LIFT ON A SPHERE IN SHEAR FLOW NEAR FLAT CHANNEL BED

by

Ker-Jen Ying

Dissertation submitted to the Faculty of the

Virginia Polytechnic Institute and State University

in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

in

Civil Engineering

APPROVED:

Dr. C. Y. Kuo, Chairman

Dr. P. Diplas, Co-Chairman

Dr. D. P. Telionis

Dr. S. A. Ragab

Dr. J. M. Wiggert

September, 1991

Blacksburg, Virginia

LIFT ON A SPHERE IN SHEAR FLOW NEAR FLAT CHANNEL BED

by

Ker-Jen Ying

Committee Chairman: C. Y. Kuo

Civil Engineering

(Abstract)

The lift and drag forces exerting on a sphere immersed in a shear flow above a flat channel bed are evaluated by solving the steady three-dimensional Navier-Stokes equations. The numerical technique which combines the Newton iteration method and the finite element method is used to solve the non-linear Navier-Stokes equations. The technique first linearizes the non-linear terms in the partial differential equations, then solves the linearized equations by the finite element method. The Newton iteration method is used to linearize the non-linear equations. Since the iteration method requires a good initial guess, the linear solution of the partial differential equations is used for the initial guess, where the linear solution is the obtained by solving the differential equations without non-linear terms. The computer model developed can evaluate the lift coefficients of a sphere stationed at various distance from the channel bed. The computational results agree very well with the experimental measurements cited in the literature. The lift coefficient of the sphere changes with the undisturbed approaching velocity profile as well as the

gap ratio which is the ratio of the distance between the sphere and the channel bed and the diameter of sphere. For fixed gap ratios, higher Reynolds number gives smaller lift coefficient than that of the lower Reynolds number. On the other hand, the lift coefficient also changes with the diameter of sphere for each fixed gap ratio. For small gap ratios, the lift coefficient increases as the diameter of sphere increases. For large gap ratios, the lift coefficient increases in the negative (downward) direction as the diameter of sphere increases.

Acknowledgements

The idea for this work was first suggested by Dr. C. Y. Kuo, my major advisor. Subsequently, the author received assistance from Dr. S. Ragab for the development of the methodology. Dr. D. Telionis offered a really good piece of advice: "If you want to do it, you have to do a good job." which helped me in overcoming many difficulties. Dr. P. Diplas and Dr. J. Wiggert always show their enthusiasms to help.

Building a computer code for a three-dimensional flow problem is really an eternal job. The author has to constantly deal with not only the bugs in the programs, but also the computer facilities during the past two years. As a serious computer user, it has been the author's dream to posses a super computer with huge size of memory, lightening speed and, the best of all, unlimited free CPU time. In the past, it was not unusual for the author to receive the message "out of memory" in the output after several sleepless nights. Sometimes, the program was cut off in the middle when the computer time limit for batch job was exceeded. Searching for the computer fund for the mainframe computer is always difficult in the Department of Civil Engineering. The author thanks Dr. R. Krebs for the infusion of fund in order to complete the work. During the early stage, the programs were running on the Apollo workstation for debugging. The author owes to Dr. G. Loganathan who authorizes me for exclusive use of this computer.

Other important individuals who supported my graduate study are my wife — Li-Cheng, my Daughter — Sharon, and both of our parents. Without them, this work would be impossible.

Table of Content

List of Tables

List of Illustrations

List of Symbols

A = aspect area of sphere (L^2)

ΔA_x = projection of the area of an element on the x-plane (L^2)

ΔA_y = projection of the area of an element on the y-plane (L^2)

ΔA_z = projection of the area of an element on the z-plane (L^2)

b = shortest distance between the channel bed and the surface of sphere (L)

C_L = lift coefficient

C_D = drag coefficient

D = drag force (MLT^{-2})

D_s = diameter of sphere (L)

J = Jocobian matrix

L = lift force (MLT^{-2})

$L(\underset{\sim}{x})$ = mapping from an arbitrary element to the reference element

p = pressure ($ML^{-1}T^{-2}$)

r, s, t = natural coordinate within the reference element

Re = Reynolds number

u = x-component of velocity (LT^{-1})

u^n = unknown x-component of velocity for the current iteration (LT^{-1})

u^{n-1} = x-component of velocity obtained from the preceding iteration (LT^{-1})

u_* = shear velocity (LT^{-1})

U_x = x-component of the free stream velocity (LT^{-1})

U_y = y-component of the free stream velocity (LT^{-1})

U_z = z-component of the free stream velocity (LT^{-1})

U_∞ = mean velocity of the free stream flow (LT^{-1})

v = y-component of velocity (LT^{-1})

v^n = unknown y-component of velocity for the current iteration (LT^{-1})

v^{n-1} = y-component of velocity obtained from the preceding iteration (LT^{-1})

w = z-component of velocity (LT^{-1})

w^n = unknown z-component of velocity for the current iteration (LT^{-1})

w^{n-1} = z-component of velocity obtained from the preceding iteration (LT^{-1})

x = longitudinal distance along the centerline of channel (L)

$\underset{\sim}{x}$ = coordinate vector for any point within an element (L)

$\underset{\sim}{x}_m$ = coordinate vector of the i-th nodal point within an element (L)

y = transverse distance in the horizontal plane from the center of sphere(L)

z = vertical distance from the center of sphere (L)

α_i = coefficient for the i-th term in the basis function ϕ_i

β_i = coefficient for the i-th term in the basis function ψ_i

Γ = boundary of the element

δ' = thickness of the wall boundary layer (L)

δ_{ij} = Kronecker delta

ν = kinematic viscosity of fluid (L^2T^{-1})

ϕ_i = basis function for pressure at the i-th node within the element

ρ = density of fluid (ML^{-3})

σ_x = normal stress in x-direction ($ML^{-1}T^{-2}$)

σ_y = normal stress in y-direction ($ML^{-1}T^{-2}$)

σ_z = normal stress z-direction ($ML^{-1}T^{-2}$)

τ_0 = shear stress ($ML^{-1}T^{-2}$)

τ_{xy}, τ_{yz}, τ_{zx} = shear stresses ($ML^{-1}T^{-2}$)

Ψ_m = basis function for pressure at the m-th node within the element

Ω = region enclosed by an element

Chapter I
Introduction

1.1 Background

The study of the behavior of a spherical particle in a flow near a stationary or moving boundary has found many uses in engineering problems. One immediate application is to the problem of incipient motion of particles related to the sediment transport. Other applications can be found in the biomechanics for the movement of blood cell, in the aerodynamics for particle movement in the combustion processes, and in chemical engineering for particulate system.

The application in the sediment transport is related to the settling and dislodging processes of a particle in a channel or a pipe flow. The settling process plays an important role in the deposition of sediment in the natural channel. The deposition of sediment usually changes the bed formation in the river. In some severe cases, it blocks the flow and causes flooding. The damages caused by this type of sedimentation problems are extensive in many areas. It claims thousand of lives and washes off numerous

properties each year. The dislodging process entrains the particles which are rested on the channel bed. Typical phenomena observed are the river bank erosion and scouring at the outlet around hydraulic structures. Many bridges have failed due to the scour of channel bed around the bridge piers. Other type of damage is caused by the combination of the erosion and settling of sediment. The commonly seen example is the sediment deposited in reservoirs, navigable waterways, and harbors which must be removed in order to maintain required storage capacity or depths. Fundamental knowledge of the sediment erosion, transport, and deposition is needed to understand these physical processes and to formulate feasible control measures.

By a close examination, it is found that the above stated problems are mainly governed by the entrainment or incipient motion of sediment (Sutherland, 1966; ASCE Task Committee, 1975). Most of the past research related to the threshold movement of sediment fall into two categories: the critical velocity concept and the critical shear stress concept. The critical velocity concept has long been of significant importance in the design of earth channels. Regarding to the critical force concept, lift force on a particle has received less attention than drag force, despite the fact that studies (Jeffreys, 1929; Einstein and El-Samni, 1949; Coleman, 1967) have proved its significant role in the dislodging process. Furthermore, very few literatures have dealt with the forces acting on a saltating particle immediately after the dislodging. It is necessary to fully understand the development of forces during the entire dislodging process in order to study the incipient motion of the sediment particles. In this study, attempts will

be made to numerically determine the lift force exerting on a sediment particle in a shear flow near the wall. Relationship will be established between the velocity profiles and the lift/drag forces. The lift forces to be determined numerically will be compared with various experimental results reported in the literature. Cases to be investigated include a sphere near a flat channel bed at various distances from the boundary. No rotation or movement of the sphere will be considered in the present study. The details of the assumptions, governing equations, and boundary conditions for the problem will be discussed in Chapter 2.

1.2 Literature Review

Over the past three decades, there have been moderate amount of studies for the wall effect on the lift force experienced by spheres immersed in a fluid flow. Most of them are done by the experimental method. Only limited number of analytical or numerical studies have been carried out. Among these are Saffman (1965), Coleman (1967, 1979), Benedict and Christensen (1972), Kral and McLaughlin (1986), Fornburg (1988), Kim and Pearlstein(1990), and Dandy and Dwyer (1990). Saffman (1965) applied the solution technique of perturbation to momentum equations and solved for the velocity and pressure distributions by matching the inner expansion to the outer expansion (Van Dyke, 1975) to the first order terms. The force on a particle was found by integrating the pressure over the surface of the particle. The lift was taken to be the force component which was normal to

the free streamlines. The results were considered to be valid for spheres in pure shear flow at particle Reynolds number much less than unity. Coleman (1967) expressed lift as the product of a lift coefficient and the immersed particle weight. By using the balance of moments among drag, lift and immersed particle weight for the initial motion, the lift coefficient was expressed in terms of the particle Reynolds number and was determined experimentally. The lift coefficient for a sphere resting on a horizontal channel bed which was composed of uniform non-cohesive particles was found to be negative, i.e., towards the bed, for particle Reynolds number less than 100. Benedict and Christensen (1972) used the potential flow theory to find the analytical solution for the lift of a particle resting on an idealized horizontal bed which was composed of a series of uniform hemispheres. The velocities over the surface of hemispheres were calculated and used to find the pressures by means of the Bernoulli's equation. These vertical components of pressures were integrated over the surface area of the sphere to obtain a total vertical force due to the hydrodynamic lift. Kral and Mclaughlin (1986) computed the lift force by means of 2-D Navier-Stokes equations for steady flow around a horizontal circular cylinder located inside the wall region of a boundary layer flow. A finite-difference solution technique was used for a 2-D grid system with approaching ambient flow field being assumed to have a linear velocity distribution. The lift force was determined by integrating the vertical components of normal and shear stresses on the surface of the cylinder. The Reynolds number based on particle diameter and shear velocity $u_* = \sqrt{\tau_o/\rho}$ was 8 in their computation. Fornburg (1988) solved the problem

of steady uniform flow past a sphere at high Reynolds number by a variable transformation technique. In the formulation, the governing equations were written in cylindrical form and a conformal mapping was used to map the computational region onto a complex plane. The governing equations were then solved by a centered second-order finite difference scheme. The results were obtained for the case of Reynolds number up to 5000 where the Reynolds number is defined in terms of the diameter of sphere. Kim and Pearlstein (1990) used a spectral technique to solve the problem of instability of flow past a sphere not near the wall. Both the base flow and the linear disturbance were solved by the same spectral method. Their results showed a value of 175 for the critical Reynolds number at which the flow loses its stability and becomes unsteady. This information is useful to the present work in terms of the formulation of the problem. Dandy and Dwyer (1990) solved the problem of a sphere in a linear shear flow by an integral method. Again, the sphere is not near the wall. The governing equations, i.e., the Navier-Stokes equations were discretized by the finite volume formulation procedure for the solution approximation. A relation was established for the effect of shear flow on the lift and drag forces for Reynolds numbers between 0.1 and 100. Again, the Reynolds number is defined in terms of the diameter of sphere.

Saffman's results were for a particle in shear flow, but not located in the vicinity of the wall. Coleman's study only showed the case of a fixed sphere in contact with stream bed, but not detached with a small distance away from the bed. The detached case represents the particle suspension immediately after the incipient motion. There was no vertical velocity

gradient considered in the Benedict and Christensen's computation. The shear flow with velocity gradient near the stream bed has a significant influence on the particle lift force. The work of Kral and McLaughlin was for 2-D cylinder, not for the 3-D sphere which resembles most of the sediment particles. The boundary layer flow field around the sphere and near the wall region is 3-D in nature and more complex as compared to the 2-D flow around the 2-D cylinder. Fornberg's study assumed that the flow is uniform in the computation domain, which does not apply to the flow near a stationary boundary. Dandy and Dwyer (1990) assumed a linear shear flow for the approaching flow, which does not represent the flow pattern near a boundary unless only a very small region within the boundary layer is considered.

There have been several publications pertaining to the experimental studies on the lift force for a sphere which is fixed either in contact with, or at a small distance away from a solid boundary. The research results of Coleman (1967), Watters and Rao (1971), Aksoy (1973), and Davies and Samad (1978) have provided the lift on a sphere fixed in contact with a boundary. Thomschke (1971), Chen and Carstens (1973), Bagnold (1974), Willetts and Murray (1981), and Willetts and Naddeh (1986) measured the lift on a fixed sphere which was located at some distance from the boundary.

The experimental results of Coleman (1967) have shown that the lift force on a sphere resting on a bed composed of uniform non-cohesive spheres depends on the particle Reynolds number, and that the sign of the lift force

may be determined by the thickness of the upstream laminar sublayer thickness. The range of particle Reynolds number in his study was from 5 to 9,400. Positive lift forces were found for Re > 100. Watters and Rao (1971), however, discovered the lift on a sphere resting on a layer of closely packed uniform spheres to be negative for all particle Reynolds numbers between 15 and 60. Thomschke (1971) used a sphere fixed at different gaps from a channel bed to study the lift and drag forces under various particle Reynolds numbers ranging from 126,000 to 500,000. The experiment were carried out in two steps. First, the velocity profile was fixed and the gap ratio (gap distance to particle diameter) was allowed to change; second, the gap ratio was fixed and the velocity profile was allowed to vary. For each fixed velocity profile, he found that the maximum lift occurred at the gap ratio 0.1. For each fixed gap ratio, the maximum lift occurred at the velocity profile with particle Reynolds number around 380,000. All the lift forces were positive in his results. Aksoy (1973) measured the lift on a sphere at rest on a smooth boundary and found the lift to be positive for particle Reynolds number over the range of 2,700 to 6,600. Chen and Carstens (1973) investigated a sphere at a small distance away from a wall in an air flow and concluded with positive lift forces for the range of particle Reynolds number 30,000 to 50,000. However, no explicit form was given for the lift coefficient in terms of Reynolds number in their results. Davies and Samad (1978) tested a sphere mounted in contact with a smooth bed and found that the negative lift occurred for values of Reynolds numbers to be less than 5. Bagnold (1974) made an extensive investigation on the lift force on a sphere at various distance away from a channel bed. Two cases were

considered. First, the sphere was allowed to drift with the flow. Second, the sphere was fixed in position. In the first case, the sphere was either allowed to rotate or prevented from rotating. Bagnold defined the gap ratio to be the ratio of the nearest distance from the wall to the sphere surface to the sphere diameter. The first case was run for the gap ratio ranging from 0.25 to 1.2. The second case covered the gap ratio from zero to 3.25. His results showed that the lift forces were all positive for the first case. For the second case, positive lift was obtained for gap ratio of zero and negative values were obtained for gap ratio 2.6 and higher. The particle Reynolds number was about 12 for the second case. For the first case, particle Reynolds number ranged from 12 to 800. Willetts and Murray (1981) observed the lift on a sphere fixed near a channel bed at several values of gap ratio. They showed that a maximum positive lift occurred at gap ratio of zero, and it decreased rapidly to a small negative value as the gap ratio increased to 0.06. As the gap ratio increased to 0.1 there was an increase in lift followed by a decrease as the gap ratio increased up to 0.3. For gap ratio higher than 0.3, the lift only changed slightly. The negative values were reported for gap ratio greater than 0.25. Willetts and Naddeh (1986) made an experiment on the lift exerting on a sphere at particle Reynolds numbers between 43 and 250. They summarized that the maximum lift occurred when the sphere was in contact with the boundary, and monotonically decreased to a negative value at a gap ratio of 0.7. The lift coefficient reached its maximum negative value at a gap ratio of about 2 and decayed to zero at a gap ratio of about 4. No general conclusions can be drawn from the previous studies. Each investigator has examined a certain parameters

with a certain range of values. Therefore, it is the intent of this study to investigate the problem by numerical method to encompass all variables of interest.

Chapter II
Governing Equations

2.1 Introduction

The characteristics of the flow close to the vicinity of a sphere are extremely complicated. The presence of a sphere in the wall boundary layer zone induces another boundary layer surrounding the sphere, which interacts with the wall boundary layer and creates more complicated flow field in that zone. The streamlines of the wall boundary layer are forced to change by the blockage of the sphere in the flow field. Meanwhile, the boundary layer generated by the sphere thrusts into the wall boundary layer and influences the flow inside the wall boundary layer. These effects are directly related to the distance, or the gap, between the sphere and the wall. The flow in the region between the sphere and the wall is subjected to the effect of the gap ratio more than the characteristics of flow field in other regions. This is due to the integration of the two boundary layers in this region and the constraint of the wall and the sphere on the flow. If separation occurs in this region, the flow in this region will become even more complicated. The same phenomena will occur in the region at the

downstream side of the sphere. Application of boundary layer theory to the flow in this region is very difficult due to the presence of two boundary layers in one region. The selection of the order of magnitude in the boundary layer equation for one boundary layer will not necessary meet the condition in the other boundary layer. For example, the radial component of the velocity near the surface of the sphere and the vertical component of the velocity near the channel are assumed to be negligible within these boundary layers. It is not easy to decide which component to be neglected for the flow in the region between the sphere and channel bed. Another problem in the application of the boundary layer theory is the calculation of the outer flow of each boundary layer. Since part of the outer flow of the sphere boundary layer is in the wall boundary layer zone where the viscous effect dominates, the inviscid assumption can not be adapted. Therefore, the flow around the sphere can not be calculated by the two-layer (viscous-inviscid) theory. To avoid these difficulties, the alternative has been turned to the direct solution of the Navier-Stokes equations. The solvability of the Navier-Stokes equations has been made possible by the progress in mathematics and the development of the new computer technology and numerical techniques in recent years. The efforts made by the mathematicians in solving the non-linear partial differential equations have lead to several ways for solving the Navier-Stokes equations by numerical approaches. Among various numerical methods, the finite element methods has been proved to be an effective method for engineering problems in different fields. The invention of the new generation of high speed computers in recent years has significantly improved the feasibility of

the numerical solutions to the hitherto intractable non-linear problems including the Navier-Stokes problems. Hence, the numerical solutions of the Navier-Stokes equations by finite element methods will be used in this study.

2.2 General Assumptions

For a sphere in a shear flow near the wall, the upstream undisturbed flow is assumed to be a two-dimensional boundary layer flow. The velocity can be obtained from the velocity profile given in conjunction with experiments cited in the literature. The sphere is assumed to be completely immersed within the channel flow. The sphere is placed along the center line of a rectangular channel. The effect of the channel side walls and the secondary channel flow are considered to be insignificant. Therefore, the problem is idealized to be a two-dimensional flow passing a sphere near a channel bed. The flow can be assumed to be symmetric with respect to the vertical plane passing through the center line of the channel.

For the numerical computation in this study, the particle Reynolds number will cover a range between 5 and 150 over which most of the existing experimental data for lift with different gap ratios are available.

Only steady flow will be considered in this study. Experiment shows that the steady flow past a sphere experiences vortex shedding or development of

unsteadiness for 120 < Re < 300 (Monkewitz, 1988). The characteristic length used for Re is the diameter of sphere. Nakamura (1976) also reported that steady flow past a sphere can be found up to Re = 190. In the numerical study performed by Kim and Pearlstein (1990), the value of 175 was reported for the critical Re at which the instability of flow begins. It is decided to narrow the problem to the case of steady flow for the present work. The flow close to the vicinity of the sphere and the channel bed is assumed to be laminar (Chow, 1969; Dwyer, 1981). In natural channel with tranquil flow, the flow close to the channel bottom is practically laminar and the particles resting on the channel bed are essentially immersed within the boundary layer of the flow.

2.3 Formulations

As shown in Figure 2.1 (Appendix B), a sphere with radius R is fixed at a location with a distance b from the channel bed to the nearest point on the surface of sphere. A Cartesian coordinate system (x, y, z) is chosen to denote the point positions in the three dimensional space for this study. The origin of the coordinate system is chosen at the center of sphere. The sphere is assumed to be completely submerged in the flow. The focal point of the study is to examine the case where the sphere is totally or partially immersed in the channel boundary layer with a thickness of δ', i.e., $b < \delta'$. The translational motion or rotation of the sphere will not be modeled for

the present work. For moving or rotating particles, the kinematic effect must be taken into consideration, which is beyond the scope of this study. The governing equations for the problem are given by the following equations.

Continuity equation:

$$\frac{\partial u}{\partial x}+\frac{\partial v}{\partial y}+\frac{\partial w}{\partial z}=0 \tag{2.1}$$

Navier-Stokes equations:

$$u\frac{\partial u}{\partial x}+v\frac{\partial u}{\partial y}+w\frac{\partial u}{\partial z}=-\frac{1}{\rho}\frac{\partial p}{\partial x}+\nu\left(\frac{\partial^2 u}{\partial x^2}+\frac{\partial^2 u}{\partial y^2}+\frac{\partial^2 u}{\partial z^2}\right) \tag{2.2}$$

$$u\frac{\partial v}{\partial x}+v\frac{\partial v}{\partial y}+w\frac{\partial v}{\partial z}=-\frac{1}{\rho}\frac{\partial p}{\partial y}+\nu\left(\frac{\partial^2 v}{\partial x^2}+\frac{\partial^2 v}{\partial y^2}+\frac{\partial^2 v}{\partial z^2}\right) \tag{2.3}$$

$$u\frac{\partial w}{\partial x}+v\frac{\partial w}{\partial y}+w\frac{\partial w}{\partial z}=-\frac{1}{\rho}\frac{\partial p}{\partial z}+\nu\left(\frac{\partial^2 w}{\partial x^2}+\frac{\partial^2 w}{\partial y^2}+\frac{\partial^2 w}{\partial z^2}\right) \tag{2.4}$$

where u, v, w = velocity component in the x, y, and z direction, respectively;

ρ = density of fluid;

ν = kinematic viscosity of fluid.

The boundary conditions must be satisfied by Equations (2.1) - (2.4) are:

$u = 0$, $v = 0$, $w = 0$ at the surface of sphere, and channel bed and walls,

$u = U_x$, $v = U_y$, $w = U_z$ at locations far upstream and downstream from the sphere,

and

$p = 0$ at the free surface of flow.

U_x, U_y and U_z are the velocity components of the undisturbed stream flow in the channel. The following reference quantities are used to make Equations (2.1) - (2.4) non-dimensional:

U_∞ = velocity parameter = the mean velocity of the free stream flow in the channel,

ρU_∞^2 = pressure parameter,

and D_s = diameter of sphere = length parameter.

Denoting the dimensionless parameters by:

$$x^* = \frac{y}{D_s}, \quad y^* = \frac{x}{D_s}, \quad z^* = \frac{z}{D_s}, \quad u^* = \frac{u}{U_\infty}, \quad v^* = \frac{v}{U_\infty},$$

$$w^* = \frac{w}{U_\infty}, \quad p^* = \frac{p}{\rho U_\infty^2}, \quad Re = \frac{\rho U_\infty D_s}{\mu},$$

$\mu = \rho\nu$ = dynamic viscosity of fluid.

The governing equations (2.1) - (2.4) can be written in the dimensionless form as following.

Continuity equation:

$$\frac{\partial u^*}{\partial x^*} + \frac{\partial v^*}{\partial y^*} + \frac{\partial w^*}{\partial z^*} = 0 \tag{2.5}$$

Navier-Stokes equations:

$$u^* \frac{\partial u^*}{\partial x^*} + v^* \frac{\partial u^*}{\partial y} + w^* \frac{\partial u^*}{\partial z^*} = -\frac{\partial p^*}{\partial x^*} + \frac{1}{Re}\left(\frac{\partial^2 u^*}{\partial x^{*2}} + \frac{\partial^2 u^*}{\partial y^{*2}} + \frac{\partial^2 u^*}{\partial z^{*2}}\right) \quad (2.6)$$

$$u^* \frac{\partial v^*}{\partial x^*} + v^* \frac{\partial v^*}{\partial y} + w^* \frac{\partial v^*}{\partial z^*} = -\frac{\partial p^*}{\partial y^*} + \frac{1}{Re}\left(\frac{\partial^2 v^*}{\partial x^{*2}} + \frac{\partial^2 v^*}{\partial y^{*2}} + \frac{\partial^2 v^*}{\partial z^{*2}}\right) \quad (2.7)$$

$$u^* \frac{\partial w^*}{\partial x^*} + v^* \frac{\partial w^*}{\partial y} + w^* \frac{\partial w^*}{\partial z^*} = -\frac{\partial p^*}{\partial z^*} + \frac{1}{Re}\left(\frac{\partial^2 w^*}{\partial x^{*2}} + \frac{\partial^2 w^*}{\partial y^{*2}} + \frac{\partial^2 w^*}{\partial z^{*2}}\right) \quad (2.8)$$

The boundary conditions become:

$$\begin{aligned} &u^* = 0, \quad v^* = 0, \quad w^* = 0 && \text{at the surface of sphere;} \\ &u^* = 0, \quad v^* = 0, \quad w^* = 0 && \text{at the channel bed and walls;} \\ &u^* = U_x^*, \quad v^* = U_y^*, \quad w^* = U_z^* && \text{at far upstream and downstream.} \end{aligned} \quad (2.9)$$

The entire region of flow field are governed by Equations (2.5)-(2.9). It is not necessary to make distinctions between the boundary layers and the inviscid layers as encountered in the solution of boundary layer flow problems. Hence, there is only one system of equations to be solved. Even for the separation zone of the sphere, the same system of equations can be applied. The problems of velocity singularities near the separation line, which usually occur in the boundary layer computation, can be avoided by solving Equations (2.5) - (2.9) instead of the boundary layer equations. The procedure to solve these equations will be discussed in the later sections.

2.4 Solution Methods

Equations (2.5) - (2.8) are non-linear partial differential equations. The conventional solution methods of partial differential equations can not be applied to these equations. A solution technique consisting of an iteration procedure and the finite element method will be used to solve these equations. First part of this method, i.e., the iterative procedure, is for the the non-linear terms in equations. It will be discussed later in this section. The second part is a procedure of variational formulation for the system of equations. The derivation of the finite element formulation will be presented in the next chapter.

As mentioned earlier, the system of Equations (2.5) - (2.8) can not be solved directly due to the presence of the non-linear terms, i.e., the convective terms. The available solution techniques for these equations include the perturbation method, asymptotic method, integral method, penalty method, spectral method and iterative method. Each of these methods has been well developed and tested. The discussion and comparison of these methods can be found in Smith (1985), Cuvelier et al. (1986), Carrier and Pearson (1988), and Kevorkian (1990). The iterative methods have been widely accepted in recent years. These methods usually start with the linearization of the governing equations before finite element formulation or finite difference method is applied. These methods can be further divided into linear methods and non-linear methods. The linear iterative methods include the well known Gauss-Seidel method (Isaacson and Keller, 1966; Atkinson,

1978) which is also called the block iteration, the Uzawa method (Thomasset, 1981; Cuvelier et al., 1986; Girault and Raviart, 1986) also known as the simple gradient method, and the Picard iteration. Both of the Gauss-Seidel and Uzawa methods converge very slowly (Atkinson, 1978; Thomasset, 1981; Cuvelier et al., 1986; Girault and Raviart, 1986). The Picard iteration, also known as the successive substitution, is characterized by substituting the values obtained from the preceding iteration for the non-linear terms in equation for each iteration. An initial guess, usually obtained from the linear solution, is needed for the first iteration. Once the non-linear terms are replaced by some known values from the preceding iteration, the equation becomes a linear equation and can be solved easily. The Picard method is easy to use but, like the other linear iterative methods, it converges linearly. The non-linear iterative methods normally have higher speed of convergence than the linear methods. The quadratic methods are the only non-linear methods which receive wide attentions. The Newton method and its alternative forms (Thomasset, 1981; Engelman et al., 1981; Smith, 1985; Cuvelier et al., 1986; Girault and Raviart, 1986; Johnson, 1987) are among the well developed quadratic methods. The Newton method linearizes the non-linear terms of equations by using the values obtained from the preceding iteration plus a corrector. The Newton method requires a good initial guess for the solution but converges quadratically. The computation time can be substantially reduced by using the Newton method. As suggested by Engelman et al. (1981), the values obtained from the the first iteration of Picard iteration is a good selection for the initial guess in Newton method. An alternative form of the Newton

method is chosen to linearize the equations, which will give the same result as the Newton method. Using this method, the velocity at the new level of iteration can be expressed as the sum of the preceding level and a correction.

$$u^n = u^{n-1} + du^{n-1} \tag{2.10}$$

When u^n is close to the exact solution, u, then du^{n-1} is small. For convenience, the asterisk used in the previous sections to denote the dimensionless variables will be omitted from now on. Substituting Eq. (2.10) into Eqs. (2.5) - (2.8) results in the following linearized form:

$$\frac{\partial u^n}{\partial x} + \frac{\partial v^n}{\partial y} + \frac{\partial w^n}{\partial z} = 0 \tag{2.11}$$

$$\left(u^n \frac{\partial u^{n-1}}{\partial x} + u^{n-1} \frac{\partial u^n}{\partial x} - u^{n-1} \frac{\partial u^{n-1}}{\partial x} \right) + \left(v^n \frac{\partial u^{n-1}}{\partial y} + v^{n-1} \frac{\partial u^n}{\partial y} - v^{n-1} \frac{\partial u^{n-1}}{\partial y} \right) + \left(w^n \frac{\partial u^{n-1}}{\partial z} + w^{n-1} \frac{\partial u^n}{\partial z} - w^{n-1} \frac{\partial u^{n-1}}{\partial z} \right) = -\frac{\partial p^n}{\partial x} + \frac{1}{Re} \left(\frac{\partial^2 u^n}{\partial x^2} + \frac{\partial^2 u^n}{\partial y^2} + \frac{\partial^2 u^n}{\partial z^2} \right) \tag{2.12}$$

$$\left(u^n \frac{\partial v^{n-1}}{\partial x} + u^{n-1} \frac{\partial v^n}{\partial x} - u^{n-1} \frac{\partial v^{n-1}}{\partial x} \right) + \left(v^n \frac{\partial v^{n-1}}{\partial y} + v^{n-1} \frac{\partial v^n}{\partial y} - v^{n-1} \frac{\partial v^{n-1}}{\partial y} \right) + \left(w^n \frac{\partial v^{n-1}}{\partial z} + w^{n-1} \frac{\partial v^n}{\partial z} - w^{n-1} \frac{\partial v^{n-1}}{\partial z} \right) = -\frac{\partial p^n}{\partial y} + \frac{1}{Re} \left(\frac{\partial^2 v^n}{\partial x^2} + \frac{\partial^2 v^n}{\partial y^2} + \frac{\partial^2 v^n}{\partial z^2} \right) \tag{2.13}$$

$$\left(u^n \frac{\partial w^{n-1}}{\partial x} + u^{n-1} \frac{\partial w^n}{\partial x} - u^{n-1} \frac{\partial w^{n-1}}{\partial x} \right) + \left(v^n \frac{\partial w^{n-1}}{\partial y} + v^{n-1} \frac{\partial w^n}{\partial y} - v^{n-1} \frac{\partial w^{n-1}}{\partial y} \right)$$
$$+ \left(w^n \frac{\partial w^{n-1}}{\partial z} + w^{n-1} \frac{\partial w^n}{\partial z} - w^{n-1} \frac{\partial w^{n-1}}{\partial z} \right) = - \frac{\partial p^n}{\partial z} + \frac{1}{Re} \left(\frac{\partial^2 w^n}{\partial x^2} + \frac{\partial^2 w^n}{\partial y^2} \right.$$
$$\left. + \frac{\partial^2 w^n}{\partial z^2} \right) \qquad (2.14)$$

The linearized Eqs. (2.11) - (2.14) can be discretized by the finite element procedure which will be described in the next chapter. After the discretization, a system of equations in terms of the unknown velocity components and pressures can be obtained by the finite element procedure. This system of equations will be solved for the unknown velocity components and pressures by a spare matrix solver. The solution will be substituted into Eqs.(2.11) - (2.14) for the next iteration, and so on.

Chapter III
Numerical Techniques

3.1 Introduction

The mathematical model described in the previous chapter takes the form of non-linear partial differential equations. Only for the very simplest cases is it possible to find the analytical solutions of these equations. In general, one has to employ the numerical approaches to find the approximate solutions of these non-linear differential equations. The numerical approaches rely on computers and numerical methods. In the last three decades, the computer technology has grown dramatically. Both the computer speed and memory have increased by three order of magnitude since the mid 1960's. In addition to the improvement of speed and memory, the development of parallel processing technology also leads to a faster performance on the new generation of computer. Accompanied by the improved computer technology was the rapid advance of numerical methods to solve hitherto intractable problems in physics and engineering. The most well known methods are the finite difference method and the finite element method. Both methods are widely accepted and utilized to solve problems in just about every branch of the physics and engineering. The application of finite difference methods to solve the partial differential

equations was first discussed by Courant et al. in 1928. The practical use of the methods was only made possible after the first digital computer was built in the 1940's. For general discussion of finite difference methods, see Richtmyer and Morton (1967), Mitchell and Griffith (1980), Hildebrand (1968), Isaacson and Keller (1966), Smith (1985), and Strikwerda (1989).

The term "*finite element method*" first appeared in 1960 (Clough, 1960). However, the ideas of the finite element method can be dated back much further. From a mathematical standpoint the finite element method is one of the variational methods which first appeared in the eighteenth century (Courant and Hilbert, 1937). The forerunner of the method is the well-known Ritz method which was developed in the first decade of the twentieth century (Ritz, 1909). The first application of the concept of finite element appeared in a paper by Courant (1943). The method was later introduced to the engineering community as the *direct stiffness method* to solve the aircraft structure problems (Turner et al., 1956). In the 1960's the method was refined and found to be applicable to problems in other field of engineering (Zienkiewicz and Cheung, 1965; Zienkiewicz, 1967). The rapid growth of computer technology after the mid 1960's helped opening the way to the solution of complex problems and lead to widespread use in physics and engineering. Up-to-date, the finite element method has been recognized as a general technique to handle problems that can be cast into variational form.

The solution procedure for a problem by the finite element method consists of several orderly steps. The present study basically follows these steps. They will be discussed in detail later.

1. Discretization of the computational domain
2. Selection of the interpolation functions
3. Formulation of the element equations
4. Assembly of the element equations to obtain the system equations
5. Solution of the system equations.
6. Post processor.

3.2 Discretization of the Computational Domain

The first step is to divide the computational region into a finite number of subregions. These subregions are called the *elements*. In each element, the equations of continuity and momentum must be satisfied. In the present study the typical computational domain, as shown in Figure 3.1, consists of a sphere enclosed by a rectangular prism boundary. The boundary of the region includes the surface of sphere, the side walls of channel, the bottom of channel, the free-surface, and the upstream and downstream cross-sections of channel flow. Compared to the one- and two-dimensional cases, there are relatively few literatures available for the three-dimensional finite elements. The two commonly used shapes of

elements are the tetrahedron and the hexahedron (Figure 3.2). Each of these two elements can be straight sided or curve sided. Straight sided elements are simple to use. However, problems are normally encountered when applying this type of elements to curved boundaries. The elements with curved sides have been developed to correct these problems. These elements are known to be the *isoparametric elements* (Figure 3.3). For the boundary at the surface of sphere, the isoparametric elements are preferable to the straight sided ones. On the other hand, these elements can be used for other boundaries as well. For consistency and simplicity, the isoparametric elements are chosen to discretize the whole region.

In each element, the nodal point distribution can determine the mathematical property of element. Examples are shown in Figure 3.4. The simplest possible elements are the *tri-linear* elements (Figures 3.4(a) and 3.4(b)). All the nodal points are located at corners of element. The values of velocity components at each node are taken as nodal unknown field variables. The pressure is taken to be one unknown field variable over the element. These elements are characterized by the linear velocity and constant pressure. Each component of the velocity is a linear function of values of field variables at nodal points. The pressure is fixed within an element. Thus, the discontinuity of pressure exists between any pair of connected elements. The use of these linear elements is simple which is the only reason to use them. Indeed, they are not adequate to compute complex problems. For the present study, the tri-linear tetrahedron has been once tested for the computation. The results show that the pressure is

unstable throughout the computational region. The same problem was reported by Fortin (1981) for the tri-linear hexahedral element.

The tri-quadratic elements (Figures 3.4 (c) and 3.4(d)) have nodal points at vertices, mid-points of sides, the centroids of faces and the center of element. In these elements, velocity components are approximated by tri-quadratic functions based on the values at nodes of elements. The pressure is approximated by the tri-linear function based on the values at the corner nodes of element. So far there has been no report of any problem encountered using these elements. With the higher order of accuracy, these elements appear to be attractive choices for the 3D computation. For the present study, the hexahedron has the adequate shape for the channel flow region. With a little difficulty, these elements can still be used to approximate the surface of sphere. The typical discretization of the region is shown in Figure 3.5. The discretized region will form the grid system for the finite element method. Figure 3.6 illustrates the cross-sectional profiles of the finite element grid system with nodal points. Within the grid system, a representative element is shown in Figure 3.7. The property of this type of element will be discussed later in this chapter.

The tri-cubic elements have more nodes than the corresponding quadratic elements. The degrees of approximation functions for the velocity and pressure within the tri-cubic elements are also higher than those of the tri-quadratic elements. The velocity components are approximated by the tri-cubic functions and the pressure is approximated by the tri-quadratic function for these elements. It needs more effort to use these elements than

the tri-quadratic elements. Although the order of accuracy is higher than that of tri-quadratic elements, the computation may not be cost effective. More studies are needed to address this question.

3.3 Selection of the Interpolation Functions

The basic idea of finite element method is to divide the field into small regions and approximate the solution over each small region by a simple function. As stated in the previous section, the tri-quadratic hexahedral element has been selected to discretize the computational domain. In this section, the selection of interpolation functions (also known as the basis functions) for this type of element will be discussed.

In the numerical analysis, given a list of known values at several points in the geometric space, an approximate function passing through these points can be found to produce values at other points. The approximate function is normally chosen to be a polynomial or spline function. This procedure is known as the interpolation. In the finite element method, similar procedure can be applied within one element to evaluate a variable. The value of the variable at any point within the element can be obtained by interpolating the values of the corresponding field variable at nodal points of the element. For the hexahedral element under considerations, there are 27 nodes for the velocity components and 8 nodes for the pressure. Each of the velocity components at any point within the element can be expressed

as a linear combination of the product of the velocity components and the basis functions associated with the nodal points.

$$u(\underset{\sim}{x}) = \sum_{i=1}^{27} u_i \phi_i(\underset{\sim}{x}), \qquad v(\underset{\sim}{x}) = \sum_{i=1}^{27} v_i \phi_i(\underset{\sim}{x}), \qquad w(\underset{\sim}{x}) = \sum_{i=1}^{27} w_i \phi_i(\underset{\sim}{x}) \tag{3.1}$$

where u,v and w = the velocity components in the x-, y- and z- direction, respectively;

$u_i = u(\underset{\sim}{x}_i)$; $v_i = v(\underset{\sim}{x}_i)$; $w_i = w(\underset{\sim}{x}_i)$;

$\underset{\sim}{x}_i$ = coordinate vector of the i-th nodal point;

$\phi_i(\underset{\sim}{x})$ = the basis function corresponding to i-th nodal point;

$\underset{\sim}{x}$ = any arbitrary point within the element.

Similarly, the pressure at any point within element can be written as:

$$p(\underset{\sim}{x}) = \sum_{m=1}^{8} p_m \psi_m(\underset{\sim}{x}) \tag{3.2}$$

where $p_m = p(\underset{\sim}{x}_m)$ = pressure at node $\underset{\sim}{x}_m$;

$\psi_m(\underset{\sim}{x})$ = the basis function corresponding to nodal point $\underset{\sim}{x}_m$.

The basis functions $\phi_i(\underset{\sim}{x})$ and $\psi_m(\underset{\sim}{x})$ in Equations (3.1) and (3.2) are defined over the element. To retain the values of velocity and pressure at the nodal points, these basis functions are required to satisfy the following conditions.

$$\phi_i(\underset{\sim}{x}_j) = \delta_{ij} \qquad \text{for } i, j = 1, 2, ..., 27 \tag{3.3}$$

$$\psi_m(\underset{\sim}{x}_n) = \delta_{mn} \qquad \text{for } m, n = 1, 2, ..., 8 \tag{3.4}$$

in which δ_{ij} = Kronecker delta.

Since the solutions of Equations (2.1) - (2.4) are differentiable functions over the domain of flow, the basis functions in Equations (3.1) and (3.2) must be differentiable over the domain of each element. The polynomial functions always satisfy this requirement and are suitable form for the basis functions. The possible polynomial form for the function ϕ_i defined over the 27 nodal points of one element is given by:

$$\phi_i(\underset{\sim}{x}) = \alpha_1^{(i)} + \alpha_2^{(i)}x + \alpha_3^{(i)}y + \alpha_4^{(i)}z + \alpha_5^{(i)}x^2 + \alpha_6^{(i)}y^2 + \alpha_7^{(i)}z^2 + \alpha_8^{(i)}xy + \alpha_9^{(i)}xz$$

$$+ \alpha_{10}^{(i)}yz + \alpha_{11}^{(i)}xyz + \alpha_{12}^{(i)}x^2y + \alpha_{13}^{(i)}x^2z + \alpha_{14}^{(i)}y^2x + \alpha_{15}^{(i)}y^2z + \alpha_{16}^{(i)}z^2x$$

$$+ \alpha_{17}^{(i)}z^2y + \alpha_{18}^{(i)}x^2yz + \alpha_{19}^{(i)}y^2xz + \alpha_{20}^{(i)}z^2xy + \alpha_{21}^{(i)}x^2y^2 + \alpha_{22}^{(i)}x^2z^2 + \alpha_{23}^{(i)}y^2z^2$$

$$+ \alpha_{24}^{(i)}xy^2z^2 + \alpha_{25}^{(i)}x^2yz^2 + \alpha_{26}^{(i)}x^2y^2z + \alpha_{27}^{(i)}x^2y^2z^2 \qquad i = 1, 2, \ldots, 27 \quad (3.5)$$

By using the condition (3.3) for each of the 27 nodes, the coefficients $\{\alpha_1^{(i)}, \alpha_2^{(i)}, \ldots, \alpha_{27}^{(i)}\}$ can be determined for the function $\phi_i(\underset{\sim}{x})$. Note that each of $\phi_i(\underset{\sim}{x})$ has different set of coefficients $\{\alpha_1^{(i)}, \alpha_2^{(i)}, \ldots, \alpha_{27}^{(i)}\}$.

Similarly, $\psi_m(\underset{\sim}{x})$ can be written as a tri-linear polynomial function.

$$\psi_m(\underset{\sim}{x}) = \beta_1^{(m)} + \beta_2^{(m)}x + \beta_3^{(m)}y + \beta_4^{(m)}z + \beta_8^{(m)}xy + \beta_6^{(m)}xz + \beta_7^{(m)}yz + \beta_8^{(m)}xyz$$

$$m = 1, 2, \ldots, 8 \quad (3.6)$$

The coefficients $\{\beta_1^{(m)}, \beta_2^{(m)}, \ldots, \beta_8^{(m)}\}$ for each of $\psi_m(\underset{\sim}{x})$ can be determined by using the condition (3.4) for 8 vertices within the element.

The procedure described above has to repeat for each element. For each element, therefore, a 27 x 27 matrix equation must be solved for each of the basis functions $\phi_i(\underset{\sim}{x})$, and an 8 x 8 matrix equation must be solved for each of $\psi_m(\underset{\sim}{x})$. Thus there are twenty-seven of 27 x 27 matrix equations and eight of 8 x 8 matrix equations to be solved for each element. None of the calculation can be reused for other elements, resulting in repeated effort.

A less expensive way to evaluate the basis functions is to make use of the *reference* element. In this procedure, the basis functions are defined on the reference element and then transformed into basis functions on the element under consideration. The most attractive feature of this method is that all computations for each finite element can be carried out on one reference element. It does not require to solve systems of equations for the basis function as described above. For the present study, the reference element is chosen to be a unit cube (Figure 3.8) with 27 nodal points listed below.

$$\underset{\sim}{x}_1 = (0, 0, 0), \quad \underset{\sim}{x}_2 = (1, 0, 0), \quad \underset{\sim}{x}_3 = (1, 0, 1), \quad \underset{\sim}{x}_4 = (0, 0, 1),$$

$$\underset{\sim}{x}_5 = (\tfrac{1}{2}, 0, 0), \quad \underset{\sim}{x}_6 = (1, 0, \tfrac{1}{2}), \quad \underset{\sim}{x}_7 = (\tfrac{1}{2}, 0, 1), \quad \underset{\sim}{x}_8 = (0, 0, \tfrac{1}{2}),$$

$$\underset{\sim}{x}_9 = (\tfrac{1}{2}, 0, \tfrac{1}{2}), \quad \underset{\sim}{x}_{10} = (0, 1, 0), \quad \underset{\sim}{x}_{11} = (1, 1, 0), \quad \underset{\sim}{x}_{12} = (1, 1, 1),$$

$$\underset{\sim}{x}_{13} = (0, 1, 1), \quad \underset{\sim}{x}_{14} = (\tfrac{1}{2}, 1, 0), \quad \underset{\sim}{x}_{15} = (1, 1, \tfrac{1}{2}), \quad \underset{\sim}{x}_{16} = (\tfrac{1}{2}, 1, 1), \qquad (3.7)$$

$$\underset{\sim}{x}_{17} = (0, 1, \tfrac{1}{2}), \quad \underset{\sim}{x}_{18} = (\tfrac{1}{2}, 1, \tfrac{1}{2}), \quad \underset{\sim}{x}_{19} = (0, \tfrac{1}{2}, 0), \quad \underset{\sim}{x}_{20} = (1, \tfrac{1}{2}, 0),$$

$$\underset{\sim}{x}_{21} = (1, \tfrac{1}{2}, 1), \quad \underset{\sim}{x}_{22} = (0, \tfrac{1}{2}, 1), \quad \underset{\sim}{x}_{23} = (\tfrac{1}{2}, \tfrac{1}{2}, 0), \quad \underset{\sim}{x}_{24} = (1, \tfrac{1}{2}, \tfrac{1}{2}),$$

$$\underset{\sim}{x}_{25} = (\tfrac{1}{2}, \tfrac{1}{2}, 1), \quad \underset{\sim}{x}_{26} = (0, \tfrac{1}{2}, \tfrac{1}{2}), \quad \underset{\sim}{x}_{26} = (0, \tfrac{1}{2}, \tfrac{1}{2}), \quad \underset{\sim}{x}_{27} = (\tfrac{1}{2}, \tfrac{1}{2}, \tfrac{1}{2}).$$

The basis functions for this unit cube take the tri-quadratic polynomial form.

$$\hat{\phi}_{1+9i}(\underset{\sim}{r}) = \theta_1(r)\,\theta_1(t)\,\theta_{i+1}(s)$$

$$\hat{\phi}_{2+9i}(\underset{\sim}{r}) = \theta_2(r)\,\theta_1(t)\,\theta_{i+1}(s)$$

$$\hat{\phi}_{3+9i}(\underset{\sim}{r}) = \theta_2(r)\,\theta_2(t)\,\theta_{i+1}(s)$$

$$\hat{\phi}_{4+9i}(\underset{\sim}{r}) = \theta_1(r)\,\theta_2(t)\,\theta_{i+1}(s)$$

$$\hat{\phi}_{5+9i}(\underset{\sim}{r}) = \theta_3(r)\,\theta_1(t)\,\theta_{i+1}(s) \qquad i = 0, 1, 2 \qquad (3.8)$$

$$\hat{\phi}_{6+9i}(\underset{\sim}{r}) = \theta_2(r)\,\theta_3(t)\,\theta_{i+1}(s)$$

$$\hat{\phi}_{7+9i}(\underset{\sim}{r}) = \theta_3(r)\,\theta_2(t)\,\theta_{i+1}(s)$$

$$\hat{\phi}_{8+9i}(\underset{\sim}{r}) = \theta_1(r)\,\theta_3(t)\,\theta_{i+1}(s)$$

$$\hat{\phi}_{9+9i}(\underset{\sim}{r}) = \theta_3(r)\,\theta_3(t)\,\theta_{i+1}(s)$$

in which $\theta_1(q) = (1-q)\,(1-2q)$, $\theta_2(q) = q\,(2q-1)$, $\theta_3(q) = 4q\,(1-q)$,

$\underset{\sim}{r} = (r, s, t)$ = arbitrary point within the unit cube.

