contours in Flow between Parallel Plates
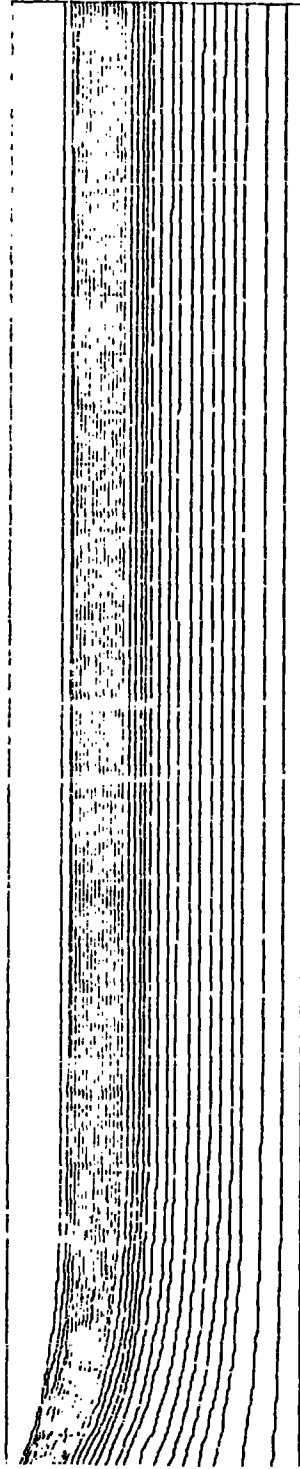
TURFl = 0.1   Re = 2.0

FIG. D - 4

Streamline Contours in Flow between Parallel Plates
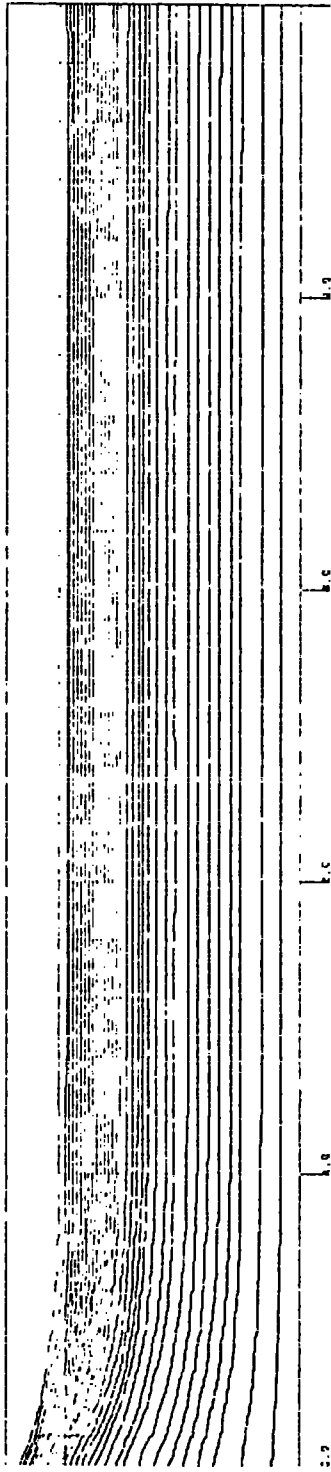
TURF1 = 1.0   Re = 3.0

FIG. D - 5

Streamline Contours in Flow between Parallel Plates
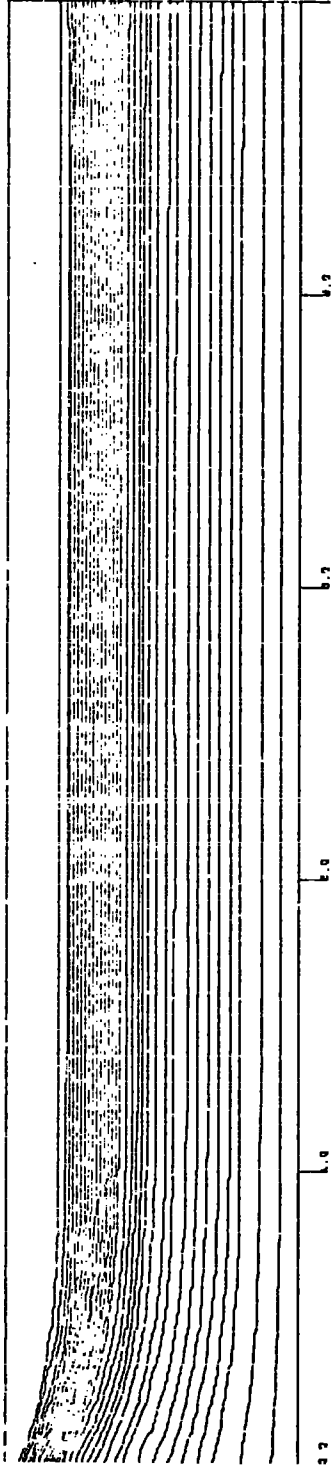
STREAM LINES