These basis functions are defined on the reference element $\hat{e}$ corresponding to the nodal points described in Eq.(3.7). It can be observed that the condition

$$\hat{\phi}_i(\underset{\sim}{r}_j) = \delta_{ij} \qquad i, j = 1, 2, \ldots, 27 \tag{3.9}$$

is satisfied by each of $\hat{\phi}_i(\underset{\sim}{r})$. Besides, all of these functions are differentiable over the reference element.

The connection between an arbitrary hexahedral element **e** and the reference element $\hat{e}$ is based on the coordinate transform expressed by:

$$\underset{\sim}{x} = \sum_{i=1}^{27} \underset{\sim}{x}_i \, \hat{\phi}_i(\underset{\sim}{r}) \tag{3.10}$$

or in the scalar form:

$$x = \sum_{i=1}^{27} x_i \, \hat{\phi}_i(\underset{\sim}{r}), \qquad y = \sum_{i=1}^{27} y_i \, \hat{\phi}_i(\underset{\sim}{r}), \qquad z = \sum_{i=1}^{27} z_i \, \hat{\phi}_i(\underset{\sim}{r}) \tag{3.11}$$

in which $\underset{\sim}{r}$ is an arbitrary point within the reference element, $\underset{\sim}{x}$ is the image of $\underset{\sim}{r}$ onto the arbitrary element. It can be observed that the mapping by Equation (3.10) is chosen such that vertices are mapped onto vertices, the mid-points of sides are mapped onto mid-points of sides, and the centers of faces are mapped onto centers of faces. The image of the reference element is a curve sided element, i.e., an isoparametric element **e**. Denote the mapping Equation (3.10) by:

$$L(\underset{\sim}{r}) = \sum_{i=1}^{27} \underset{\sim}{x}_i \hat{\phi}_i(\underset{\sim}{r}) , \qquad \text{i.e.,} \qquad \underset{\sim}{x} = L(\underset{\sim}{r}) \tag{3.12}$$

Let Ω denotes the closed region enclosed by element **e** and $\hat{\Omega}$ denote the closed region enclosed by $\hat{\mathbf{e}}$. The transformation L is one to one, i.e., $L(\underset{\sim}{r}_s) = L(\underset{\sim}{r}_t)$ implies that $\underset{\sim}{r}_s = \underset{\sim}{r}_t$, and onto Ω. Thus there exists a transformation function S on Ω to $\hat{\Omega}$ defined as follows: $S(\underset{\sim}{x})$ is a vector in $\hat{\Omega}$ such that

$$L(S(\underset{\sim}{x})) = \underset{\sim}{x} ,$$

i.e., S is the *inverse* of L and is denoted by $S \equiv L^{-1}$.

The basis functions $\phi_i(\underset{\sim}{x})$ on the isoparametric element corresponding to the points $\underset{\sim}{x}_i = L(\underset{\sim}{r}_i)$ are defined by:

$$\phi_i(\underset{\sim}{x}) = \hat{\phi}_i(L^{-1}(\underset{\sim}{x})) \tag{3.13a}$$

or

$$\phi_i(\underset{\sim}{x}) = \hat{\phi}_i(\underset{\sim}{r}) \tag{3.13b}$$

for any point $\underset{\sim}{x} = L(\underset{\sim}{r})$ on **e**.

Since $\underset{\sim}{x}$ depends linearly on $\underset{\sim}{r}$, the degree of the basis functions is preserved.

The condition (3.9) leads to

$$\phi_i(\underset{\sim}{x}_j) = \hat{\phi}_i(\underset{\sim}{r}_j) = \delta_{ij} \qquad \text{for } i, j = 1, 2, \ldots , 27 . \tag{3.14}$$

The same procedure described above applies to the interpolation functions of pressure variable. Instead of 27 nodes, only 8 nodes at corners are used for this case (Figure 3.9). The basis functions are given by:

$$\hat{\psi}_1(\underset{\sim}{r}) = (1-r)\,(1-s)\,(1-t)$$

$$\hat{\psi}_2(\underset{\sim}{r}) = r\,(1-s)\,(1-t)$$

$$\hat{\psi}_3(\underset{\sim}{r}) = r\,(1-s)\,t$$

$$\hat{\psi}_4(\underset{\sim}{r}) = (1-r)\,(1-s)\,t \qquad (3.15)$$

$$\hat{\psi}_5(\underset{\sim}{r}) = (1-r)\,s\,(1-t)$$

$$\hat{\psi}_6(\underset{\sim}{r}) = r\,s\,(1-t)$$

$$\hat{\psi}_7(\underset{\sim}{r}) = r\,s\,t$$

$$\hat{\psi}_8(\underset{\sim}{r}) = (1-r)\,s\,t$$

These basis functions are tri-linear functions. The mapping applies to these basis functions and the images of these functions are also tri-linear functions.

3.4 Formulation of the Element Equations

Once the finite elements and their interpolation functions have been selected, one can determine the coefficients of the governing equations for the field variables at each element. The continuity and the momentum equations defined over the entire computational domain as described in the previous chapter must be discretized by a variational procedure or other

methods before they can be applied to the subregion formed by an element. As pointed out earlier, the computational procedure can be simpler and less expensive by using a reference element, $\hat{\mathbf{e}}$, rather than using individual element **e** within the grid. Thus the 27-node reference element $\hat{\mathbf{e}}$ (Figure 3.8) will be used throughout the entire computation. For the discretization of the governing equations, a variational procedure will be applied with the interpolating functions defined in the previous section. Let Ω denote the closed region enclosed by element **e** and Γ the boundary of Ω. Consider the Equations (2.11) - (2.14). Each of them includes the unknown variables u, v, w and p which can be expressed in terms of nodal point values and the interpolation functions as shown in Eqs.(3.1) and (3.2). Eq. (2.11) is multiplied by ψ_m and Eqs. (2.12) through (2.14) by ϕ_i, respectively. Then taking the partial integration over the domain Ω leads to the discretized forms of equations.

$$\sum_{j=1}^{27} \left(u_j^n \int_\Omega \psi_m \frac{\partial \phi_j}{\partial x}\, d\Omega + v_j^n \int_\Omega \psi_m \frac{\partial \phi_j}{\partial y}\, d\Omega + w_j^n \int_\Omega \psi_m \frac{\partial \phi_j}{\partial z}\, d\Omega\right) = 0$$

$$m = 1, 2, \ldots, 8 \qquad (3.16)$$

$$\sum_{j=1}^{27} \Bigg[u_j^n \Bigg\{ \int_\Omega \Bigg[\phi_i \left(\frac{\partial u^{n-1}}{\partial x} \phi_j + u^{n-1} \frac{\partial \phi_j}{\partial x} + v^{n-1} \frac{\partial \phi_j}{\partial y} + w^{n-1} \frac{\partial \phi_j}{\partial z}\right) + \frac{1}{Re}\left(\frac{\partial \phi_j}{\partial x} \frac{\partial \phi_i}{\partial x} + \frac{\partial \phi_j}{\partial y} \frac{\partial \phi_i}{\partial y} + \frac{\partial \phi_j}{\partial z} \frac{\partial \phi_i}{\partial z}\right)\Bigg] d\Omega - \int_\Gamma \frac{\phi_i}{Re} \left(\frac{\partial \phi_j}{\partial x} \mathbf{n_x} + \frac{\partial \phi_j}{\partial y} \mathbf{n_y} + \frac{\partial \phi_j}{\partial z} \mathbf{n_z}\right) d\Gamma \Bigg\} + v_j^n \int_\Omega \phi_i$$

$$\phi_j \frac{\partial u^{n-1}}{\partial y}\, d\Omega + w_j^n \int_\Omega \phi_i \phi_j \frac{\partial u^{n-1}}{\partial z}\, d\Omega] + \sum_{m=1}^{8} [\, p_m^n (- \int_\Omega \psi_m \frac{\partial \phi_i}{\partial x} d\Omega + \int_\Gamma \psi_m$$

$$\phi_i \, \mathbf{n_x}\, d\Gamma)] = \int_\Omega \phi_i \, (u^{n-1} \frac{\partial u^{n-1}}{\partial x} + v^{n-1} \frac{\partial u^{n-1}}{\partial y} + w^{n-1} \frac{\partial u^{n-1}}{\partial z}) d\Omega$$

$$i = 1, 2, \ldots , 27 \qquad (3.17)$$

$$\sum_{j=1}^{27} [\, u_j^n \int_\Omega \phi_i \phi_j \frac{\partial v^{n-1}}{\partial x}\, d\Omega + v_j^n \{ \int_\Omega [\, \phi_i \, (u^{n-1} \frac{\partial \phi_j}{\partial x} + \frac{\partial v^{n-1}}{\partial y} \phi_j + v^{n-1} \frac{\partial \phi_j}{\partial y} + w^{n-1}$$

$$\frac{\partial \phi_j}{\partial z}) + \frac{1}{Re} (\frac{\partial \phi_j}{\partial x} \frac{\partial \phi_i}{\partial x} + \frac{\partial \phi_j}{\partial y} \frac{\partial \phi_i}{\partial y} + \frac{\partial \phi_j}{\partial z} \frac{\partial \phi_i}{\partial z})] \, d\Omega - \int_\Gamma \frac{\phi_i}{Re} (\frac{\partial \phi_j}{\partial x} \mathbf{n_x} + \frac{\partial \phi_j}{\partial y} \mathbf{n_y} +$$

$$\frac{\partial \phi_j}{\partial z} \mathbf{n_z})\, d\Gamma \} + w_j^n \int_\Omega \phi_i \phi_j \frac{\partial v^{n-1}}{\partial z}\, d\Omega] + \sum_{m=1}^{8} [\, p_m^n (- \int_\Omega \psi_m \frac{\partial \phi_i}{\partial y} d\Omega + \int_\Gamma \psi_m$$

$$\phi_i \, \mathbf{n_y}\, d\Gamma)] = \int_\Omega \phi_i \, (u^{n-1} \frac{\partial v^{n-1}}{\partial x} + v^{n-1} \frac{\partial v^{n-1}}{\partial y} + w^{n-1} \frac{\partial v^{n-1}}{\partial z}) d\Omega$$

$$i = 1, 2, \ldots , 27 \qquad (3.18)$$

$$\sum_{j=1}^{27} [\, u_j^n \int_\Omega \phi_i \phi_j \frac{\partial w^{n-1}}{\partial x}\, d\Omega + v_j^n \int_\Omega \phi_i \phi_j \frac{\partial w^{n-1}}{\partial y}\, d\Omega + w_j^n \{ \int_\Omega [\, \phi_i \, (u^{n-1} \frac{\partial \phi_j}{\partial x} + v^{n-1}$$

$$\frac{\partial \phi_j}{\partial y} + \frac{\partial w^{n-1}}{\partial z} \phi_j + w^{n-1} \frac{\partial \phi_j}{\partial z}) + \frac{1}{Re} (\frac{\partial \phi_j}{\partial x} \frac{\partial \phi_i}{\partial x} + \frac{\partial \phi_j}{\partial y} \frac{\partial \phi_i}{\partial y} + \frac{\partial \phi_j}{\partial z} \frac{\partial \phi_i}{\partial z})] \, d\Omega -$$

$$\int_\Gamma \frac{\phi_i}{Re} (\frac{\partial \phi_j}{\partial x} \mathbf{n_x} + \frac{\partial \phi_j}{\partial y} \mathbf{n_y} + \frac{\partial \phi_j}{\partial z} \mathbf{n_z})\, d\Gamma \}] + \sum_{m=1}^{8} [\, p_m^n (- \int_\Omega \psi_m \frac{\partial \phi_i}{\partial z} d\Omega +$$

$$\int_\Gamma \psi_m \phi_i \, \mathbf{n_z}\, d\Gamma)] = \int_\Omega \phi_i \, (u^{n-1} \frac{\partial w^{n-1}}{\partial x} + v^{n-1} \frac{\partial w^{n-1}}{\partial y} + w^{n-1} \frac{\partial w^{n-1}}{\partial z}) d\Omega$$

$$i = 1, 2, \ldots , 27 \qquad (3.19)$$

where u^{n-1}, v^{n-1}, w^{n-1} = velocity components at the node from previous iteration or initial guess.

Note that Eq.(3.16) represents eight equations at corners of element, and each of Eqs.(3.17) - (3.19) represents 27 equations at all nodal points of element **e**. Thus there are totally 89 equations for each element. The unknown field variables within one element include 81 velocity components and 8 pressures. Therefore, the total number of unknowns matches the number of equations for each element. Equations (3.16) - (3.19) can be written in the form of matrices equations:

$$[\mathbf{A}]\,\{\mathbf{X}\} = \{\mathbf{B}\} \tag{3.20}$$

where

$$[\mathbf{A}] = \begin{bmatrix} [K_1^1] \ldots [K_{27}^1] & \{L_1^1\} \ldots \{L_8^1\} \\ \vdots \quad \vdots \quad \vdots & \vdots \quad \vdots \quad \vdots \\ [K_1^{27}] \ldots [K_{27}^{27}] & \{L_1^{27}\} \ldots \{L_8^{27}\} \\ \{C_1^1\} \ldots \{C_{27}^1\} & 0 \ \ldots \ 0 \\ \vdots \quad \vdots \quad \vdots & \vdots \quad \vdots \quad \vdots \\ \{C_1^8\} \ldots \{C_{27}^8\} & 0 \ \ldots \ 0 \end{bmatrix},$$

$$\{X\} = \begin{Bmatrix} \{U_1^{(e)}\} \\ . \\ \{U_{27}^{(e)}\} \\ \{P_1^{(e)}\} \\ . \\ \{P_8^{(e)}\} \end{Bmatrix}, \qquad \{B\} = \begin{Bmatrix} \{F_1^{(e)}\} \\ . \\ \{F_8^{(e)}\} \\ 0 \\ . \\ 0 \end{Bmatrix}.$$

$$\{U_i^{(e)}\} = \begin{Bmatrix} u_i \\ v_i \\ w_i \end{Bmatrix}, \qquad i = 1, 2, \ldots, 27$$

u_i, v_i, w_i = velocity components at node i,

$$\{P_j^{(e)}\} = p_j^n, \qquad j = 1, 2, \ldots, 8$$

p_j^n = pressure stress at corner of element,

$$\{F_1^{(e)}\} = \begin{Bmatrix} f_{ix}^{(e)} \\ f_{iy}^{(e)} \\ f_{iz}^{(e)} \end{Bmatrix},$$

$$[K_j^i] = \begin{bmatrix} k_{ij}^{(e)11} & k_{ij}^{(e)12} & k_{ij}^{(e)13} \\ k_{ij}^{(e)21} & k_{ij}^{(e)22} & k_{ij}^{(e)23} \\ k_{ij}^{(e)31} & k_{ij}^{(e)32} & k_{ij}^{(e)33} \end{bmatrix} \qquad i, j = 1, 2, \ldots, 27$$

$$\{L_m^i\} = \begin{Bmatrix} l_{i;m}^{(e)1} \\ l_{i;m}^{(e)2} \\ l_{i;m}^{(e)3} \end{Bmatrix} \qquad i=1, 2, \ldots, 27\ ,\ m = 1, 2, \ldots, 8$$

$$\{C_j^m\} = \left\{ c_{m;j}^{(e)1}\ c_{m;j}^{(e)2}\ c_{m;j}^{(e)3} \right\} \qquad j = 1, 2, \ldots, 27\ ,\ m = 1, 2, \ldots, 8$$

The coefficients in the equation above are given by:

$$k_{ij}^{(e)11} = \int_\Omega [\, \phi_i \left(\frac{\partial u^{n-1}}{\partial x} \phi_j + u^{n-1} \frac{\partial \phi_j}{\partial x} + v^{n-1} \frac{\partial \phi_j}{\partial y} + w^{n-1} \frac{\partial \phi_j}{\partial z}\right) + \frac{1}{Re}\left(\frac{\partial \phi_j}{\partial x} \frac{\partial \phi_i}{\partial x} + \frac{\partial \phi_j}{\partial y} \frac{\partial \phi_i}{\partial y} + \frac{\partial \phi_j}{\partial z} \frac{\partial \phi_i}{\partial z}\right)] \, d\Omega - \int_\Gamma \frac{\phi_i}{Re} \left(\frac{\partial \phi_j}{\partial x} \mathbf{n_x} + \frac{\partial \phi_j}{\partial y} \mathbf{n_y} + \frac{\partial \phi_j}{\partial z} \mathbf{n_z}\right) d\Gamma$$

$$i, j = 1, 2, \ldots, 27 \qquad (3.21a)$$

$$k_{ij}^{(e)12} = \int_\Omega \phi_i \phi_j \frac{\partial u^{n-1}}{\partial y} \, d\Omega \qquad i, j = 1, 2, \ldots, 27 \qquad (3.21b)$$

$$k_{ij}^{(e)13} = \int_\Omega \phi_i \phi_j \frac{\partial u^{n-1}}{\partial z} \, d\Omega \qquad i, j = 1, 2, \ldots, 27 \qquad (3.21c)$$

$$k_{ij}^{(e)21} = \int_\Omega \phi_i \phi_j \frac{\partial v^{n-1}}{\partial x} \, d\Omega \qquad i, j = 1, 2, \ldots, 27 \qquad (3.21d)$$

$$k_{ij}^{(e)22} = \int_\Omega [\, \phi_i \left(u^{n-1} \frac{\partial \phi_j}{\partial x} + \frac{\partial v^{n-1}}{\partial y} \phi_j + v^{n-1} \frac{\partial \phi_j}{\partial y} + w^{n-1} \frac{\partial \phi_j}{\partial z}\right) + \frac{1}{Re}\left(\frac{\partial \phi_j}{\partial x} \frac{\partial \phi_i}{\partial x} + \frac{\partial \phi_j}{\partial y} \frac{\partial \phi_i}{\partial y} + \frac{\partial \phi_j}{\partial z} \frac{\partial \phi_i}{\partial z}\right)] \, d\Omega - \int_\Gamma \frac{\phi_i}{Re} \left(\frac{\partial \phi_j}{\partial x} \mathbf{n_x} + \frac{\partial \phi_j}{\partial y} \mathbf{n_y} + \frac{\partial \phi_j}{\partial z} \mathbf{n_z}\right) d\Gamma$$

$$i\ , j = 1, 2, \ldots, 27 \qquad (3.21e)$$

$$k_{ij}^{(e)23} = \int_{\Omega} \phi_i \phi_j \frac{\partial v^{n-1}}{\partial z} d\Omega \qquad i, j = 1, 2, \ldots, 27 \qquad (3.21f)$$

$$k_{ij}^{(e)31} = \int_{\Omega} \phi_i \phi_j \frac{\partial w^{n-1}}{\partial x} d\Omega \qquad i, j = 1, 2, \ldots, 27 \qquad (3.21g)$$

$$k_{ij}^{(e)32} = \int_{\Omega} \phi_i \phi_j \frac{\partial w^{n-1}}{\partial y} d\Omega \qquad i, j = 1, 2, \ldots, 27 \qquad (3.21h)$$

$$k_{ij}^{(e)33} = \int_{\Omega} [\, \phi_i \left(u^{n-1} \frac{\partial \phi_j}{\partial x} + v^{n-1} \frac{\partial \phi_j}{\partial y} + \frac{\partial w^{n-1}}{\partial z} \phi_j + w^{n-1} \frac{\partial \phi_j}{\partial z} \right) + \frac{1}{Re} \left(\frac{\partial \phi_j}{\partial x} \frac{\partial \phi_i}{\partial x} + \frac{\partial \phi_j}{\partial y} \frac{\partial \phi_i}{\partial y} + \frac{\partial \phi_j}{\partial z} \frac{\partial \phi_i}{\partial z} \right)] \, d\Omega - \int_{\Gamma} \frac{\phi_i}{Re} \left(\frac{\partial \phi_j}{\partial x} \mathbf{n_x} + \frac{\partial \phi_j}{\partial y} \mathbf{n_y} + \frac{\partial \phi_j}{\partial z} \mathbf{n_z} \right) d\Gamma$$

$$i, j = 1, 2, \ldots, 27 \qquad (3.21i)$$

$$l_{i;m}^{(e)1} = -\int_{\Omega} \psi_m \frac{\partial \phi_i}{\partial x} d\Omega + \int_{\Gamma} \psi_m \phi_i \, \mathbf{n_x} \, d\Gamma \;, \; i = 1, 2, \ldots, 27 \;, \; m = 1, 2, \ldots, 8 \qquad (3.22a)$$

$$l_{i;m}^{(e)2} = -\int_{\Omega} \psi_m \frac{\partial \phi_i}{\partial y} d\Omega + \int_{\Gamma} \psi_m \phi_i \, \mathbf{n_y} \, d\Gamma \;, \; i = 1, 2, \ldots, 27 \;, \; m = 1, 2, \ldots, 8 \qquad (3.22b)$$

$$l_{i;m}^{(e)3} = -\int_{\Omega} \psi_m \frac{\partial \phi_i}{\partial z} d\Omega + \int_{\Gamma} \psi_m \phi_i \, \mathbf{n_z} \, d\Gamma \;, \; i = 1, 2, \ldots, 27 \;, \; m = 1, 2, \ldots, 8 \qquad (3.22c)$$

$$c_{m;j}^{(e)1} = \int_{\Omega} \psi_m \frac{\partial \phi_j}{\partial x} d\Omega \qquad j = 1, 2, \ldots, 27 \;, \; m = 1, 2, \ldots, 8 \qquad (3.23a)$$

$$c_{m;j}^{(e)2} = \int_{\Omega} \psi_m \frac{\partial \phi_j}{\partial y} d\Omega \qquad j = 1, 2, \ldots, 27 \;, \; m = 1, 2, \ldots, 8 \qquad (3.23b)$$

$$c_{m;j}^{(e)3} = \int_{\Omega} \psi_m \frac{\partial \phi_j}{\partial z} d\Omega \qquad j = 1, 2, \ldots, 27 \;, \; m = 1, 2, \ldots, 8 \qquad (3.23c)$$

$$f_{ix}^{(e)} = \int_{\Omega} \phi_i \left(u^{n-1} \frac{\partial u^{n-1}}{\partial x} + v^{n-1} \frac{\partial u^{n-1}}{\partial y} + w^{n-1} \frac{\partial u^{n-1}}{\partial z} \right) d\Omega \quad , i = 1, 2, \ldots , 27 \quad (3.24a)$$

$$f_{iy}^{(e)} = \int_{\Omega} \phi_i \left(u^{n-1} \frac{\partial v^{n-1}}{\partial x} + v^{n-1} \frac{\partial v^{n-1}}{\partial y} + w^{n-1} \frac{\partial v^{n-1}}{\partial z} \right) d\Omega \quad , i = 1, 2, ..., 27 \quad (3.24b)$$

$$f_{iz}^{(e)} = \int_{\Omega} \phi_i \left(u^{n-1} \frac{\partial w^{n-1}}{\partial x} + v^{n-1} \frac{\partial w^{n-1}}{\partial y} + w^{n-1} \frac{\partial w^{n-1}}{\partial z} \right) d\Omega \, , i = 1, 2, \ldots , 27 \quad (3.24c)$$

The integrations in the equations (3.21a) - (3.24c) can be evaluated by the Newton-Cotes formulas or the Gaussian quadrature rules (Stroud, 1971). Since the element **e** can be an image of the reference element $\hat{\mathbf{e}}$ under the mapping $L(\underset{\sim}{r}) = \sum_{i=1}^{27} x_i \hat{\phi}_i(\underset{\sim}{r})$, the integral over **e** of the form:

$$\int_{\Omega} f(\underset{\sim}{x}) \, d\Omega \quad (3.25)$$

can be transformed into the integral over $\hat{\mathbf{e}}$ of the form:

$$\int_{\hat{\Omega}} f(\underset{\sim}{r}) \, d\hat{\Omega} \quad (3.26)$$

By changing variables in Eq.(3.25), it can be written as:

$$\int_{\Omega} f(\underset{\sim}{x}) \, d\Omega = \int_{\hat{\Omega}} f(L(\underset{\sim}{r})) \left| \frac{\partial(x,y,z)}{\partial(r,s,t)} \right| d\hat{\Omega} \quad (3.27)$$

where

$$\left| \frac{\partial(x,y,z)}{\partial(r,s,t)} \right| = \begin{vmatrix} \frac{\partial x}{\partial r} & \frac{\partial x}{\partial s} & \frac{\partial x}{\partial t} \\ \frac{\partial y}{\partial r} & \frac{\partial y}{\partial s} & \frac{\partial y}{\partial t} \\ \frac{\partial z}{\partial r} & \frac{\partial z}{\partial s} & \frac{\partial z}{\partial t} \end{vmatrix} \quad (3.28)$$

= Jocobian of x, y, z, with respect to r, s, t.

Let $J(\underset{\sim}{r})$ denote the Jocobian defined above. Then, any integral over an element **e** can be written in the form:

$$\int_{\Omega} f(\underset{\sim}{x})\, d\Omega = \int_{\hat{\Omega}} f(L(\underset{\sim}{r}))J(\underset{\sim}{r})\, d\hat{\Omega} = \int_{\hat{\Omega}} f(\underset{\sim}{r})\, d\hat{\Omega} \qquad (3.29)$$

The integration can be evaluated over the unit cubic element according to one of the following formulas:

Gaussian quadrature rules (Stroud, 1971):

$$I = \int_{\hat{\Omega}} \hat{f}(\underset{\sim}{r})\, d\hat{\Omega} \approx \sum_{i=1}^{3}\sum_{j=1}^{3}\sum_{k=1}^{3} \omega_i\omega_j\omega_k\, \hat{f}\,(\xi_i, \xi_j, \xi_k) = \hat{I}\ ,$$

degree of precision = 5 (3.30)

where $\omega_1 = \omega_3 = \frac{5}{18}$, $\omega_2 = \frac{8}{18}$,

$\xi_1 = \frac{1}{2} - \frac{\sqrt{15}}{10}$, $\xi_2 = \frac{1}{2}$, $\xi_3 = \frac{1}{2} + \frac{\sqrt{15}}{10}$, $\hat{\Omega}$ is the unit cube.

Newton-Cotes formula (Stroud, 1971):

$$\begin{aligned} I(\hat{f}) = \int_{\hat{\Omega}} \hat{f}(\underset{\sim}{r})\, d\hat{\Omega} \approx \frac{1}{216}\{[\, &\hat{f}_1 + \hat{f}_2 + \hat{f}_3 + \hat{f}_4 + 4(\hat{f}_5 + \hat{f}_6 + \hat{f}_7 + \hat{f}_8) \\ &+ 16\hat{f}_9\,] + 4[\,\hat{f}_{19} + \hat{f}_{20} + \hat{f}_{21} + \hat{f}_{22} + 4(\hat{f}_{23} + \hat{f}_{24} + \hat{f}_{25} + \hat{f}_{26}) \\ &+ 16\hat{f}_{27}\,] + [\,\hat{f}_{10} + \hat{f}_{11} + \hat{f}_{12} + \hat{f}_{13} + 4(\hat{f}_{14} + \hat{f}_{15} + \hat{f}_{16} + \hat{f}_{17}) \\ &+ 16\hat{f}_{18}\,] = \tilde{I}(\hat{f})\ , \qquad \text{degree of precision} = 3 \end{aligned} \qquad (3.31)$$

where $\hat{f}_i = \hat{f}(\underset{\sim}{r}_i)$, $\underset{\sim}{r}_i$ = the i-th node of unit cube (Figure 3.8).

The Gaussian quadrature is known to result extremely accurate result and is faster than most of other methods (Stroud, 1971; Stroud and Secrest, 1966). It can reach the same accuracy with less number of nodal points of approximation. It has been recognized as one of the best numerical integration methods in the study of numerical analysis. The only drawback of this method is that it is difficult to estimate the error (Atkinson, 1978). The other method listed above is one of the Newton-Cotes formulas which are most commonly used methods. The Newton-Cotes formulas include a few famous methods, e.g., the trapezoidal rule which has degree of polynomial 1, and Simpson rules which have degree of polynomial 2. Let n denote the degree of polynomial, f. Then the numerical integration formula $\tilde{I}(f)$ that approximates I(f) is said to have degree of precision m if

(i) $\tilde{I}(f) = I(f)$ for all polynomials f(x) of degree $n \leq m$, and

(ii) $\tilde{I}(f) \neq I(f)$ for all polynomial f of degree $m+1$.

When the degree of polynomial becomes larger, for $n \geq 8$, the Newton-Cotes formulas can cause significant errors. This method should be avoided for integrand with degree higher than 8 (Atkinson, 1978).

As the computation proceeds, it will be necessary to calculate the derivatives of functions in the equations over each element. The derivatives can be evaluated numerically by utilizing the interpolation functions defined in Eq.(3.8). For any function A(x, y, z), its derivatives with respect to x, y, and z and its derivatives with respect to r, s, and t can be related by the chain rule:

$$\frac{\partial A}{\partial r} = \frac{\partial A}{\partial x}\frac{\partial x}{\partial r} + \frac{\partial A}{\partial y}\frac{\partial y}{\partial r} + \frac{\partial A}{\partial z}\frac{\partial z}{\partial r}$$

$$\frac{\partial A}{\partial s} = \frac{\partial A}{\partial x}\frac{\partial x}{\partial s} + \frac{\partial A}{\partial y}\frac{\partial y}{\partial s} + \frac{\partial A}{\partial z}\frac{\partial z}{\partial s} \tag{3.32}$$

$$\frac{\partial A}{\partial t} = \frac{\partial A}{\partial x}\frac{\partial x}{\partial t} + \frac{\partial A}{\partial y}\frac{\partial y}{\partial t} + \frac{\partial A}{\partial z}\frac{\partial z}{\partial t}$$

Or in the matrix form:

$$\begin{bmatrix} \frac{\partial A}{\partial r} \\ \frac{\partial A}{\partial s} \\ \frac{\partial A}{\partial t} \end{bmatrix} = \begin{bmatrix} \frac{\partial x}{\partial r} & \frac{\partial y}{\partial r} & \frac{\partial z}{\partial r} \\ \frac{\partial x}{\partial s} & \frac{\partial y}{\partial s} & \frac{\partial z}{\partial s} \\ \frac{\partial x}{\partial t} & \frac{\partial y}{\partial t} & \frac{\partial z}{\partial t} \end{bmatrix} \begin{bmatrix} \frac{\partial A}{\partial x} \\ \frac{\partial A}{\partial y} \\ \frac{\partial A}{\partial z} \end{bmatrix} \tag{3.33}$$

i.e.

$$\begin{bmatrix} \frac{\partial A}{\partial r} \\ \frac{\partial A}{\partial s} \\ \frac{\partial A}{\partial t} \end{bmatrix} = [J] \begin{bmatrix} \frac{\partial A}{\partial x} \\ \frac{\partial A}{\partial y} \\ \frac{\partial A}{\partial z} \end{bmatrix} \tag{3.34}$$

The Jocobian matrix can be found by using Eq.(3.1) to obtain:

$$[J] = \begin{bmatrix} \frac{\partial x}{\partial r} & \frac{\partial y}{\partial r} & \frac{\partial z}{\partial r} \\ \frac{\partial x}{\partial s} & \frac{\partial y}{\partial s} & \frac{\partial z}{\partial s} \\ \frac{\partial x}{\partial t} & \frac{\partial y}{\partial t} & \frac{\partial z}{\partial t} \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^{27} x_i \frac{\partial \hat{\phi}_i}{\partial r} & \sum_{i=1}^{27} y_i \frac{\partial \hat{\phi}_i}{\partial r} & \sum_{i=1}^{27} z_i \frac{\partial \hat{\phi}_i}{\partial r} \\ \sum_{i=1}^{27} x_i \frac{\partial \hat{\phi}_i}{\partial s} & \sum_{i=1}^{27} y_i \frac{\partial \hat{\phi}_i}{\partial s} & \sum_{i=1}^{27} z_i \frac{\partial \hat{\phi}_i}{\partial s} \\ \sum_{i=1}^{27} x_i \frac{\partial \hat{\phi}_i}{\partial t} & \sum_{i=1}^{27} y_i \frac{\partial \hat{\phi}_i}{\partial t} & \sum_{i=1}^{27} z_i \frac{\partial \hat{\phi}_i}{\partial t} \end{bmatrix} \tag{3.35}$$

in which the derivatives $\frac{\partial\hat{\phi}}{\partial r}$, $\frac{\partial\hat{\phi}}{\partial s}$ and $\frac{\partial\hat{\phi}}{\partial t}$ can be derived directly from Eq.(3.8). Note that the Jocobian matrix is defined at each node of the element. At different node, it can have different values in the matrix. It is required that the inverse of Jocobian matrix exist. Under the transformation between two Cartesian coordinate systems, this condition is always satisfied (Carmo, 1976; Coddington and Levinson, 1955). The inverse of Jocobian matrix is often needed in the calculation. By multiplying the inverse of Jocobian matrix to Eq. (3.34), it can be written:

$$\begin{bmatrix} \frac{\partial A}{\partial x} \\ \frac{\partial A}{\partial y} \\ \frac{\partial A}{\partial z} \end{bmatrix} = [J^{-1}] \begin{bmatrix} \frac{\partial A}{\partial r} \\ \frac{\partial A}{\partial s} \\ \frac{\partial A}{\partial t} \end{bmatrix} \tag{3.36}$$

In Eqs.(3.21a) - (3.24c), the derivatives can be written in the explicit forms:

$$\begin{bmatrix} \frac{\partial \phi_i}{\partial x} \\ \frac{\partial \phi_i}{\partial y} \\ \frac{\partial \phi_i}{\partial z} \end{bmatrix} = [J^{-1}] \begin{bmatrix} \frac{\partial \phi_i}{\partial r} \\ \frac{\partial \phi_i}{\partial s} \\ \frac{\partial \phi_i}{\partial t} \end{bmatrix} \tag{3.37}$$

$$\frac{\partial u}{\partial x} = \sum_{i=1}^{27} u_i \frac{\partial \phi_i}{\partial x}, \qquad \frac{\partial u}{\partial y} = \sum_{i=1}^{27} u_i \frac{\partial \phi_i}{\partial y}, \qquad \frac{\partial u}{\partial z} = \sum_{i=1}^{27} u_i \frac{\partial \phi_i}{\partial z} \tag{3.38}$$

$$\frac{\partial v}{\partial x} = \sum_{i=1}^{27} v_i \frac{\partial \phi_i}{\partial x}, \qquad \frac{\partial v}{\partial y} = \sum_{i=1}^{27} v_i \frac{\partial \phi_i}{\partial y}, \qquad \frac{\partial v}{\partial z} = \sum_{i=1}^{27} v_i \frac{\partial \phi_i}{\partial z} \tag{3.39}$$

$$\frac{\partial w}{\partial x} = \sum_{i=1}^{27} w_i \frac{\partial \phi_i}{\partial x}, \qquad \frac{\partial w}{\partial y} = \sum_{i=1}^{27} w_i \frac{\partial \phi_i}{\partial y}, \qquad \frac{\partial w}{\partial z} = \sum_{i=1}^{27} w_i \frac{\partial \phi_i}{\partial z} \tag{3.40}$$

which represent the discrete form of the derivatives.

The element matrices include 89 x 89 entries for the coefficient matrix, 89 unknowns including 81 velocity components and 8 pressure unknowns, and 89 entries for the force vector on the right hand side.

3.5 Assembly of the Element Equations

The element matrices obtained from Eq.(3.20) represent the parts of coefficients contributed by the element. To complete the formulation for the problem, each of the elements in the computational domain must be considered. Each entry of the element matrices will be combined according to the nodal numbering scheme of the element and the entire grid system. Within one element each node must have a unique different number as the local identification. This number is known as the *local nodal number.* Since one node can be shared by several elements, the same node may have different local nodal numbers in different elements. It is necessary to assign each node a unique, non-repeated number, or the *global nodal number* within the grid system. For each element, a table is generated to link the local nodal number to the global number for each nodal point. This

table will be used to determine the position in the assembled matrix for each entry of element matrices.

The numbering scheme of nodal points strongly affect the profile of the global matrix. A good scheme may result in a small band width of the assembled matrix and therefore saves the computation time. In the computation of three-dimensional finite element problem, there are very few rules available for the nodal numbering for the grid mesh in the literature. One rule applied in this study is that minimizing the difference of global nodal numbers between any two neighboring nodes. The other rule applied is that the difference between the element numbers of neighboring elements must be as small as possible. A typical element numbering method is shown in Figure 3.10. The first element of the grid mesh is located at the corner in the 8-th octant (i.e., the partition of space formed by all the points (x, y, z) with $x<0$, $y<0$ and $z<0$). The element number increases monotonically with increment one in the direction in which there are fewest number of elements in one row (or column) among the three directions. After filling up one row (or column) of elements, it then moves one place toward the direction which has the next fewest number of elements in one row (or column) and fills up another row (or column) of element in the first direction. Once one slice is filled up, the next immediate slice parallel to it will be numbered in the same way, and so on until all the elements are numbered.

The nodal numbering will be done elementwise. Since the local nodal numbering scheme is basically the same in each element as shown in

Figure 3.11, it is felt that the global numbering scheme should be built on the base of local numbering system. In such way, the resulted banded matrix will not have large band width. In the first element, the global nodal number will be the same as the local number for each nodal point. For the second element and thereafter, the global nodal numbers will be assigned to nodal points following the order of local numbers within each element. For the nodal points shared by several elements, the global numbers will be assigned at the first appearance of the node. Only one unique, non-repeated global number is assigned for each node in the grid system.

In the assembled coefficient matrices, each node reserves three rows and three columns of positions of entries for the three velocity components. One more row and column will be reserved for the pressure if the node is located at a corner of any element. Each element has 27 nodes for the velocity components and 8 nodes for pressures. Thus the coefficient matrix for the element must be an 89 x 89 matrix. The field variable vector for the element is given in an nodal pointwise sequence.

$$[\ \underbrace{u_1\ v_1\ w_1\ p_1}_{\text{local node 1}}\ \cdots\ \underbrace{u_4\ v_4\ w_4\ p_4}_{\text{local node 4}}\ \underbrace{u_5\ v_5\ w_5}_{\text{local node 5}}\ \cdots\ \underbrace{u_{27}\ v_{27}\ w_{27}}_{\text{local node 27}}\ ]^T \qquad (3.40)$$

local node 1 — global node n_1; local node 4 — global node n_4; local node 5 — global node n_5; local node 27 — global node n_{27}

On the other hand, in the global matrix the sequence of unknowns is also chosen to be nodal pointwise:

$$[\, u_1 \;\; v_1 \;\; w_1 \;\; p_1 \;\; \cdots \;\; u_j \;\; v_j \;\; w_j \;\; p_j \qquad u_k \;\; v_k \;\; w_k \;\; \cdots \;\; u_N \;\; v_N \;\; w_N]^T \quad (3.41)$$

global node 1	global node j, if j is at a corner of an element.	global node k, if k is not at corner of any element.	global number N

Each entry of the element coefficient matrix will be added to the proper row and column position in the global coefficient matrix during the assembly procedure. The complete form of the assembled coefficient matrix is shown in Figure 3.12. It takes the form of banded matrix. The profile or the band width of the matrix can be determined by the sequence of the field variables in the vector (3.41). Other possible sequence of unknowns in the vector is

$$[\, u_1 \;\; v_1 \;\; w_1 \;\; u_2 \;\; v_2 \;\; w_2 \;\; \cdots \;\; u_N \;\; v_N \;\; w_N \;\; p_1 \;\; p_2 \;\; \cdots \;\; p_M]^T \quad (3.42)$$

The sequence shown in (3.42) will result in a large band width of assembled matrix which is very time consuming to solve. The nodal pointwise sequence generates a banded matrix which has a much smaller profile and which can be solved more efficiently.

Boundary conditions:

Once the matrix is assembled, the field variables which have known values must be excluded from the equations. These known values are normally obtained from the boundary conditions. The boundary conditions for the current study belong to the type of Dirichlet boundary condition, i.e., the values of some variables are predetermined at the boundary. These includes: the velocity components at the channel bed and side walls are taken to be the same velocity of the boundary (which is zero for this case),

the velocity components at the far upstream and downstream cross sections of flow are taken to be the velocity distribution of undisturbed channel flow, and the pressure at the free surface is taken to be the atmospheric pressure. In the assembled equation, the corresponding columns for these boundary nodes will be subtracted from both sides of equations, and the rows for these nodes will be deleted from the matrix equations.

The banded matrix equation is stored in a packed form in the computer memory. Only the non-zero entries are stored. For each entry, the value is saved in an array along with an integer index number in another array which indicates the row and column position of the entry. The coefficient matrix is stored to a direct access disk file for later use.

3.6 Solution of the Systems of Equations

Among various approaches developed in the past centuries, Gaussian elimination method has been proved to be one of the most effective methods for solving systems of equations. For the current study, there are two aspects of considerations for the solution method of the assembled equations: the computer memory and the computing time required. For a typical case, the size of coefficient matrix ranges from 2500 to 5000 rows with band width ranging from 400 to 2200. The required computer core memory for each case ranges from 8 megabytes to 120 megabytes. The requirement well fits into the range of the memory capacity of the IBM-3090

300 series computer; however, some problem were encountered during the computation for the cases with row number larger than 4000. The values of variables were lost in the memory. The cause of the problem was not clear. It could be related to the VS FORTRAN compiler which was originally designed for FORTRAN programs which require less than 16 megabytes of memory. For any program requiring more than 16 megabytes of memory, the compiler stores the large arrays in the extension memory of computer which may be lost or interrupted by other unknown factor during the transmission. For the current study, due to the requirement of the large size of memory, the program can only be run under the batch facility on IBM which makes it difficult to trace the problem. It is felt that the memory requirement can be reduced by some way to avoid such problem. Based on this idea, the program was designed to reduce the required memory size during the computation. The coefficient matrix is stored on the disk at the begin of computation. The program reads only one row of entries into the core memory at a time. For the first row, each of the entries will be divided by the main diagonal entry; for the second row and thereafter, the row will be subtracted by the previous rows following the forward elimination procedure of Gaussian elimination method. Each row will then be divided by its main diagonal entry. By discarding the eliminated entries from each row right after the forward elimination, it can reduce the size of array by one-half; i.e., the required computer memory size is reduced by 50 percent. One other method, the Gaussian-Legendre method, can also reduce the memory size by eliminating the non-diagonal entries from each row during the procedure of forward elimination, but it

requires 50 percent more of computation time than the Gaussian elimination method.

After the forward elimination, the backward substitution will be proceeded as the regular way of Gaussian elimination. The result will be stored in the force vector on the right hand side of the equations.

3.7 Postprocessor

The result stored in the force vector will be rearranged to obtain the velocity components and pressures at nodal points of grid. The components of stresses in the x-, y- and z- directions at each node on the surface of sphere can be calculated according to the formulas:

$$\sigma_x = -p + 2\mu \frac{\partial u}{\partial x}$$

$$\sigma_y = -p + 2\mu \frac{\partial v}{\partial y} \qquad (3.43)$$

$$\sigma_z = -p + 2\mu \frac{\partial w}{\partial z}$$

and

$$\tau_{xy} = \tau_{yx} = \mu \left(\frac{\partial v}{\partial x} + \frac{\partial u}{\partial y} \right)$$

$$\tau_{yz} = \tau_{zy} = \mu \left(\frac{\partial w}{\partial y} + \frac{\partial v}{\partial z} \right) \qquad (3.44)$$

$$\tau_{zx} = \tau_{xz} = \mu \left(\frac{\partial u}{\partial z} + \frac{\partial w}{\partial x} \right)$$

where σ_x , σ_y , σ_z = normal stresses (Figure 3.13),

τ_{xy} , τ_{yz} , τ_{zx} = shear stresses (Figure 3.13).

To find the total drag and lift forces, these stress components must be evaluated at each node on the surface of sphere and then integrated over the surface area to obtain the forces. The following expressions represent the contributing components to the total forces by these stresses at any point (x,y,z) on the surface of sphere (Figure 3.14).

$$L_i = \sigma_z \Delta A_z + \tau_{xy}\Delta A_x + \tau_{yz}\Delta A_y \tag{3.45}$$

$$D_i = \sigma_x \Delta A_x + \tau_{zx}\Delta A_z + \tau_{yx}\Delta A_y \tag{3.46}$$

where L_i = lift component contributed by the i-th element,

D_i = drag component contributed by the ith element,

ΔA_x , ΔA_y , and ΔA_z = the projection of the area of the element on the x- , y- and z-plane, respectively.

The total forces of drag and lift can be obtained by the summations:

$$L = \sum L_i = \sum (\sigma_z \Delta A_z + \tau_{xy}\Delta A_x + \tau_{yz}\Delta A_y) \tag{3.47}$$

$$D = \sum D_i = \sum (\sigma_x \Delta A_x + \tau_{zx}\Delta A_z + \tau_{yx}\Delta A_y) \tag{3.48}$$

Chapter IV
Computer Programming

4.1 Basic Concepts

The basic concepts associated with the computer programming of the finite element approach for the present study are summarized in the flow chart as shown in Figure 4.1. There are six fundamental parts in the structure of the computer program with the following functions:

(1) Grid generator

(2) Defining the boundary conditions

(3) Input of the velocity distributions for the initial guess of the iteration

(4) Calculation of the element matrix

(5) Assembly of global matrix

(6) The matrix equation solver.

Each part consists of one or more subroutines written in FORTRAN 77 language. The general structure of the program is shown in Figure 4.2. All the relevant routines are called from the main routine. The required input data are read in by the main routine. A description will be given for each of the subroutine. The program is list in the Appendix C.

4.2 Subroutines

4.2.1 Subroutine GRID

The procedure of dividing the entire computational domain into elements lays the foundation for the structure of finite element method. The computational domain includes a sphere located in the channel flow (Figure 3.1). The coordinate reference is chosen to be a rectangular Cartesian coordinate system with origin fixed at the center of sphere, x-axis pointing into the flow direction, y-axis pointing away from the origin horizontally toward a vertical wall of channel, and z-axis pointing upward vertically (Figure 3.1). The delineation starts with the sphere itself. Because of the curved surface on the sphere, it is difficult to generate the grid mesh directly. Improper dividing may result in some non-hexahedral element near the surface of sphere. Figure 4.3 shows a cross-section of the mesh where some elements with other shape occur. With the different

shapes of elements in the grid system, it will require different set of shape functions for these elements in the computation. To ensure the proper division of the grid system, some simple procedure is added to the delineation. It is observed that the problem would be easier to handle if the sphere were replaced by a cube. Thus a cube internally connected to the sphere is selected as the reference for the delineation (Figure 4.4). The intersection of the cube and the sphere consists of eight points, i.e., the eight vertices of the cube. These points are used as the reference points for the discretization of the domain. Figure 4.5 shows a cross-section of the mesh with the cube at the center. All the lines are chosen to be parallel to one of the coordinate axes. A finer division is selected for the region close to the surface of sphere. The cube is later replaced by the sphere in the grid mesh with the boundary points moving outward from the surface of cube to that of sphere. The entire procedure is really straight forward and is built in the subroutine GRID. After the completion of the delineation of hexahedral elements, the subroutine INDTA and INTDB are called to locate all the interior nodes for each element. For each element with eight known vertices, the interior nodes, i.e., the nodes not at corners, are given by the following relations (Figure 4.6):

$$\underset{\sim}{x}_9 = \frac{1}{2}(\underset{\sim}{x}_1 + \underset{\sim}{x}_2) \quad \underset{\sim}{x}_{10} = \frac{1}{2}(\underset{\sim}{x}_2 + \underset{\sim}{x}_3) \quad \underset{\sim}{x}_{11} = \frac{1}{2}(\underset{\sim}{x}_3 + \underset{\sim}{x}_4)$$

$$\underset{\sim}{x}_{12} = \frac{1}{2}(\underset{\sim}{x}_1 + \underset{\sim}{x}_4) \quad \underset{\sim}{x}_{13} = \frac{1}{4}(\underset{\sim}{x}_9 + \underset{\sim}{x}_{10} + \underset{\sim}{x}_{11} + \underset{\sim}{x}_{12})$$

$$\underset{\sim}{x}_{14} = \frac{1}{2}(\underset{\sim}{x}_5 + \underset{\sim}{x}_6) \quad \underset{\sim}{x}_{15} = \frac{1}{2}(\underset{\sim}{x}_6 + \underset{\sim}{x}_7) \quad \underset{\sim}{x}_{16} = \frac{1}{2}(\underset{\sim}{x}_7 + \underset{\sim}{x}_8)$$

$$\underset{\sim}{x}_{17} = \frac{1}{2}(\underset{\sim}{x}_5 + \underset{\sim}{x}_8) \quad \underset{\sim}{x}_{18} = \frac{1}{4}(\underset{\sim}{x}_{14} + \underset{\sim}{x}_{15} + \underset{\sim}{x}_{16} + \underset{\sim}{x}_{17}) \qquad (4.1)$$

$$\underset{\sim}{x}_{19} = \frac{1}{2}(\underset{\sim}{x}_5 + \underset{\sim}{x}_1) \quad \underset{\sim}{x}_{20} = \frac{1}{2}(\underset{\sim}{x}_2 + \underset{\sim}{x}_6) \quad \underset{\sim}{x}_{21} = \frac{1}{2}(\underset{\sim}{x}_3 + \underset{\sim}{x}_7)$$

$$\underset{\sim}{x}_{22} = \frac{1}{2}(\underset{\sim}{x}_4 + \underset{\sim}{x}_8) \quad \underset{\sim}{x}_{23} = \frac{1}{2}(\underset{\sim}{x}_{19} + \underset{\sim}{x}_{20}) \quad \underset{\sim}{x}_{24} = \frac{1}{2}(\underset{\sim}{x}_{20} + \underset{\sim}{x}_{21})$$

$$\underset{\sim}{x}_{25} = \frac{1}{2}(\underset{\sim}{x}_{21} + \underset{\sim}{x}_{22}) \quad \underset{\sim}{x}_{26} = \frac{1}{2}(\underset{\sim}{x}_{19} + \underset{\sim}{x}_{22})$$

$$\underset{\sim}{x}_{27} = \frac{1}{4}(\underset{\sim}{x}_{23} + \underset{\sim}{x}_{24} + \underset{\sim}{x}_{25} + \underset{\sim}{x}_{26})$$

Each node in the element has a global nodal number as well as a local nodal number within the element. The schemes for nodal numbering and element numbering were described in the previous chapter. The output of these subroutine includes the tables of global nodal number for each element, the coordinates for each nodal point, and properties for each nodal point which show whether the node is a boundary point of the region, a vertex point or interior points of an element.

4.2.2 Subroutine INITBD

This subroutine performs all of the calculations for the velocity profile at the boundary. The velocity profile is based on the user supplied profile or the log-profile if there is no profile provided. For cases studied, the velocity profiles are obtained from the observation reported in the same literature. The log-profile is built in as an option according to the relation given by Nezu and Rodi (1986).

For the user supplied profiles, interpolation is usually needed to obtain the values in between of any two given points of data. The subroutine BSPLN is used to interpolate the velocity components with the Newton forward or backward interpolating polynomials for nonuniform space. Other interpolation methods such as the spline method and higher degree polynomial methods were carefully examined in this study. For some cases, the oscillation became a serious problem for those methods. The Newton's difference method showed a better behavior than other methods.

4.2.3 Subroutine BASIS

This subroutine computes the basis functions for each element according to Eqs (3.8) and (3.15). It also evaluates the values of the derivatives of the basis functions at each nodal point within an element. The Jocobian matrices which are essential in the coordinate transformation and the computation of derivatives and integrals are determined by this subroutine as well. For each element, all of the basis functions are based on the ones for the reference element, the unit cube. A transformation from the reference element to each element is performed in the subroutine for these basis functions. The mapping is expressed in Eq.(3.13). All of the outputs from this subroutine are stored in the computer memory for the later use for the calculation of element matrices.

4.2.4 Subroutine ELEMENT

This subroutine extracts all the necessary data from the results of GRID and INITBD for each element. The required data include the velocity components from the previous iteration or initial guess, the coordinates of each nodal point within the element, and the basis functions computed from the subroutine BASIS. The major function of this subroutine is to calculate the element matrix for each element which plays the important role for the calculation of coefficient matrices. Most of the coefficients of the governing equations require the calculation of derivatives and the integration over each element. The derivatives are obtained from the result of subroutine BASIS. The numerical integration is based on Eqs.(3.30) or (3.31). Both the derivatives and the integrations are evaluated on the reference element; therefore, a transformation is necessary to obtain the values for the element. The transformation is based upon Equations (3.29) and (3.33). The subroutine SURINF is used to calculate the surface integrals for elements located at the boundary of domain. The subroutine TFACE is used to find the element surfaces which are located at the boundary of domain. The results of this subroutine are stored in a two-dimensional array which will be used to assemble the global matrix.

4.2.5 Subroutine ASSEM

All the element matrices calculated by ELEMENT will be assembled to form the global matrix according to the method described in Chapter 3. Each entry in the element matrices will be added to its corresponding position in the global matrix during the procedure of assembly. The assembled matrix is stored in two arrays, one stores the values of entries and another stores the position of the entries in matrix.

4.2.6 Subroutine STORE

The assembled banded matrix and force vector are stored in compact forms. In order to save the disk space and reduce the truncation error, the file is stored in binary format. It is stored in the form of machine code and can be accessed quickly for the input and output. Other data includes the nodal coordinates, element-node relation array, boundary conditions and nodal information are stored in another file for the calculation of next iteration.

4.2.7 Program SOLVER

This subroutine utilizes the subroutine BAND to solve the banded matrix equation. The Gaussian elimination method as described in Chapter 3 is included in subroutine BAND. The calculation can be divided into two

parts: forward elimination and backward substitution. Both are already described in the previous chapter. Since the matrix is in a spare banded form, the program will automatically skip the rows and column of entries which has values of zero and save the computing time. The convergent test for the computation is built in this program. After each iteration, the program will determine the differences of the variables between the current and preceding iterations. The absolute values of deviations in velocity components and pressures will be summed up and divided by the total number of unknowns. The mean value will be used for the convergence test with a user supplied tolerance.

4.2.8 Program PROFILE

This subroutine reads the output from the program SOLVER and arranges all the entries in the force vector to obtain the values for velocity components and pressure at each nodal point. The lift and drag forces are evaluated for the sphere by using the Equations (3.43) - (3.47). The velocity profiles can be found by this subroutine as well. The results are stored in a disk file which will be used for plotting and analysis.

Chapter V
Results

5.1 Numerical Input

Twenty-four numerical experiments are presented,

(a) Fourteen cases for the first free stream velocity profile, profile A, as shown in Figure 5.1 with gap ratios of 0.1, 0.25, 0.4, 0.5, 0.75, 1, 1.25, 1.5, 1.75, 2, 2.5, 3, 3.5 and 4.

(b) Five cases for the second free stream velocity profile, profile B, with gap ratios of 0.5, 1, 2, 3 and 4.

(c) Five cases for the third free stream velocity profile, profile C, with gap ratios of 0.5, 1, 2, 3 and 4.

The output of these cases are to be compared with the experimental measurements by Willetts and Naddeh (1986). For each case, the input data reflect the values of parameters used in their laboratory experiment. These data are listed in Table 5.1 for each case. As a requirement of the solution

procedure for the problem, the boundary conditions must be given at all of the boundaries of flow domain. These boundary conditions include: the upstream and downstream velocity profiles, no-slip conditions at the side walls and the channel bed, and the atmospheric pressure at the free surface of flow. Also, the physical properties of fluid such as viscosity and density must be specified in the input data for the calculation of parameters in the governing equations. Mineral oil was used in Willetts and Naddeh's experiment (1986) and the properties of the fluid were not given in their paper, the liquid properties of mineral oil were obtained from literatures (Daily, 1966; White, 1991). The temperature was reported to be 20°C throughout the experiment. Therefore, the liquid properties are considered to be constant in the computations for all of the cases.

The computational domain is taken to be one reach of the channel with the dimension of 40 cm long x 10 cm wide x 20 cm high with the sphere (2 cm diameter) being located at the mid-length of the channel. For most of the cases, the grid mesh consists of 17 nodes in x-direction, 13 nodes in y-direction, and 15 nodes in z-direction. In some cases, due to the limitations of geometric and flow conditions, it is necessary to adjust the grid system for computation. The number of nodes in each coordinate direction will be slightly different from the one described above. In general, this grid system is considered not fine enough for the calculation of the vortex shedding or wake fluctuation; however, the refinement will result in dramatical increase both in the computer memory requirement and the computing cost. For the current grid system, the computer memory requirement already reaches the size of 120 megabytes. For better results, the number of

grid nodes has to be doubled or even more in each direction, which will require eight or more times of computer memory for the computation. The computing cost will also increase proportionally for the refined grid system. It should be noted here that the present study focus on the low Reynolds numbers within which the wake and vortex shedding phenomena do not normally occur. For this reason, the grid system is considered to be fine enough. This can be seen from the comparison of numerical results to the experimental results to be discussed next.

5.2 Results and Discussions

The results provided by the numerical model include the lift and drag coefficients for the sphere, velocity and pressure distributions in the flow domain. The lift and drag coefficients are defined by the following expressions.

$$C_L = \frac{L}{\frac{1}{2}\rho A v^2} \tag{5.1}$$

$$C_D = \frac{D}{\frac{1}{2}\rho A v^2} \tag{5.2}$$

where L = lift force,

D = drag force,

ρ = density of fluid,

A = aspect area of sphere = $\frac{\pi}{4} D_s^2$,

D_s = diameter of sphere,

v = free stream velocity along the central line of sphere.

The computed lift coefficients are listed in Table 5.2 for each case along with the laboratory measured values obtained by Willetts and Naddeh (1986). In Table 5.2, the first 14 cases are for velocity profile A; cases 15 - 19 are for velocity profile B; and cases 20 - 24 are for velocity profile C. Figure 5.2 shows the relationship between the lift coefficients and the gap ratios. In the figure, one pair of simulated and measured curves are shown for each velocity profile. Same scale is used for all of the velocity profiles. As can be seen from Table 5.2 and Figure 5.2, the simulated values are very close to the measured values for most of the cases. The results from the numerical model agree with the experiment study in terms of the relationship between the lift force and the gap ratios. It shows that, for velocity profile A, the lift forces are positive (upward) for the gap ratios less than 0.7 and are negative for the gap ratios greater than 0.7. It decreases monotonically from the maximum positive values to zero as the gap ratio increases from 0 to 0.7, then increases steadily in the negative direction as gap ratio increases from 0.7 to 2. It reaches the maximum negative value at gap ratio of 1.75 to 2, then reaches zero at gap ratio of 4. The maximum positive lift coefficient occurs at the gap ratio near zero (within the simulated result) and the

extreme negative lift coefficient occurs at gap ratio of 1.75 to 2. It can be seen that the absolute maximum positive value is almost as twice as much of the absolute maximum negative value. Similar results can be observed for the other two velocity profiles. For both of the velocity profiles B and C, the lift coefficient changes sign at gap ratio around 0.75 which is about the same as the case of the velocity profile A. The maximum negative lift coefficients also occur at gap ratio of 2 for both velocity profiles B and C. The lift forces also remain negative for gap ratios larger than 0.75 in these two cases. In the same figure, one can observe an interesting phenomenon. For each fixed gap ratio, the magnitude of lift coefficient exerting on the same sphere decreases as the Reynolds number increases. In other words, the magnitude of the lift coefficient on the sphere decreases as the upstream velocity increases from profile A to B, from profile B to C, or from A to C. This can be explained as follows. The thickness of wall boundary layer at the channel bed is known to decrease as the magnitude of the velocity of the upstream flow increases (Figure 5.1). As the thickness of the wall boundary layer decreases, the sphere will have less part immersed in the wall boundary layer and more part immersed in the free stream flow. Thus the wall boundary layer will have less effect on the sphere and the sphere is more affected by the free stream flow. For a sphere completely immersed in the free steam flow, the lift force exerting on the sphere is normally negligible. As the thickness of boundary layer decreases, the sphere approaches to the conditions of free stream flow gradually. Thus, the magnitude of the lift force decreases as a result.

The computed drag coefficients have not been compared with the measurements due to the unavailability of measurements reported in the same literature. Figure 5.3 shows the relationship between the drag coefficients and the gap ratios. It indicates that there is a tendency of decrease in drag coefficient as the gap ratio increases. For the velocity profile A, the maximum drag coefficient occurs at the gap ratio near zero. It decreases monotonically from the maximum value to the minimum value as the gap ratio increases. For velocity profiles B and C, similar results can be observed from Figure 5.3. It is interesting to note that the drag coefficient decreases as the Reynolds number increases. For the same gap ratio, the drag coefficient decreases as the upstream velocity increase from profile A to B, from profile B to C, or from profile A to C. This is very similar to the phenomena observed in the case of lift coefficient. Figures 5.4 - 5.19 show the velocity distribution for two vertical planes in the longitudinal direction at y=0 and y=0.5D_s , and two horizontal planes at z=0 and z=0.5D_s for each known undisturbed upstream velocity profile. For velocity profile A, cases with gap ratios of 0.1, 0.5, 1, 2, 3 and 4 are shown in these figures. For both velocity profiles B and C, cases with gap ratios of 0.5, 1, 2, 3 and 4 are shown in these figures. The corresponding stress distributions are shown in Figures 5.20 - 5.35. Each figure includes the distribution of vertical component of stress for a vertical plane passing through the center of the sphere in x-direction and the distribution of the longitudinal component of stress for the same plane. In these figures, the vertical component of stress at any point on the plane of sphere is represented by an arrow tailed at the position of the point of interest with tip

pointing upward or downward. The length of the arrow is the magnitude of the stress at that point and the direction of the arrow indicates the direction of the stress. Similar arrow is used to represent the longitudinal component of stress at any point on the plane of the sphere except that the arrow is pointing to left or right horizontally from the position of the point. Same as above, the direction of the arrow is the direction of the stress which constitutes the drag force. The unit indicated in the figures is in kg/m^2. These vertical or longitudinal components include the contribution from both the viscous stress and pressure in the flow field.

As can be seen from Figures 5.4 to 5.19, for all three approaching velocity profiles, there is no large scale of vortex or separation of flow occurred in the flow field which indicates that the flow maintains laminar flow pattern for all the cases. This confirms with the laboratory observation as by Willetts and Naddeh (1986) in which the flow was reported to maintain laminar for all cases. For each case, the upstream stagnation zone is displaced slightly away from the channel bed especially for cases with small gap ratios. This displacement of stagnation zone is caused by the presence of channel bed which blocks the flow toward the lower half of sphere. This effect becomes negligible as the sphere moves away from the channel bed. The blockage of the flow by the channel bed can be related to the positive lift forces for cases with small gap ratios. Since the flow is blocked by the channel bed, it has to change its direction. The vertical component of flow has to change from downward to upward or horizontal for the flow in the proximity of channel bed. As a result, the flow loses its downward component of momentum in this region and causes an upward

lift force on the sphere. This phenomenon can be seen in Figures 5.20 to 5.35 in which the distributions of vertical components of stresses show with larger values for the lower hemisphere of the sphere for cases with small gap ratios. The blockage effect gradually decays as the sphere moves away from the channel bed.

In figure 5.4 - 5.19, the flow downstream of the sphere shows a weak wake pattern. For small gap ratios, the downstream vortex from the sphere is strongly interfered by the channel bed (Figures 5.4, 5.5, 5.10 and 5.15). The flow behind the sphere is influenced by the wall boundary layer and changes its direction or decays within the wall proximity for the cases with small gap ratios. For profile A, which has small free-stream velocity, the wake behind the sphere is relatively weak and is dominated by the wall boundary layer (Figures 5.4 and 5.5). For velocity profiles B and C, which have higher free-stream velocities, the wake behind the sphere is stronger and is less affected by the wall boundary layer (Figures 5.10 and 5.15). For large gap ratios, the wall effects on the sphere becomes less significant. As gap ratio increases, the sphere moves away from the wall. The wake behind the sphere is gradually separated from the wall boundary layer (Figures 5.6 - 5.9, 5.11 - 5.14, and 5.16 - 5.19). The flow around the sphere gradually restores the free-shear flow pattern as the gap ratio increases. To the extreme cases (Figure 5.9, 5.14, and 5.19), the wake almost completely stay away from the wall boundary layer. For these cases, the flow around the sphere is very close to the free-shear flow pattern.

For small gap ratios, the lift force acting on the sphere is due primarily to the wall effect. As the sphere moves away from the channel bed, the wall effect becomes less and less significant. For large gap ratios, the lift force on the sphere is mainly caused by the velocity gradient in the free-stream flow. However, at the region far away from the channel bed, the velocity distribution is normally close to the uniform flow distribution; i.e., the velocity gradient is near zero. Therefore, the lift forces are expected to be small for cases with large gap ratios. As can be seen from Figure 5.2, the lift forces become smaller in magnitude at large gap ratios.

Figures 5.20(a) - 5.25(a) show the distribution of the vertical component of stress, which constitutes the lift force, for one vertical cross-section of the sphere for velocity profile A. For gap ratio of 0.1 (Figure 5.20(a)), the vertical component of stress at the lower hemisphere is larger than that of the upper hemisphere. As the gap ratio increases, the distribution of the vertical stress becomes more symmetric with respect to the horizontal plane passing through the center of the sphere (Figures 5.21(a) - 5.25(a)). Figures 5.26(a) - 5.30(a) and Figures 5.31(a) - 5.35(a) show the distribution of the vertical component of stress at one vertical cross-section of the sphere for velocity profiles B and C, respectively. From these figures, it also can be seen, for both velocity profiles B and C, that the distribution approaches symmetric pattern with respect to the horizontal plane passing through the center of the sphere as the gap ratio increases.

From Figure 5.2, the absolute magnitude of lift coefficients for approaching velocity profiles B and C are smaller than that of velocity profile A. As can

be observed from Figure 5.1, the thickness of the wall boundary layer for velocity profile A is larger than those of the velocity profiles B and C. Thus, at the wall proximity, the wall effect for velocity profile A is larger than the wall effects for the velocity profiles B and C. Hence, the magnitude of the lift coefficient for the velocity profile A is larger than those of the velocity profiles B and C at the wall proximity. On the other hand, at the region far away from the channel bed, the wall effect becomes insignificant. The lift force is due primarily to the velocity gradient of upstream flow. Since the velocity gradients for velocity profiles B and C are smaller than the gradient of velocity profile A (Figure 5.1), the magnitudes of lift forces on sphere are smaller for velocity profiles B and C than that of velocity profile A. As shown in Figure 5.2, the lift coefficient has smaller magnitude for velocity profiles B and C than that of velocity profile A for a given gap ratio. This evidence also supports the discussion presented in the previous paragraph.

In Tables 5.1 and 5.2, cases with $Re \leq 44$ are the results from the iteration procedure after the number of iterations shown in Table 5.2. These numbers of iterations are the number of loops in the Newton iteration. The criteria used for convergence was fixed as 0.0005. The cases with $Re > 44$ are the results of the linear solutions of the partial differential equations where the non-linear (Newton) iteration failed to converge. The linear solutions are obtained by solving the linear part of the partial differential equations. The finite element method as described in Chapter III is applied to solve these linear equations. The grid system remains the same for both of the linear and non-linear solutions. In the iteration procedure, these linear solutions are used as the initial guesses for the iteration.

Mathematically, the non-linear solutions can be obtained by adding the correctors to the linear solutions as described in Chapter III. The Newton iteration method is applied to obtain the correctors for the non-linear solution. For cases with Reynolds numbers greater than 44, the iteration does not yield the solution. The divergence could be caused by the downstream flow instability which becomes unsteady as the upstream flow velocity increases. Other possible reasons include the initial values guessed for the Newton iteration method and the coarse grid size for the computation. The Newton iteration becomes more sensitive to the initial guess as the Reynolds number increases (Engelman, et al., 1981). It can easily diverge for some initial guesses even though they are good guesses for the cases with low Reynolds numbers. Since there is no simple way to assess how good the initial guess is for this problem, it is not easy to modify the initial guess if the iteration diverges. This presents the major difficulty to solve this type of problem. Of course, the size of the grid system could be another cause for the divergence. A finer grid system should yield more satisfactory results. It may improve the smoothness of the velocity distribution within the grid. With a coarse grid, sudden change of the velocity distribution may occur which could result in the violation of continuity principle in some case. With the current grid system adapted, the computer program already required the memory close to the available capacity of the existing computing facility in this Institution. The further refinement of the grid system will increase the requirement of memory size by at least one order of magnitude.

5.3 Parametric study

For the problem studied, the size of the sphere also plays an important role for the lift coefficient. The diameter of sphere has been used to non-dimensionlize the variables in the governing equations. The particle Reynolds number and the gap ratio have included the diameter of the sphere for this study. Therefore, the effect of particle size has been demonstrated. However, it is not quite clear from the previous discussion how significant the particle size is on the lift coefficient. In fact, the interaction between the flow and the sphere is governed by the size of the sphere. Large size sphere would create the bigger blockage effect on the flow field while small size sphere would be easily immersed in the wall boundary layer. From Equation (5.1), the size of sphere is one of the factors used to calculate the lift coefficient. Therefore, it is decided to examine more in detail the effect of the size of sphere on the lift coefficient.

Twelve cases presented for this part of study are four different sizes of diameters for each of the three fixed gap ratios 0.5, 1 and 3. For each case, only the velocity profile A shown in Figure 5.1 is used as the approaching velocity profile. Since the cases with large gap ratios, according to the results discussed in the previous sections, have very small magnitude of lift coefficients, it is felt that the effect may not be clear for these cases. Hence, only the small and medium gap ratios are chosen for this particular investigation. The diameters chosen are 0.5 cm, 1 cm, 2 cm and 3 cm. The

third diameter, i.e., the diameter of 2 cm is the same as the diameter used in the previous studies.

The results are shown in Tables 5.3 - 5.5 and Figure 5.36. The lift coefficient decreases as the diameter of sphere increases for the constant gap ratio of 0.5 (Table 5.3 and Figure 5.36). The lift forces are all positive (upward) for each of the sizes. It can be seen that, for the gap ratio of 0.5, the smallest sphere is the most affected by the wall effect associated the channel bed than the other cases. The largest sphere is least affected by the wall boundary layer among all of the four sizes of spheres. For the gap ratio of 1, the lift coefficients are all negative for all four sizes of the spheres. The lift coefficient increases in the negative (downward) direction as the diameter of sphere increases. For the gap ratio of 3, the lift coefficients are also all negative. The lift coefficient increases steadily in the negative direction as the diameter of sphere increases. From Figure 5.36, it can be observed that, for each fixed diameter of sphere, the absolute value of the lift coefficient for the gap ratio of 1 is larger than that of the gap ratio of 3, and the lift coefficient for the gap ratio of 0.5 are all positive. These basically agree with the result shown in Figure 5.2 for the corresponding gap ratios.

On the other hand, it can be concluded from the results shown in Figure 5.36 and Tables 5.3 - 5.5 that the small particles have larger lift coefficients than those of the large particles near the wall. In the region far away from the wall, the large particles have larger negative lift coefficients than those of the small particles. This may help explaining why the small particles are easy to suspend in flows for the sediment transport problem. It is also

known that small particles in a flow are not easy to settle. As pointed out in the first chapter, most of the study for the dislodging of sediment particle are limited to the concepts of critical shear or critical velocity. The relationship among the lift coefficient, approaching velocity profile, sphere size, and gap ratio should provide some insights to the problem.

The side wall of the channel may have some effect on the sphere for some cases with narrow channel or large sized particle. As long as the channel is wide (say 10 times of the water depth), the side wall effect is insignificant on the lift and drag. In the extreme case in which the particle immersed in the side wall boundary layer, there will be side wall effect. In the present study, this is not the case.

5.4 Future study

The computer model developed for the present study has been shown to be able to simulate the lift coefficient for the sphere located completely or partially within a wall boundary layer. The numerical results agree quite well with the experimental measurements reported in the literature. However, there are some limitations to this computer model which needs improvements in the future. For example, the refinement of grid system should improve the results of the current model if larger computer facility is available. For future study, the present model needs to be extended to include:

(a). Unsteady flow. The governing equations presented in this study do not include the unsteady term. This prohibits the model from the calculation of the downstream unsteady flow caused by the flow separation for high Reynolds numbers. With the addition of the unsteady term, the model should be able to perform the time march for unsteady flow simulation.

(b). Dynamic simulation of the particle motion. The sphere is not allowed to rotate or drift in the flow for the present study. The movement of the sphere will affect the flow pattern and change the lift coefficient. With this improvement, the model can simulate the successive movement of the sphere in a flow near the channel bed.

(c). Inclusion of better iteration scheme with *wider range* and *faster speed* of convergence. The Newton iteration is limited by the small range (or radius) of convergence. Other iteration method may have wide range of convergence but very slow speed of convergence. For future applications, a better iteration scheme which combines all the advantages from each of the iteration methods should be explored.

(d). Turbulent flow. The current model can only handle the laminar flows. For high Reynolds numbers, the turbulent flow model will be necessary to properly simulate the flow separation and vortex shedding.

(e). Wavy channel bed. The current model only addresses the flat channel bed. For real channels, the wavy bed forms play an important role in the sediment bed-load transport process.

(f). Shallow channel flow. The vertical velocities vary significantly with respect to the water depth in a channel. Therefore, the velocity gradients change accordingly. The lift and drag forces should be investigated for the cases with particle partially submerged in the flow and particle near the free surface. In both cases, surface wave will play an important role.

Chapter VI
Conclusions

The lift and drag forces play important roles in the particle dislodging process of the bed-load sediment transport in open channels. The lift and drag forces exerting on a spherical particle in a shear flow near the channel bed have been calculated by solving the Navier-Stokes equations numerically. A numerical technique which combines the finite element method and the iteration procedure has been applied to solve the non-linear Navier-Stokes equations. The technique first linearizes the non-linear terms in the partial differential equations, then solves the linearized equations by the finite element method. The iteration procedure requires initial guesses for the solutions. The linear solutions of the partial differential equations are used for the initial guesses. The finite element method as described in Chapter III has been applied to solve the linearized Navier-Stokes equations. The tri-quadratic hexahedron is chosen to discretize the computational domain. The tri-quadratic interpolation functions are used to discretize the governing equations. Then the element matrix is calculated for each finite-element. These element matrices are later assembled to obtain the global matrix. The global matrix is in banded form which are solved by a banded matrix solver.

The particle Reynolds numbers for the cases studies are less than 134. Previous study shows that steady flow can be maintained up to particle Reynolds number of 175 (Kim and Pearlstein, 1990). Thus only steady flow is considered in this study.

The numerical results from the model developed are compared to the experimental results conducted by Willetts and Naddeh (1986). As shown in Figure 5.2, the calculated lift coefficients agree quite well with the measured values in the laboratory for various gap ratios. The results show that, for the velocity profile A (Figure 5.1), the lift coefficients are positive for the gap ratios less than 0.7 and are negative for the gap ratios greater than 0.7. It decreases monotonically from the maximum positive values to zero as the gap ratio increase from zero to 0.7, then increases steadily in the negative direction as the gap ratio increases from 0.7 to 2. As the gap ratio increases from 2 to 4, the lift coefficient decreases to zero gradually. For both the velocity profiles B and C, the lift coefficient change from positive to negative value at gap ratio around 0.75. The results are similar to the case of velocity profile A. However, the absolute magnitude of the lift coefficient (on the sphere at a fixed gap ratio) decreases as the upstream velocity increases from profile A to B, from profile B to C, or from profile A to C. In other words, for a given gap ratio, the absolute value of the lift coefficient decreases as the Reynolds number increases. The lift coefficient is governed by two major factors: the wall effect and the velocity gradient of the approaching flow. For small gap ratios, the wall effect dominates. As the gap ratio increases, the wall effect on the sphere decreases. In the

region far away from the wall, the velocity gradients are normally small; therefore, the magnitude of the lift forces are small.

The velocity distributions show that the stagnation zone is displaced upward for cases with small gap ratios. This is due to the blockage effect between the sphere and the channel bed. The wall effect on the lift coefficient is also caused by the blockage of flow in the region between the sphere and the channel bed.

The distributions of the vertical component of stress at vertical planes (Figures 5.20(a) - 5.35(a)) show that the vertical component of stress at the lower hemisphere larger than that of the upper hemisphere for small gap ratios. The distribution becomes symmetric with respect to the horizontal plane passing through the center of the sphere as the gap ratio increases.

The diameter of the sphere also plays an important role in the determination of the lift coefficient. For the gap ratio of 0.5, the lift coefficient decreases as the diameter increases. For gap ratios of 1 and 3, the lift coefficient increases in the negative direction as the diameter increases. For the gap ratio of 0.5, the lift coefficient maintains the positive value for all the diameters of spheres studied. For gap ratios of 1 and 3, the lift coefficient becomes negative for all the diameters of spheres studied. The absolute value of the magnitude of the lift coefficient for the gap ratio of 3 is smaller than that of the gap ratio of 1 for all the diameters.

The numerical method used in this study does encounter some difficulties. The main problem is the divergence in the iteration procedure. It has been

concluded in the current study that the Newton iteration method requires a good initial guess. The iteration becomes more sensitive to the initial guess as the Reynolds number increases. The size of grid system may also be one of the factors to cause the divergence. Finer grid system should be adapted if the computing facility permits.

For the future study, the computer model can be extended to include: (a) unsteady flow, (b) dynamic simulation of the particle motion, (c) a better iteration scheme, (d) turbulent flow, (e) wavy channel bed, and (f) shallow channel flow.

References

Abbot, M. (1979). *Computational hydraulics*, Pitman.

Aksoy, S. (1973). Fluid force acting on a sphere near a solid boundary. *5th Cong.* IAHR, Istanbul, Turkey, Vol.1, Paper A29.

Argyris, J. H. (1955). Energy theorems and structural analysis. *Aircraft Eng.*, Vol. 27, Febr.-March-April-May.

ASCE Task Committee (1975). *Sedimentation engineering*. V.A. Vanoni, ed., ASCE M & R, No. 54.

Atkinson, K. E. (1978). *An introduction to numerical analysis*, John Wiley and sons, New York.

Bagnold, R. A. (1974). Fluid forces on a body in shear flow; experimental use of stationary flow. *Proc. Roy. Soc.* London, A 340, 147-171.

Benedict, B. C. and B. A. Christensen (1972). Hydrodynamic lift on a streambed. *Sedimentation*, H. W. Shen, ed., Water Resources Publications, Fort Collins, Colorado.

Carey, G. F., and J. T. Oden (1986). *Finite elements: Fluid mechanics*, Vol. VI. Prentice-Hall, Englewood Cliffs, N. J.

Carrier, G., and C. E. Pearson (1988). *Partial differential equations: theory and technique*, second ed., Academic Press.

Carmo, M. P. D. (1976). *Differential geometry of curves and surfaces*, Prentice-Hall, Englewood Cliffs, N. J.

Chen, C. N. and M. R. Carsten (1973). Mechanic of removal of a spherical particle from a flat bed. *5th Cong.* IAHR, Istanbul, Turkey, Vol. 1, A30.

Chow, V. T. (1969). *Open-channel hydraulics.* McGraw-Hill, pp.192-194.

Clough, R. W. (1960). The finite element method in plane stress analysis. *Proc. of 2nd ASCE Conference on electronic computation.* Pittsburgh, PA, Sept. 8&9.

Coddington, E. A., and N. Levinson (1955). *Theory of ordinary differential equations*, Krieger Publishing.

Coleman, N. L. (1967). A theoretical and experimental study of drag and lift forces acting on a sphere resting on a hypothetical streambed. *Proc. 12th Cong.* IAHR, Fort Collins, Paper C.22.

Coleman, N. L (1979). Bed particle Reynolds modelling for fluid drag. *J. Hydraulic Research*, 17, No. 2, 91-105.

Courant, R. (1943). Variational methods for the solutions of problems of equilibrium and vibration, *Bull. Am. Math. Soc.*, Vol. 49, pp. 1-23.

Courant, R., K. Friedrichs, and H. Lewy (1928). Über Die Partiellen Differenzengleichungen der mathematischen Physik. *Mathematische Annalen*, Vol. 100, pp. 32-74.

Courant, R., and D. Hilbert (1943). *Methods of mathematical physics*, Vol. 1, Interscience Publisher, New York.

Cuvelier, C.,A. Segal, and A.A. van Steenhoven (1986). *Finite element methods and Navier-Stokes equations.* D. Reidel.

Daily, J. W. and D. R. Harleman (1966). *Fluid dynamics*, Addison Wesley.

Dandy, D. S., and H. A. Dwyer (1990). A sphere in shear flow at finite Reynolds number: effect of shear on particle lift, drag, and heat transfer. *J. Fluid Mech.*, Vol. 216, pp. 381-410.

Davies, T. R. H. and M. F. A. Samad (1978). Fluid dynamic lift on a bed particle. *J. Hydraulics Division, Proc. ASCE.* 104, HY8, 1171-1182.

Dwyer, H. A. (1981). Some aspects of three-dimensional laminar boundary layers. *Annual Review of Fluid Mechanics*, 13, 217-229.

Einstein, H. A., and E. S. A. El-Samni (1949). Hydrodynamic forces on a rough wall. *Review of Modern Physics*, Vol. 21, pp. 520-524.

Engelman, M. S., G. Strang, and K.-J. Bathe (1981). The application of quasi-Newton methods in fluid mechanics. *International Journal for Numerical Methods in Engineering*, Vol. 17, 707-718.

Fornburg, B. (1988). Steady viscous flow past a sphere at high Reynolds numbers. *J. Fluids Mech.*, Vol. 190, pp. 471-489.

Fortin, M. (1981). Old and new finite elements for incompressible flows. *International Journal for Numerical Methods in Fluids*, Vol. 1, pp. 347-364.

Girault, V. and P. A. Raviart (1986). *Finite element methods for Navier-Stokes equations: theory and algorithm.* Springer-Verlag. pp. 352-362.

Hildebrand, F. B. (1968). *Finite-difference equations and simulations*, Prentice, Englewood Cliffs, N. J.

Isaacson, E., and H. B. Keller (1966). *Analysis of numerical methods*, Wiley, New York.

Jeffreys, H. (1929). Transport of sediments by streams. *Proceedings of Cambridge Philosophical Society*, Vol. 25, Cambridge, England, pp.272-276.

Johnson, C. (1987). *Numerical solution of partial differential equations by the finite element method*, Cambridge University Press, Cambridge.

Kevorkian, J. (1990). *Partial differential equations: analytical solution techniques*, Wadsworth & Brooks/Cole, Pacific Grove, CA.

Kim, I., and A. J. Pearlstein (1990). Stability of the flow past a sphere. *J. Fluid Mech.*, Vol. 211, pp. 73-93.

Kral, E. R. and D. K. McLaughlin (1986). Analytical modeling of sediment particle lift-up. *Advancements in Aerodynamics, Fluid Mechanics, and Hydraulics*. R. E. A. Arndt et al., ed., ASCE, pp. 128-135. TA357 A277

Mager, A. (1964). Three-dimensional laminar boundary layers. *Theory of Laminar flows*. F. K. Moore, ed., Princeton University Press. Sec. C.

Mitchell, A. R., and D. F. Griffiths (1980). *The finite difference method in partial difference equations*, John Wiley & sons, New York.

Monkewitz, P. A. (1988). A note on vortex shedding from axisymmetric bluff bodies. *J. Fluid Mechanics*, 192, 561- 575.

Nakamura, I. (1976). Steady wake behind a sphere. *Phys. Fluids*, 19, 5-8.

Pironneau, O. (1989). *Finite element methods for fluids*, John Wiley & sons.

Ritchmeyer, R., and K. Morton (1967). *Difference methods for initial value problems,* 2nd edition, John Wiley & son.

Ritz, W. (1909). Über eine neue Methode zur Lösung gewissen Variations probleme der mathematischen Physik. *Journ. f. d. Reine und Angew. Math.*, Vol. 135, pp. 1-61; *Gesammelte Werke*, pp.192-250, Gauthier-Villars, Paris, 1911.

Saffman, P. G. (1965). The lift on a small sphere in a slow shear flow. *J. Fluid Mechanics*, 22, part 2, 385-400. QA911 S27

Saffman, P. G. (1968). Corrigendum. *J. Fluid Mechanics*, 31, 624.

Smith, G. D. (1985). *Numerical solution of partial differential equations: finite difference methods*, third edition, Clarendon Press, Oxford.

Strikwerda, J. C. (1989). *Finite difference schemes and partial differential equations*, Wadsworth & Brook/Cole, Pacific Grove, CA.

Stroud, A. H. (1971). *Approximate calculation of multiple integrals*, Prentice-Hall, Englewood Cliffs, N. J.

Sutherland, A. J. (1966). Entrainment of fine sediments by turbulent flows. *Report* No. KH-R-13, California Institute of Technology, Pasadena, CA.

Teman, R. (1977). *Theory and numerical analysis of the Navier-Stokes equations*, North-Holland.

Thomasset, F. (1981). *Implementation of finite element methods for Navier-Stokes equations.* Springer Verlag.

Thomschke, H. (1971). Experimentelle Untersuchung der stationären Umströmung von Kugel und Zylinder in Wandnähe. *Dissertation*, Unversität Karlsruhe, Karlsruhe, Germany.

Turner, M. J., R. W. Clough, H. C. Martin, and L. J. Topp (1956). Stiffness and deflection analysis of complex structures. *J. Aeronautical Sci.*, 23, 805-824.

Van Dyke, M. (1975). *Perturbation methods in fluid mechanics.* Parabolic Press, Stanford, California.

Watters, G. Z. and M. V. P. Rao (1971). Hydrodynamic effects of seepage on bed particles. *J. Hydraulics Division, Proc.* ASCE, 97, HY3, 421-439.

White, F. M. (1991). *Viscous fluid flow*. 2nd ed., McGraw-Hill, New York.

Willetts, B. B. and C. G. Murray (1981). Lift exerted on stationary spheres in turbulent flow. *J. Fluid Mechanics*, 105, 487-505.

Willetts, B. B. and K. F. Naddeh (1986). Measurements of lift on spheres fixed in low Reynolds number flows. *J. Hydra. Research*, 24, No. 5 425-435.

Zienkiewicz, O. C. (1967). *The finite element method in structural and continuum mechanics*, McGraw-Hill.

Zienkiewicz, O. C. (1971). *The finite element method in engineering science*, McGraw-Hill.

Zienkiewicz, O. C., and Y. K. Cheung (1965). Finite elements in the solution of field problems. *Engineer*, Vol. 220, Sept. 24.

Appendix A

Tables

Table 5.1 Input data information for each case.

Case	Velocity profile	Gap ratio	$Re = \frac{VD_s}{\nu}$ †
1	A	0.10	30
2	A	0.25	33
3	A	0.40	36
4	A	0.50	37
5	A	0.75	40
6	A	1.00	43
7	A	1.25	44
8	A	1.50	47
9	A	1.75	48
10	A	2.0	50
11	A	2.5	52
12	A	3.0	54
13	A	3.5	57
14	A	4.0	58
15	B	0.5	69
16	B	1.0	77
17	B	2.0	80
18	B	3.0	82
19	B	4.0	84
20	C	0.5	117
21	C	1.0	122
22	C	2.0	129
23	C	3.0	132
24	C	4.0	133

† Following the notation in Willetts and Naddeh (1986), V is the upper stream velocity at the center line of sphere.

Table 5.2 Comparison of simulated and measured values of lift coefficients.

Case	Gap ratio	C_L, simulated	C_L, measured†	Remark
1	0.10	0.252	0.268	9 iterations
2	0.25	0.138	0.149	8 iterations
3	0.40	0.073	0.107	9 iterations
4	0.50	0.067	0.071	9 iterations
5	0.75	–0.035	–0.020	8 iterations
6	1.00	–0.100	–0.083	9 iterations
7	1.25	–0.104	–0.100	8 iterations
8	1.50	–0.125	–0.140	linear solution
9	1.75	–0.133	–0.153	linear solution
10	2.00	–0.140	–0.152	linear solution
11	2.50	–0.103	–0.102	linear solution
12	3.00	–0.057	–0.067	linear solution
13	3.50	–0.058	–0.063	linear solution
14	4.00	–0.006	0	linear solution
15	0.5	0.016	0.015	linear solution
16	1.0	–0.038	–0.040	linear solution
17	2.0	–0.061	–0.058	linear solution
18	3.0	–0.041	–0.034	linear solution
19	4.0	–0.008	–0.009	linear solution
20	0.5	0.014	0.015	linear solution
21	1.0	–0.024	–0.019	linear solution
22	2.0	–0.033	–0.028	linear solution
23	3.0	–0.016	–0.018	linear solution
24	4.0	–0.005	–0.004	linear solution

† Source: Willetts and Naddeh (1986).

Table 5.3 Effect of the diameter of sphere on the lift coefficient at gap ratio of 0.5.

Case	Diameter (cm)	Lift coefficient, C_L
1	0.5	0.076
2	1.0	0.070
3	2.0	0.067
4	3.0	0.030

Table 5.4 Effect of the diameter of sphere on the lift coefficient at gap ratio of 1.

Case	Diameter (cm)	Lift coefficient, C_L
1	0.5	–0.082
2	1.0	–0.087
3	2.0	–0.100
4	3.0	–0.116

Table 5.5 Effect of the diameter of sphere on the lift coefficient at gap ratio of 3.

Case	Diameter (cm)	Lift coefficient, C_L
1	0.5	–0.051
2	1.0	–0.054
3	2.0	–0.057
4	3.0	–0.058

Appendix B
Figures

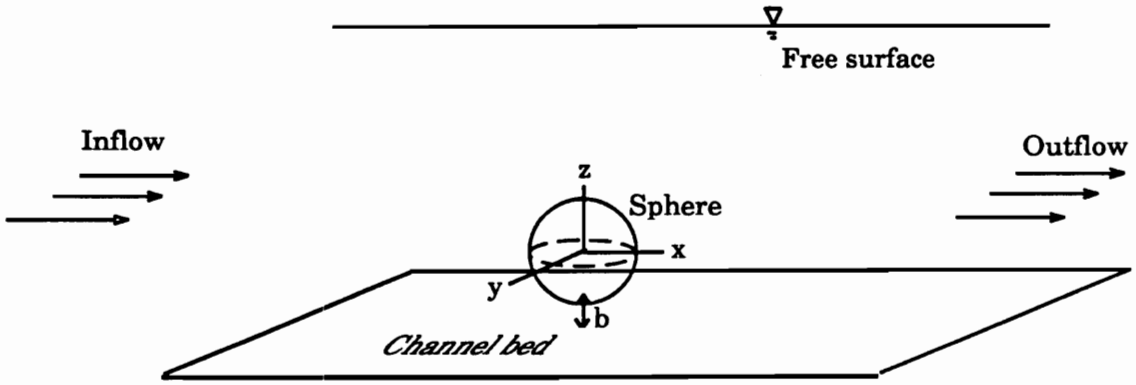


Figure 2.1 Sphere located in a flow near the channel bed.

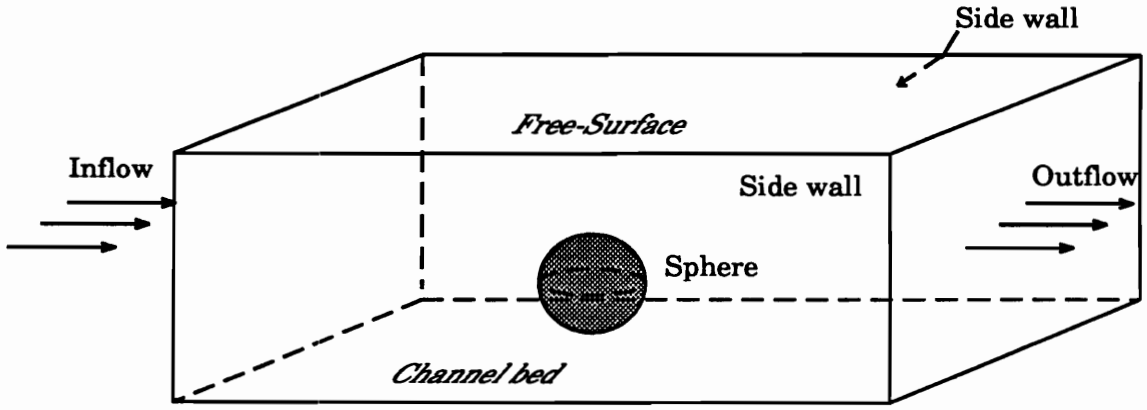


Figure 3.1 Computational domain.

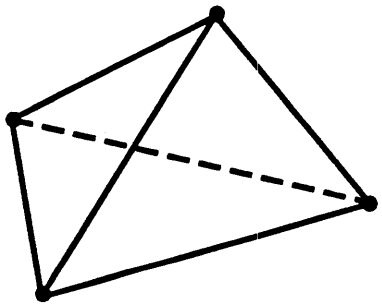
(a) (b)

Figure 3.2 Three-dimensional elements. (a) Tetrahedron (b) Hexahedron

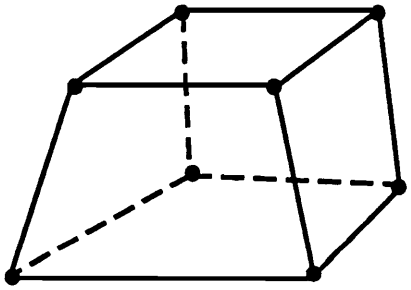
• Nodal point of element

(a) (b)

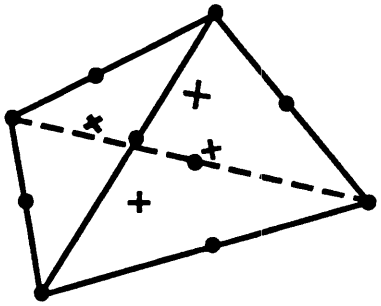
Figure 3.3 Isoparametric elements



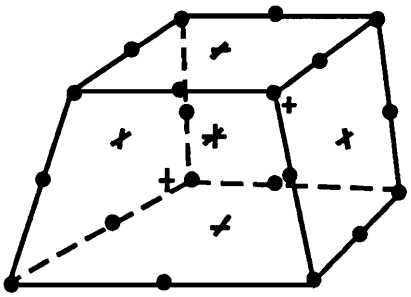
(a) Tri-linear tetrahedral element



(b) Tri-linear hexahedral element



(c) Tri-quadratic tetrahedral element



(d) Tri-quadratic hexahedral element

- Nodal points on the edges and vertices of element
- + Nodal points at centers of surfaces of element

Figure 3.4. Examples of nodal point distribution.