TURF1 = 0.1   Re = 3.0

FIG.   D - 6

Streamline Contours in Flow between Parallel Plates
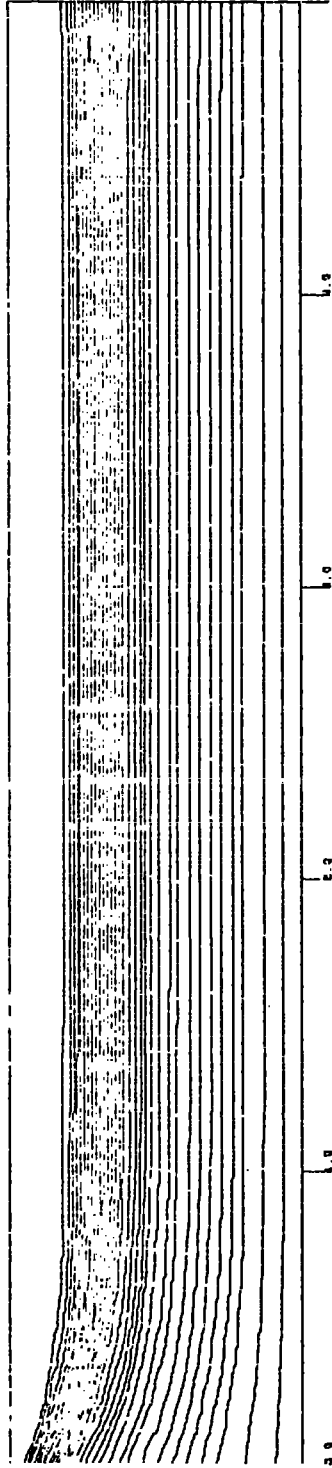
STREAM LINES

TURFl = 1.0   Re = 4.0

FIG. D - 7

Streamline Contours in Flow between Parallel Plates
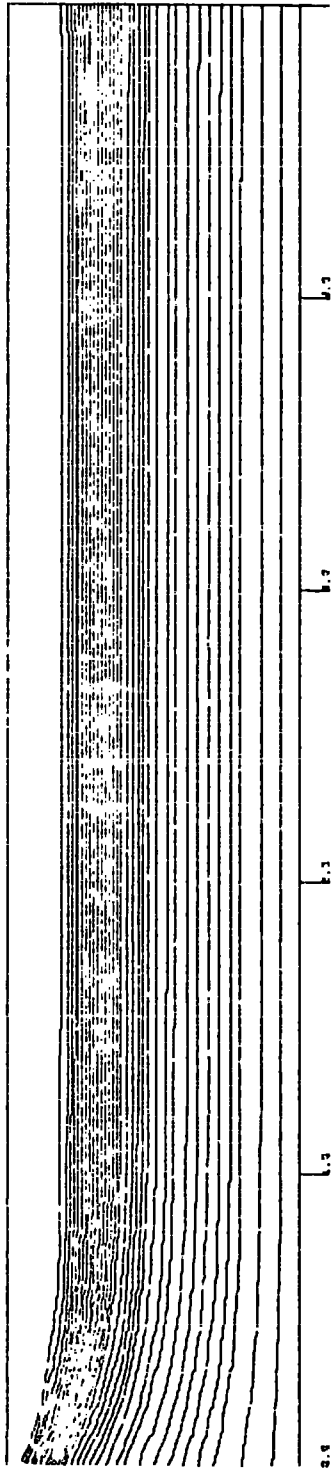
STREAM LINES