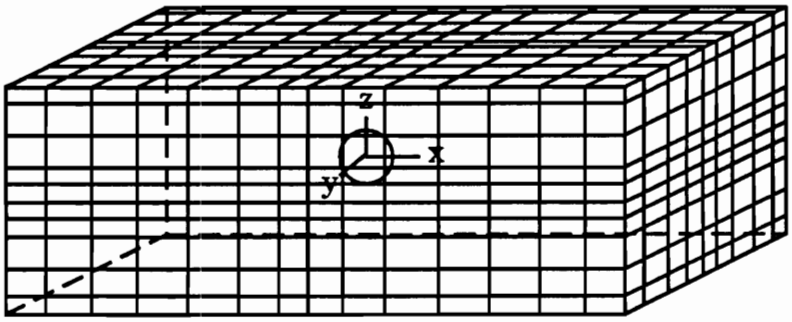


Figure 3.5 Finite element discretization of the computational domain.

x : Node for velocity only

• : Node for velocity and pressure

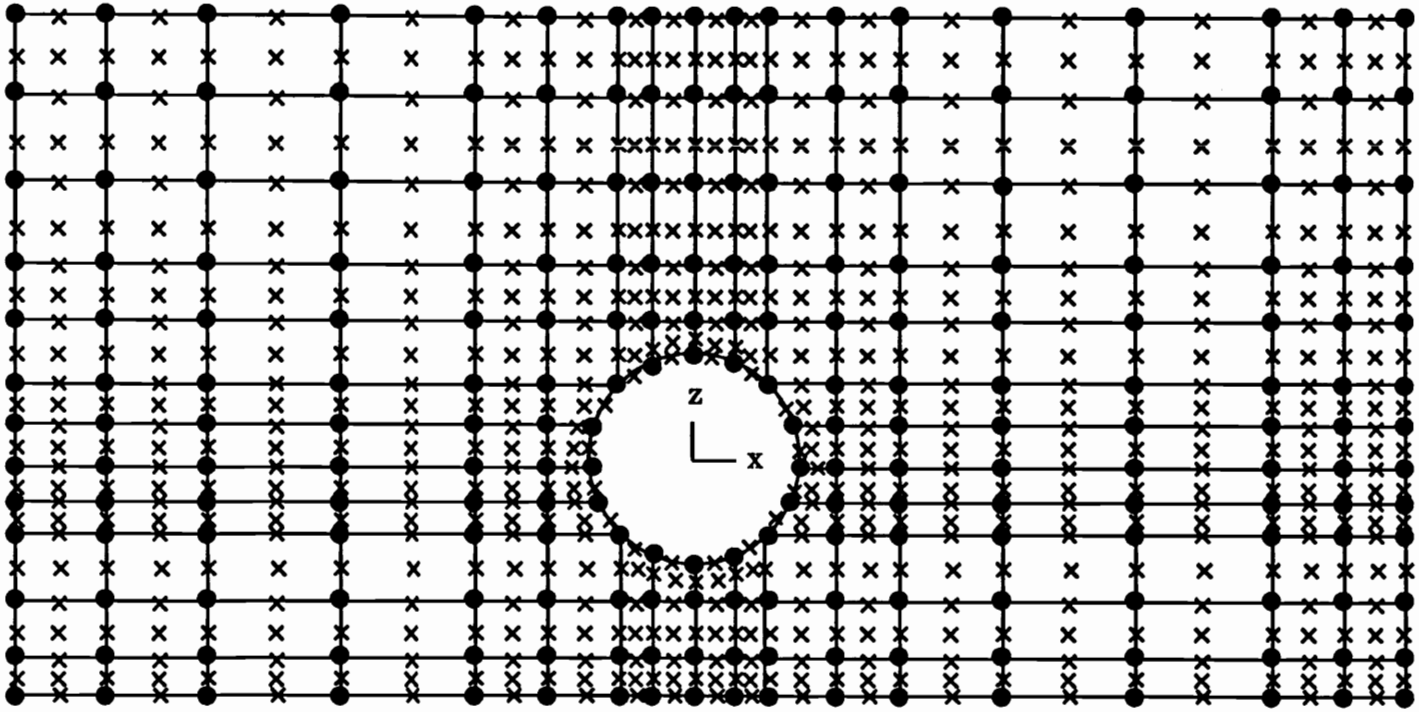


Figure 3.6(a) Typical cross-section of the grid system (x-z plane).

x : Node for velocity only

• : Node for velocity and pressure

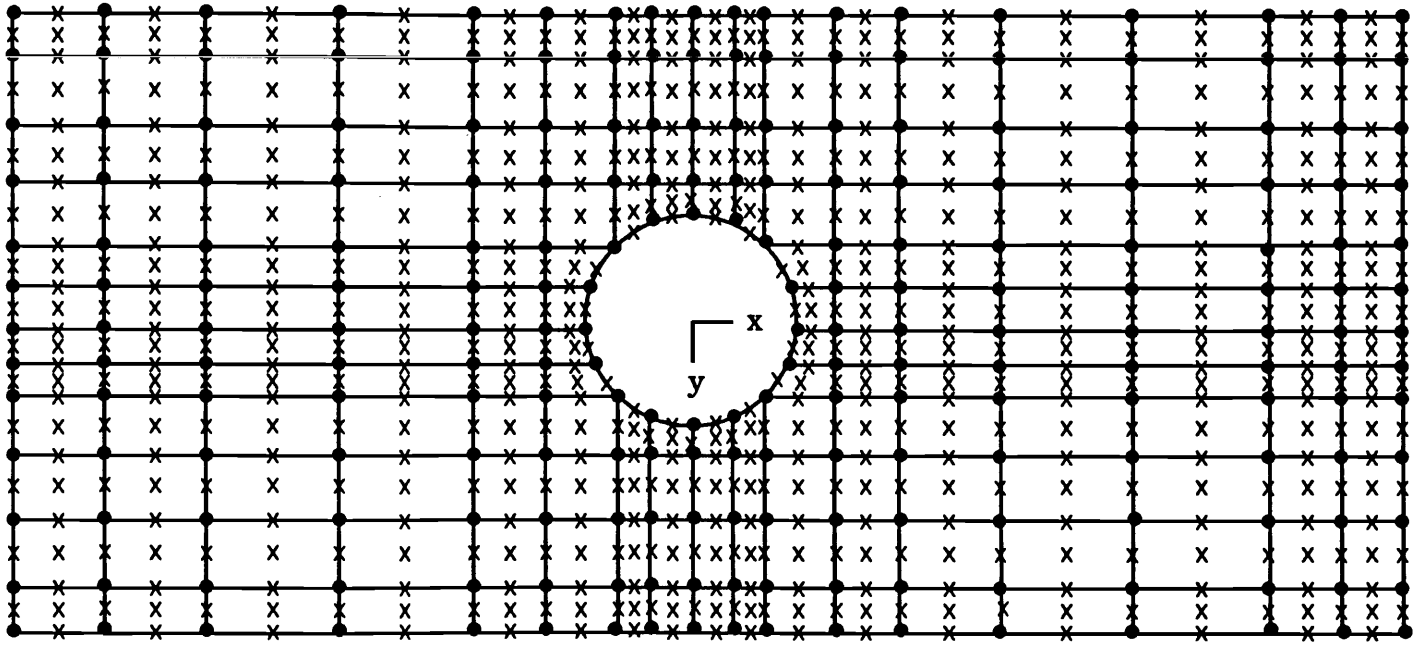
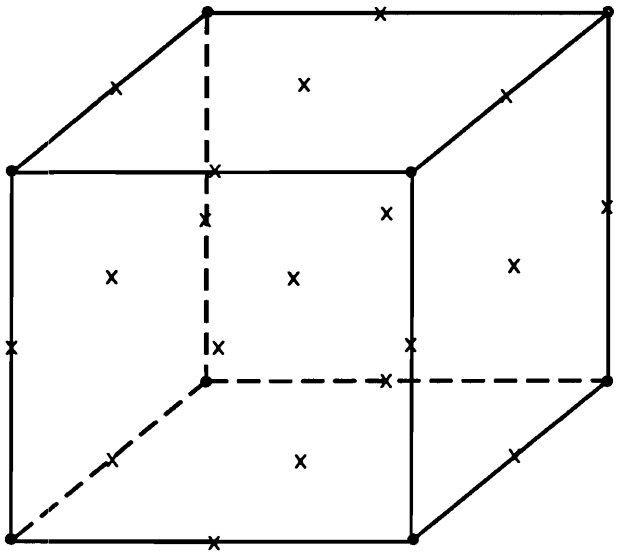


Figure 3.6(b) Typical cross-section of the grid system (x-y plane).



× Nodal points for velocity only

• Nodal points for velocity and pressure

Figure 3.7 Representative finite element with nodal points for the present study.

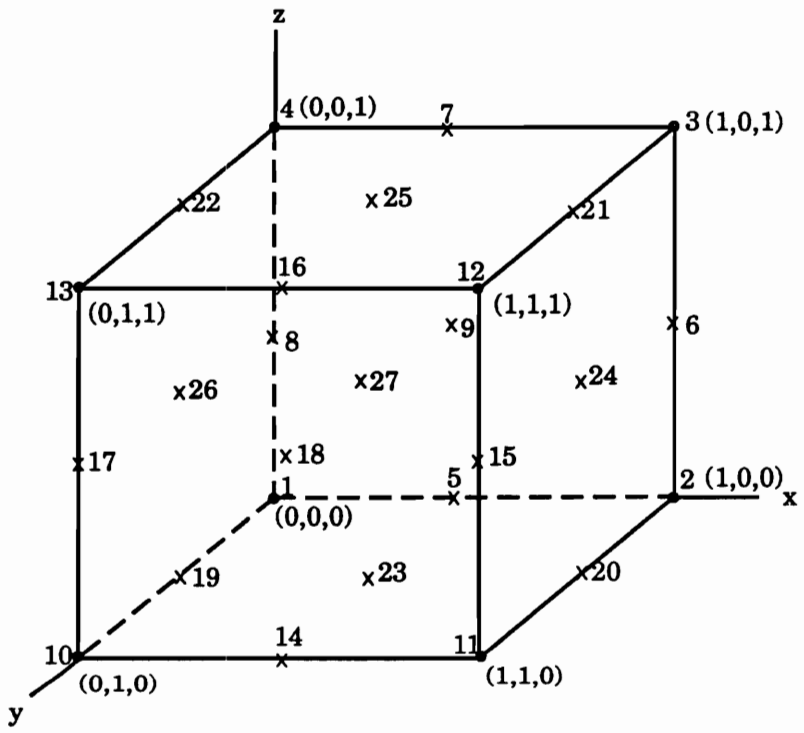


Figure 3.8 Reference element for the computation of velocity components.

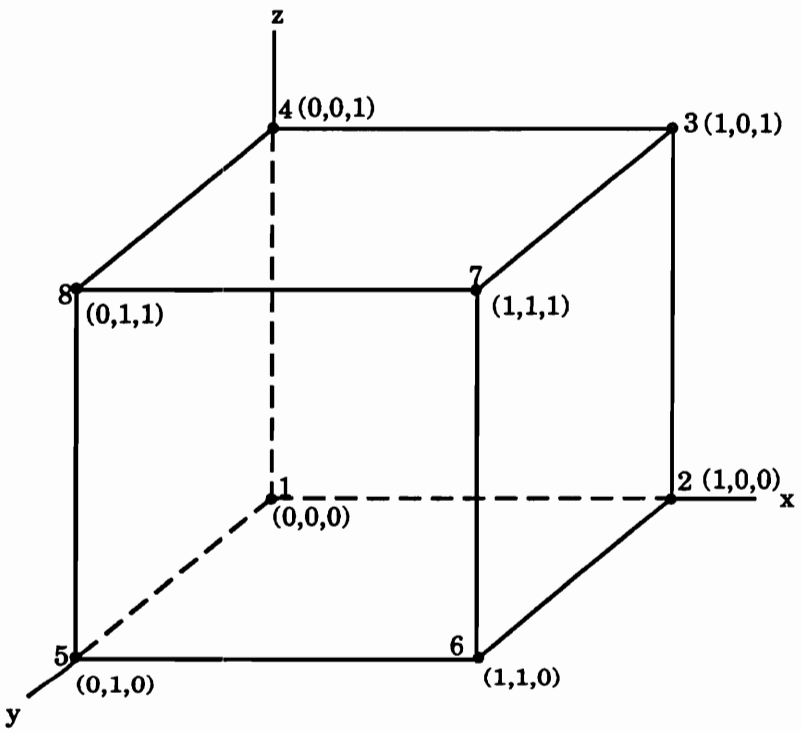


Figure 3.9 Reference element for the computation of pressure.

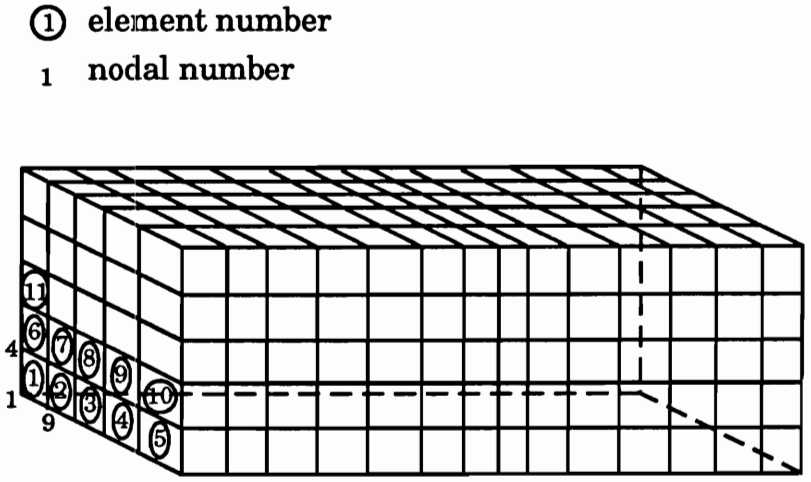


Figure 3.10 Element numbering of the grid system.

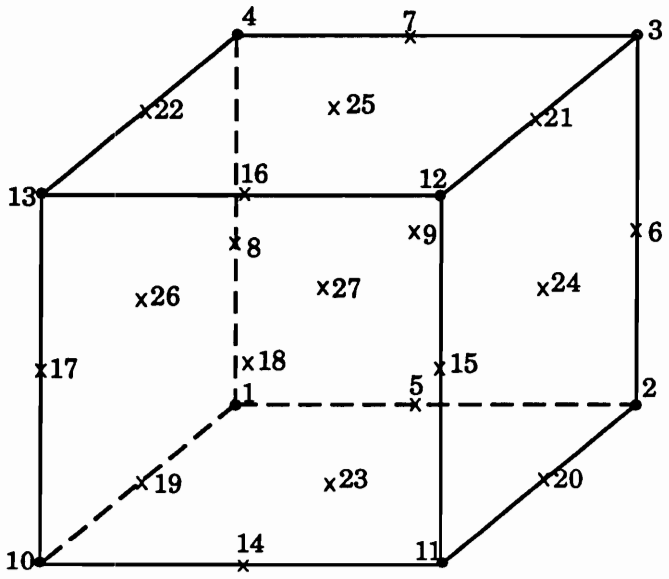


Figure 3.11 Local nodal numbering for an element.

$$
\begin{bmatrix}
\left[K_{11}^{(e)}\right] & \{L_{11}^{(e)}\} & \cdots & \left[K_{1j}^{(e)}\right] & \{L_{1j}^{(e)}\} & \cdots & \left[K_{1n}^{(e)}\right] & \{L_{1n}^{(e)}\} \\
\{C_{11}^{(e)}\} & 0 & \cdots & \{C_{1j}^{(e)}\} & 0 & \cdots & \{C_{1n}^{(e)}\} & 0 \\
\cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\
\left[K_{i1}^{(e)}\right] & \{L_{i1}^{(e)}\} & \cdots & \left[K_{ij}^{(e)}\right] & \{L_{ij}^{(e)}\} & \cdots & \left[K_{in}^{(e)}\right] & \{L_{in}^{(e)}\} \\
\{C_{i1}^{(e)}\} & 0 & \cdots & \{C_{ij}^{(e)}\} & 0 & \cdots & \{C_{in}^{(e)}\} & 0 \\
\cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\
\left[K_{n1}^{(e)}\right] & \{L_{n1}^{(e)}\} & \cdots & \left[K_{nj}^{(e)}\right] & \{L_{nj}^{(e)}\} & \cdots & \left[K_{nn}^{(e)}\right] & \{L_{nn}^{(e)}\} \\
\{C_{n1}^{(e)}\} & 0 & \cdots & \{C_{nj}^{(e)}\} & 0 & \cdots & \{C_{nn}^{(e)}\} & 0
\end{bmatrix}
\begin{bmatrix}
u_1 \\ v_1 \\ w_1 \\ p_1 \\ . \\ u_i \\ v_i \\ w_i \\ p_i \\ . \\ u_n \\ v_n \\ w_n \\ p_n
\end{bmatrix}
=
\begin{bmatrix}
f_{1x} \\ f_{1y} \\ f_{1z} \\ 0 \\ . \\ f_{ix} \\ f_{iy} \\ f_{iz} \\ 0 \\ . \\ f_{nx} \\ f_{ny} \\ f_{nz} \\ 0
\end{bmatrix}
$$

Figure 3.12 Form of the assembled global matrix.

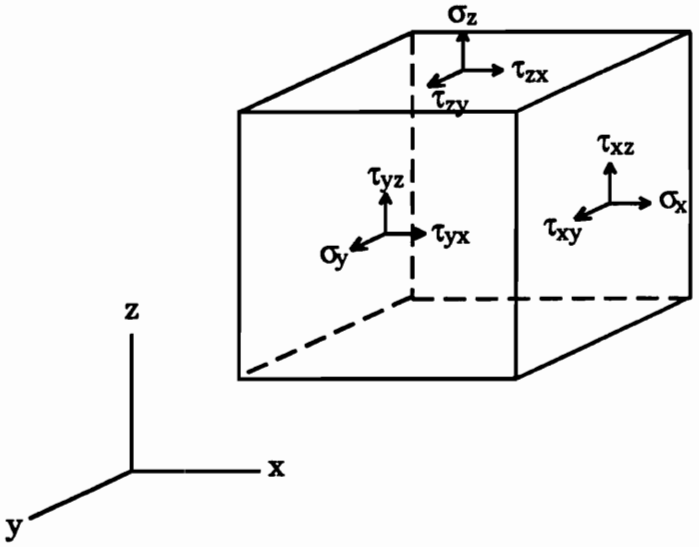


Figure 3.13. Shear stresses on the surface of an element.

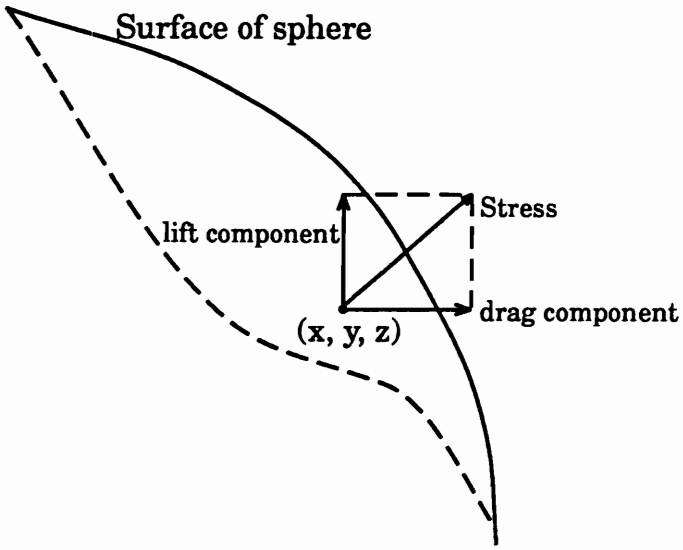


Figure 3.14 Decomposition of stress into lift and drag components.

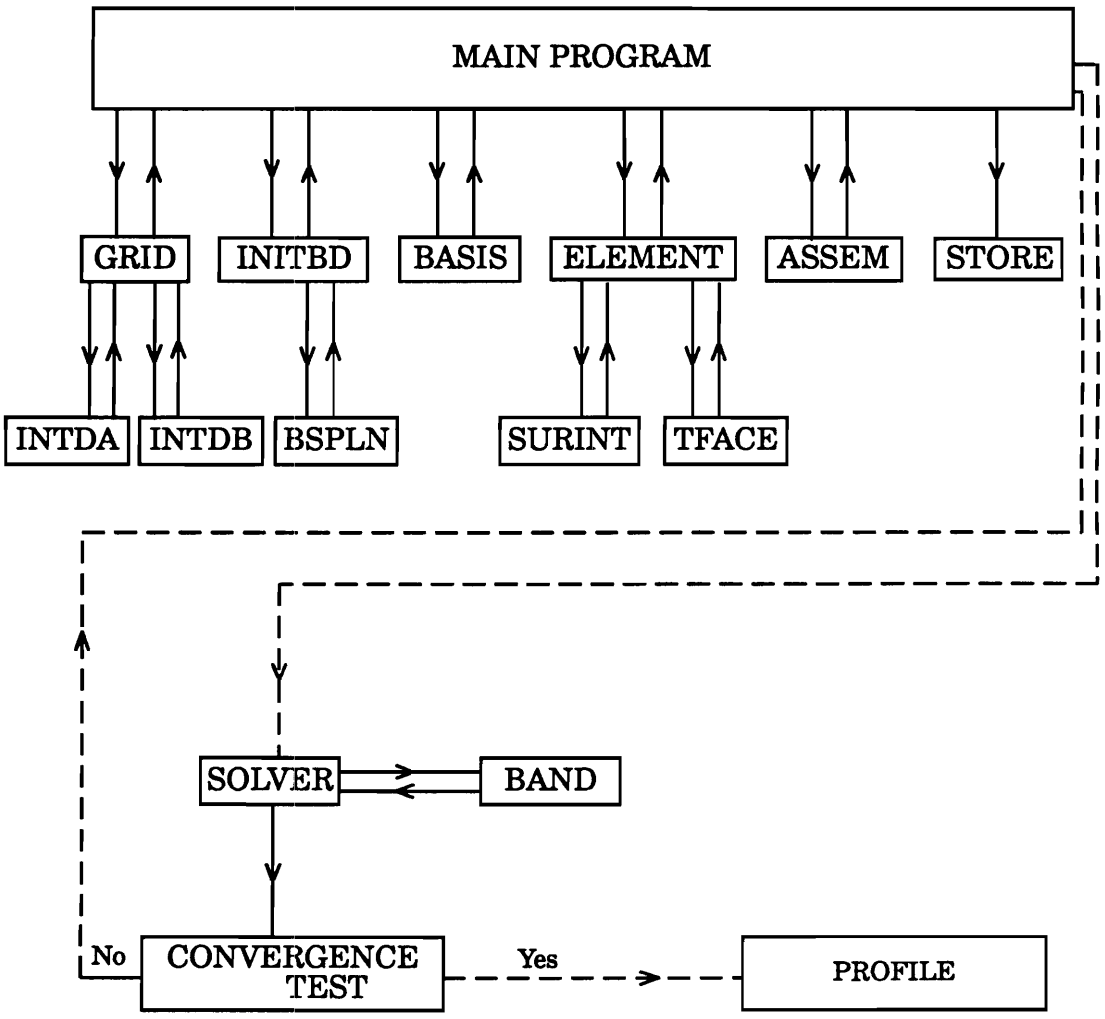


Figure 4.1. Structure of the computer model.

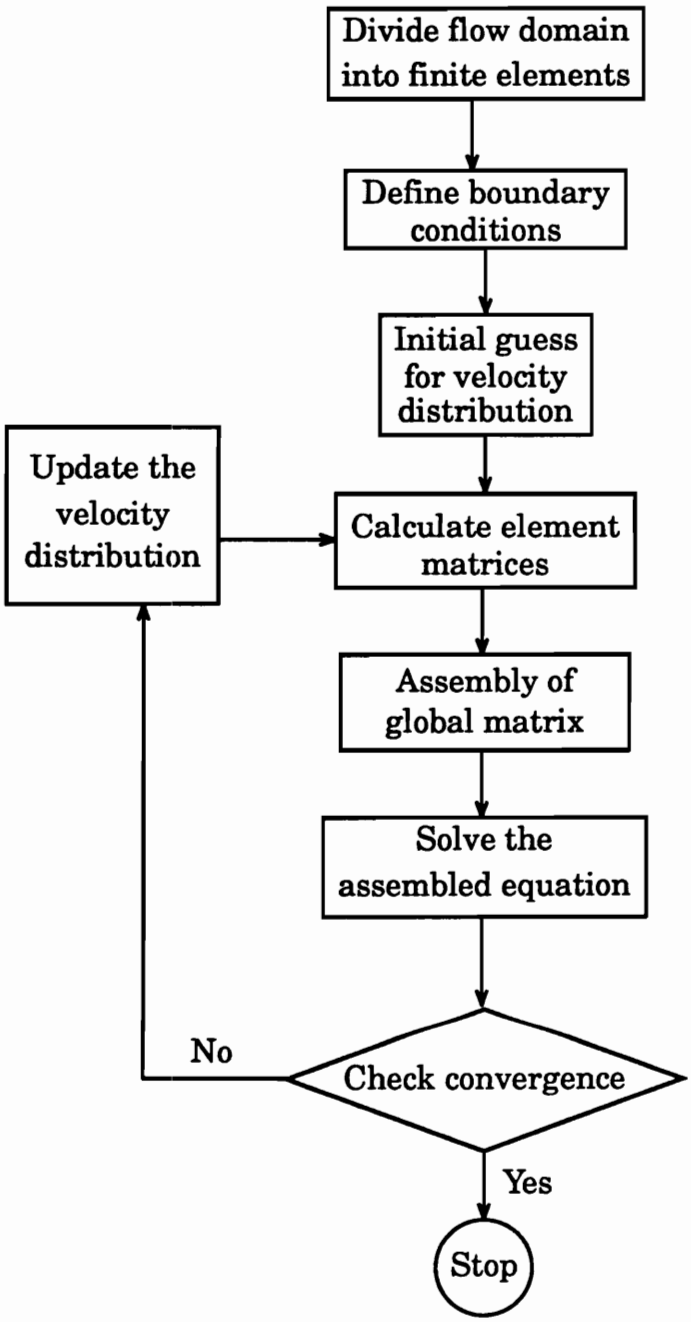


Figure 4.2 Flow chart of the computer program.

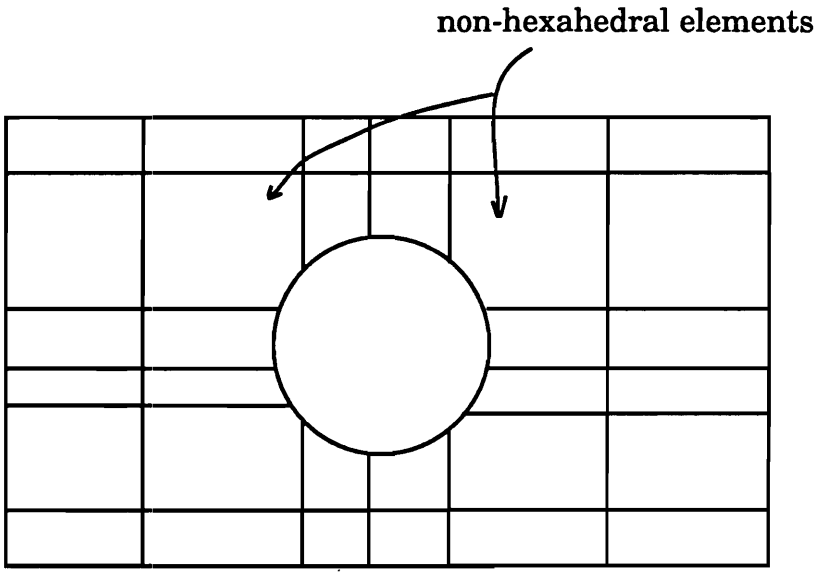


Figure 4.3 Cross-section of an improperly divided mesh.

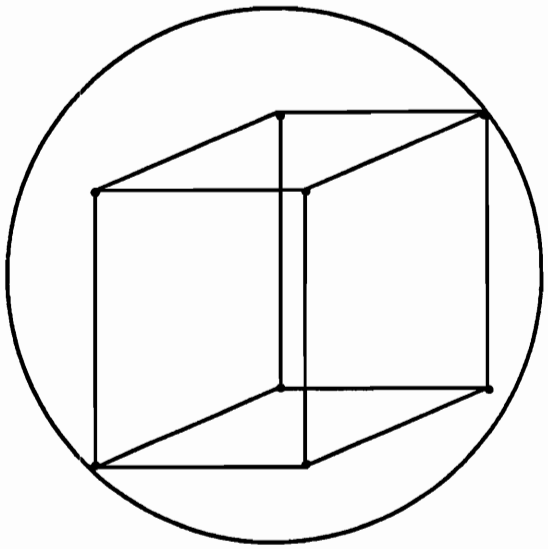


Figure 4.4 Unit cube interconnected to the sphere.

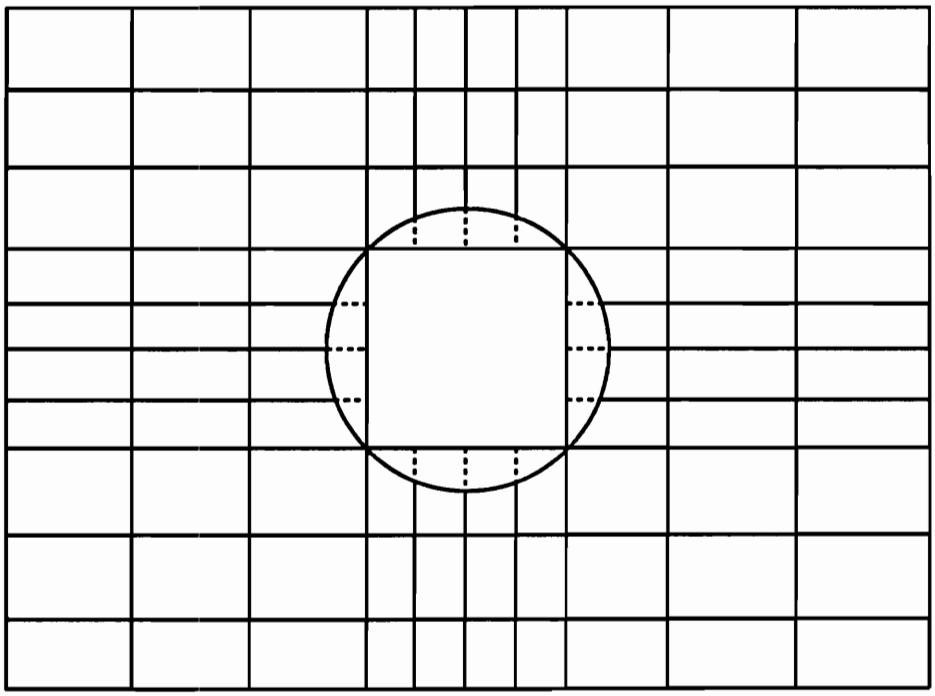


Figure 4.5 A cross-section of the mesh with the cube at the center.

• Nodes with known coordinates

x Nodes with unknown coordinates

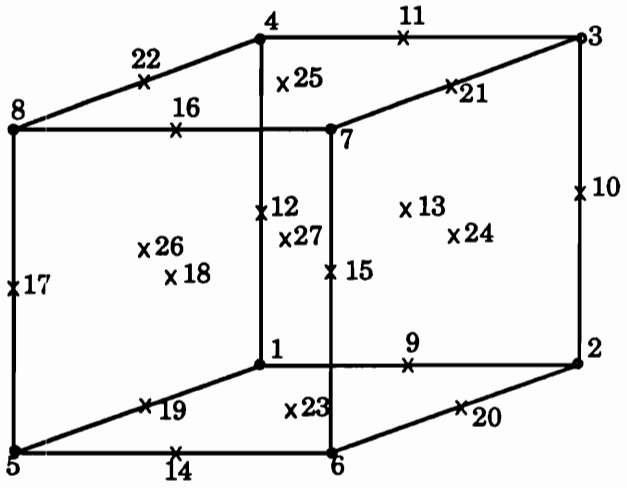
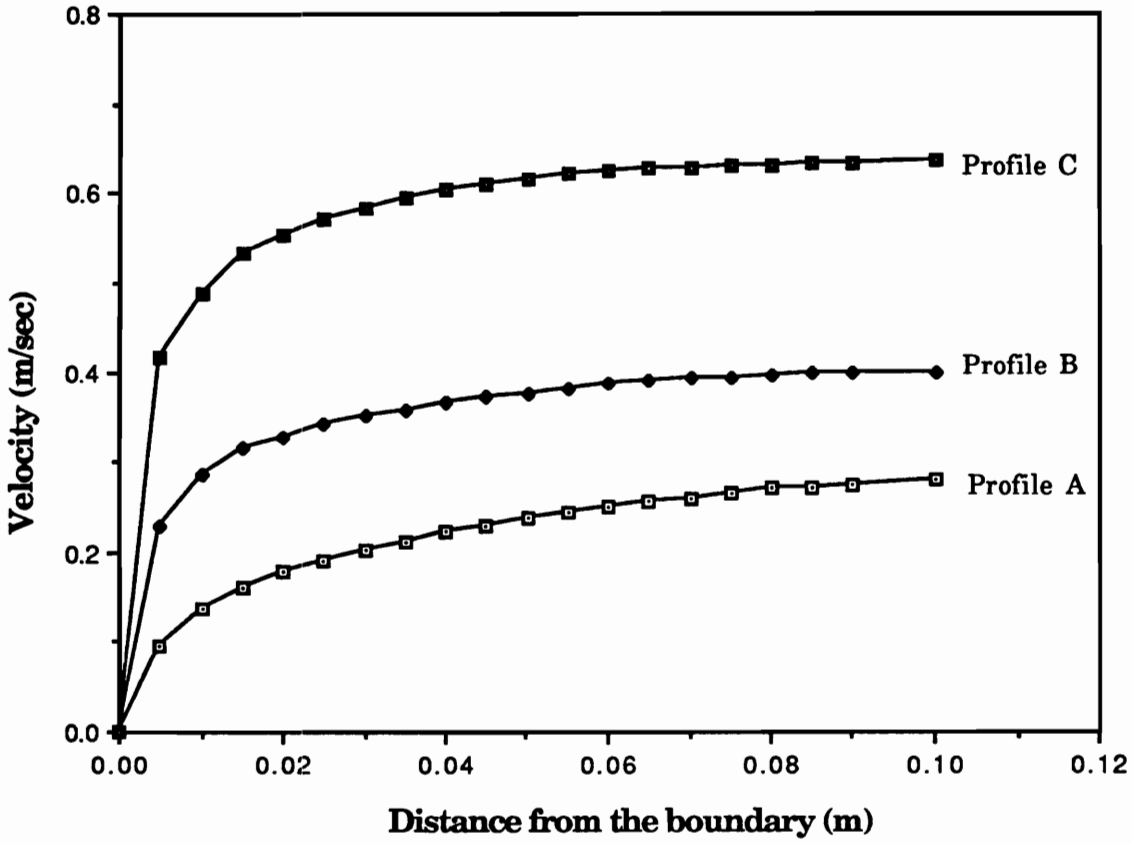


Figure 4.6 Nodal points with known coordinates and unknown coordinates within an finite element in the grid mesh.



Source: Willetts and Naddeh (1986)

Figure 5.1 Velocity profiles for the input data.

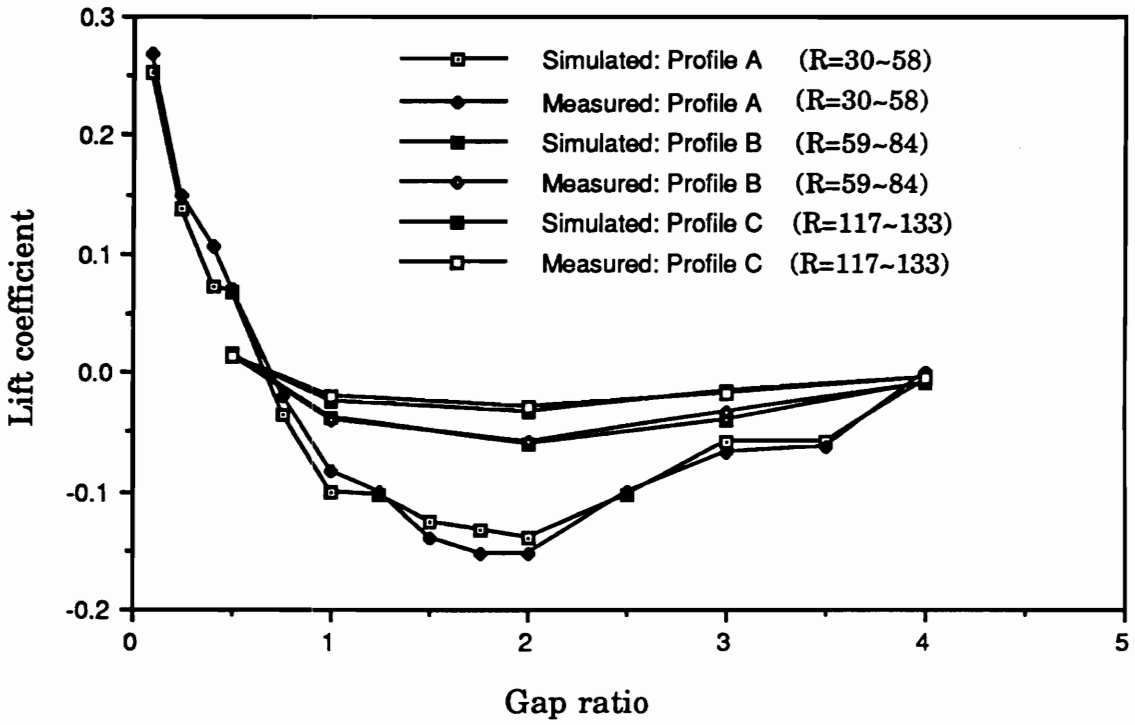


Figure 5.2 Comparison of lift coefficients from simulated and measured results.

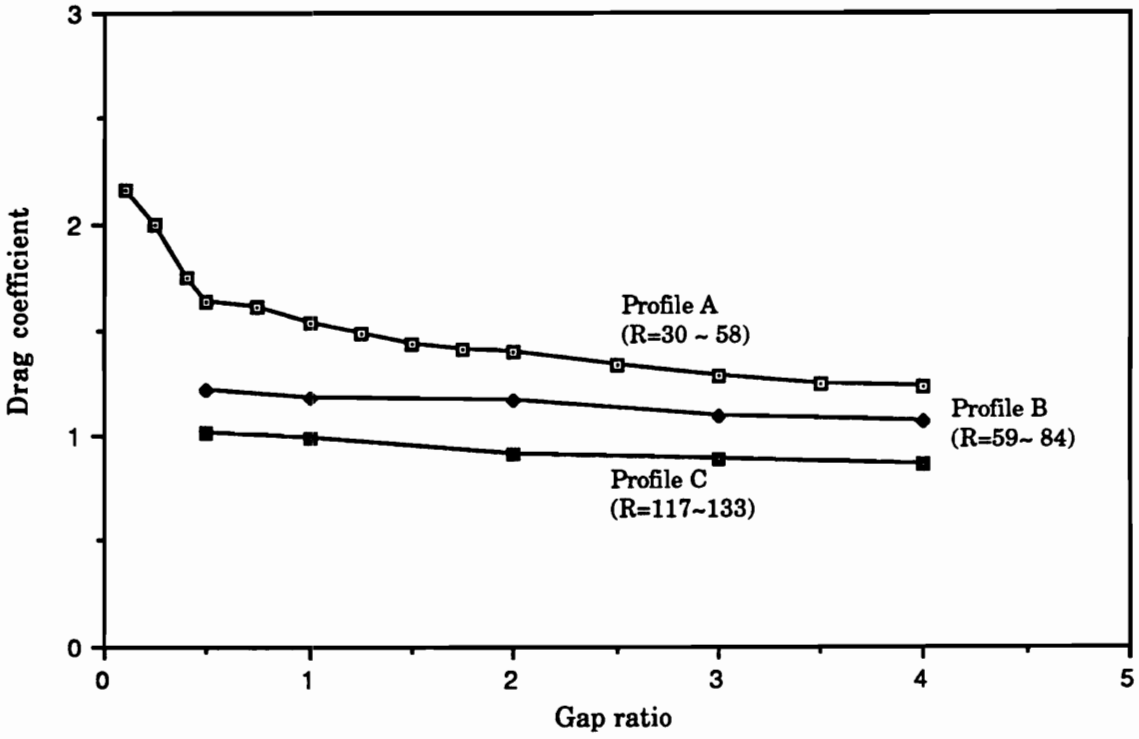


Figure 5.3 Relation between the drag coefficient and gap ratio.

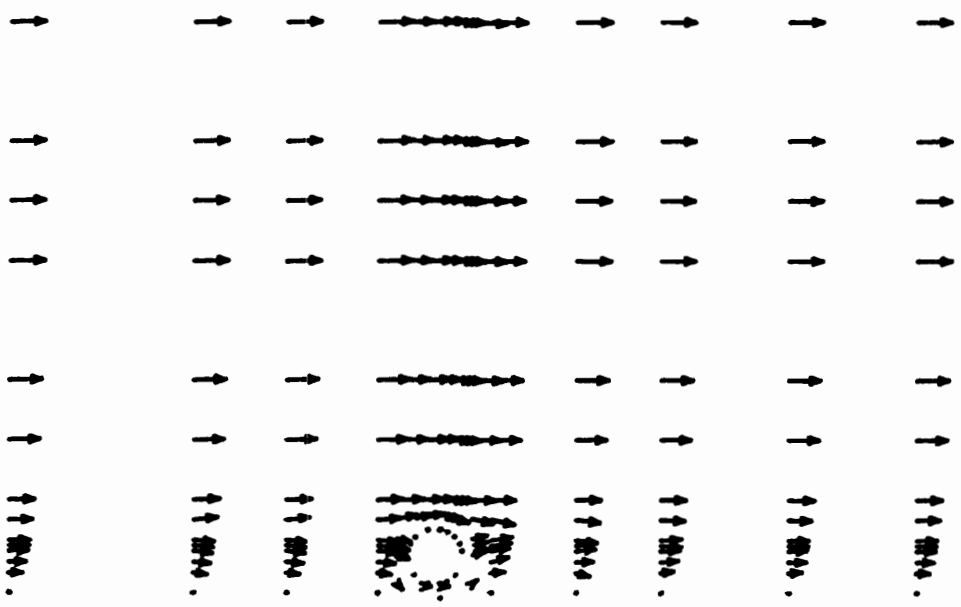


Figure 5.4(a) Vertical velocity distribution at y=0, profile A, gap ratio 0.1.

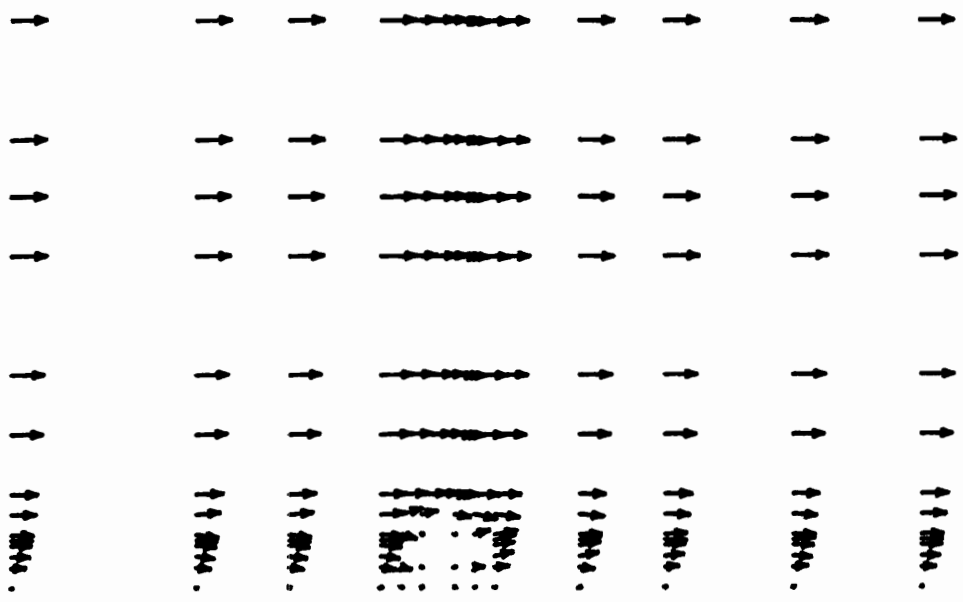


Figure 5.4(b) Vertical velocity distribution at y=0.5D_S, profile A, gap ratio 0.1.

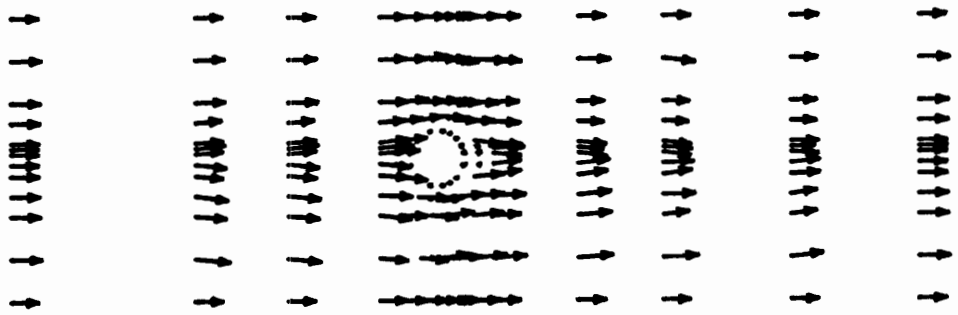


Figure 5.4(c) Horizontal velocity distribution at z=0, profile A, gap ratio 0.1.

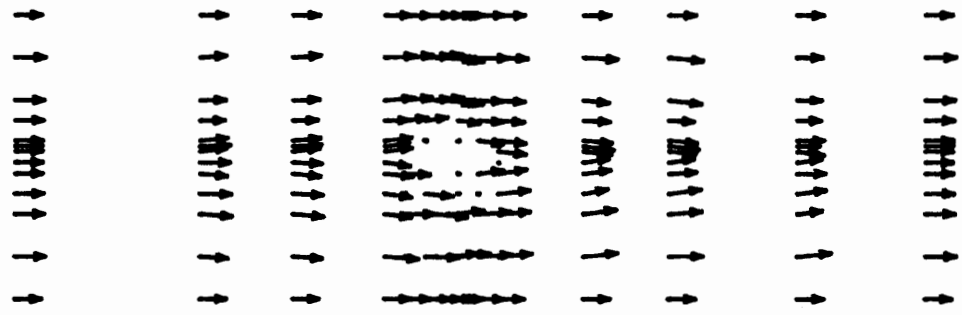


Figure 5.4(d) Horizontal velocity distribution at $z=0.5D_S$, profile A, gap ratio 0.1.

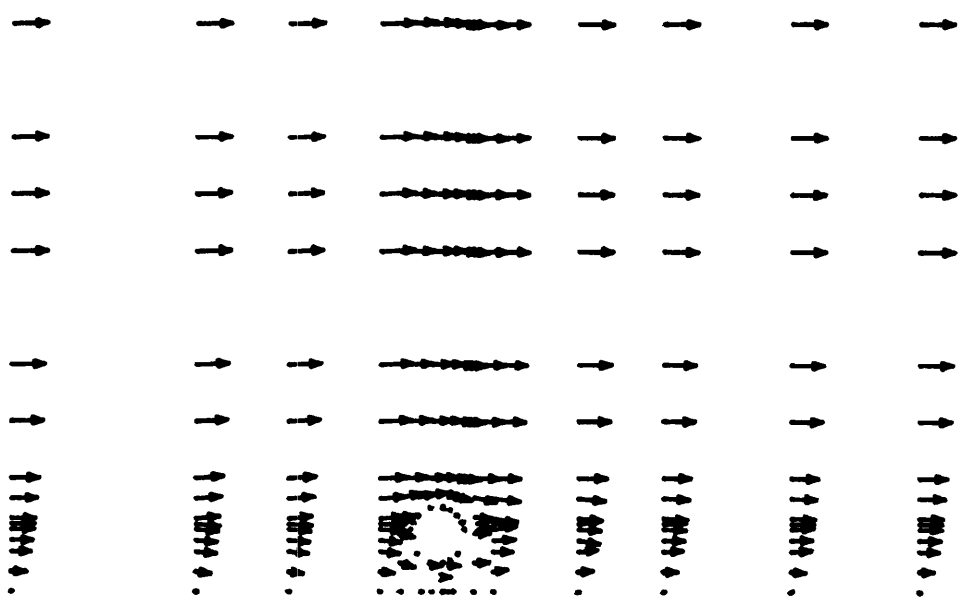


Figure 5.5(a) Vertical velocity distribution at y=0, profile A, gap ratio 0.5.

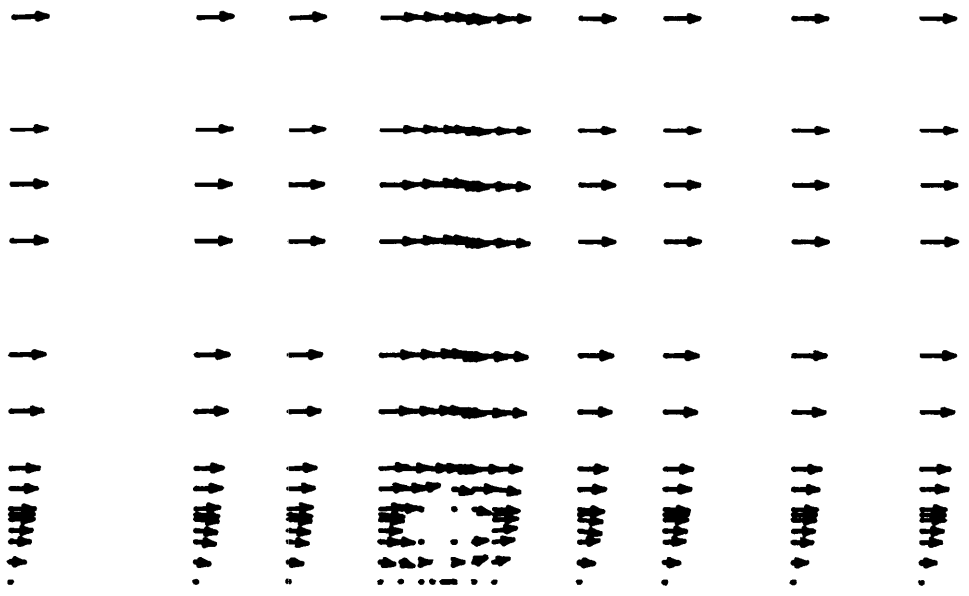


Figure 5.5(b) Vertical velocity distribution at $y=0.5D_S$, profile A, gap ratio 0.5.

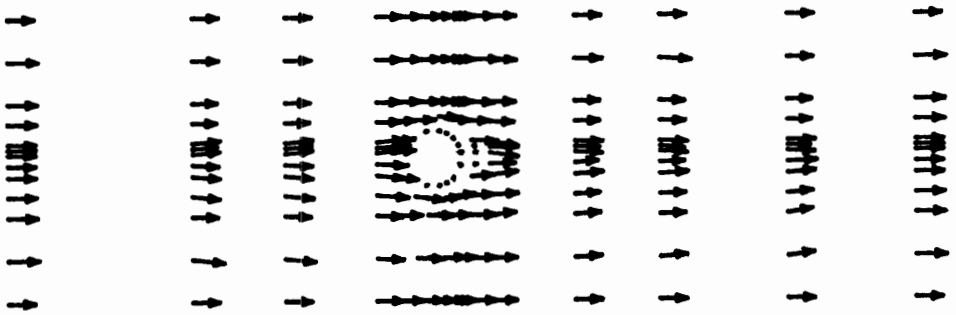


Figure 5.5(c) Horizontal velocity distribution at z=0, profile A, gap ratio 0.5.

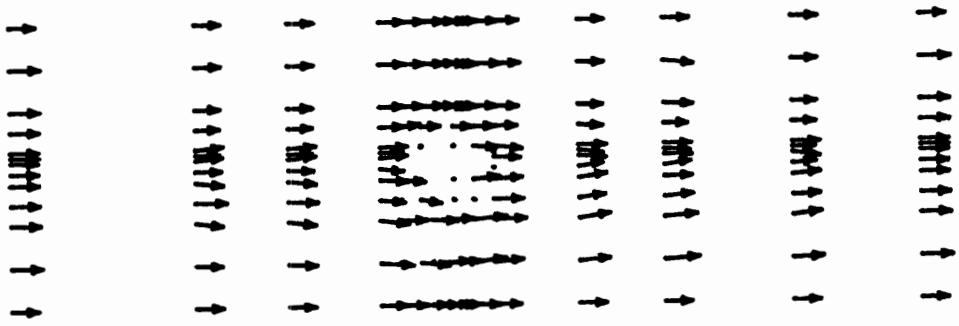


Figure 5.5(d) Horizontal velocity distribution at $z=0.5D_S$, profile A, gap ratio 0.5.

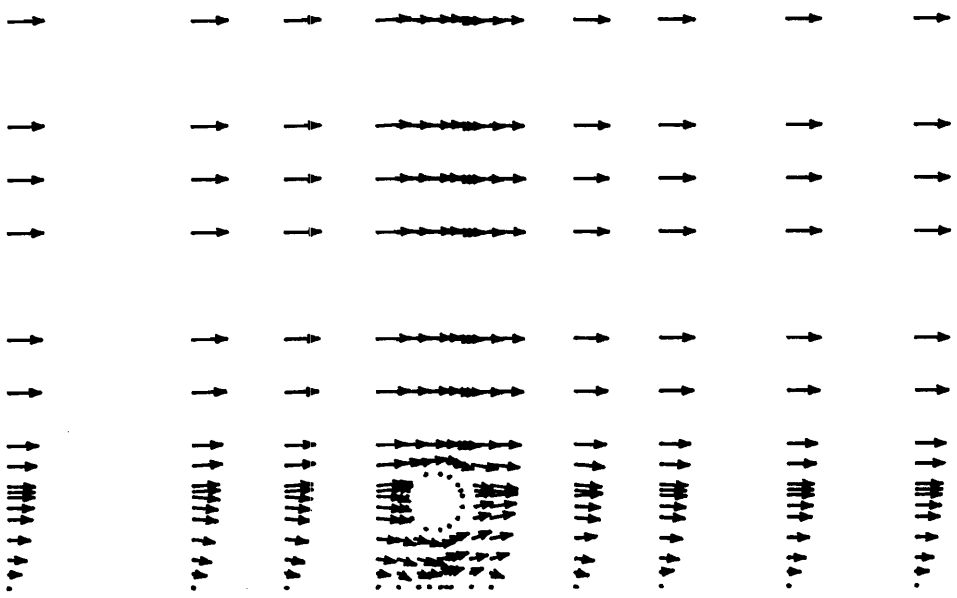


Figure 5.6(a) Vertical velocity distribution at y=0, profile A, gap ratio 1.

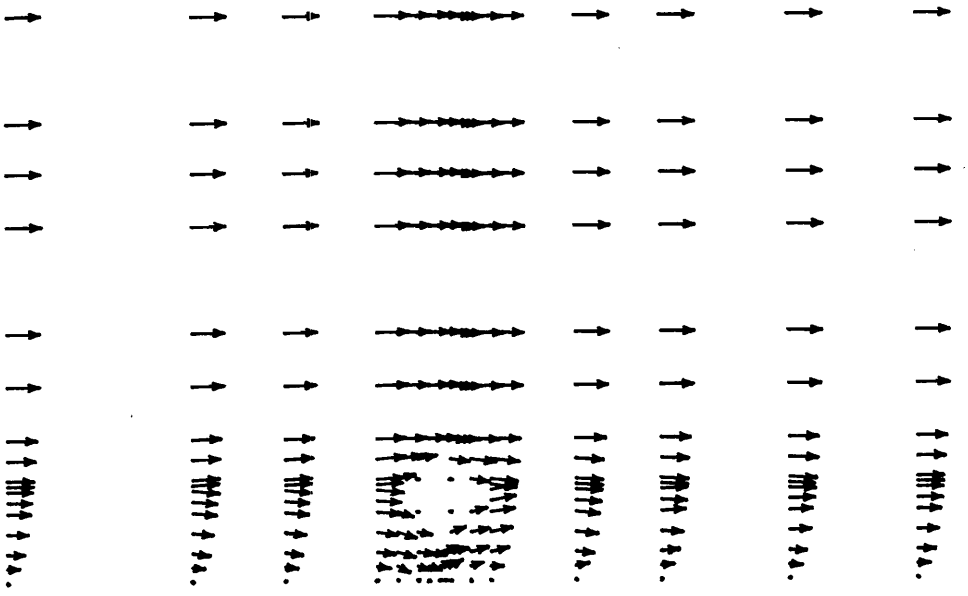


Figure 5.6(b) Vertical velocity distribution at y=0.5D_S, profile A, gap ratio 1.

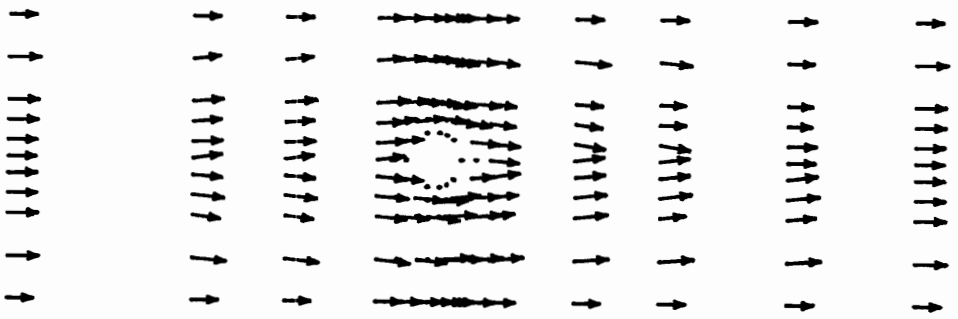


Figure 5.6(c) Horizontal velocity distribution at z=0, profile A, gap ratio 1.

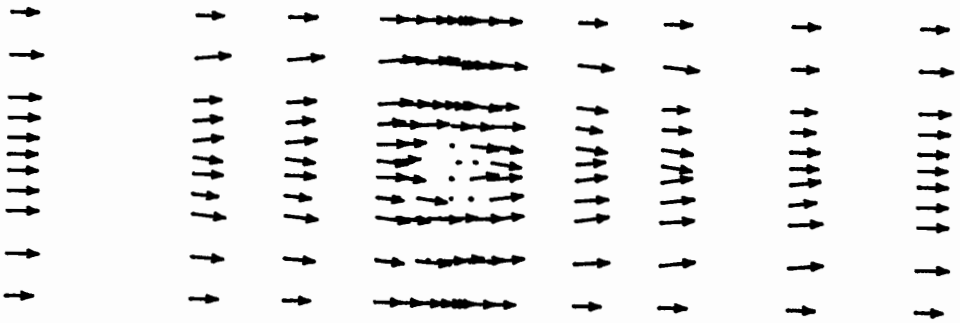


Figure 5.6(d) Horizontal velocity distribution at $z=0.5D_S$, profile A, gap ratio 1.

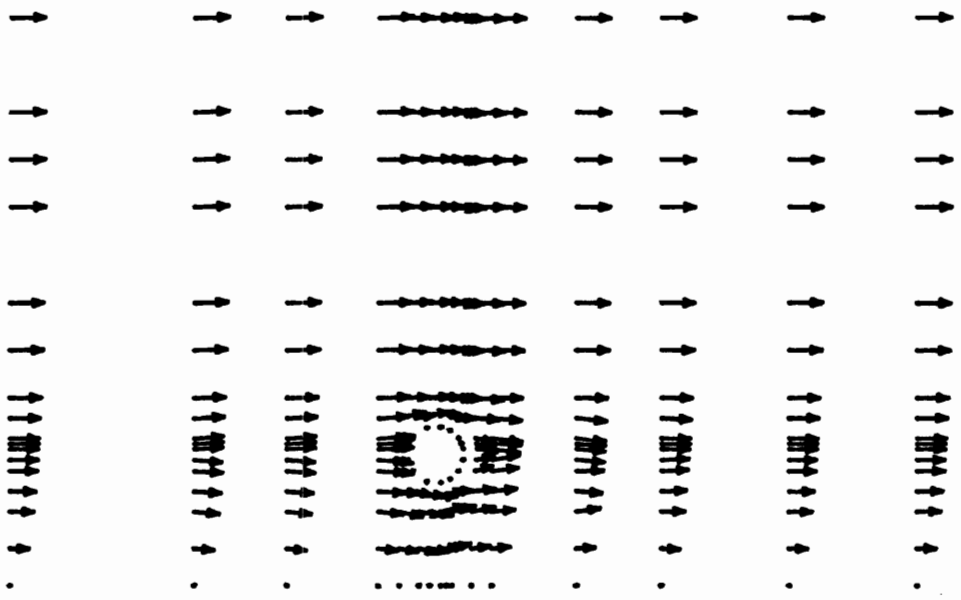


Figure 5.7(a) Vertical velocity distribution at y=0, profile A, gap ratio 2.

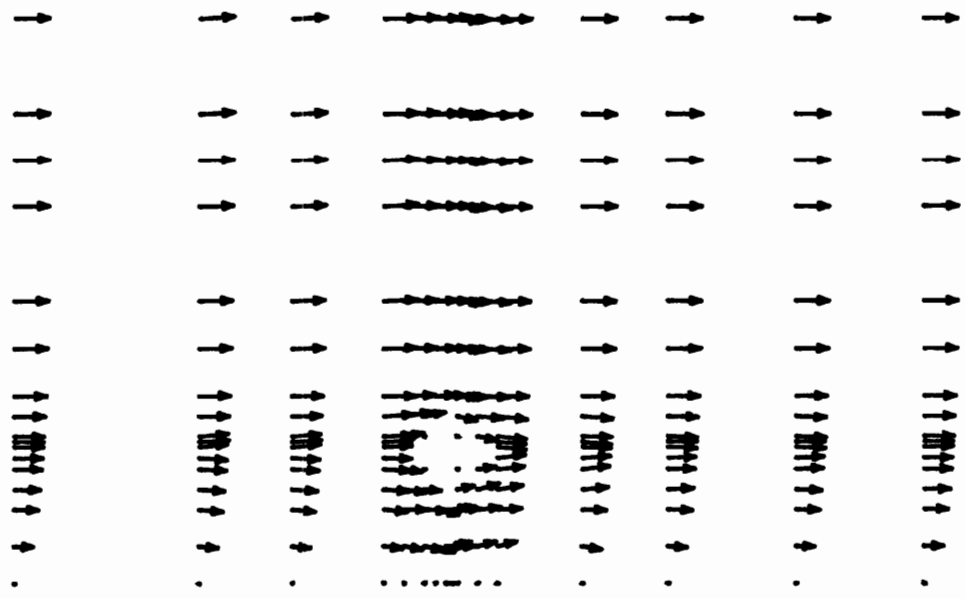


Figure 5.7(b) Vertical velocity distribution at $y=0.5D_S$, profile A, gap ratio 2.

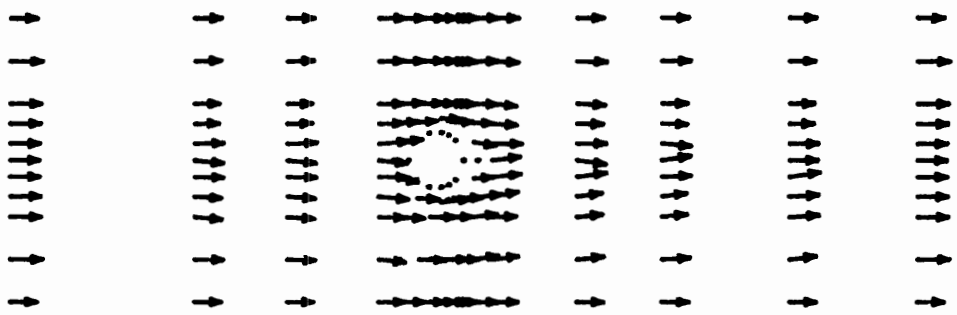


Figure 5.7(c) Horizontal velocity distribution at z=0, profile A, gap ratio 2.

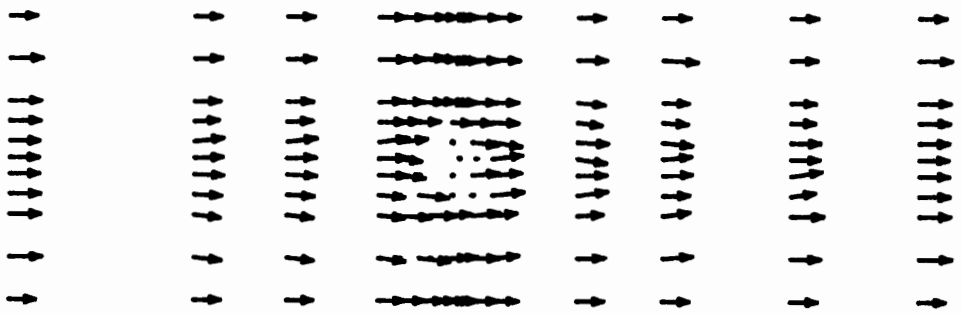


Figure 5.7(d) Horizontal velocity distribution at $z=0.5D_S$, profile A, gap ratio 2.

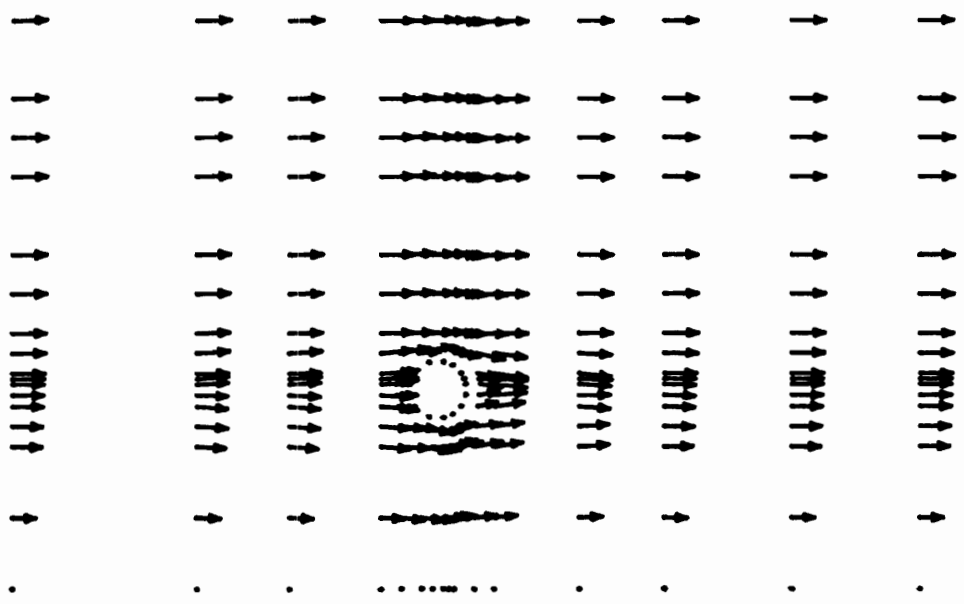


Figure 5.8(a) Vertical velocity distribution at y=0, profile A, gap ratio 3.

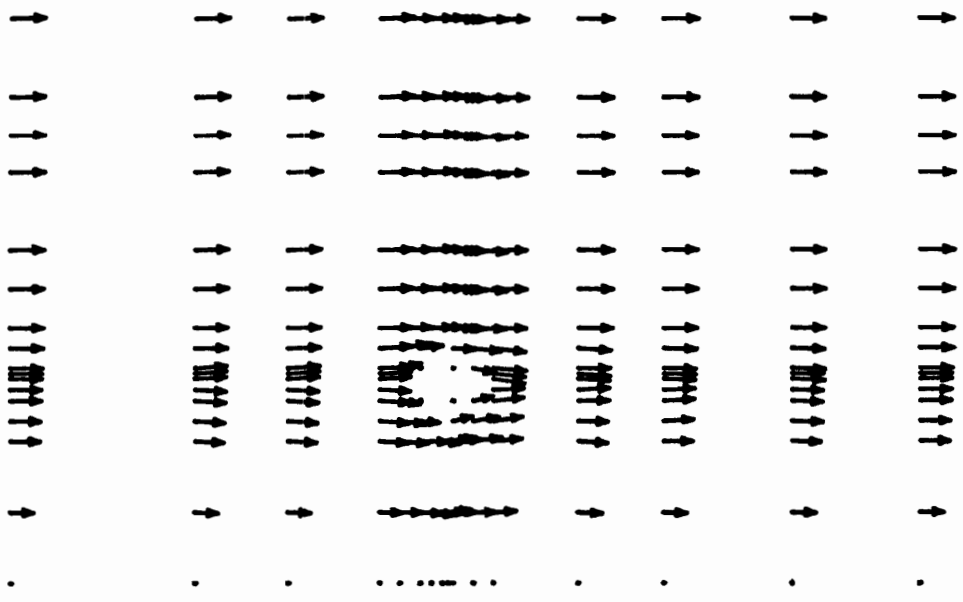


Figure 5.8(b) Vertical velocity distribution at y=0.5D_S, profile A, gap ratio 3.

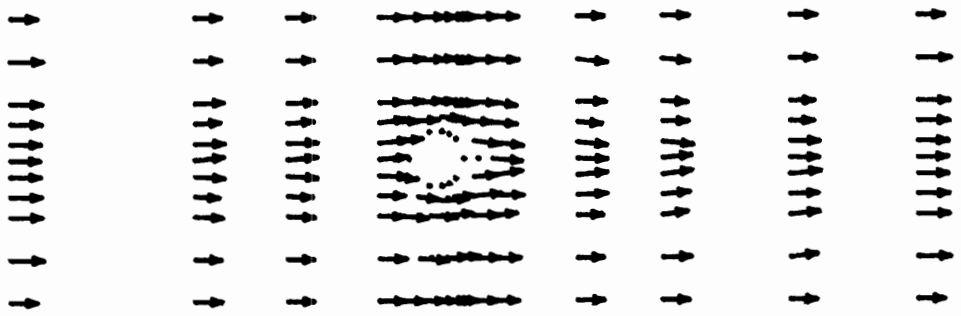


Figure 5.8(c) Horizontal velocity distribution at z=0, profile A, gap ratio 3.

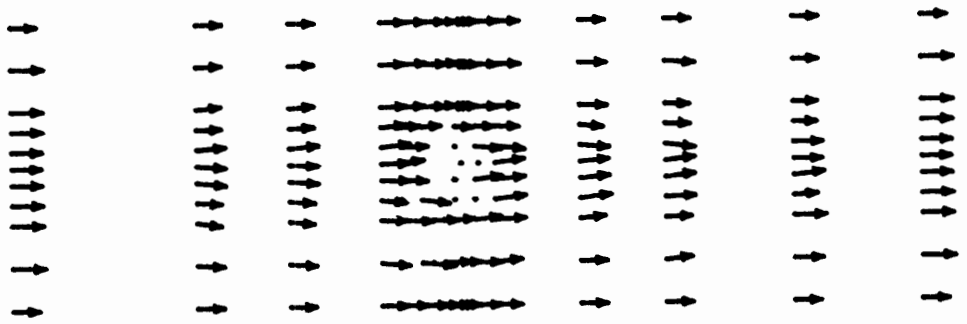


Figure 5.8(d) Horizontal velocity distribution at $z=0.5D_S$, profile A, gap ratio 3.

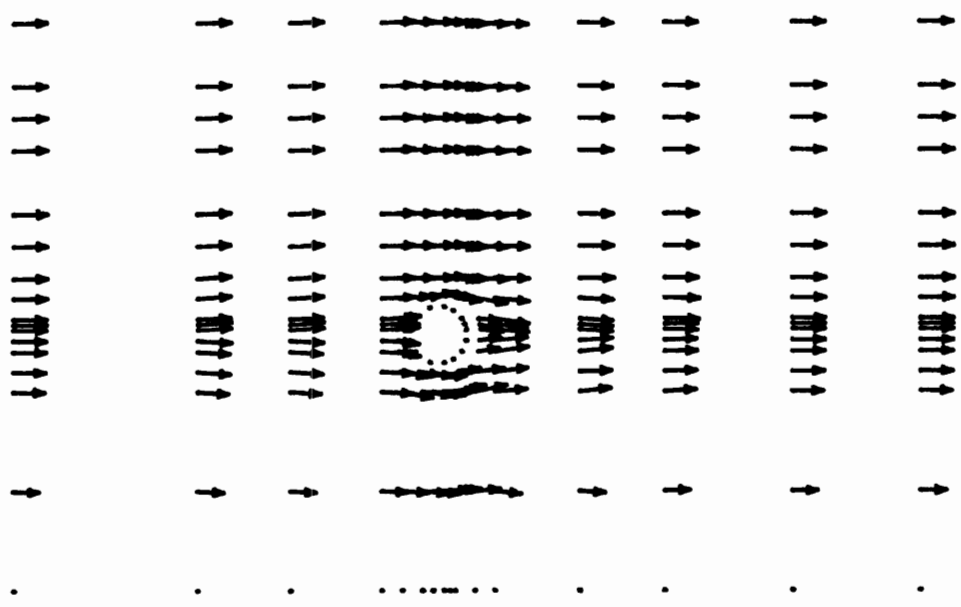


Figure 5.9(a) Vertical velocity distribution at y=0, profile A, gap ratio 4.

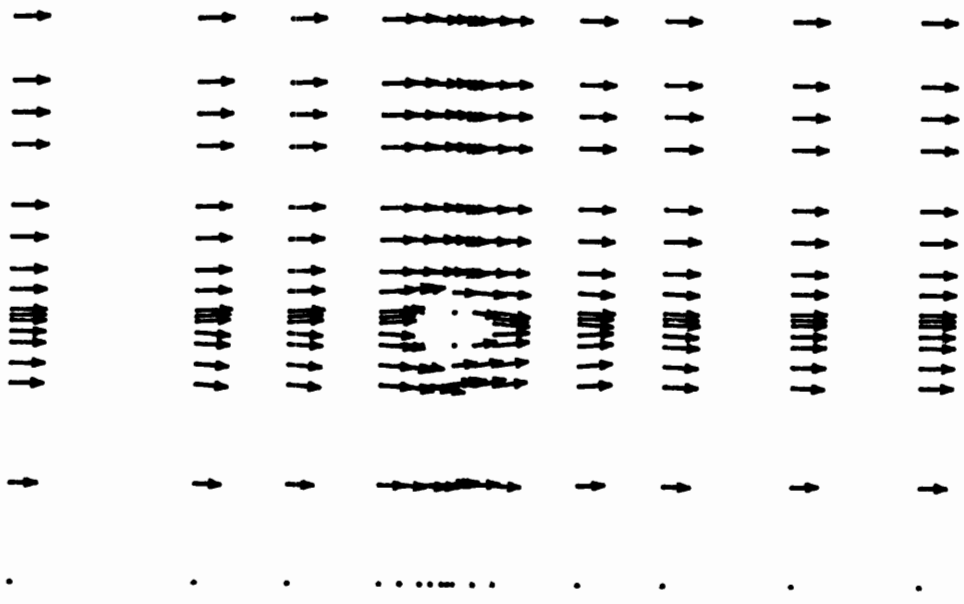


Figure 5.9(b) Vertical velocity distribution at y=0.5D_S, profile A, gap ratio 4.

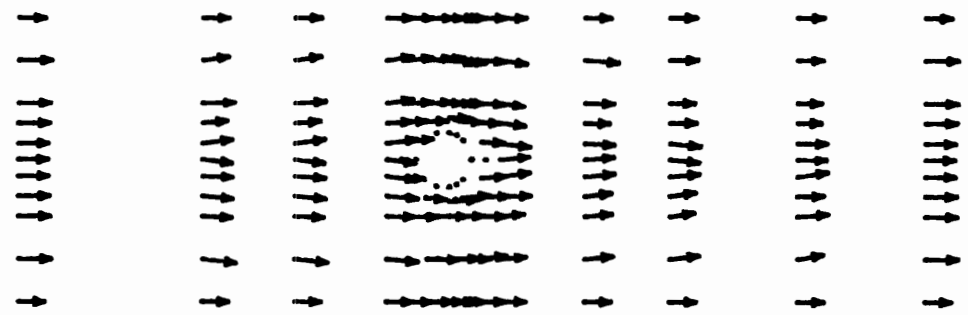


Figure 5.9(c) Horizontal velocity distribution at z=0, profile A, gap ratio 4.

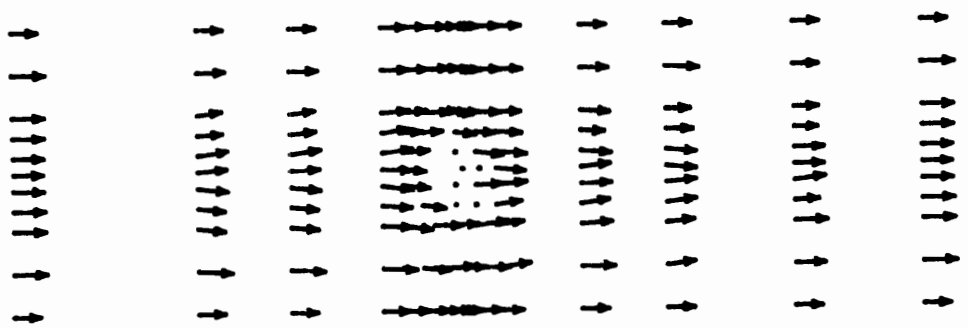


Figure 5.9(d) Horizontal velocity distribution at z=0.5D_S, profile A, gap ratio 4.

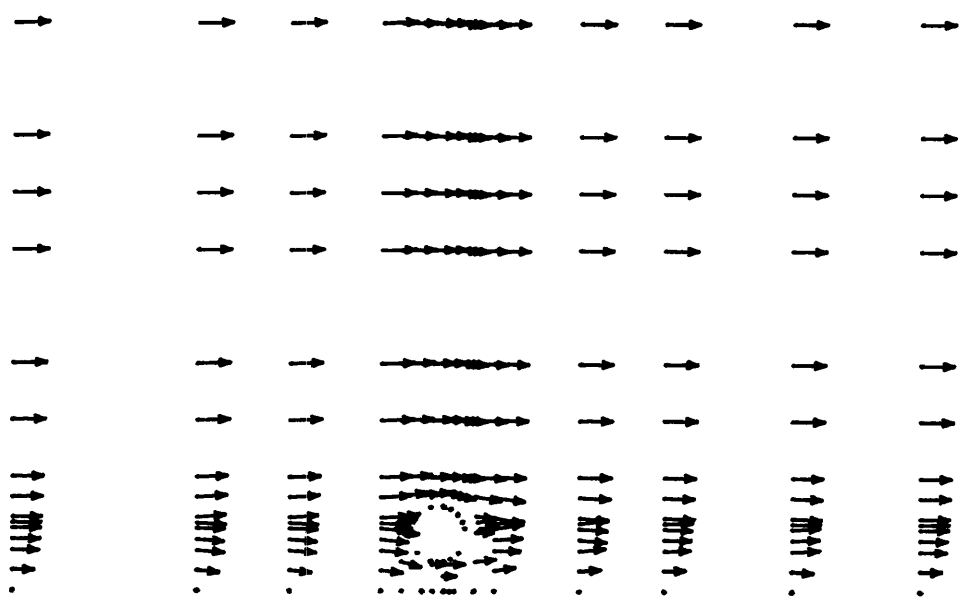


Figure 5.10(a) Vertical velocity distribution at y=0, profile B, gap ratio 0.5.

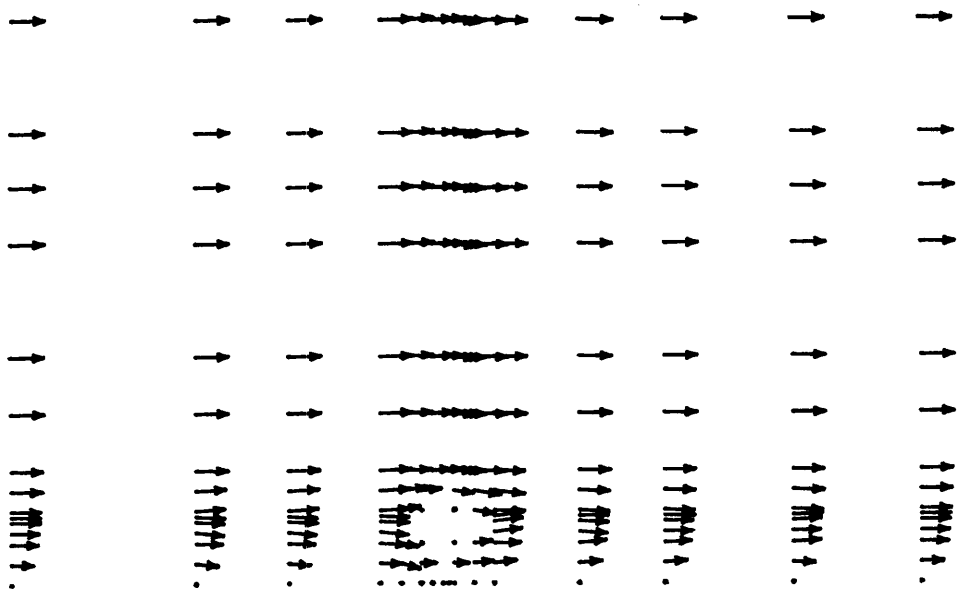


Figure 5.10(b) Vertical velocity distribution at y=0.5D_S, profile B, gap ratio 0.5.

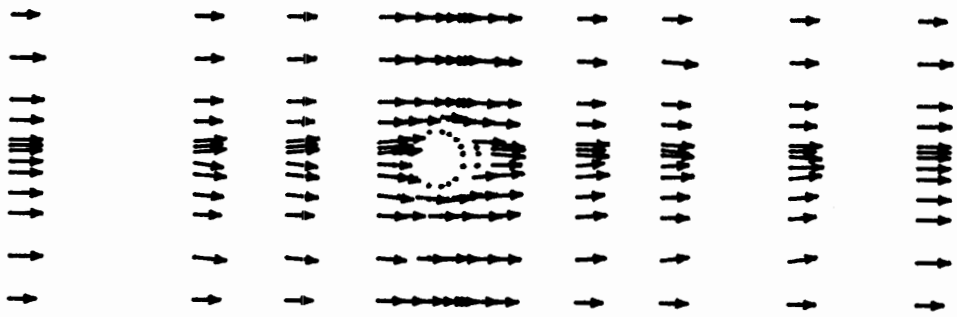


Figure 5.10(c) Horizontal velocity distribution at z=0, profile B, gap ratio 0.5.

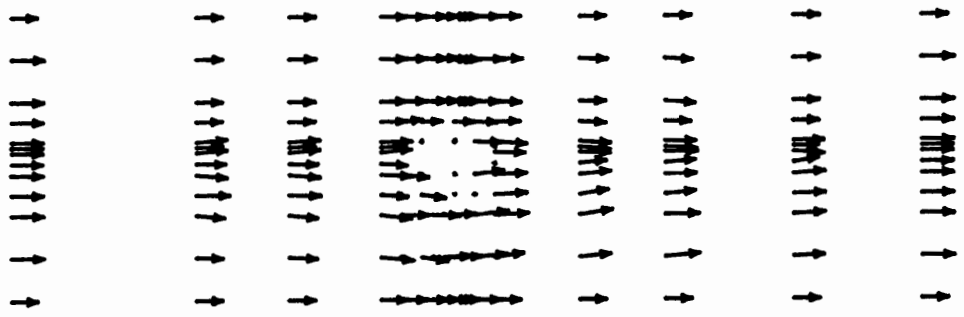


Figure 5.10(d) Horizontal velocity distribution at z=.5D_S, profile B, gap ratio 0.5.

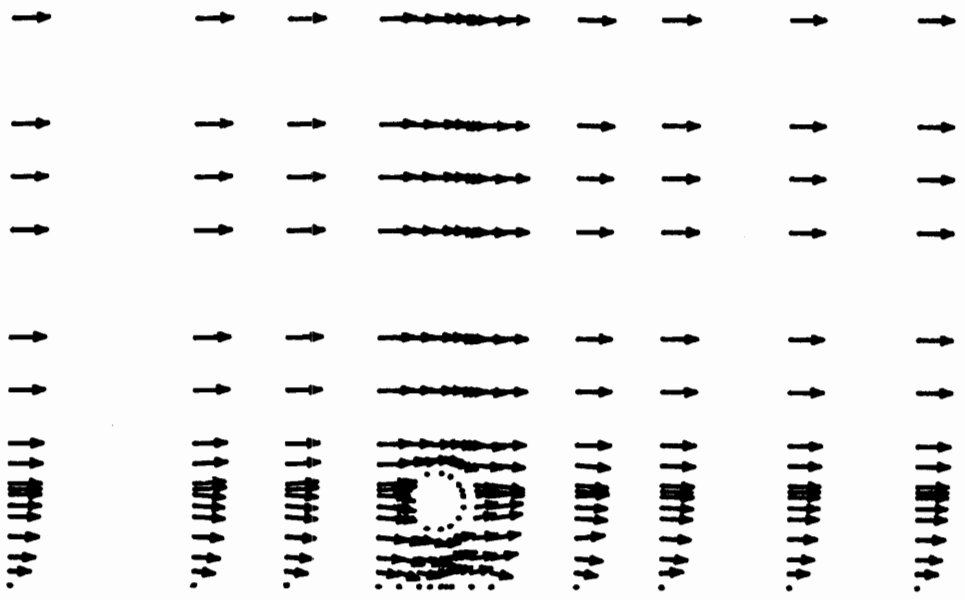


Figure 5.11(a) Vertical velocity distribution at y=0, profile B, gap ratio 1.

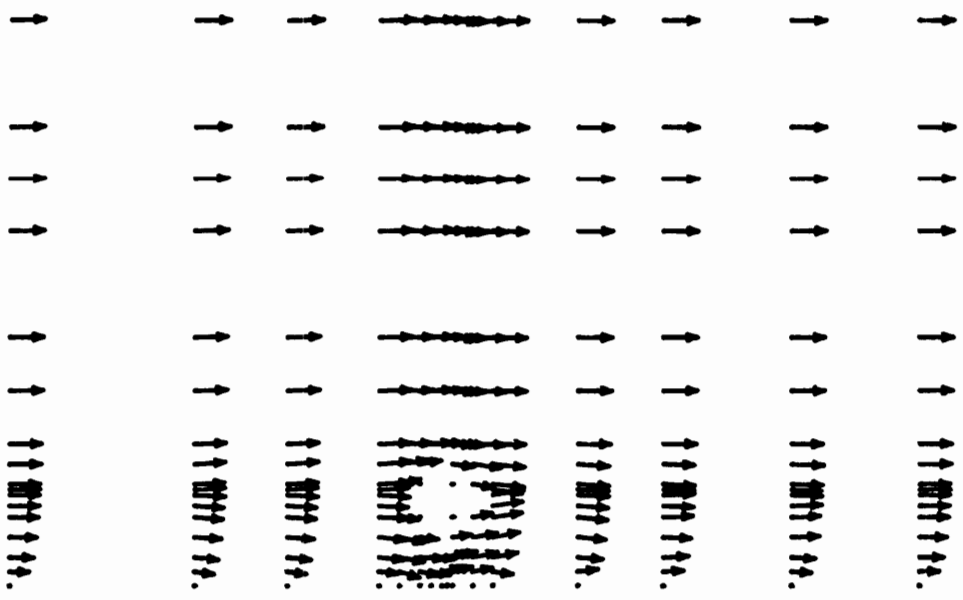


Figure 5.11(b) Vertical velocity distribution at $y=0.5D_S$, profile B, gap ratio 1.

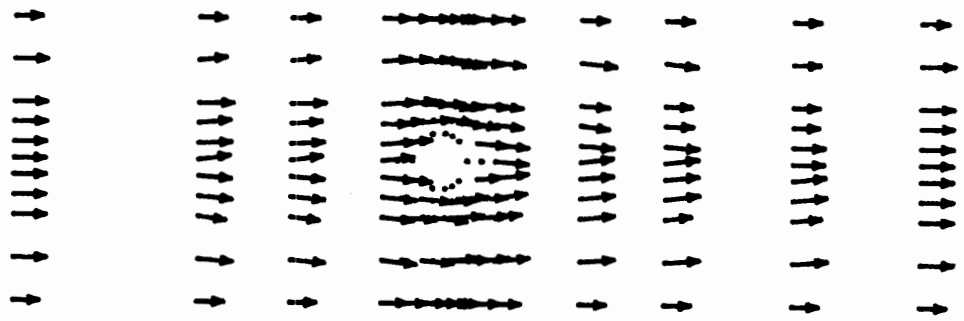


Figure 5.11(c) Horizontal velocity distribution at z=0, profile B, gap ratio 1.

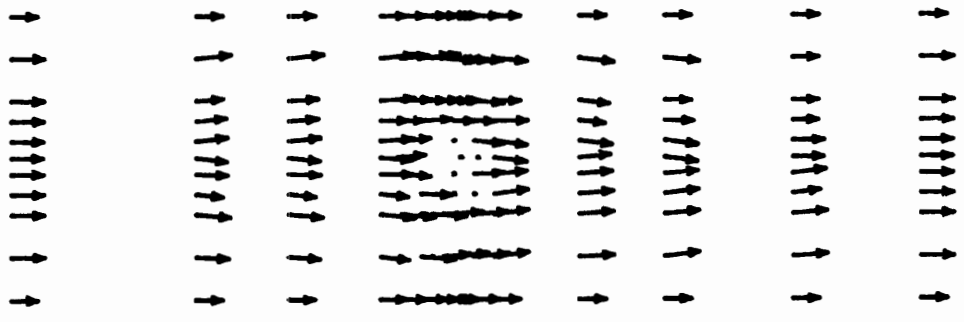


Figure 5.11(d) Horizontal velocity distribution at $z=0.5D_S$, profile B, gap ratio 1.

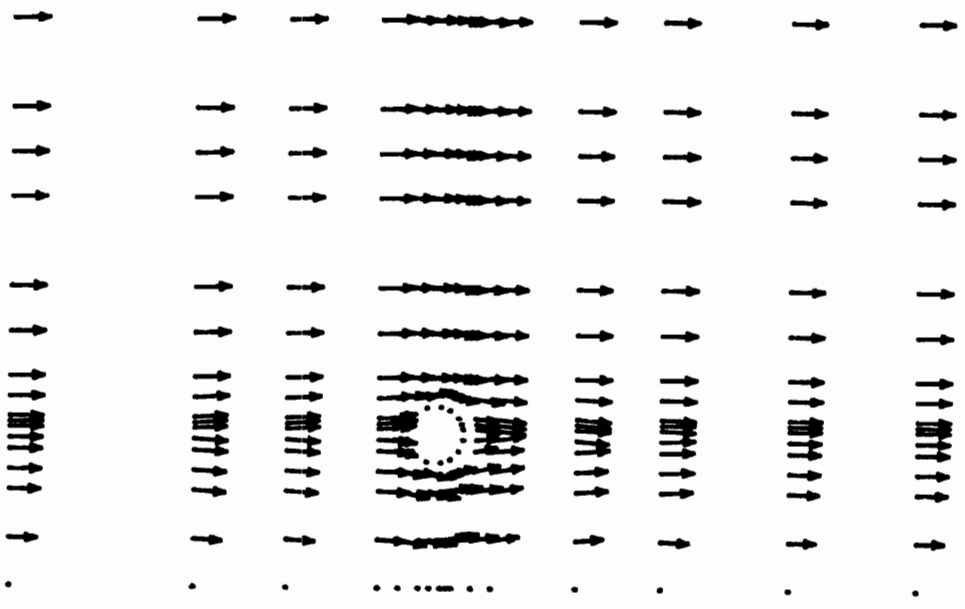


Figure 5.12(a) Vertical velocity distribution at y=0, profile B, gap ratio 2.

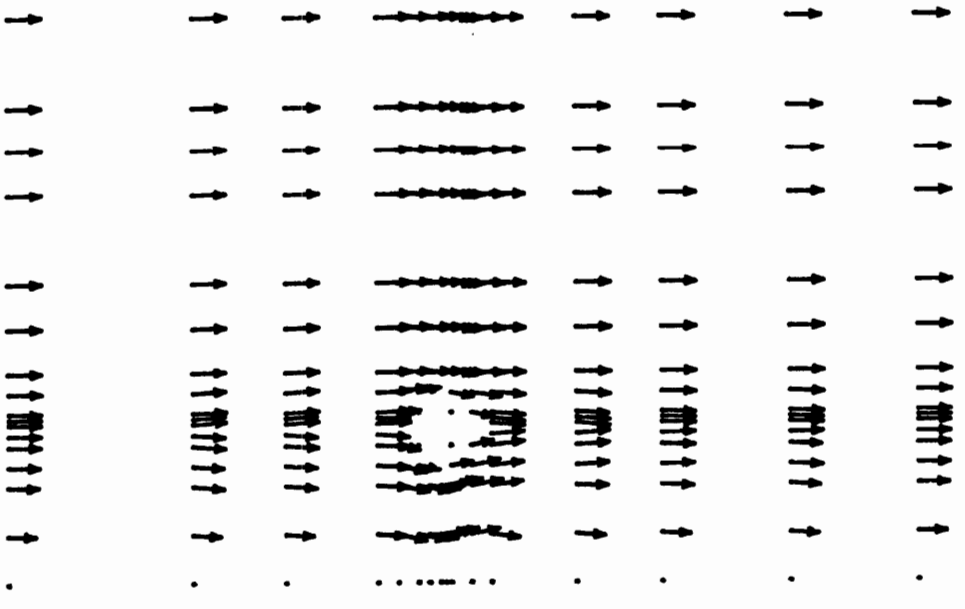


Figure 5.12(b) Vertical velocity distribution at y=0.5D_S, profile B, gap ratio 2.

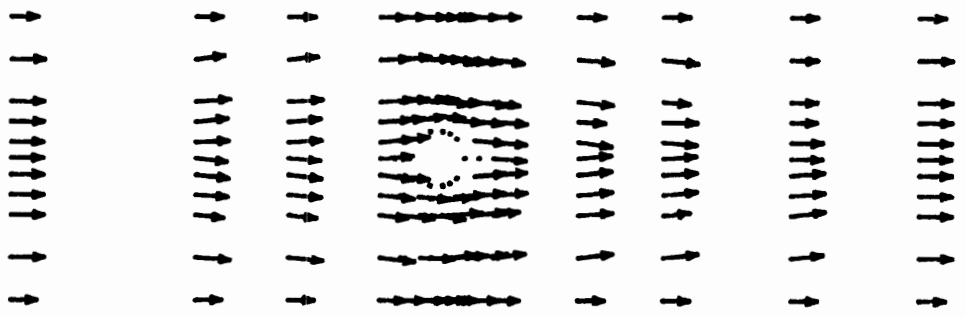


Figure 5.12(c) Horizontal velocity distribution at z=0, profile B, gap ratio 2.