TURF1 = 0.1   Re = 4.0

FIG. D - 8

Streamline Contours in Flow between Parallel Plates
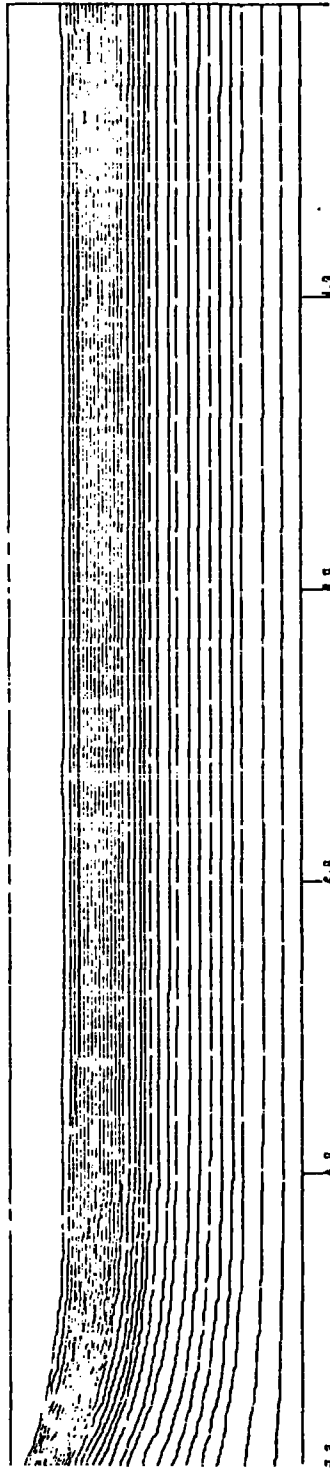
TURF1 = 1.0   Re = 5

FIG. D - 9
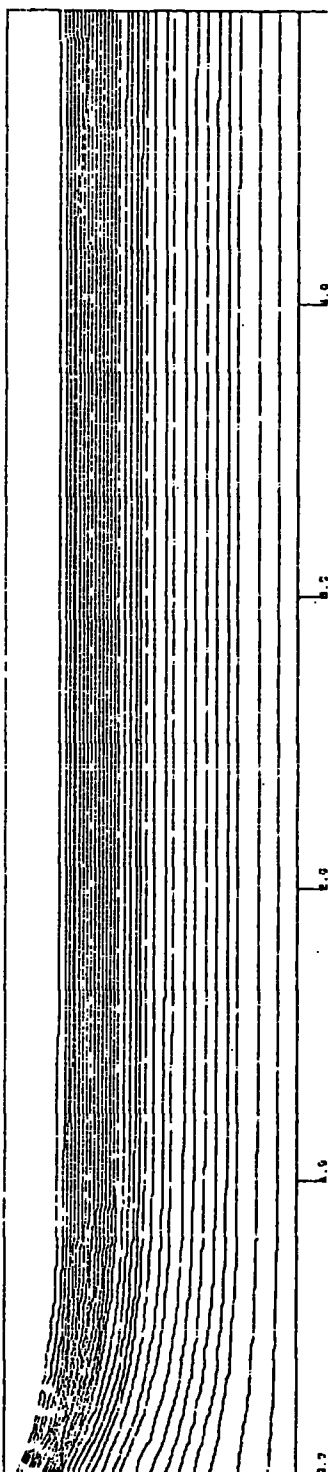
Streamline Contours in Flow between Parallel Plates

TURFL = 0.1   Re = 5

FIG. D - 10

Streamline Contours in Flow between Parallel Plates

Appendix E   Re = 200

Steamline Contours

APPENDIX F

VTFI
VTFI
VTFI
VTFI
VTFI
VTFI
VTFI
VTFI
VTFI
VTFI
VTFI
STCP

STCP
VLLF
VELF
VELF
VELF
VELF
VELF
VELF
VLLF
VELF
VLLF
VELF
VELF

RTC1

STCP
STCP
STOP

0.1
1.0
0.06
0.0001
3.0

0.1

1.0
0.0

VELF
VELF
STOP

APPENDIX G

Nodal points	Streamfunction	Vorticity
1	1.	2.934
2	0.995	2.820
3	0.942	2.387
4	0.839	1.953
5	0.695	1.519
6	0.519	1.085
7	0.296	0.560
8	0.0	0.0
9	1.0	2.857
10	0.994	2.679
17	1.0	2.633
18	0.995	2.467
25	1.0	2.177
26	0.996	2.071
33	1.0	1.271
34	0.998	1.368
42	0.999	0.588
49	1.0	0.007
72	1.0	20.785
77	1.0	29.091
78	0.913	14.613
79	0.908	14.665
84	1.0	21.323
111	1.0	0.005
113	0.999	0.585
120	1.0	1.267
121	0.998	1.364
128	1.0	2.173
129	0.996	2.068
136	1.0	2.631
137	0.995	2.465
144	1.0	2.855
145	0.994	2.678
152	1.0	2.934
153	0.995	2.820