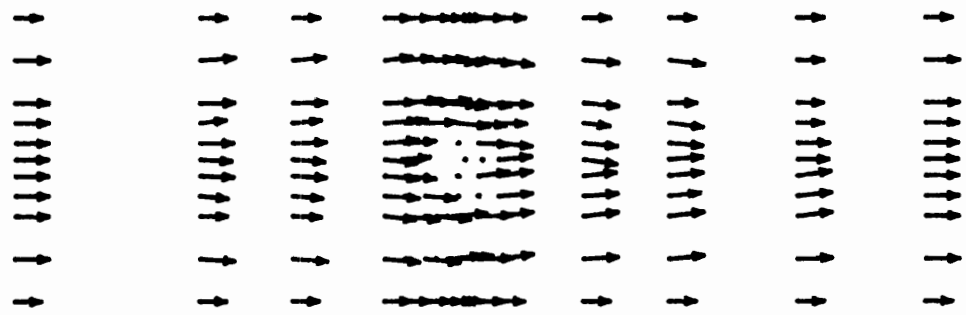


Figure 5.12(d) Horizontal velocity distribution at $z=0.5D_S$, profile B, gap ratio 2.

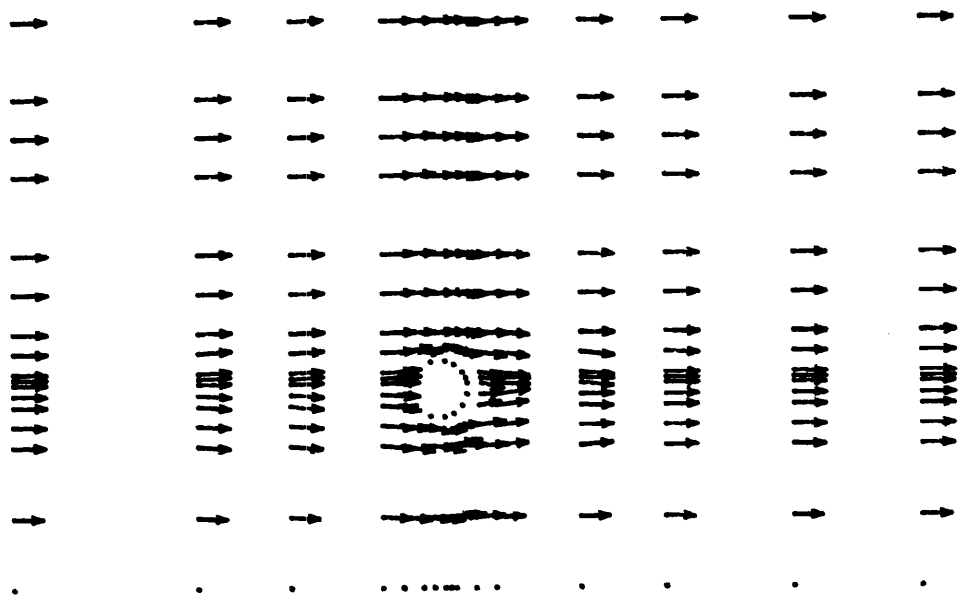


Figure 5.13(a) Vertical velocity distribution at y=0, profile B, gap ratio 3.

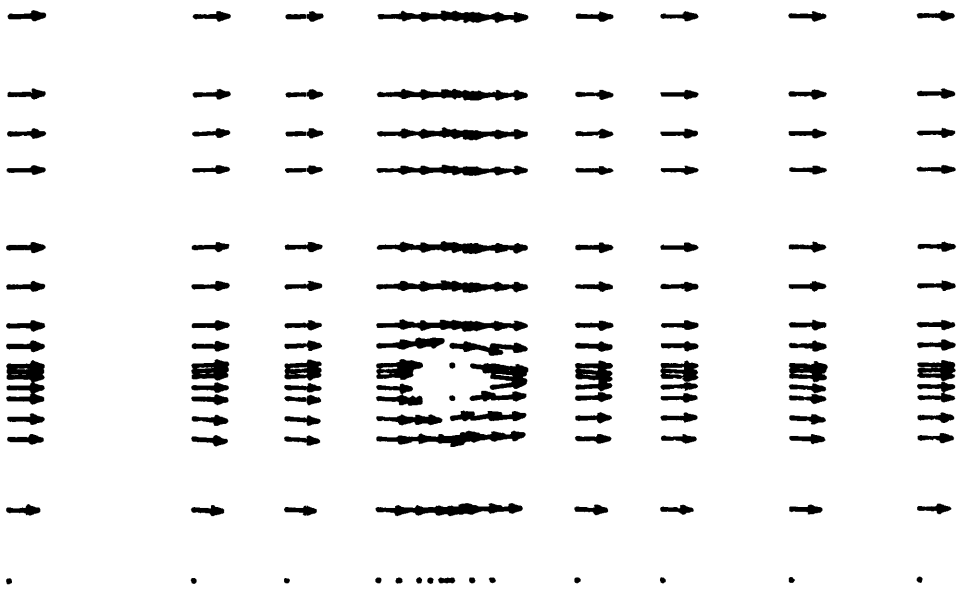


Figure 5.13(b) Vertical velocity distribution at y=0.5D_s, profile B, gap ratio 3.

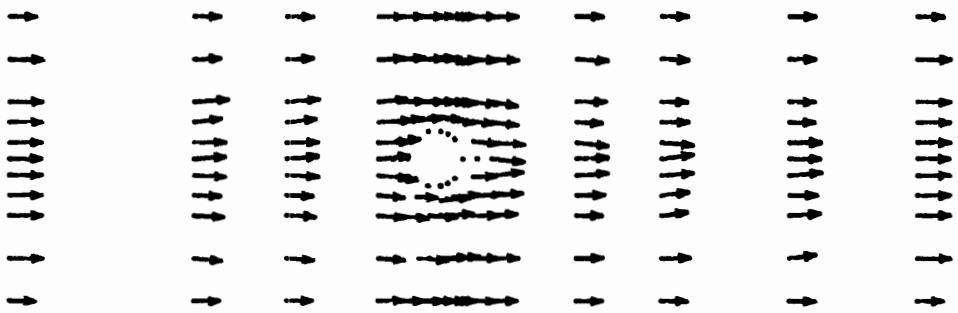


Figure 5.13(c) Horizontal velocity distribution at z=0, profile B, gap ratio 3.

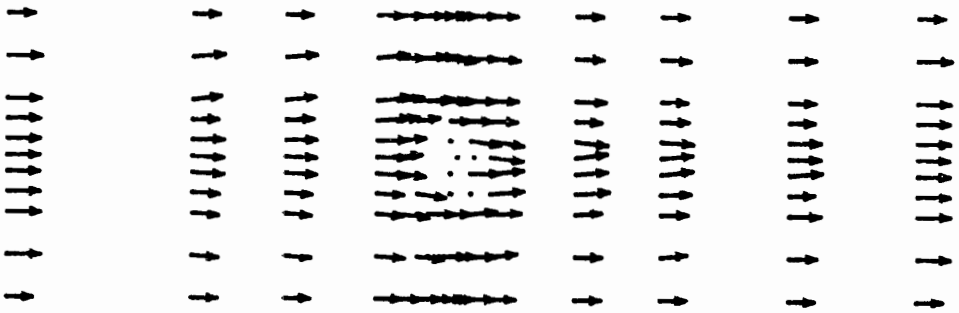


Figure 5.13(d) Horizontal velocity distribution at $z=0.5D_S$, profile B, gap ratio 3.

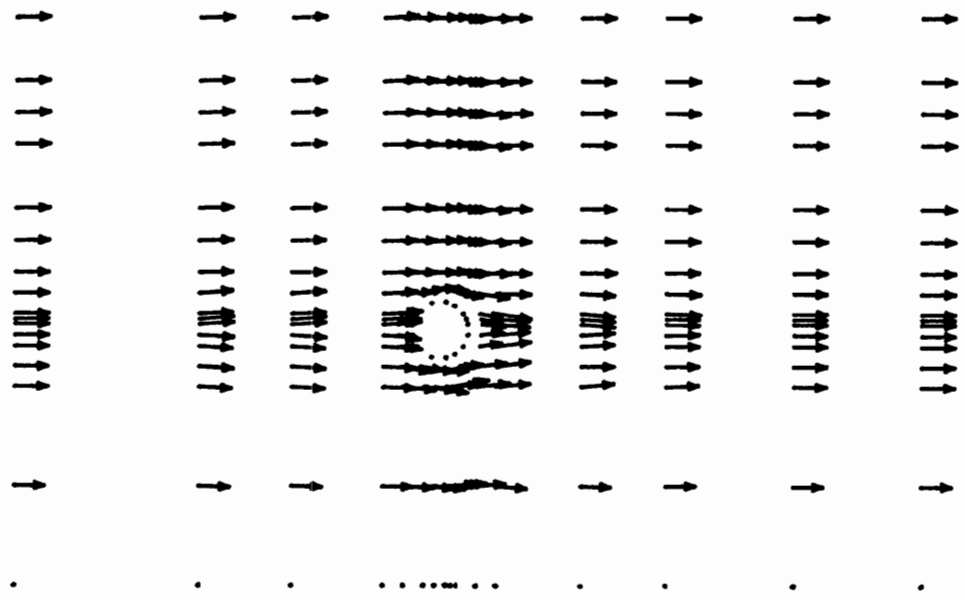


Figure 5.14(a) Vertical velocity distribution at y=0, profile B, gap ratio 4.

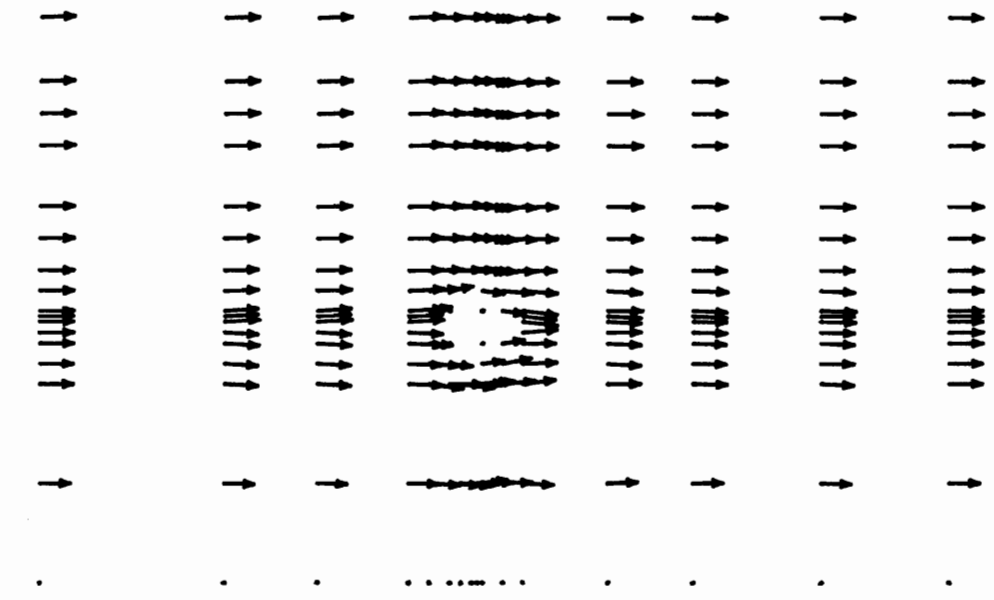


Figure 5.14(b) Vertical velocity distribution at y=0.5D_S, profile B, gap ratio 4.

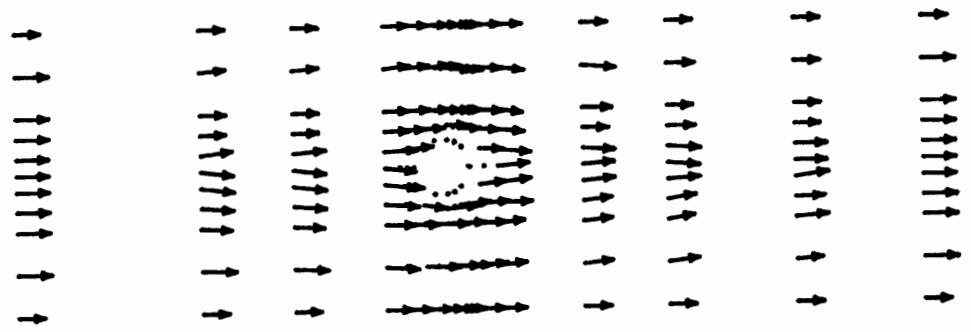


Figure 5.14(c) Horizontal velocity distribution at z=0, profile B, gap ratio 4.

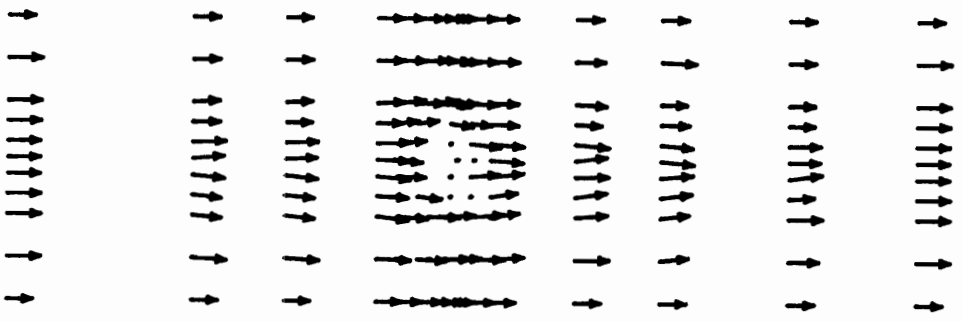


Figure 5.14(d) Horizontal velocity distribution at $z=0.5D_S$, profile B, gap ratio 4.

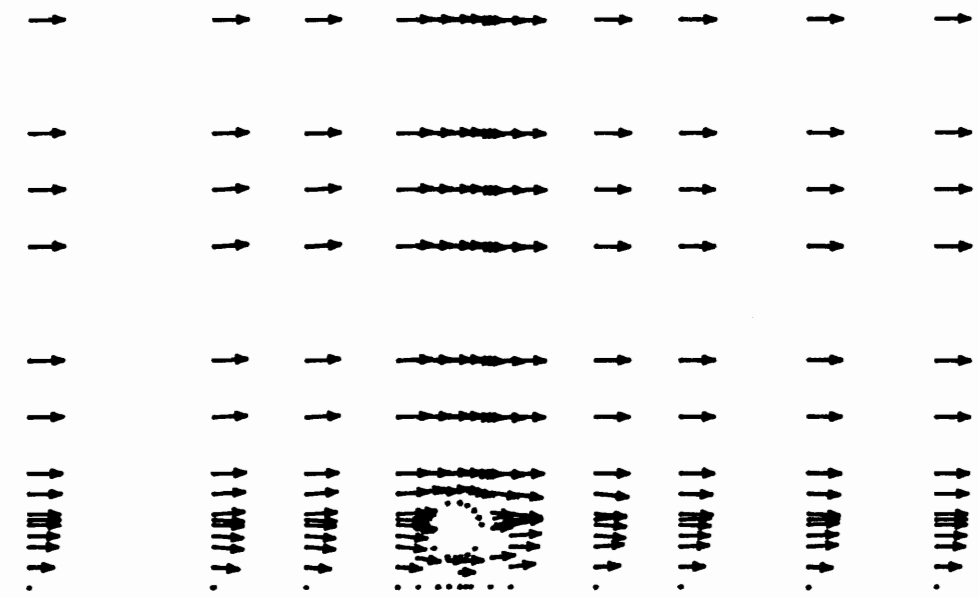


Figure 5.15(a) Vertical velocity distribution at y=0, profile C, gap ratio 0.5.

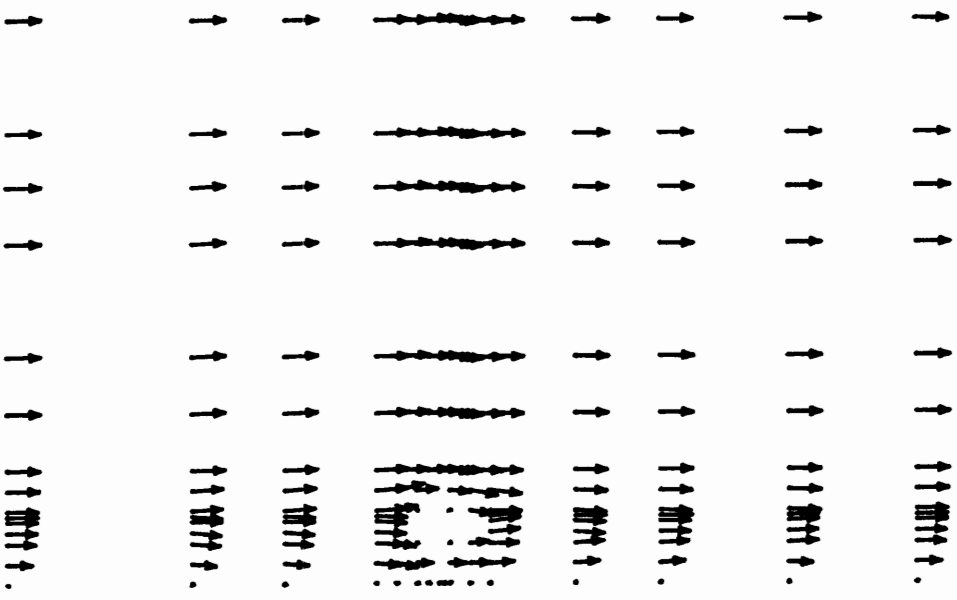


Figure 5.15(b) Vertical velocity distribution at y=0.5D_S, profile C, gap ratio 0.5.

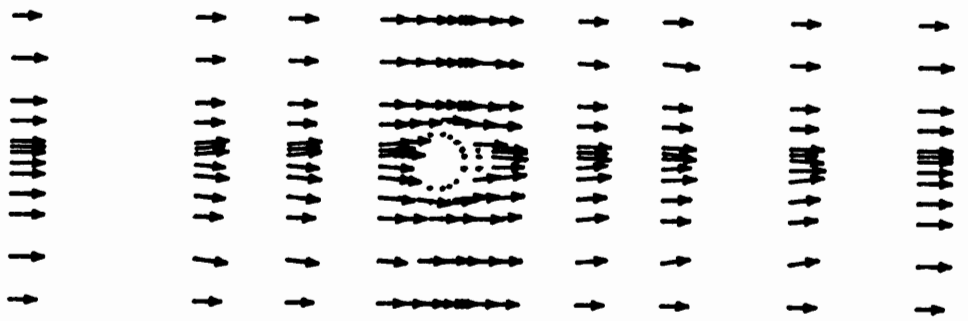


Figure 5.15(c) Horizontal velocity distribution at z=0, profile C, gap ratio 0.5.

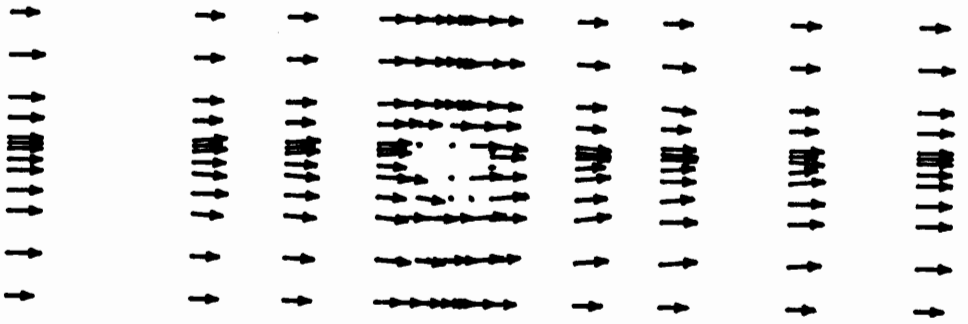


Figure 5.15(d) Horizontal velocity distribution at z=.5D_S, profile C, gap ratio 0.5.

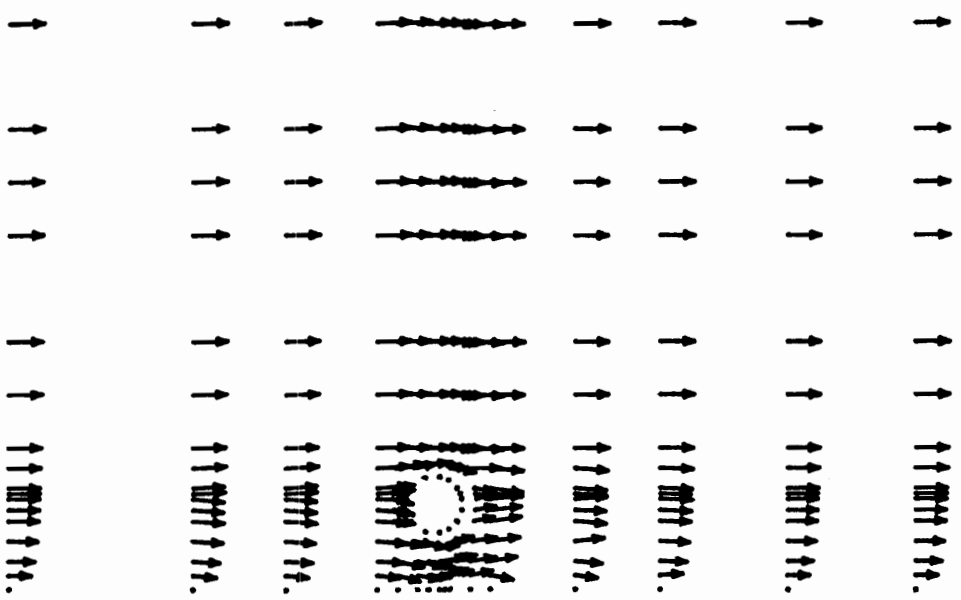


Figure 5.16(a) Vertical velocity distribution at y=0, profile C, gap ratio 1.

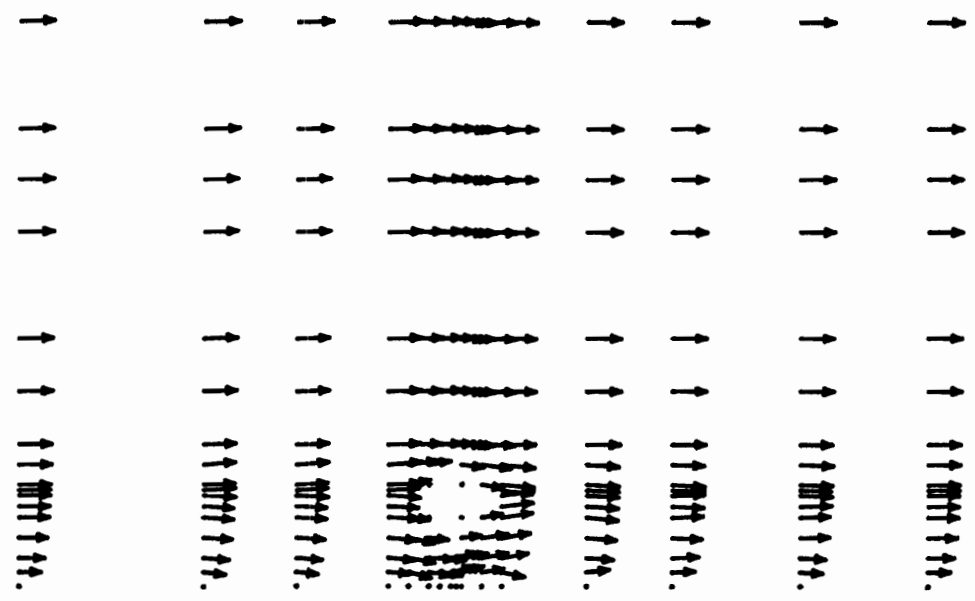


Figure 5.16(b) Vertical velocity distribution at y=0.5D_s, profile C, gap ratio 1.

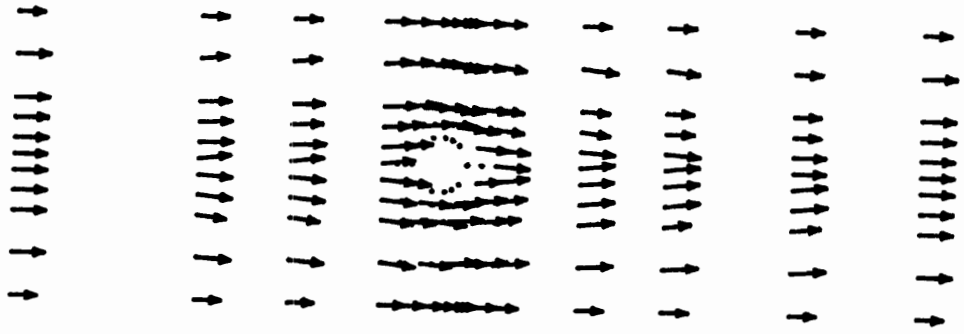


Figure 5.16(c) Horizontal velocity distribution at z=0, profile C, gap ratio 1.

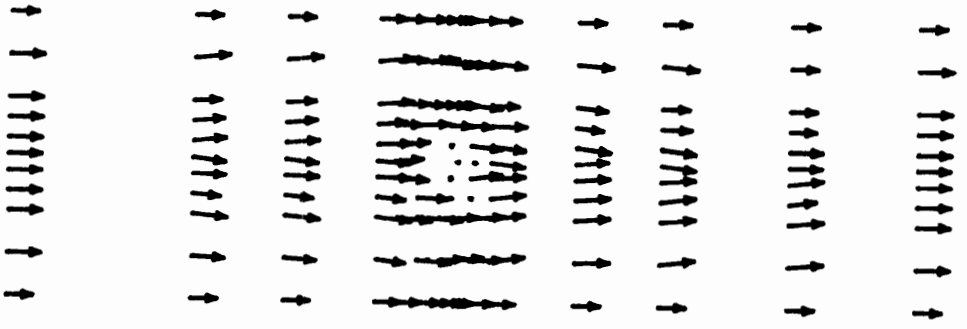


Figure 5.16(d) Horizontal velocity distribution at $z=0.5D_S$, profile C, gap ratio 1.

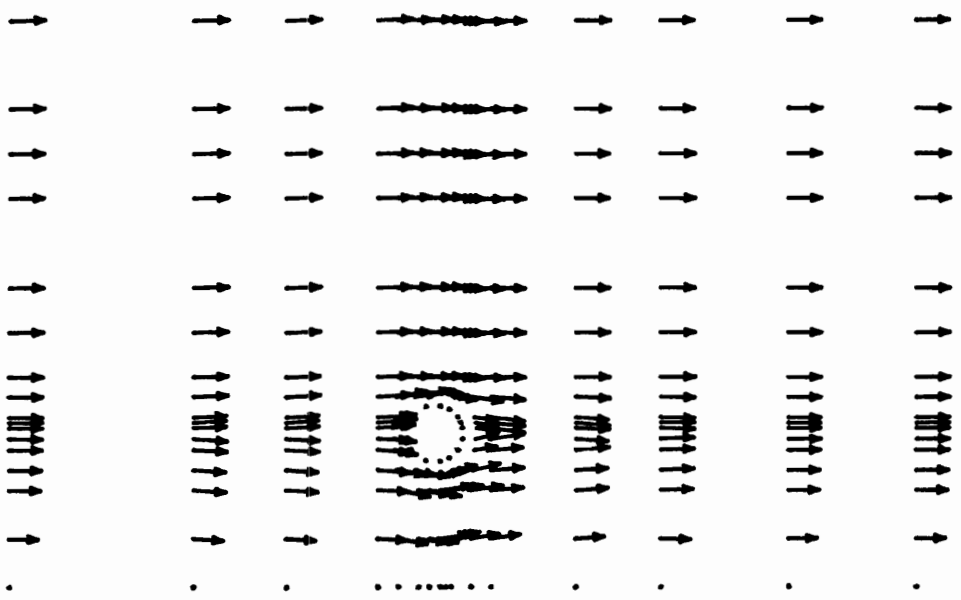


Figure 5.17(a) Vertical velocity distribution at y=0, profile C, gap ratio 2.

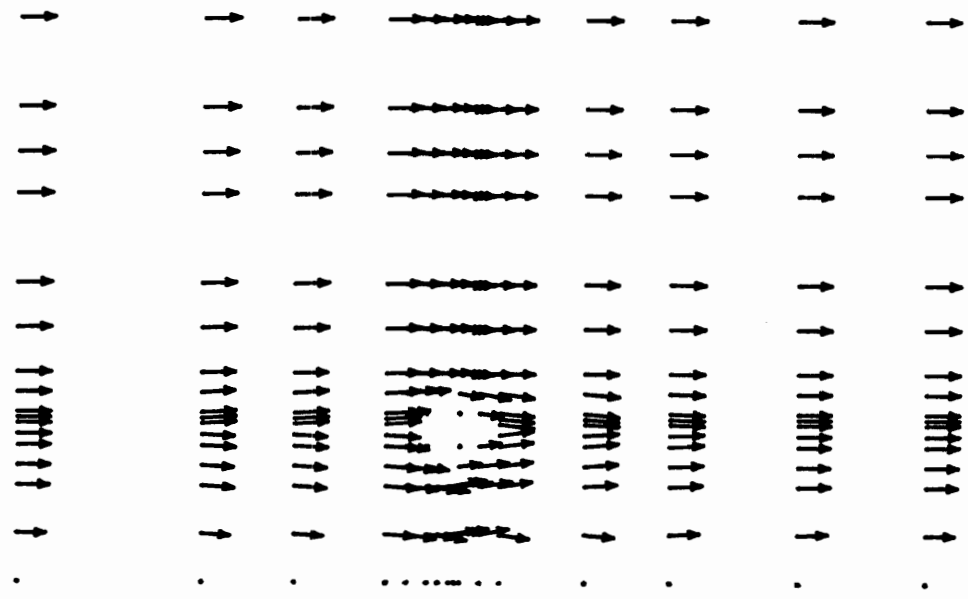


Figure 5.17(b) Vertical velocity distribution at y=0.5D_S, profile C, gap ratio 2.

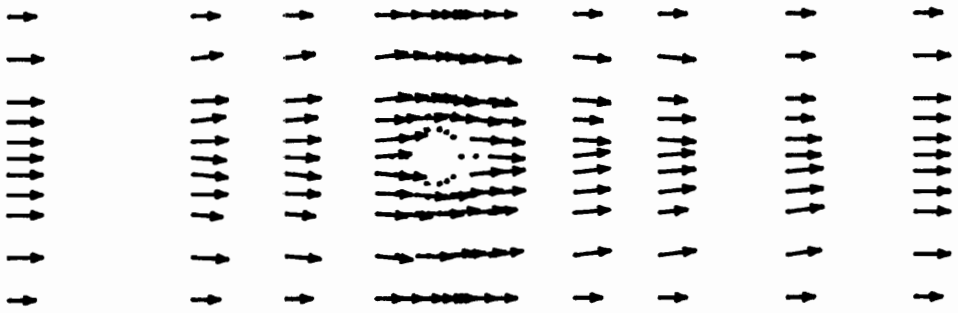


Figure 5.17(c) Horizontal velocity distribution at z=0, profile C, gap ratio 2.

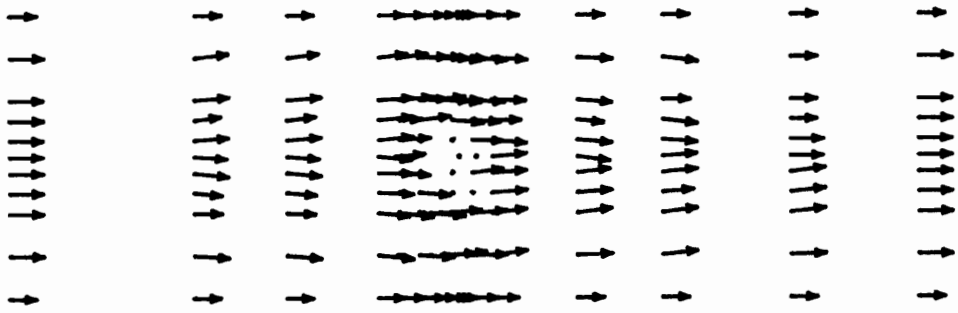


Figure 5.17(d) Horizontal velocity distribution at z=0.5D_S, profile C, gap ratio 2.

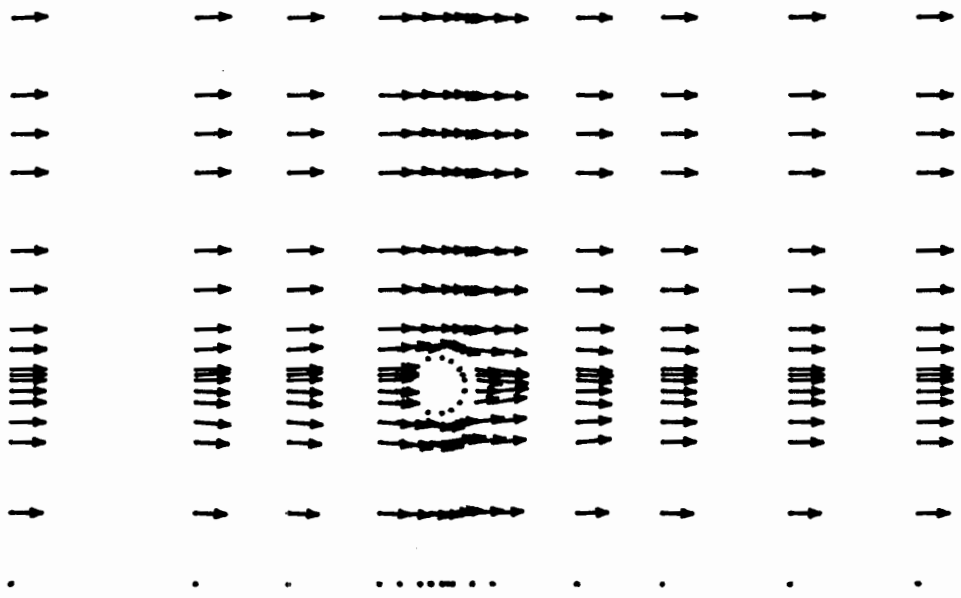


Figure 5.18(a) Vertical velocity distribution at y=0, profile C, gap ratio 3.

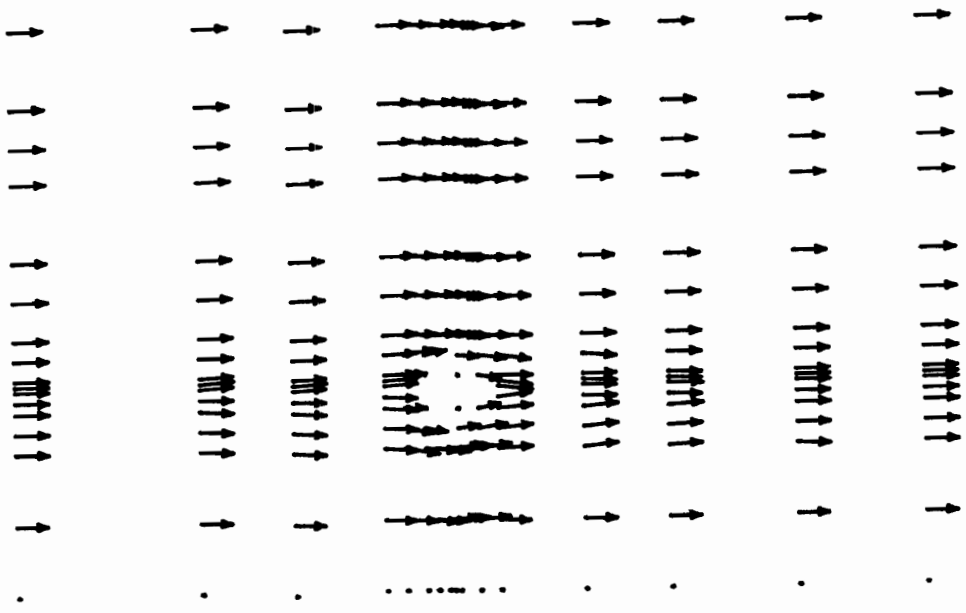


Figure 5.18(b) Vertical velocity distribution at y=0.5D_S, profile C, gap ratio 3.

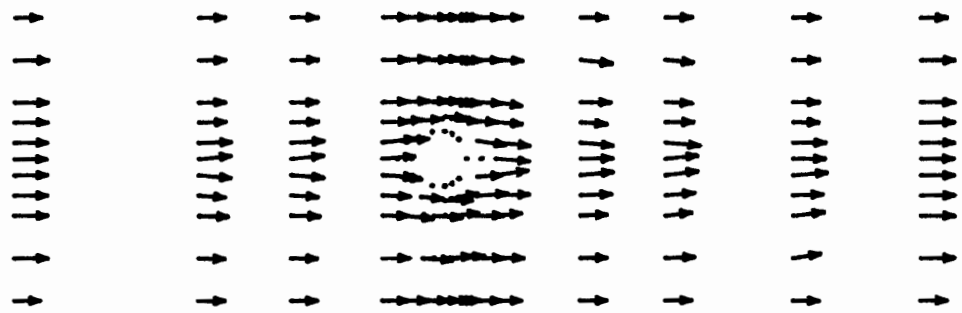


Figure 5.18(c) Horizontal velocity distribution at z=0, profile C, gap ratio 3.

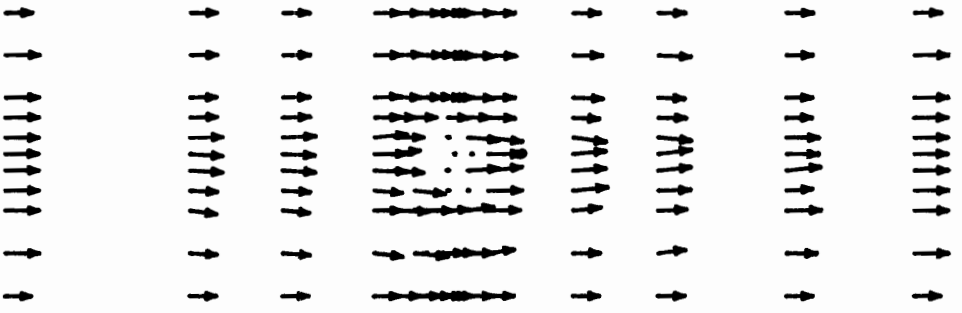


Figure 5.18(d) Horizontal velocity distribution at $z=0.5D_S$, profile C, gap ratio 3.

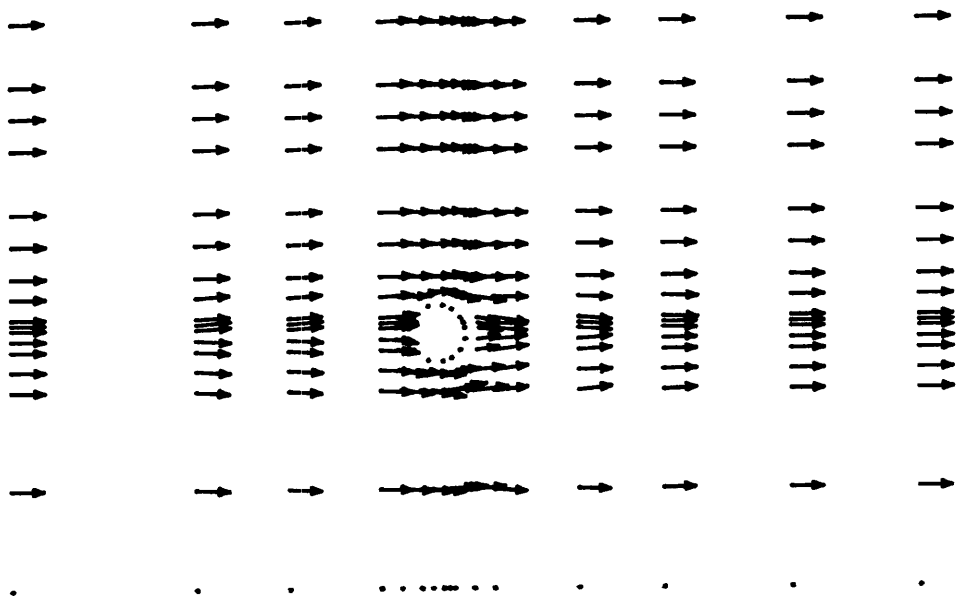


Figure 5.19(a) Vertical velocity distribution at y=0, profile C, gap ratio 4.

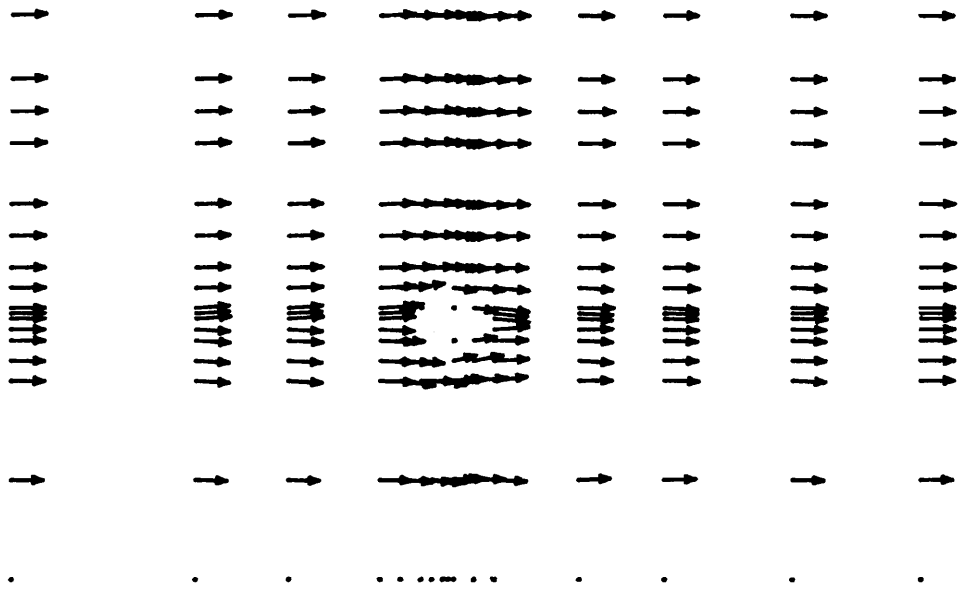


Figure 5.19(b) Vertical velocity distribution at y=0.5D_S, profile C, gap ratio 4.

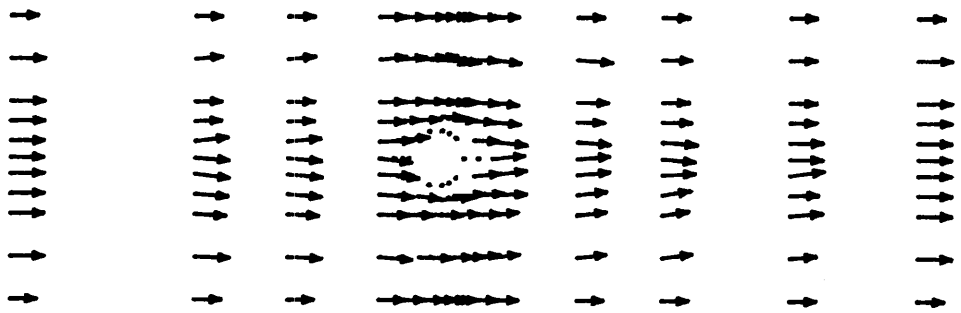


Figure 5.19(c) Horizontal velocity distribution at z=0, profile C, gap ratio 4.

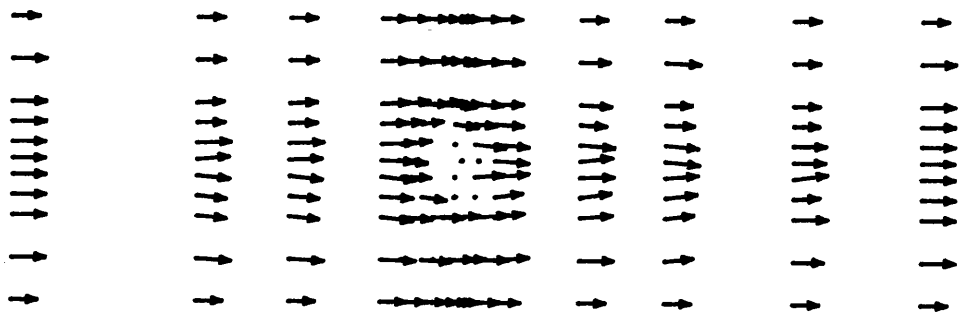


Figure 5.19(d) Horizontal velocity distribution at $z=0.5D_S$, profile C, gap ratio 4.

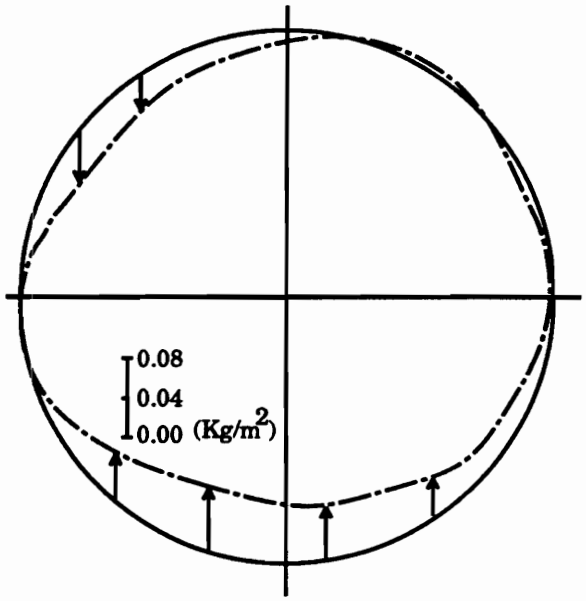


Figure 5.20(a) Distribution of vertical component of stress for velocity profile A, gap ratio 0.1.

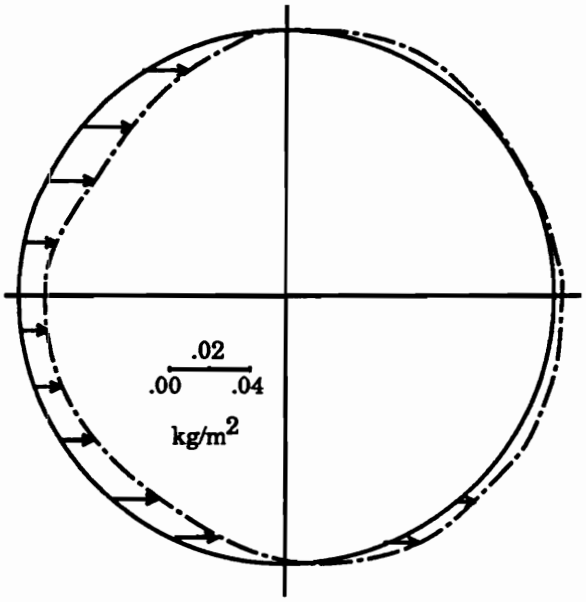


Figure 5.20(b) Distribution of longitudinal component of stress for velocity profile A, gap ratio 0.1.

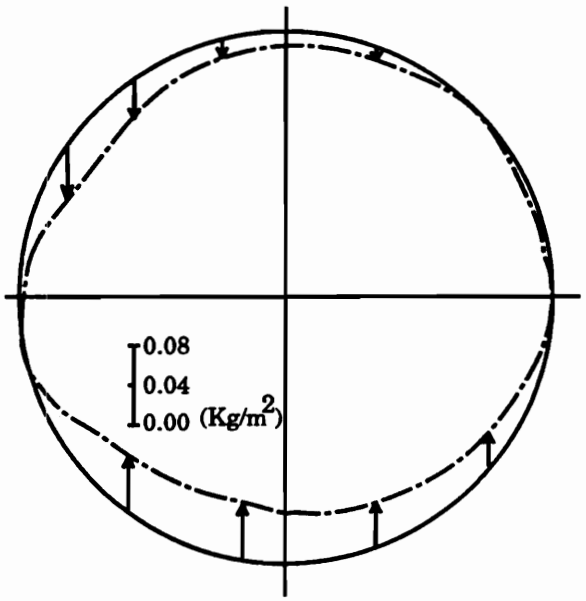


Figure 5.21(a) Distribution of vertical component of stress for velocity profile A, gap ratio 0.5.

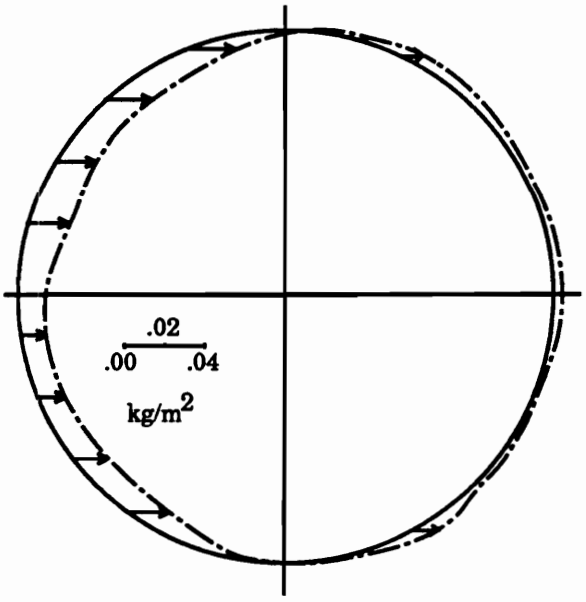


Figure 5.21(b) Distribution of longitudinal component of stress for velocity profile A, gap ratio 0.5.

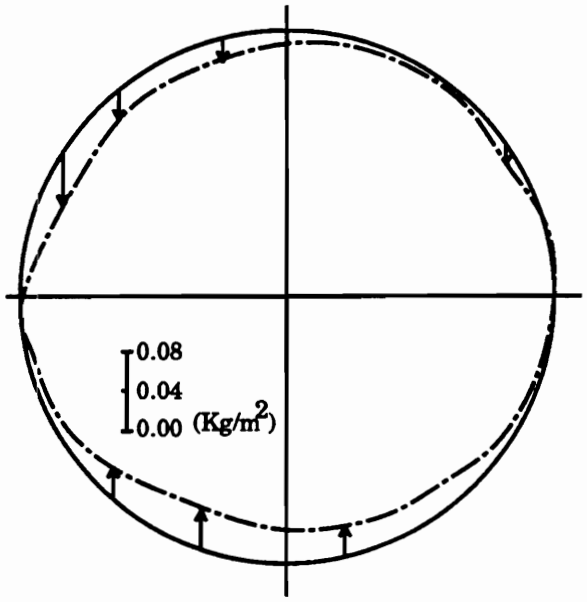


Figure 5.22 (a) Distribution of vertical component of stress for velocity profile A, gap ratio 1.

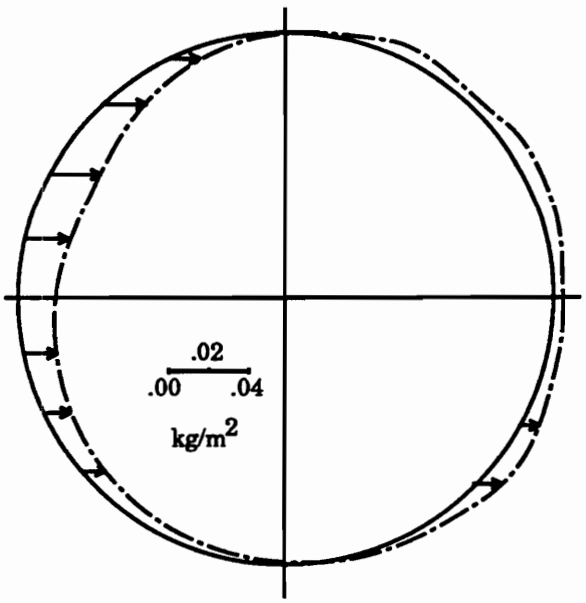


Figure 5.22(b) Distribution of longitudinal component of stress for velocity profile A, gap ratio 1.

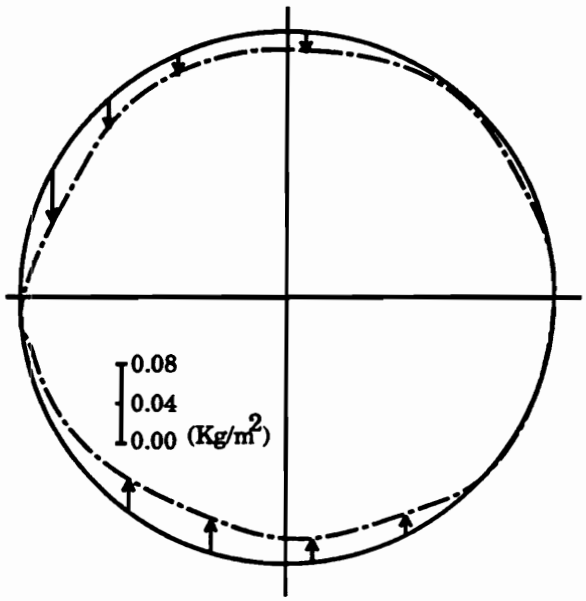


Figure 5.23(a) Distribution of vertical component of stress for velocity profile A, gap ratio 2.

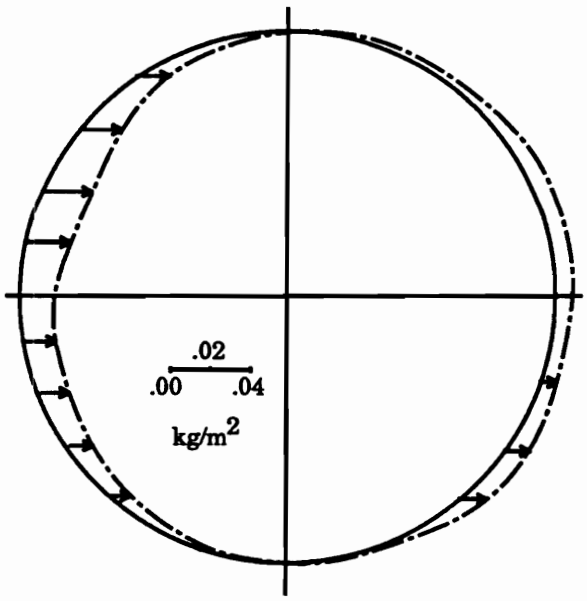


Figure 5.23(b) Distribution of longitudinal component of stress for velocity profile A, gap ratio 2.

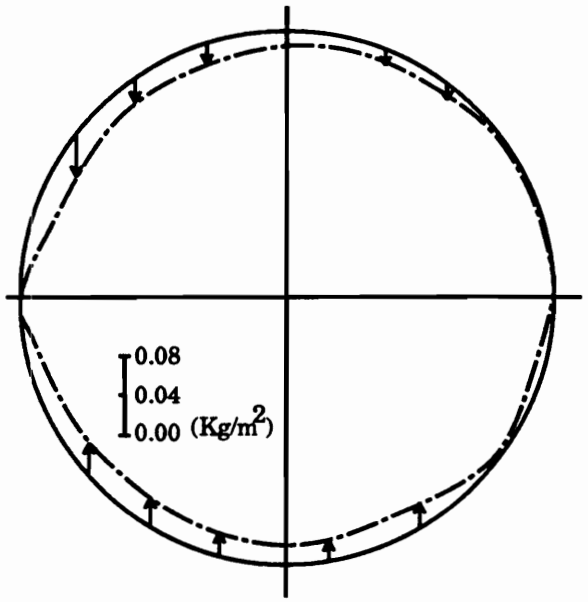


Figure 5.24(a) Distribution of vertical component of stress for velocity profile A, gap ratio 3.

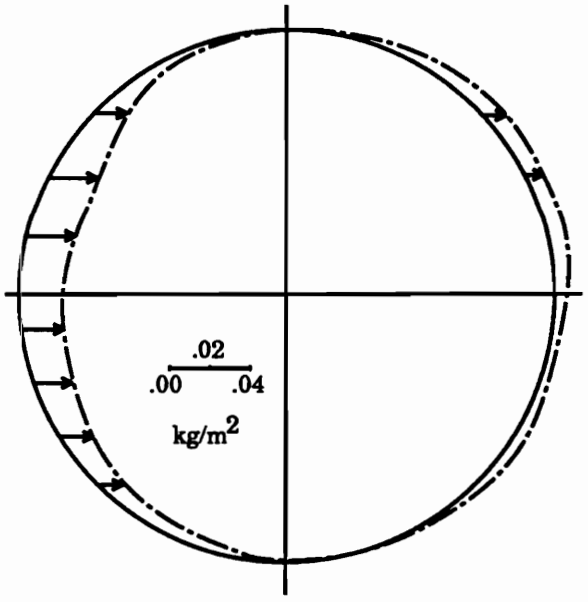


Figure 5.24(b) Distribution of longitudinal component of stress for velocity profile A, gap ratio 3.

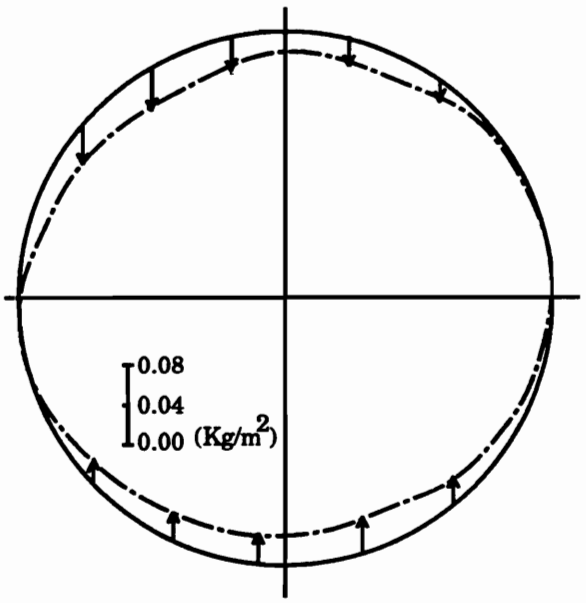


Figure 5.25(a) Distribution of vertical component of stress for velocity profile A, gap ratio 4.

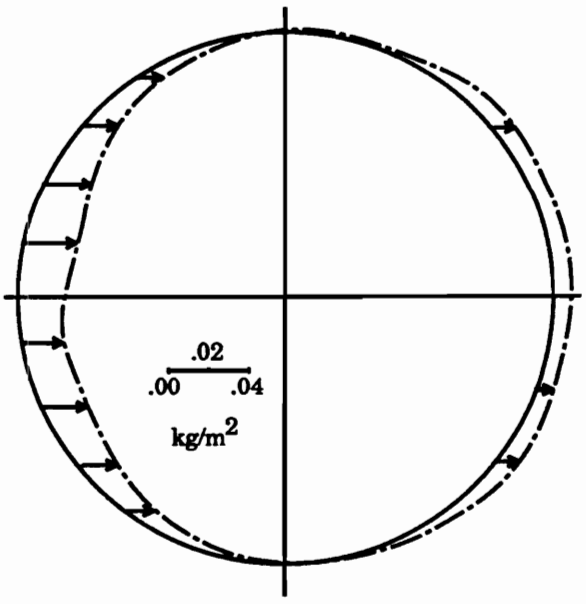


Figure 5.25(b) Distribution of longitudinal component of stress for velocity profile A, gap ratio 4.

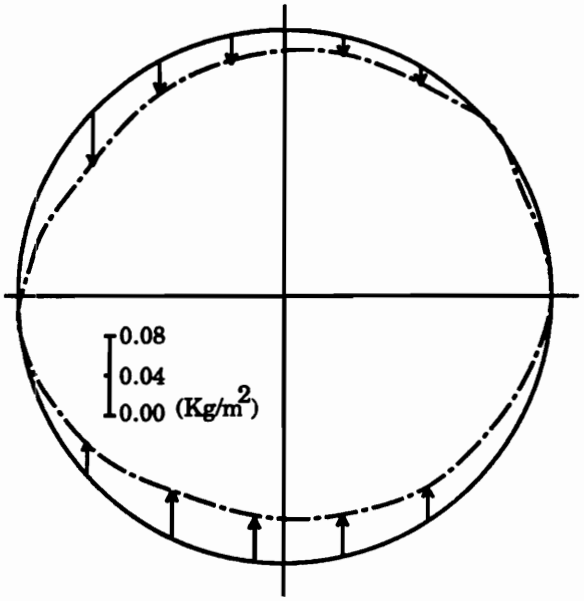


Figure 5.26(a) Distribution of vertical component of stress for velocity profile B, gap ratio 0.5.

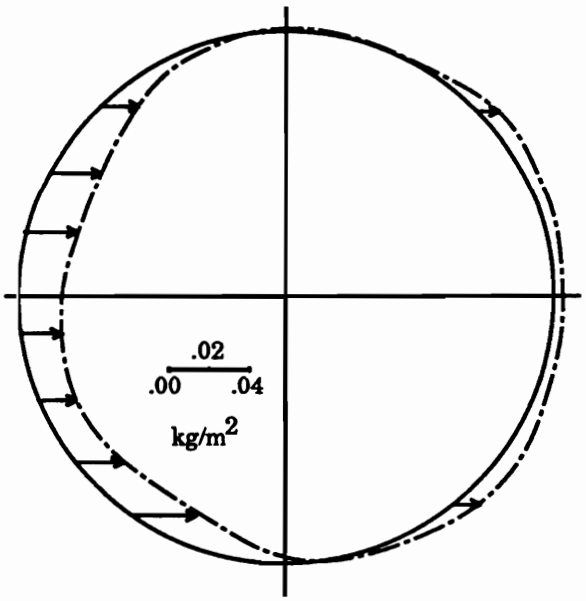


Figure 5.26(b) Distribution of longitudinal component of stress for velocity profile B, gap ratio 0.5.

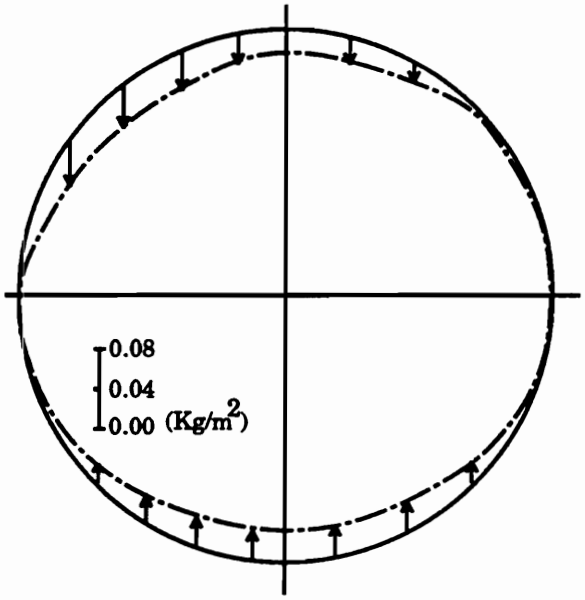


Figure 5.27(a) Distribution of vertical component of stress for velocity profile B, gap ratio 1.

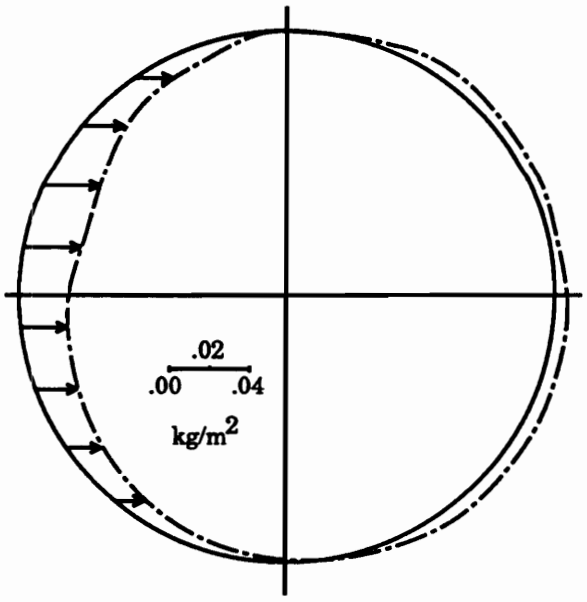


Figure 5.27(b) Distribution of longitudinal component of stress for velocity profile B, gap ratio 1.

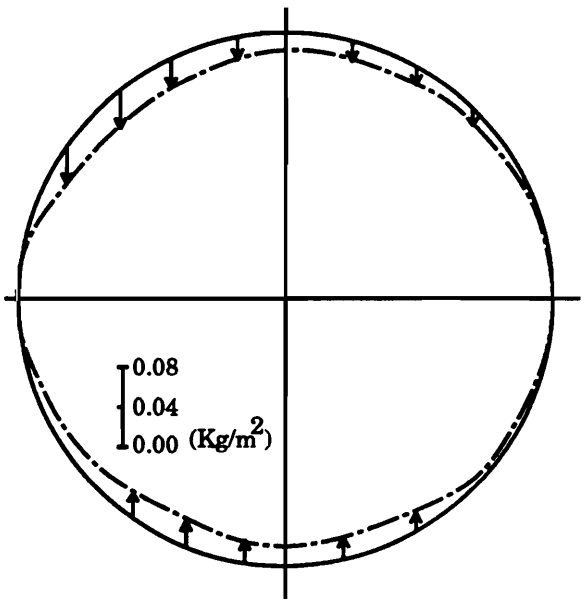


Figure 5.28(a) Distribution of vertical component of stress for velocity profile B, gap ratio 2.

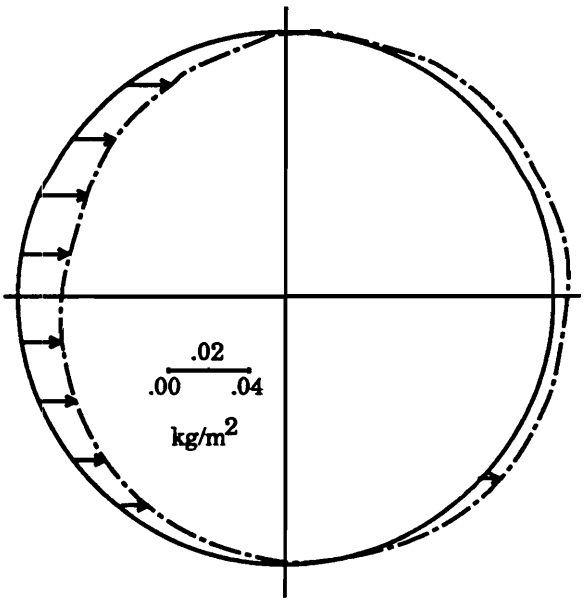


Figure 5.28(b) Distribution of longitudinal component of stress for velocity profile B, gap ratio 2.

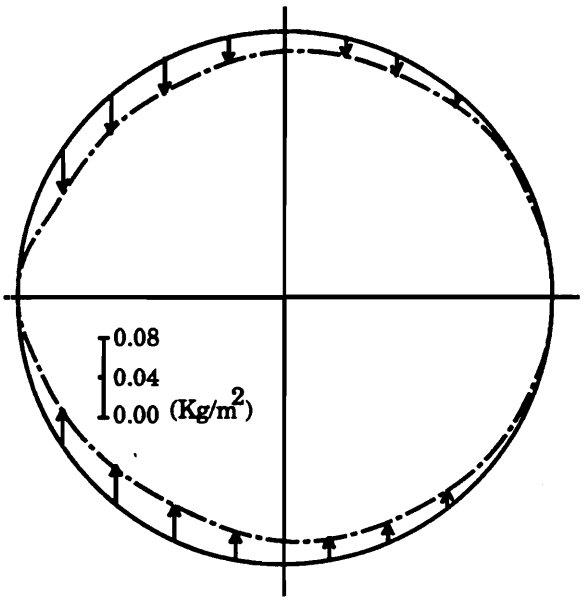


Figure 5.29(a) Distribution of vertical component of stress for velocity profile B, gap ratio 3.

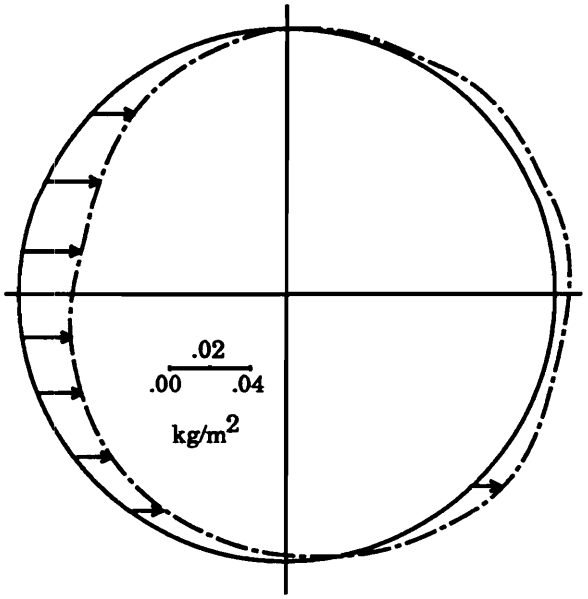


Figure 5.29(b) Distribution of longitudinal component of stress for velocity profile B, gap ratio 3.

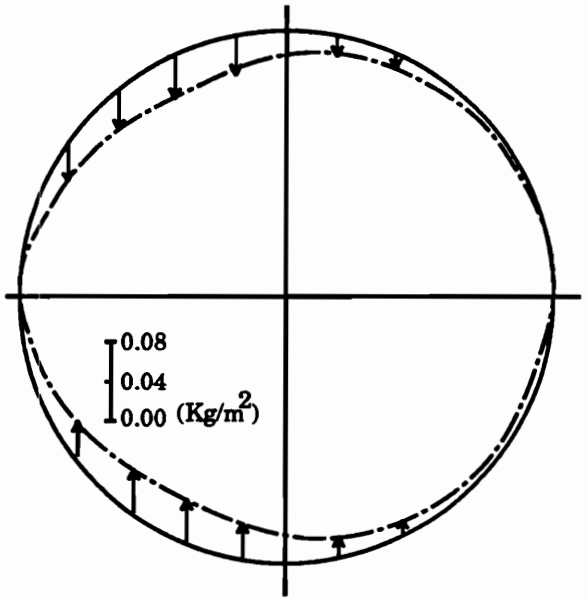


Figure 5.30(a) Distribution of vertical component of stress for velocity B, gap ratio 4.

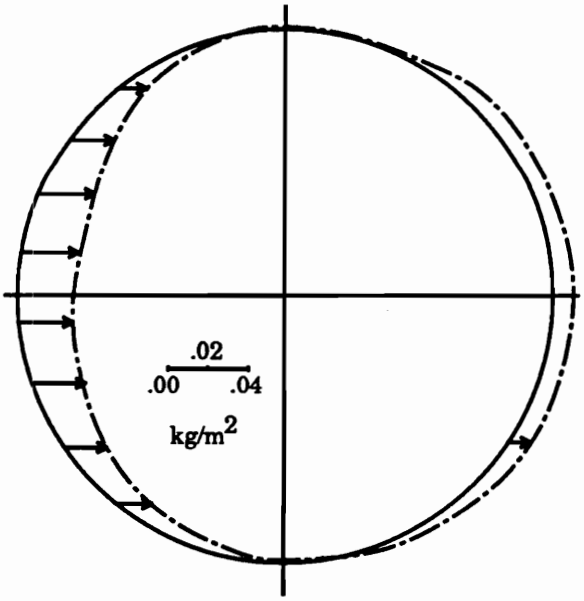


Figure 5.30(b) Distribution of longitudinal component of stress for velocity profile B, gap ratio 4.

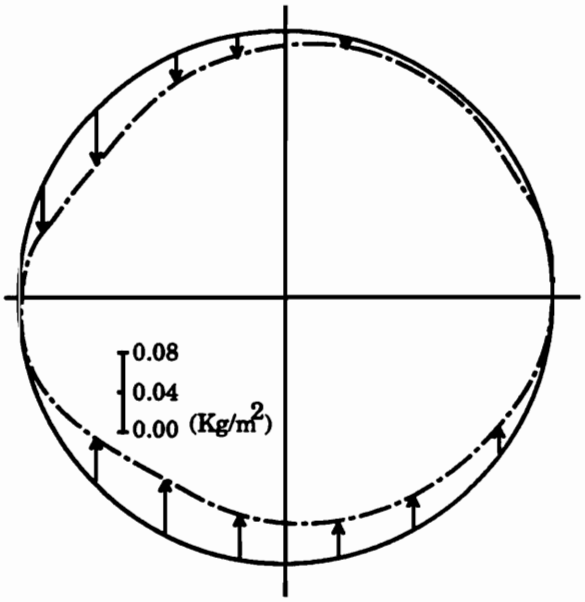


Figure 5.31(a) Distribution of vertical component of stress for velocity profile C, gap ratio 0.5.

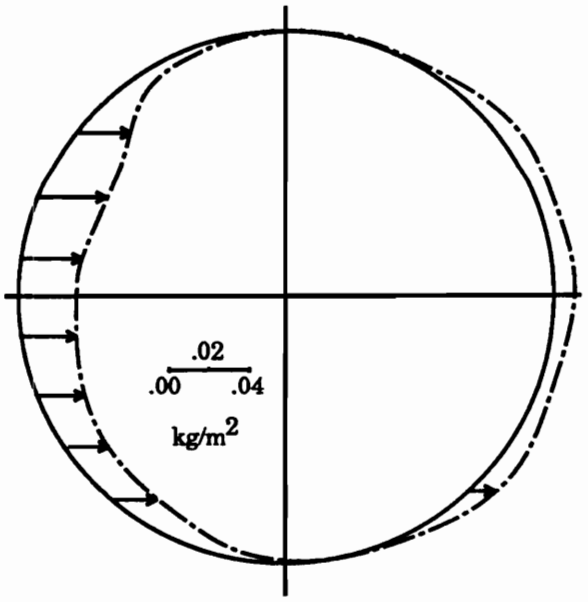


Figure 5.31(b) Distribution of longitudinal component of stress for velocity profile C, gap ratio 0.5.

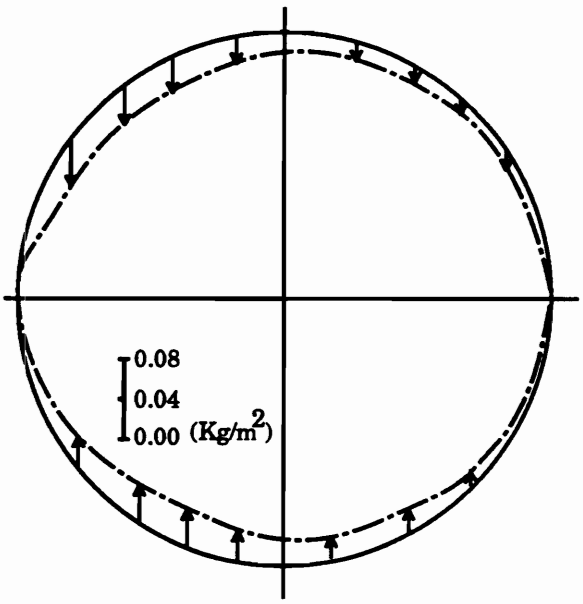


Figure 5.32(a) Distribution of vertical component of stress for velocity profile C, gap ratio 1.

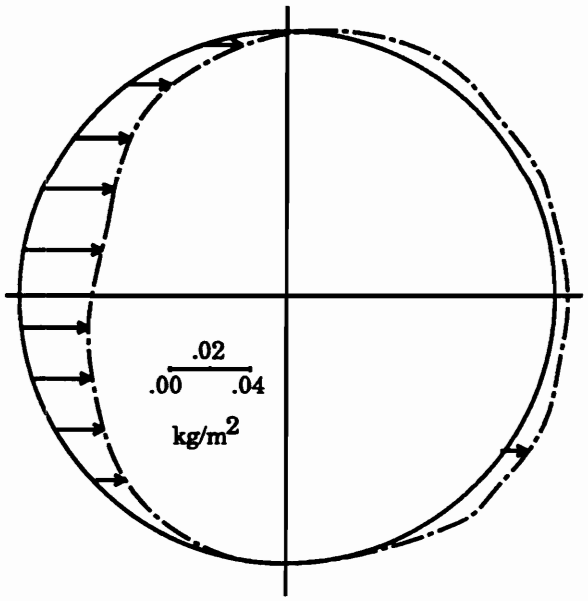


Figure 5.32(b) Distribution of longitudinal component of stress for velocity profile C, gap ratio 1.

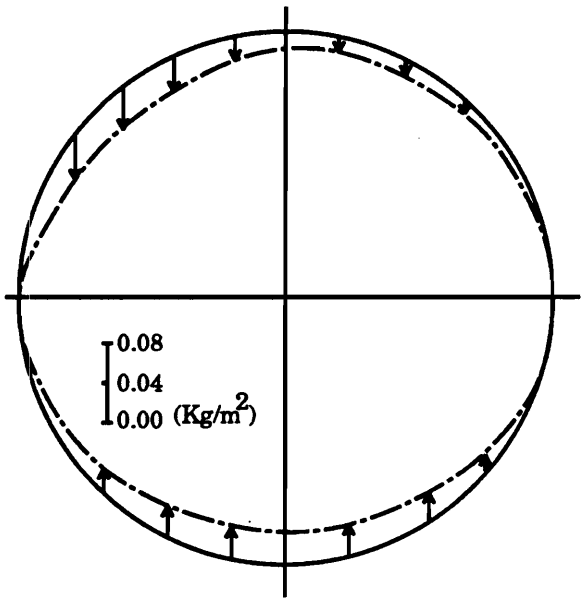


Figure 5.33(a) Distribution of vertical component of stress for velocity profile C, gap ratio 2.

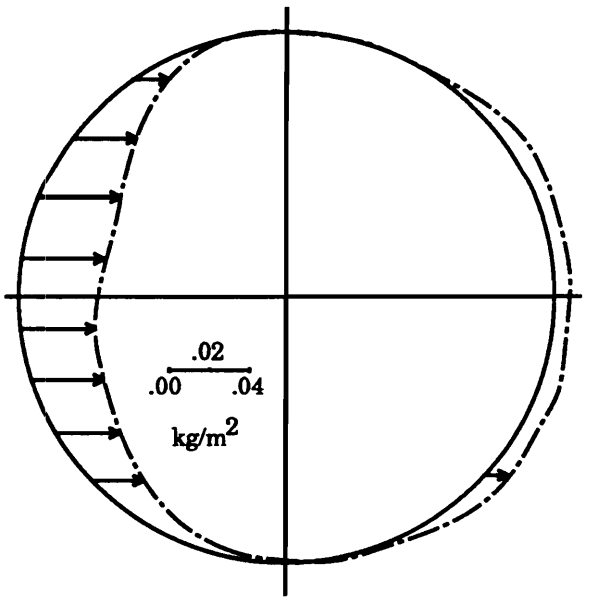


Figure 5.33(b) Distribution of longitudinal component of stress for velocity profile C, gap ratio 2.

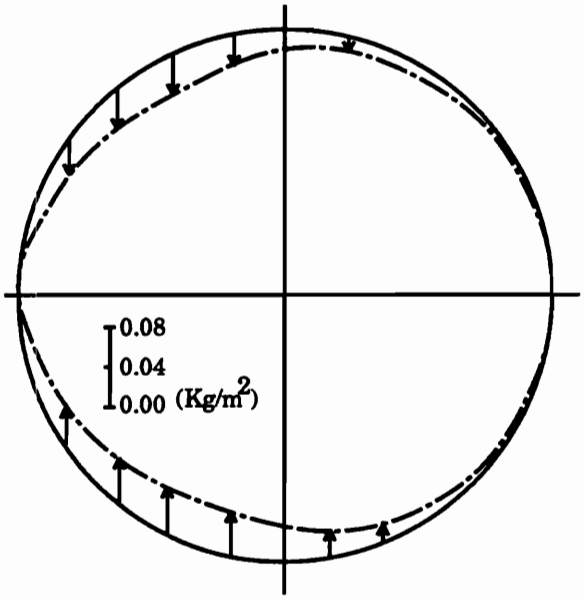


Figure 5.34(a) Distribution of vertical component of stress for velocity profile C, gap ratio 3.

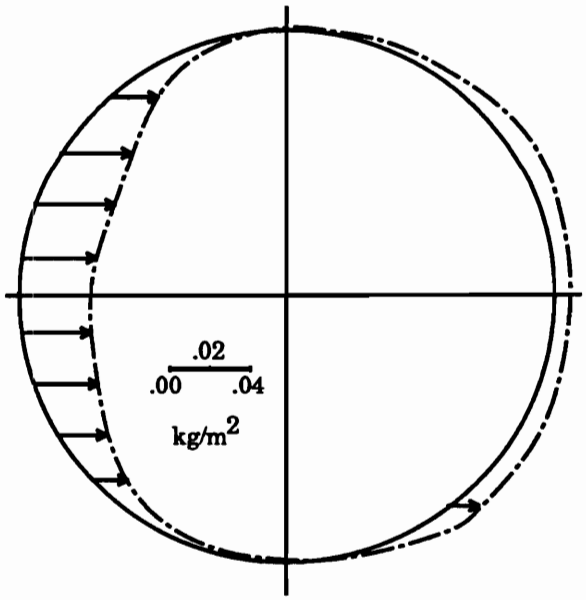


Figure 5.34(b) Distribution of longitudinal component of stress for velocity profile C, gap ratio 3.

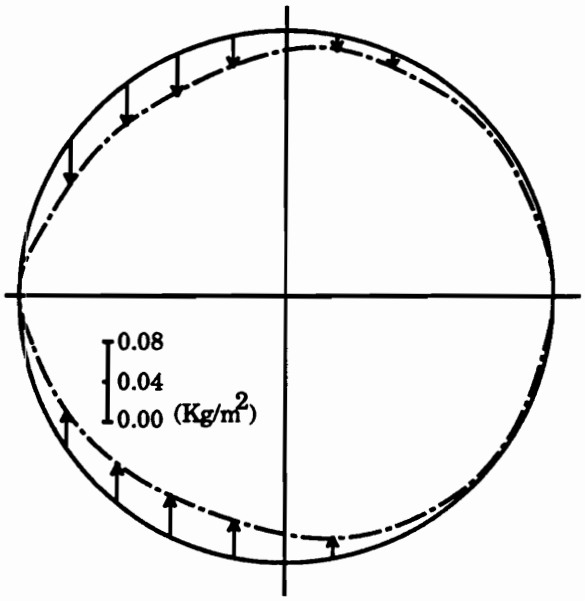


Figure 5.35(a) Distribution of vertical component of stress for velocity profile C, gap ratio 4.

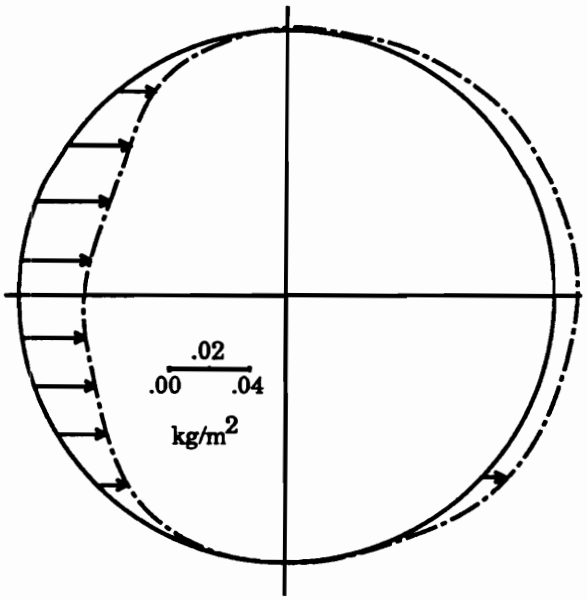


Figure 5.35(b) Distribution of longitudinal component of stress for velocity profile C, gap ratio 4.

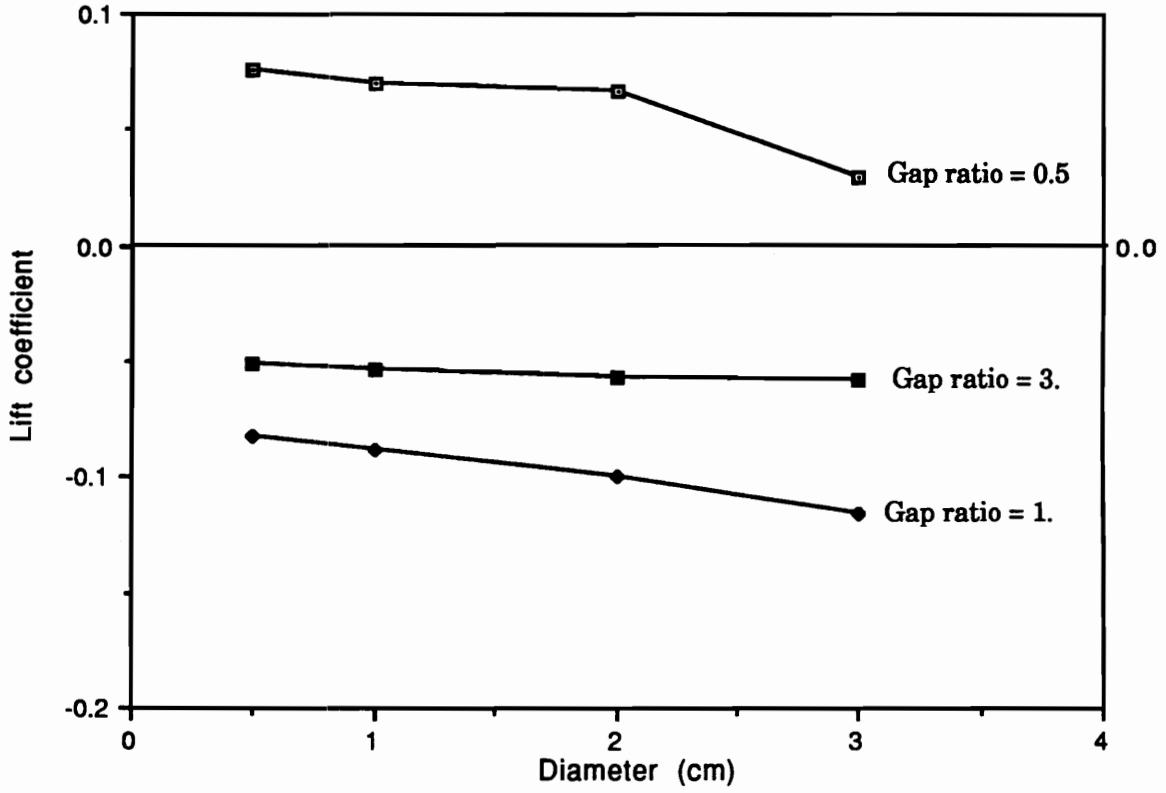


Figure 5.36 Relation between the diameter of sphere and the lift coefficient.