Nodal points	Streamfunction	Vorticity
154	0.942	2.387
155	0.839	1.953
156	0.695	1.519
157	0.519	1.085
158	0.296	0.600
159	0.0	0.0

Nodal points	Streamfunction	Vorticity

```
C     TARGES OF CONTOUR LINE VALUES SOMEWHERE NEAR
C     THE CONTOUR LINE ITSELF.
C
C     CALL CONTUR (X,IX,Y,IY,Z,IDX,CZ, NC, SWCHES, &9910, &9920, &9930 )
C
C     CALL CONTUR TO DO THE PLOT.
C
      CALL IGIXTI('STREAM LINES<F>')
      CALL IGTRAN('4MP*','WINDOW',0.0,2.0,0.0,3.0,'MOVE',0.0,1.0
     &  'SCALE',1.0)
      CALL IGORCN('CALCOMP')
      CALL CCMEAC( NREGEN, NALLOC, NUSED )
C
      STOP
 9810 PAUSE ' CCNLBL RETURN 1'
 9820 PAUSE ' CCNLPL RETURN 2'
 9910 PAUSE ' CONTUR RETURN 1'
 9920 PAUSE ' CCNTLP RETURN 2'
 9930 PAUSE ' CONTUR RETURN 3'
      END
```

APPENDIX H

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C     THIS PROGRAM CAN ONLY BE USED FOR PRODUCING
C     CONTOUR LINES OF DATA SURFACE DEFINED ON
C     RECTANGULAR GRID.
C
C     CONTOUR ARE PLOTTED AS A SERIES OF STRAIGHT LINES
C     OVER EACH ELEMENT CONSISTENT WITH THIS NOCA DATA.
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      EXTERNAL IGRA, IGDA
      EXTERNAL CLABEL
      LOGICAL SWCHES(5) / 4* .TRUE., .TRUE. /
C
      REAL X(20), Y(8), Z(20, 8), CZ(26)
      REAL YLABS(11,XLABS(2)
      REAL XC(15),YC(7)
C
      DATA X / 0.0, 0.2, 0.4, 0.6, 0.8, 0.972,
     &  1.0923, 1.2122, 1.3232, 1.4539, 1.5461, 1.6669,
     &  1.7372, 1.9077, 2.028, 2.2, 2.4, 2.6, 2.8, 3.0 /
C
      DATA Y / 0.0, 0.2, 0.3616, 0.5063, 0.65C9, 0.7955, 0.94, 1.0 /
C
      DATA CZ / 0.1, 0.3, 0.5, 0.7, 0.8,0.85, 0.9,0.95,0.05,
     &  0.2,0.25,0.37,0.54,0.94,0.93,C.92,C.91,
     &  C.90,0.88,0.87,C.86,0.83,0.84,0.82,0.81,
     &  C.76,C.76,C.74,C.72,C.67,C.64,0.63,0.6,0.58,
     &  C.55,0.4/
C
      DATA XLABS/ 1.0, 2.0 /
C
      NXLABS = 1
      NYLABS=0
      DIST=0.0
      NC = 36
      IDX = 20
      IY = 8
      IX = 20
C
      DO 90 JC=1,IY
      J=IY+1-JC
      READ(4,70) ( Z(I,J), I = 1, IX )
 70   FORMAT( 10(F4.2,2X) )
      WRITE(8,70) ( Z(I,J), I = 1, IX )
 90   CONTINUE
C
      CALL IGINIT
      CALL CCASETI IGRA, IGDA, 6 )
C
C     CCNSET IS CALLED TO SET UP THE NAMES OF THE FUNCTIONS
C     TO DISPOSE OF PICTURE COORDINATES, AND TO PASS IN THE
C     LOGICAL I/O UNIT UPON WHICH ERROR COMMENTS (IF ANY)
C     WILL BE WRITTEN.
C
      CALL CCNLBL(CLABEL,DIST,NXLABS,XLABS,NYLABS,YLABS, &9810, &9820
```

RETURN
END

APPENDIX I

STOP

STGP

STGP

STGP