Appendix C

Program List

```
C Program for Newton iteration method
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION XX(20),YY(20),ZZ(20),NOD(12),P(4000),NROWP(4000)
      COMMON/ELE/X(4000),Y(4000),Z(4000),NT(4000)
      COMMON /ELNOD/NED(4000,27)
      COMMON /ASEMB/NROW(4000)
      COMMON/VELK/VX(4000),VY(4000),VZ(4000)
      COMMON/COEF/A(4000,473),KA(4000,473),F(4000)
C GAP=gap ratio
C     OPEN(11,FILE='TEMP.FIL',STATUS='OLD',FORM='UNFORMATTED')
C     OPEN(7,FILE='RSIDE.COE',STATUS='OLD')
C     OPEN(8,FILE='LOW.COE',STATUS='OLD',FORM='UNFORMATTED',ERR=5)
C     CLOSE(8,STATUS='SCRATCH')
C INITIALIZE THE VARIABLES
    5 DO 10 I=1,4000
      VX(I)=0.
      VY(I)=0.
      VZ(I)=0.
   10 CONTINUE
      DO 20 I=1,4000
       NROWP(I)=0
   20 NROW(I)=0
      DO 30 I=1,4000
      DO 35 J=1,473
      A(I,J)=0.
      KA(I,J)=0
   35 CONTINUE
   30 CONTINUE
      DO 40 I=1,4000
      F(I)=0.
   40 CONTINUE
      READ(11) NTELE,REYNOLD,NX,NY,NZ,NROWT,NODT,GR
      DO 110 L=1,NODT
      READ(11) X(L),Y(L),Z(L),NT(L)
  110 CONTINUE
      DO 120 K=1,NODT
      READ(11) NROW(K)
  120 CONTINUE
      DO 130 I=1,NTELE
      READ(11) (NED(I,K),K=1,27)
  130 CONTINUE
      DO 140 J=1,NODT
      READ(11) VX(J),VY(J),VZ(J)
  140 CONTINUE
      READ(17,'(3I10)') MSIZE,MBND,NPR
      READ(17,'(15D15.8)') (F(I),I=1,NROWT)
C      CLOSE(17,STATUS='SCRATCH')
      REWIND(17)
      NRVD=0
      DO 200 I=1,NTELE
      DO 210 J=1,27
C For nodal points at the vertices of elements, there are four
C unknowns:  Vx,Vy,Vz and P.  (For these nodes, NT(I) > 0 )
C For nodes not at the vertices of elements, there are three unknowns:
C  Vx,Vy,Vz.  (For these nodes, NT(I) < 0 )
```

```
C For nodes at boundary: Move the values to the right hand side and
C  delete the corresponding rows and columns. (NROW(I)=0)
      ND=NED(I,J)
      IF(NROWP(ND).GT.0) GOTO 210
      IF (NT(ND).EQ.1) THEN
       NRVD=NRVD+4
C Note: NROW(I)= the row number of the last unknown at the current
C   node.   The first unknown will start at row/column number
C  (NROW(ND)-3) or (NROW(ND)-2).
      NROWP(ND)=NRVD
       P(I)=F(NRVD)
       VZ(I)=F(NRVD-1)
       VY(I)=F(NRVD-2)
       VX(I)=F(NRVD-3)
      GOTO 210
      ELSEIF (NT(ND).EQ.-1) THEN
        NRVD=NRVD+3
        NROWP(ND)=NRVD
       VZ(I)=F(NRVD)
       VY(I)=F(NRVD-1)
       VX(I)=F(NRVD-2)
       GOTO 210
      ELSEIF (NT(ND).EQ.2.OR.NT(I).EQ.3) THEN
        NRVD=NRVD+1
        NROWP(ND)=NRVD
        P(I)=F(NRVD)
       GOTO 210
      ELSEIF (NT(ND).EQ.4.OR.NT(I).EQ.-4) THEN
        NRVD=NRVD+3
        NROWP(ND)=NRVD
       VZ(I)=F(NRVD)
       VY(I)=F(NRVD-1)
       VX(I)=F(NRVD-2)
      ENDIF
  210 CONTINUE
  200 CONTINUE
      DO 100 I=1,NTELE
      CALL BASIS(I)
      CALL ELEMENT(REYNOLD,GR)
      CALL ASSEM(I)
  100 CONTINUE
      CALL STORE(NROWT,NPR)
      REWIND(11)
      WRITE(11) NTELE,REYNOLD,NX,NY,NZ,NROWT,NODT,GR
      DO 150 L=1,NODT
      WRITE(11) X(L),Y(L),Z(L),NT(L)
  150 CONTINUE
      DO 160 K=1,NODT
      WRITE(11) NROW(K)
  160 CONTINUE
      DO 170 I=1,NTELE
      WRITE(11)  (NED(I,K),K=1,27)
  170 CONTINUE
      DO 180 J=1,NODT
      WRITE(11) VX(J),VY(J),VZ(J)
```

```
  180 CONTINUE
      STOP
      END
C
C  ***************************************************************
C
      SUBROUTINE ASSEM (NEL)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION NR(4),NC(4),NVTX(8)
      COMMON/LOCAL/XE(27),YE(27),ZE(27),VXE(27),VYE(27),VZE(27),
     & NTE(27)
      COMMON/ELE/X(4000),Y(4000),Z(4000),NT(4000)
      COMMON/VELK/VX(4000),VY(4000),VZ(4000)
      COMMON/ELMX/EK(3,3,27,27),EP(3,27,8),EC(3,8,27),EF(3,27)
      COMMON/ASEMB/NROW(4000)
      COMMON/ELNOD/NED(4000,27)
      COMMON/COEF/A(4000,473),KA(4000,473),F(4000)
      DATA NVTX/1,2,3,4,10,11,12,13/
C
      DO 10 I=1,27
      NDI=NED(NEL,I)
      NR(1)=0
      NR(2)=0
      NR(3)=0
      NR(4)=0
      IF (NROW(NDI).EQ.0) GOTO 10
      DO 15 II=1,8
      IF (I.EQ.NVTX(II)) IVTX=II
   15 CONTINUE
      IF (NT(NDI).EQ.1) THEN
      INR=4
      NR(4)=NROW(NDI)
      NR(3)=NROW(NDI)-1
      NR(2)=NROW(NDI)-2
      NR(1)=NROW(NDI)-3
      ELSEIF(NT(NDI).EQ.-1.OR.NT(NDI).EQ.-4.OR.NT(NDI).EQ.4) THEN
      INR=3
      NR(3)=NROW(NDI)
      NR(2)=NROW(NDI)-1
      NR(1)=NROW(NDI)-2
      ELSEIF (NT(NDI).NE.2.OR.NT(NDI).EQ.3) THEN
      INR=110
      NR(1)=NROW(NDI)
      ENDIF
      DO 20 J=1,27
      ND=NED(NEL,J)
      NC(1)=0
      NC(2)=0
      NC(3)=0
      NC(4)=0
      IF (NT(ND).EQ.-3) GOTO 20
      IF (NT(ND).EQ.-2) GOTO 30
      DO 25 JJ=1,8
      IF (J.EQ.NVTX(JJ)) JVTX=JJ
   25 CONTINUE
```

```
      IF (NT(ND).EQ.1) THEN
      INC=4
      NC(4)=NROW(ND)
      NC(3)=NROW(ND)-1
      NC(2)=NROW(ND)-2
      NC(1)=NROW(ND)-3
      ELSEIF (NT(ND).EQ.-1.OR.NT(ND).EQ.-4.OR.NT(ND).EQ.4) THEN
      INC=3
      NC(3)=NROW(ND)
      NC(2)=NROW(ND)-1
      NC(1)=NROW(ND)-2
      ELSEIF (NT(ND).EQ.2.OR.NT(ND).EQ.3) THEN
      INC=1
      NC(1)=NROW(ND)
      ENDIF
      DO 40 K=1,INR
      IF (K.EQ.4) GOTO 60
      IF(INR.EQ.1) GOTO 60
      DO 50 L=1,INC
      IF (L.EQ.4) GOTO 70
      IF(INC.EQ.1) GOTO 70
      DO 80 M=1,473
      IF(KA(NR(K),M).EQ.NC(L)) THEN
      A(NR(K),M)=A(NR(K),M)+EK(K,L,I,J)
      GOTO 50
      ENDIF
   80 CONTINUE
      DO 90 N=1,473
      IF(KA(NR(K),N).EQ.0) THEN
      A(NR(K),N)=EK(K,L,I,J)
      KA(NR(K),N)=NC(L)
      GOTO 50
      ENDIF
   90 CONTINUE
      WRITE(*,'(A)') ' ARRAY A EXCEEDS THE DIMENSION LIMIT'
      STOP
   70 DO 100 M=1,473
      IF (KA(NR(K),M).EQ.NC(L)) THEN
      A(NR(K),M)=A(NR(K),M)+EP(K,I,JVTX)
      GOTO 50
      ENDIF
  100 CONTINUE
      DO 110 N=1,473
      IF(KA(NR(K),N).EQ.0) THEN
      A(NR(K),N)=EP(K,I,JVTX)
      KA(NR(K),N)=NC(L)
      GOTO 50
      ENDIF
  110 CONTINUE
   50 CONTINUE
      GOTO 40
   60 DO 120 LC=1,INC
C This part assembles the Continuity equation.
      DO 130 M=1,473
      IF (KA(NR(K),M).EQ.NC(LC)) THEN
```

```
      A(NR(K),M)=A(NR(K),M)+EC(LC,IVTX,J)
      GOTO 120
      ENDIF
  130 CONTINUE
      DO 140 N=1,473
      IF(KA(NR(K),N).EQ.0) THEN
      A(NR(K),N)=EC(LC,IVTX,J)
      KA(NR(K),N)=NC(LC)
      GOTO 120
      ENDIF
  140 CONTINUE
  120 CONTINUE
   40 CONTINUE
      IF (NT(ND).EQ.2) GOTO 30
      GOTO 20
   30 DO 150 K=1,3
      F(NR(K))=F(NR(K))-EK(K,1,I,J)*VXE(J)-EK(K,2,I,J)*VYE(J)-
     &EK(K,3,I,J)*VZE(J)
  150 CONTINUE
      IF (NT(NDI).EQ.2) F(NR(4))=F(NR(4))-EC(1,IVTX,J)*VXE(J)-
     &EC(2,IVTX,J)*VYE(J)-EC(3,IVTX,J)*VZE(J)
   20 CONTINUE
      IF (INR.EQ.1) GOTO 10
      DO 160 K=1,3
      F(NR(K))=F(NR(K))+EF(K,I)
  160 CONTINUE
   10 CONTINUE
      RETURN
      END
C
C ***********************************************************************
C
      SUBROUTINE BSPLN (N,X,YI,T,V)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION X(30),YI(30)
      IF(T.GE.X(N)) THEN
      V=YI(N)
      RETURN
      ENDIF
      DO 10 I=1,N-1
      IF(T.GE.X(I).AND.T.LT.X(I+1)) THEN
      V=YI(I)+(YI(I+1)-YI(I))*(T-X(I))/(X(I+1)-X(I))
      RETURN
      ENDIF
   10 CONTINUE
      RETURN
      END
C
C ***********************************************************************
C
      SUBROUTINE ELEMENT (RE,GR)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION NVTX(8),FACM(27),NDF(6,9),NTYP(6)
      COMMON/LOCAL/XE(27),YE(27),ZE(27),VXE(27),VYE(27),VZE(27),
     &NTE(27)
```

```
      COMMON/ELE/X(4000),Y(4000),Z(4000),NT(4000)
      COMMON/VELK/VX(4000),VY(4000),VZ(4000)
      COMMON/VELDX/ VXDX(27,3),VYDX(27,3),VZDX(27,3)
      COMMON/ELMX/EK(3,3,27,27),EP(3,27,8),EC(3,8,27),EF(3,27)
      COMMON/JOCOB/DETJ(27),TJ(3,3,27),TJNV(3,3,27)
      COMMON/ELNOD/NED(4000,27)
      COMMON/BASE/PF(27,27),PFDX(27,27,3),PS(8,27)
      DATA NVTX/1,2,3,4,10,11,12,13/
      DATA FACM/1.D0,1.D0,1.D0,1.D0,4.D0,4.D0,4.D0,4.D0,16.D0,
     &1.D0,1.D0,1.D0,1.D0,4.D0,4.D0,4.D0,4.D0,16.D0,
     &4.D0,4.D0,4.D0,4.D0,16.D0,16.D0,16.D0,16.D0,64.D0/
      DATA ((NDF(I,J),J=1,9),I=1,6)/1,2,3,4,5,6,7,8,9,
     &2,11,12,3,20,15,    &21,6,24,10,11,12,13,14,15,16,17,18,1,10,
     &13,4,19,17,22,8,26,1,2,11,10,5,20,14,19,23,4,3,12,13,7,
     &21,16,22,25/
      DATA NTYP/2,2,1,1,1,2/
C NVTX(I) = Nodal points at the vertices.  Each of these nodes has
C three velocity components.  There is no pressure unknowns at these
C       points.
C Evaluate the volume integrals.
      DO 5 I=1,27
      DO 5 J=1,27
      DO 5 K=1,3
      DO 5 L=1,3
   5  EK(K,L,I,J)=0.
      DO 10 I=1,27
      DO 20 J=1,27
      DO 30 K=1,27
      EK(1,1,I,J)=EK(1,1,I,J)+FACM(K)*DETJ(K)*(PF(I,K)*(VXDX(K,1)
     &*PF(J,K)+VXE(K)*PFDX(J,K,1)+VYE(K)*PFDX(J,K,2)+
     &VZE(K)*PFDX(J,K,3))+(PFDX(J,K,1)*PFDX(I,K,1)+PFDX(J,K,2)*
     &PFDX(I,K,2)+PFDX(J,K,3)*PFDX(I,K,3))/RE)
      EK(1,2,I,J)=EK(1,2,I,J)+FACM(K)*DETJ(K)*PF(I,K)*
     &PF(J,K)*VXDX(K,2)
      EK(1,3,I,J)=EK(1,3,I,J)+FACM(K)*DETJ(K)*
     &PF(I,K)*PF(J,K)*VXDX(K,3)
      EK(2,1,I,J)=EK(2,1,I,J)+FACM(K)*DETJ(K)*
     &PF(I,K)*PF(J,K)*VYDX(K,1)
      EK(2,2,I,J)=EK(2,2,I,J)+FACM(K)*DETJ(K)*(PF(I,K)*(VXE(K)
     &*PFDX(J,K,1)+VYDX(K,2)*PF(J,K)+VYE(K)*PFDX(J,K,2)+VZE(K)*
     &PFDX(J,K,3))+(PFDX(J,K,1)*PFDX(I,K,1)+PFDX(J,K,2)*PFDX(I,K,2)
     &+PFDX(J,K,3)*PFDX(I,K,3))/RE)
      EK(2,3,I,J)=EK(2,3,I,J)+FACM(K)*DETJ(K)*
     &PF(I,K)*PF(J,K)*VYDX(K,3)
      EK(3,1,I,J)=EK(3,1,I,J)+FACM(K)*DETJ(K)*
     &PF(I,K)*PF(J,K)*VZDX(K,1)
      EK(3,2,I,J)=EK(3,2,I,J)+FACM(K)*DETJ(K)*
     &PF(I,K)*PF(J,K)*VZDX(K,2)
      EK(3,3,I,J)=EK(3,3,I,J)+FACM(K)*DETJ(K)*(PF(I,K)*(VXE(K)
     &*PFDX(J,K,1)+VYE(K)*PFDX(J,K,2)+VZDX(K,3)*PF(J,K)+VZE(K)*
     &PFDX(J,K,3))+(PFDX(J,K,1)*PFDX(I,K,1)+PFDX(J,K,2)*PFDX(I,K,2)+
     &PFDX(J,K,3)*PFDX(I,K,3))/RE)
  30  CONTINUE
      DO 40 L=1,3
      DO 50 M=1,3
```

```
      EK(L,M,I,J)=EK(L,M,I,J)/216.
   50 CONTINUE
   40 CONTINUE
   20 CONTINUE
   10 CONTINUE
      DO 60 K=1,27
      DO 60 I=1,8
      DO 60 J=1,3
      EP(J,K,1)=0.
      EC(J,I,K)=0.
   60 CONTINUE
      DO 70 I=1,27
      DO 80 J=1,8
      DO 90 K=1,27
      EP(1,I,J)=EP(1,I,J)+FACM(K)*DETJ(K)*PS(J,K)*PFDX(I,K,1)
      EP(2,I,J)=EP(2,I,J)+FACM(K)*DETJ(K)*PS(J,K)*PFDX(I,K,2)
      EP(3,I,J)=EP(3,I,J)+FACM(K)*DETJ(K)*PS(J,K)*PFDX(I,K,3)
      EC(1,J,I)=EC(1,J,I)+FACM(K)*DETJ(K)*PS(J,K)*PFDX(I,K,1)
      EC(2,J,I)=EC(2,J,I)+FACM(K)*DETJ(K)*PS(J,K)*PFDX(I,K,2)
      EC(3,J,I)=EC(3,J,I)+FACM(K)*DETJ(K)*PS(J,K)*PFDX(I,K,3)
   90 CONTINUE
      DO 95 L=1,3
      EP(L,I,J)=-EP(L,I,J)/216.
      EC(L,J,I)=EC(L,J,I)/216.
   95 CONTINUE
   80 CONTINUE
   70 CONTINUE
      DO 100 I=1,27
      DO 100 J=1,3
      EF(J,I)=0.
  100 CONTINUE
      DO 110 I=1,27
      DO 120 K=1,27
      EF(1,I)=EF(1,I)+FACM(K)*DETJ(K)*PF(I,K)*
     &(VXE(K)*VXDX(K,1)+VYE(K)*VXDX(K,2)+VZE(K)*VXDX(K,3))
      EF(2,I)=EF(2,I)+FACM(K)*DETJ(K)*PF(I,K)*
     &(VXE(K)*VYDX(K,1)+VYE(K)*VYDX(K,2)+VZE(K)*VYDX(K,3))
      EF(3,I)=EF(3,I)+FACM(K)*DETJ(K)*PF(I,K)*
     &(VXE(K)*VZDX(K,1)+VYE(K)*VZDX(K,2)+VZE(K)*VZDX(K,3))
  120 CONTINUE
      DO 125 J=1,3
      EF(J,I)=EF(J,I)/216.
  125 CONTINUE
  110 CONTINUE
C Find the surface integrals.
      DO 200 I=1,6
      IF(NTE(NDF(I,1)).LT.2.OR.NTE(NDF(I,2)).LT.2.OR.
     &NTE(NDF(I,3)).LT.2) GOTO 200
C Find the direction cosines of the surface.
      IF(I.EQ.1) THEN
      CX=0.
      CY=-1.
      CZ=0.
      ELSEIF (I.EQ.2) THEN
      CX=1.
```

```
      CY=0.
      CZ=0.
      ELSEIF (I.EQ.3) THEN
      CX=0.
      CY=1.
      CZ=0.
      ELSEIF (I.EQ.4) THEN
      CX=-1.
      CY=0.
      CZ=0.
      ELSEIF (I.EQ.5) THEN
      CX=0.
      CY=0.
      CZ=-1.
      ELSEIF (I.EQ.6) THEN
      CX=0.
      CY=0.
      CZ=1.
      ENDIF
      DO 210 M=1,27
      DO 220 N=1,27
      COR=0.
      DO 230 K=1,9
      COR=COR+FACM(K)*DETJ(K)*PF(M,K)*(PFDX(N,K,1)*CX+PFDX(N,K,2)*CY+
     &PFDX(N,K,3)*CZ)
  230 CONTINUE
      COR=COR/RE/36.
      EK(1,1,M,N)=EK(1,1,M,N)-COR
      EK(2,2,M,N)=EK(2,2,M,N)-COR
      EK(3,3,M,N)=EK(3,3,M,N)-COR
  220 CONTINUE
  210 CONTINUE
      DO 240 N=1,27
      DO 250 J=1,8
      P1=0.
      P2=0.
      P3=0.
      DO 260 K=1,9
      P1=P1+FACM(K)*DETJ(K)*PS(J,K)*PF(N,K)*CX
      P2=P2+FACM(K)*DETJ(K)*PS(J,K)*PF(N,K)*CY
      P3=P3+FACM(K)*DETJ(K)*PS(J,K)*PF(N,K)*CZ
  260 CONTINUE
      P1=P1/RE/36.
      P2=P2/RE/36.
      P3=P3/RE/36.
      EP(1,N,J)=EP(1,N,J)+P1
      EP(2,N,J)=EP(2,N,J)+P2
      EP(3,N,J)=EP(3,N,J)+P3
  250 CONTINUE
  240 CONTINUE
  200 CONTINUE
      RETURN
      END
C
C  ********************************************************************
```

```
C
      SUBROUTINE BASIS(NEL)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION RR(27),SS(27),TT(27),PSDR(8,8,3),NVTX(8),PFDR(27,27,3)
      COMMON/LOCAL/XE(27),YE(27),ZE(27),VXE(27),VYE(27),VZE(27),
     &NTE(27)
      COMMON/ELE/X(4000),Y(4000),Z(4000),NT(4000)
      COMMON/VELK/VX(4000),VY(4000),VZ(4000)
      COMMON/VELDX/ VXDX(27,3),VYDX(27,3),VZDX(27,3)
      COMMON/JOCOB/DETJ(27),TJ(3,3,27),TJNV(3,3,27)
      COMMON/ELNOD/NED(4000,27)
      COMMON/BASE/PF(27,27),PFDX(27,27,3),PS(8,27)
      DATA NVTX/1,2,3,4,10,11,12,13/
      DATA RR/0.D0,1.D0,1.D0,0.D0,0.5D0,1.D0,0.5D0,0.D0,0.5D0,
     &0.D0,1.D0,1.D0,0.D0,0.5D0,1.D0,0.5D0,0.D0,
     & 0.5D0,0.D0,1.D0,1.D0,0.D0,0.5D0,1.D0,0.5D0,0.D0,0.5D0/
      DATA TT/0.D0,0.D0,0.D0,0.D0,0.D0,0.D0,0.D0,0.D0,0.D0,
     &1.D0,1.D0,1.D0,1.D0,1.D0,1.D0,1.D0,1.D0,1.D0,
     & 0.5D0,0.5D0,0.5D0,0.5D0,0.5D0,0.5D0,0.5D0,0.5D0,0.5D0/
      DATA SS/0.D0,0.D0,1.D0,1.D0,0.D0,0.5D0,1.D0,0.5D0,
     &0.5D0,0.D0,0.D0,1.D0,1.D0,0.D0,0.5D0,1.D0,0.5D0,
     & 0.5D0,0.D0,0.D0,1.D0,1.D0,0.D0,0.5D0,1.D0,0.5D0,0.5D0/
      DO 5 I=1,3
      DO 5 J=1,3
      DO 5 K=1,27
   5  TJ(I,J,K)=0.
      DO 10 I=1,27
      DO 20 J=1,27
      IF (I.EQ.J) THEN
      PF(I,J)=1.
      ELSE
      PF(I,J)=0.
      ENDIF
  20  CONTINUE
  10  CONTINUE
C Calculate the derivatives of basis functions with respect to
C variables r,s,t at each node.
      DO 30 J=1,27
      R=RR(J)
      S=SS(J)
      T=TT(J)
      PFDR(1,J,1)=(4.*R-3.)*(1.-S)*(1.-2.*S)*(1.-T)*(1.-2.*T)
C PFDR(I,J,N) = Derivatives of basis function Gi with respect to
C r (N=1), s (N=2), t (N=3) at node Xj.
      PFDR(1,J,2)=(1.-R)*(1.-2.*R)*(4.*S-3.)*(1.-T)*(1.-2.*T)
      PFDR(1,J,3)=(1.-R)*(1.-2.*R)*(1.-S)*(1.-2.*S)*(4.*T-3.)
      PFDR(2,J,1)=(4.*R-1.)*(1.-S)*(1.-2.*S)*(1.-T)*(1.-2.*T)
      PFDR(2,J,2)=R*(2.*R-1.)*(4.*S-3.)*(1.-T)*(1.-2.*T)
      PFDR(2,J,3)=R*(2.*R-1.)*(1.-S)*(1.-2.*S)*(4.*T-3.)
      PFDR(3,J,1)=(4.*R-1.)*S*(2.*S-1.)*(1.-T)*(1.-2.*T)
      PFDR(3,J,2)=R*(2.*R-1.)*(4.*S-1.)*(1.-T)*(1.-2.*T)
      PFDR(3,J,3)=R*(2.*R-1.)*S*(2.*S-1.)*(4.*T-3.)
      PFDR(4,J,1)=(4.*R-3.)*S*(2.*S-1.)*(1.-T)*(1.-2.*T)
      PFDR(4,J,2)=(1.-R)*(1.-2.*R)*(4.*S-1.)*(1.-T)*(1.-2.*T)
      PFDR(4,J,3)=(1.-R)*(1.-2.*R)*S*(2.*S-1.)*(4.*T-3.)
```

```
PFDR(5,J,1)=(4.-8.*R)*(1.-S)*(1.-2.*S)*(1.-T)*(1.-2.*T)
PFDR(5,J,2)=4.*R*(1.-R)*(4.*S-3.)*(1.-T)*(1.-2.*T)
PFDR(5,J,3)=4.*R*(1.-R)*(1.-S)*(1.-2.*S)*(4.*T-3.)
PFDR(6,J,1)=(4.*R-1.)*4.*S*(1.-S)*(1.-T)*(1.-2.*T)
PFDR(6,J,2)=R*(2.*R-1.)*(4.-8.*S)*(1.-T)*(1.-2.*T)
PFDR(6,J,3)=R*(2.*R-1.)*4.*S*(1.-S)*(4.*T-3.)
PFDR(7,J,1)=(4.-8.*R)*S*(2.*S-1.)*(1.-T)*(1.-2.*T)
PFDR(7,J,2)=4.*R*(1.-R)*(4.*S-1.)*(1.-T)*(1.-2.*T)
PFDR(7,J,3)=4.*R*(1.-R)*S*(2.*S-1.)*(4.*T-3.)
PFDR(8,J,1)=(4.*R-3.)*4.*S*(1.-S)*(1.-T)*(1.-2.*T)
PFDR(8,J,2)=(1.-R)*(1.-2.*R)*(4.-8.*S)*(1.-T)*(1.-2.*T)
PFDR(8,J,3)=(1.-R)*(1.-2.*R)*4.*S*(1.-S)*(4.*T-3.)
PFDR(9,J,1)=(4.-8.*R)*4.*S*(1.-S)*(1.-T)*(1.-2.*T)
PFDR(9,J,2)=4.*R*(1.-R)*(4.-8.*S)*(1.-T)*(1.-2.*T)
PFDR(9,J,3)=4.*R*(1.-R)*4.*S*(1.-S)*(4.*T-3.)
PFDR(10,J,1)=(4.*R-3.)*(1.-S)*(1.-2.*S)*T*(2.*T-1.)
PFDR(10,J,2)=(1.-R)*(1.-2.*R)*(4.*S-3.)*T*(2.*T-1.)
PFDR(10,J,3)=(1.-R)*(1.-2.*R)*(1.-S)*(1.-2.*S)*(4.*T-1.)
PFDR(11,J,1)=(4.*R-1.)*(1.-S)*(1.-2.*S)*T*(2.*T-1.)
PFDR(11,J,2)=R*(2.*R-1.)*(4.*S-3.)*T*(2.*T-1.)
PFDR(11,J,3)=R*(2.*R-1.)*(1.-S)*(1.-2.*S)*(4.*T-1.)
PFDR(12,J,1)=(4.*R-1.)*S*(2.*S-1.)*T*(2.*T-1.)
PFDR(12,J,2)=R*(2.*R-1.)*(4.*S-1.)*T*(2.*T-1.)
PFDR(12,J,3)=R*(2.*R-1.)*S*(2.*S-1.)*(4.*T-1.)
PFDR(13,J,1)=(4.*R-3.)*S*(2.*S-1.)*T*(2.*T-1.)
PFDR(13,J,2)=(1.-R)*(1.-2.*R)*(4.*S-1.)*T*(2.*T-1.)
PFDR(13,J,3)=(1.-R)*(1.-2.*R)*S*(2.*S-1.)*(4.*T-1.)
PFDR(14,J,1)=(4.-8.*R)*(1.-S)*(1.-2.*S)*T*(2.*T-1.)
PFDR(14,J,2)=4.*R*(1.-R)*(4.*S-3.)*T*(2.*T-1.)
PFDR(14,J,3)=4.*R*(1.-R)*(1.-S)*(1.-2.*S)*(4.*T-1.)
PFDR(15,J,1)=(4.*R-1.)*4.*S*(1.-S)*T*(2.*T-1.)
PFDR(15,J,2)=R*(2.*R-1.)*(4.-8.*S)*T*(2.*T-1.)
PFDR(15,J,3)=R*(2.*R-1.)*4.*S*(1.-S)*(4.*T-1.)
PFDR(16,J,1)=(4.-8.*R)*S*(2.*S-1.)*T*(2.*T-1.)
PFDR(16,J,2)=4.*R*(1.-R)*(4.*S-1.)*T*(2.*T-1.)
PFDR(16,J,3)=4.*R*(1.-R)*S*(2.*S-1.)*(4.*T-1.)
PFDR(17,J,1)=(4.*R-3.)*4.*S*(1.-S)*T*(2.*T-1.)
PFDR(17,J,2)=(1.-R)*(1.-2.*R)*(4.-8.*S)*T*(2.*T-1.)
PFDR(17,J,3)=(1.-R)*(1.-2.*R)*4.*S*(1.-S)*(4.*T-1.)
PFDR(18,J,1)=(4.-8.*R)*4.*S*(1.-S)*T*(2.*T-1.)
PFDR(18,J,2)=4.*R*(1.-R)*(4.-8.*S)*T*(2.*T-1.)
PFDR(18,J,3)=4.*R*(1.-R)*4.*S*(1.-S)*(4.*T-1.)
PFDR(19,J,1)=(4.*R-3.)*(1.-S)*(1.-2.*S)*4.*T*(1.-T)
PFDR(19,J,2)=(1.-R)*(1.-2.*R)*(4.*S-3.)*4.*T*(1.-T)
PFDR(19,J,3)=(1.-R)*(1.-2.*R)*(1.-S)*(1.-2.*S)*(4.-8.*T)
PFDR(20,J,1)=(4.*R-1.)*(1.-S)*(1.-2.*S)*4.*T*(1.-T)
PFDR(20,J,2)=R*(2.*R-1.)*(4.*S-3.)*4.*T*(1.-T)
PFDR(20,J,3)=R*(2.*R-1.)*(1.-S)*(1.-2.*S)*(4.-8.*T)
PFDR(21,J,1)=(4.*R-1.)*S*(2.*S-1.)*4.*T*(1.-T)
PFDR(21,J,2)=R*(2.*R-1.)*(4.*S-1.)*4.*T*(1.-T)
PFDR(21,J,3)=R*(2.*R-1.)*S*(2.*S-1.)*(4.-8.*T)
PFDR(22,J,1)=(4.*R-3.)*S*(2.*S-1.)*4.*T*(1.-T)
PFDR(22,J,2)=(1.-R)*(1.-2.*R)*(4.*S-1.)*4.*T*(1.-T)
PFDR(22,J,3)=(1.-R)*(1.-2.*R)*S*(2.*S-1.)*(4.-8.*T)
PFDR(23,J,1)=(4.-8.*R)*(1.-S)*(1.-2.*S)*4.*T*(1.-T)
```

```
      PFDR(23,J,2)=4.*R*(1.-R)*(4.*S-3.)*4.*T*(1.-T)
      PFDR(23,J,3)=4.*R*(1.-R)*(1.-S)*(1.-2.*S)*(4.-8.*T)
      PFDR(24,J,1)=(4.*R-1.)*4.*S*(1.-S)*4.*T*(1.-T)
      PFDR(24,J,2)=R*(2.*R-1.)*(4.-8.*S)*4.*T*(1.-T)
      PFDR(24,J,3)=R*(2.*R-1.)*4.*S*(1.-S)*(4.-8.*T)
      PFDR(25,J,1)=(4.-8.*R)*S*(2.*S-1.)*4.*T*(1.-T)
      PFDR(25,J,2)=4.*R*(1.-R)*(4.*S-1.)*4.*T*(1.-T)
      PFDR(25,J,3)=4.*R*(1.-R)*S*(2.*S-1.)*(4.-8.-T)
      PFDR(26,J,1)=(4.*R-3.)*4.*S*(1.-S)*4.*T*(1.-T)
      PFDR(26,J,2)=(1.-R)*(1.-2.*R)*(4.-8.*S)*4.*T*(1.-T)
      PFDR(26,J,3)=(1.-R)*(1.-2.*R)*4.*S*(1.-S)*(4.-8.*T)
      PFDR(27,J,1)=(4.-8.*R)*4.*S*(1.-S)*4.*T*(1.-T)
      PFDR(27,J,2)=4.*R*(1.-R)*(4.-8.*S)*4.*T*(1.-T)
      PFDR(27,J,3)=4.*R*(1.-R)*4.*S*(1.-S)*(4.-8.*T)
   30 CONTINUE
C Find the Jocobian matrix and its inverse.
      DO 40 I=1,27
      NTE(I)=NT(NED(NEL,I))
      VXE(I)=VX(NED(NEL,I))
      VYE(I)=VY(NED(NEL,I))
      VZE(I)=VZ(NED(NEL,I))
      XE(I)=X(NED(NEL,I))
      YE(I)=Y(NED(NEL,I))
      ZE(I)=Z(NED(NEL,I))
   40 CONTINUE
      DO 50 I=1,27
      DO 60 IR=1,3
      DO 70 ISUM=1,27
      TJ(IR,1,I)=TJ(IR,1,I)+PFDR(ISUM,I,IR)*XE(ISUM)
      TJ(IR,2,I)=TJ(IR,2,I)+PFDR(ISUM,I,IR)*YE(ISUM)
      TJ(IR,3,I)=TJ(IR,3,I)+PFDR(ISUM,I,IR)*ZE(ISUM)
   70 CONTINUE
C TJ(IR,J,I)=IR-th row and J-th column element of Jocobian matrix at
C            node I of element.
   60 CONTINUE
C Calculate the determinant of Jocobian matrix at node I and
C  the inverse matrix of Jocobian.
      DETJ(I)=TJ(1,1,I)*TJ(2,2,I)*TJ(3,3,I)+TJ(2,1,I)*TJ(3,2,I)*
     &  TJ(1,3,I)+TJ(3,1,I)*TJ(1,2,I)*TJ(2,3,I)-TJ(1,1,I)*TJ(3,2,I)*
     &  TJ(2,3,I)-TJ(2,1,I)*TJ(1,2,I)*TJ(3,3,I)-TJ(3,1,I)*TJ(2,2,I)*
     &  TJ(1,3,I)
C Inverse matrix of Jocobian.
      TJNV(1,1,I)=(TJ(2,2,I)*TJ(3,3,I)-TJ(3,2,I)*TJ(2,3,I))/DETJ(I)
      TJNV(2,1,I)=(TJ(2,3,I)*TJ(3,1,I)-TJ(2,1,I)*TJ(3,3,I))/DETJ(I)
      TJNV(3,1,I)=(TJ(2,1,I)*TJ(3,2,I)-TJ(3,1,I)*TJ(2,2,I))/DETJ(I)
      TJNV(1,2,I)=(TJ(3,2,I)*TJ(1,3,I)-TJ(1,2,I)*TJ(3,3,I))/DETJ(I)
      TJNV(2,2,I)=(TJ(1,1,I)*TJ(3,3,I)-TJ(3,1,I)*TJ(1,3,I))/DETJ(I)
      TJNV(3,2,I)=(TJ(1,2,I)*TJ(3,1,I)-TJ(1,1,I)*TJ(3,2,I))/DETJ(I)
      TJNV(1,3,I)=(TJ(1,2,I)*TJ(2,3,I)-TJ(2,2,I)*TJ(1,3,I))/DETJ(I)
      TJNV(2,3,I)=(TJ(2,1,I)*TJ(1,3,I)-TJ(1,1,I)*TJ(2,3,I))/DETJ(I)
      TJNV(3,3,I)=(TJ(1,1,I)*TJ(2,2,I)-TJ(1,2,I)*TJ(2,1,I))/DETJ(I)
   50 CONTINUE
C Find the derivatives of basis functions with respect to x,y,z.
      DO 80 I=1,27
      DO 90 J=1,27
```

```
      DO 100 K=1,3
C Here PFDX(I,J,K)=Gi(Xj) 's derivative with respect to x(K=1),y(K=2),
C  z(K=3).
      PFDX(I,J,K)=TJNV(K,1,J)*PFDR(I,J,1)+TJNV(K,2,J)*PFDR(I,J,2)+
     &TJNV(K,3,J)*PFDR(I,J,3)
  100 CONTINUE
   90 CONTINUE
   80 CONTINUE
C Evaluate the basis function for pressure stress.
      DO 110 I=1,8
      DO 120 J=1,8
      IF(I.EQ.J) THEN
       PS(I,J)=1.
      ELSE
       PS(I,J)=0.
      ENDIF
  120 CONTINUE
  110 CONTINUE
C Find the derivatives of velocity components with respect to x (N=1),
C  y (N=2), z (N=3) at node I.
      DO 200 I=1,27
      DO 200 N=1,3
      VXDX(I,N)=0.
      VYDX(I,N)=0.
      VZDX(I,N)=0.
  200 CONTINUE
      DO 210 I=1,27
      DO 220 N=1,3
      DO 230 J=1,27
      VXDX(I,N)=VXDX(I,N)+PFDX(J,I,N)*VXE(J)
      VYDX(I,N)=VYDX(I,N)+PFDX(J,I,N)*VYE(J)
      VZDX(I,N)=VZDX(I,N)+PFDX(J,I,N)*VZE(J)
  230 CONTINUE
  220 CONTINUE
  210 CONTINUE
      DO 300 I=1,27
      R=RR(I)
      S=SS(I)
      T=TT(I)
      PS(1,I)=(1.-R)*(1.-S)*(1.-T)
      PS(2,I)=R*(1.-S)*(1.-T)
      PS(3,I)=R*S*(1.-T)
      PS(4,I)=(1.-R)*S*(1.-T)
      PS(5,I)=(1.-R)*(1.-S)*T
      PS(6,I)=R*(1.-S)*T
      PS(7,I)=R*S*T
      PS(8,I)=(1.-R)*S*T
  300 CONTINUE
      RETURN
      END
C
C *********************************************************************
C
      SUBROUTINE STORE(NROW,NPR)
      IMPLICIT REAL*8(A-H,O-Z)
```

```
      COMMON/COEF/A(4000,473),KA(4000,473),F(4000)
C      OPEN(7,FILE='RSIDE.COE',STATUS='NEW')
C      OPEN(8,FILE='LOW.COE',STATUS='NEW',FORM='UNFORMATTED')
      REWIND(18)
      MSIZE=NROW
      MXD=1
      DO 10 I=1,MSIZE
      DO 110 L=1,473
      IF(KA(I,L).LE.0) GOTO 115
      MXD=MAX0(MXD,ABS(KA(I,L)-I))
  110 CONTINUE
  115 NA1=L-1
      WRITE(18) I,NA1
      IF(NA1.LE.64) THEN
      WRITE(18) (KA(I,K),K=1,NA1)
      WRITE(18) (A(I,K),K=1,NA1)
      ELSE
      DO 35 K=1,NA1,64
      KE=K+63
      IF(KE.GT.NA1) KE=NA1
   35 WRITE(18)(KA(I,KW),KW=K,KE)
      DO 45 K=1,NA1,64
      KE=K+63
      IF(KE.GT.NA1) KE=NA1
   45 WRITE(18)(A(I,KW),KW=K,KE)
      ENDIF
   10 CONTINUE
      WRITE(17,1000) MSIZE,MXD,NPR
 1000 FORMAT(3I10)
      WRITE(17,1100) (F(K),K=1,MSIZE)
 1100 FORMAT(15D15.8)
      CLOSE(17)
      CLOSE(18)
      RETURN
      END
```

```
C Linear solution program
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION XX(20),YY(20),ZZ(20)
      COMMON/ELE/X(4000),Y(4000),Z(4000),NT(4000)
      COMMON /ELNOD/NED(4000,27)
      COMMON /ASEMB/NROW(4000)
      COMMON/VELK/VX(4000),VY(4000),VZ(4000)
      COMMON/COEF/A(4000,473),KA(4000,473),F(4000)
C     CHARACTER*16 FILED,FILEI
C     WRITE(*,'(A)')' INPUT FILENAME=?'
C     READ(*,'(A)') FILEI
C     WRITE(*,'(A)')' OUTPUT FILENAME=?'
C     READ(*,'(A)') FILED
C GAP=gap ratio
C     OPEN(9,FILE=FILEI,STATUS='OLD')
C     OPEN(10,FILE=FILED,STATUS='NEW')
C     OPEN(11,FILE='TEMP.FIL',STATUS='NEW',FORM='UNFORMATTED')
C INITIALIZE THE VARIABLES
      DO 10 I=1,4000
      VX(I)=0.
      VY(I)=0.
      VZ(I)=0.
   10 CONTINUE
      DO 20 I=1,4000
   20 NROW(I)=0
      DO 30 I=1,4000
      DO 35 J=1,473
      A(I,J)=0.
      KA(I,J)=0
  35  CONTINUE
  30  CONTINUE
      DO 40 I=1,4000
      F(I)=0.
  40  CONTINUE
      READ(19,1000)DIA1,GAP,REYNOLD,VISCOUS,DEPTH1,SLOPE,VELU1,HYR1,
     &PMVEL,NX,NY,NZ
 1000 FORMAT(8F10.0,/,F10.0,3I10)
      REYNOLD=VELU1*DIA1/VISCOUS
C Dimensionless input for R(I)
      READ(19,*)(XX(I),I=1,NX)
      READ(19,*)(YY(I),I=1,NY)
      READ(19,*)(ZZ(I),I=1,NZ)
      IF(PMVEL.GT.0.) CALL READCV(VELU1,DIA1)
      R0=0.5
      GR=9.81*DIA1/VELU1/VELU1
      CALL GRID(R0,GAP,NX,NY,NZ,XX,YY,ZZ,NODT,NTELE,NROWT,NPR)
C NTELE= TOTAL NUMBER OF ELEMENTS
      WRITE(10,1200)
 1200 FORMAT(' COORDINATES OF NODES:',//,1X,'NODE',5X,'X',7X,
     %'Y',7X,'Z',5X,'BND ROW',3X,'NODE',5X,'X',7X,'Y',7X,'Z',5X,
     &'BND  ROW',
     %3X,'NODE',5X,'X',7X,'Y',7X,'Z',5X,'BND  ROW')
      DO 50 J=1,NODT,3
      J2=J+1
      J3=J+2
```

```
      IF(J2.GT.NODT) GOTO 60
      IF(J3.GT.NODT) GOTO 70
      WRITE(10,1300) J,X(J),Y(J),Z(J),NT(J),NROW(J),
     &J2,X(J2),Y(J2),Z(J2),NT(J2),NROW(J2),
     #J3,X(J3),Y(J3),Z(J3),NT(J3),NROW(J3)
 1300 FORMAT(I5,3F8.3,I3,I5,2X,I5,3F8.3,I3,I5,2X,I5,3F8.3,I3,I5)
      GOTO 50
   60 WRITE(10,1300) J,X(J),Y(J),Z(J),NT(J),NROW(J)
      GOTO 50
   70 WRITE(10,1300) J,X(J),Y(J),Z(J),NT(J),NROW(J),
     &J2,X(J2),Y(J2),Z(J2),NT(J2),NROW(J2)
   50 CONTINUE
      WRITE(10,1400)
 1400 FORMAT(//,' ELEMENT       NODES 1 THROUGH 27')
      DO 80 I=1,NTELE
      WRITE(10,1500) I,(NED(I,J),J=1,27)
 1500 FORMAT(I5,14I5,/,5X,13I5)
   80 CONTINUE
      CALL INITBD(NODT)
      DO 100 I=1,NTELE
      CALL BASIS(I)
      CALL ELEMENT(REYNOLD,GR)
      CALL ASSEM(I)
  100 CONTINUE
      CALL STORE(NROWT,NPR)
      WRITE(11) NTELE,REYNOLD,NX,NY,NZ,NROWT,NODT,GR
      DO 110 L=1,NODT
      WRITE(11) X(L),Y(L),Z(L),NT(L)
  110 CONTINUE
      DO 120 K=1,NODT
      WRITE(11) NROW(K)
  120 CONTINUE
      DO 130 I=1,NTELE
      WRITE(11)  (NED(I,K),K=1,27)
  130 CONTINUE
      DO 140 J=1,NODT
      WRITE(11) VX(J),VY(J),VZ(J)
  140 CONTINUE
      STOP
      END
C
C  ********************************************************************
C
      SUBROUTINE ASSEM (NEL)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION NR(4),NC(4),NVTX(8)
      COMMON/LOCAL/XE(27),YE(27),ZE(27),VXE(27),VYE(27),VZE(27),NTE(27
     &)
      COMMON/ELE/X(4000),Y(4000),Z(4000),NT(4000)
      COMMON/VELK/VX(4000),VY(4000),VZ(4000)
      COMMON/ELMX/EK(3,3,27,27),EP(3,27,8),EC(3,8,27),EF(3,27)
      COMMON/ASEMB/NROW(4000)
      COMMON/ELNOD/NED(4000,27)
      COMMON/COEF/A(4000,473),KA(4000,473),F(4000)
      DATA NVTX/1,2,3,4,10,11,12,13/
```

```
C
      DO 10 I=1,27
      NDI=NED(NEL,I)
      NR(1)=0
      NR(2)=0
      NR(3)=0
      NR(4)=0
      IF (NROW(NDI).EQ.0) GOTO 10
      DO 15 II=1,8
      IF (I.EQ.NVTX(II)) IVTX=II
   15 CONTINUE
      IF (NT(NDI).EQ.1) THEN
      INR=4
      NR(4)=NROW(NDI)
      NR(3)=NROW(NDI)-1
      NR(2)=NROW(NDI)-2
      NR(1)=NROW(NDI)-3
      ELSEIF(NT(NDI).EQ.-1.OR.NT(NDI).EQ.-4.OR.NT(NDI).EQ.4) THEN
      INR=3
      NR(3)=NROW(NDI)
      NR(2)=NROW(NDI)-1
      NR(1)=NROW(NDI)-2
      ELSEIF (NT(NDI).NE.2.OR.NT(NDI).EQ.3) THEN
      INR=110
      NR(1)=NROW(NDI)
      ENDIF
      DO 20 J=1,27
      ND=NED(NEL,J)
      NC(1)=0
      NC(2)=0
      NC(3)=0
      NC(4)=0
      IF (NT(ND).EQ.-3) GOTO 20
      IF (NT(ND).EQ.-2) GOTO 30
      DO 25 JJ=1,8
      IF (J.EQ.NVTX(JJ)) JVTX=JJ
   25 CONTINUE
      IF (NT(ND).EQ.1) THEN
      INC=4
      NC(4)=NROW(ND)
      NC(3)=NROW(ND)-1
      NC(2)=NROW(ND)-2
      NC(1)=NROW(ND)-3
      ELSEIF (NT(ND).EQ.-1.OR.NT(ND).EQ.-4.OR.NT(ND).EQ.4) THEN
      INC=3
      NC(3)=NROW(ND)
      NC(2)=NROW(ND)-1
      NC(1)=NROW(ND)-2
      ELSEIF (NT(ND).EQ.2.OR.NT(ND).EQ.3) THEN
      INC=1
      NC(1)=NROW(ND)
      ENDIF
      DO 40 K=1,INR
      IF (K.EQ.4) GOTO 60
      IF(INR.EQ.1) GOTO 60
```

```
      DO 50 L=1,INC
      IF (L.EQ.4) GOTO 70
      IF(INC.EQ.1) GOTO 70
      DO 80 M=1,473
      IF(KA(NR(K),M).EQ.NC(L)) THEN
      A(NR(K),M)=A(NR(K),M)+EK(K,L,I,J)
      GOTO 50
      ENDIF
   80 CONTINUE
      DO 90 N=1,473
      IF(KA(NR(K),N).EQ.0) THEN
      A(NR(K),N)=EK(K,L,I,J)
      KA(NR(K),N)=NC(L)
      GOTO 50
      ENDIF
   90 CONTINUE
      WRITE(*,'(A)') ' ARRAY A EXCEEDS THE DIMENSION LIMIT'
      STOP
   70 DO 100 M=1,473
      IF (KA(NR(K),M).EQ.NC(L)) THEN
      A(NR(K),M)=A(NR(K),M)+EP(K,I,JVTX)
      GOTO 50
      ENDIF
  100 CONTINUE
      DO 110 N=1,473
      IF(KA(NR(K),N).EQ.0) THEN
      A(NR(K),N)=EP(K,I,JVTX)
      KA(NR(K),N)=NC(L)
      GOTO 50
      ENDIF
  110 CONTINUE
   50 CONTINUE
      GOTO 40
   60 DO 120 LC=1,INC
C This part assembles the Continuity equation.
      DO 130 M=1,473
      IF (KA(NR(K),M).EQ.NC(LC)) THEN
      A(NR(K),M)=A(NR(K),M)+EC(LC,IVTX,J)
      GOTO 120
      ENDIF
  130 CONTINUE
      DO 140 N=1,473
      IF(KA(NR(K),N).EQ.0) THEN
      A(NR(K),N)=EC(LC,IVTX,J)
      KA(NR(K),N)=NC(LC)
      GOTO 120
      ENDIF
  140 CONTINUE
  120 CONTINUE
   40 CONTINUE
      IF (NT(ND).EQ.2) GOTO 30
      GOTO 20
   30 DO 150 K=1,3
      F(NR(K))=F(NR(K))-EK(K,1,I,J)*VXE(J)-EK(K,2,I,J)*VYE(J)-
     &EK(K,3,I,J)*VZE(J)
```

```
  150 CONTINUE
      IF (NT(NDI).EQ.2) F(NR(4))=F(NR(4))-EC(1,IVTX,J)*VXE(J)-
     &EC(2,IVTX,J)*VYE(J)-EC(3,IVTX,J)*VZE(J)
   20 CONTINUE
      IF (INR.EQ.1) GOTO 10
      DO 160 K=1,3
      F(NR(K))=F(NR(K))+EF(K,I)
  160 CONTINUE
   10 CONTINUE
      RETURN
      END
C
C ***************************************************************************
C
      SUBROUTINE BSPLN (N,X,YI,T,V)
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION X(30),YI(30)
      IF(T.GE.X(N)) THEN
      V=YI(N)
      RETURN
      ENDIF
      DO 10 I=1,N-1
      IF(T.GE.X(I).AND.T.LT.X(I+1)) THEN
      V=YI(I)+(YI(I+1)-YI(I))*(T-X(I))/(X(I+1)-X(I))
      RETURN
      ENDIF
   10 CONTINUE
      RETURN
      END
C
C ***************************************************************************
C
      SUBROUTINE ELEMENT (RE,GR)
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION NVTX(8),FACM(27),NDF(6,9),NTYP(6)
      COMMON/LOCAL/XE(27),YE(27),ZE(27),VXE(27),VYE(27),VZE(27),
     &NTE(27)
      COMMON/ELE/X(4000),Y(4000),Z(4000),NT(4000)
      COMMON/ELMX/EK(3,3,27,27),EP(3,27,8),EC(3,8,27),EF(3,27)
      COMMON/JOCOB/DETJ(27),TJ(3,3,27),TJNV(3,3,27)
      COMMON/ELNOD/NED(4000,27)
      COMMON/BASE/PF(27,27),PFDX(27,27,3),PS(8,27)
      DATA NVTX/1,2,3,4,10,11,12,13/
      DATA FACM/1.D0,1.D0,1.D0,1.D0,4.D0,4.D0,4.D0,4.D0,16.D0,
     &1.D0,1.D0,1.D0,1.D0,4.D0,4.D0,4.D0,4.D0,16.D0,
     &4.D0,4.D0,4.D0,4.D0,16.D0,16.D0,16.D0,16.D0,64.D0/
      DATA ((NDF(I,J),J=1,9),I=1,6)/1,2,3,4,5,6,7,8,9,2,11,12,3,20,15,
     &21,6,24,10,11,12,13,14,15,16,17,18,1,10,13,4,19,17,22,
     #8,26,1,2,11,10,5,20,14,19,23,4,3,12,13,7,21,16,22,25/
      DATA NTYP/2,2,1,1,1,2/
C NVTX(I) = Nodal points at the vertices.  Each of these nodes has
C three velocity components.  There is no pressure unknowns at these
C           points.
C Evaluate the volume integrals.
      DO 5 I=1,27
```

```
      DO 5 J=1,27
      DO 5 K=1,3
      DO 5 L=1,3
    5 EK(K,L,I,J)=0.
      DO 10 I=1,27
      DO 20 J=1,27
      DO 30 K=1,27
      EK(1,1,I,J)=EK(1,1,I,J)+FACM(K)*DETJ(K)*(
     &(PFDX(J,K,1)*PFDX(I,K,1)+PFDX(J,K,2)*PFDX(I,K,2)+PFDX(J,K,3)*
     &PFDX(I,K,3))/RE)
      EK(2,2,I,J)=EK(2,2,I,J)+FACM(K)*DETJ(K)*(
     &(PFDX(J,K,1)*PFDX(I,K,1)+PFDX(J,K,2)*PFDX(I,K,2)
     &+PFDX(J,K,3)*PFDX(I,K,3))/RE)
      EK(3,3,I,J)=EK(3,3,I,J)+FACM(K)*DETJ(K)*(
     &(PFDX(J,K,1)*PFDX(I,K,1)+PFDX(J,K,2)*PFDX(I,K,2)+
     &PFDX(J,K,3)*PFDX(I,K,3))/RE)
   30 CONTINUE
      DO 40 L=1,3
      DO 50 M=1,3
      EK(L,M,I,J)=EK(L,M,I,J)/216.
   50 CONTINUE
   40 CONTINUE
   20 CONTINUE
   10 CONTINUE
      DO 60 K=1,27
      DO 60 I=1,8
      DO 60 J=1,3
      EP(J,K,1)=0.
      EC(J,I,K)=0.
   60 CONTINUE
      DO 70 I=1,27
      DO 80 J=1,8
      DO 90 K=1,27
      EP(1,I,J)=EP(1,I,J)+FACM(K)*DETJ(K)*PS(J,K)*PFDX(I,K,1)
      EP(2,I,J)=EP(2,I,J)+FACM(K)*DETJ(K)*PS(J,K)*PFDX(I,K,2)
      EP(3,I,J)=EP(3,I,J)+FACM(K)*DETJ(K)*PS(J,K)*PFDX(I,K,3)
      EC(1,J,I)=EC(1,J,I)+FACM(K)*DETJ(K)*PS(J,K)*PFDX(I,K,1)
      EC(2,J,I)=EC(2,J,I)+FACM(K)*DETJ(K)*PS(J,K)*PFDX(I,K,2)
      EC(3,J,I)=EC(3,J,I)+FACM(K)*DETJ(K)*PS(J,K)*PFDX(I,K,3)
   90 CONTINUE
      DO 95 L=1,3
      EP(L,I,J)=-EP(L,I,J)/216.
      EC(L,J,I)=EC(L,J,I)/216.
   95 CONTINUE
   80 CONTINUE
   70 CONTINUE
      DO 100 I=1,27
      DO 100 J=1,3
      EF(J,I)=0.
  100 CONTINUE
      DO 200 I=1,6
      IF(NTE(NDF(I,1)).LT.2.OR.NTE(NDF(I,2)).LT.2.OR.
     & NTE(NDF(I,3)).LT.2) GOTO 200
C Find the direction cosines of the surface.
      IF(I.EQ.1) THEN
```

```
      CX=0.
      CY=-1.
      CZ=0.
      ELSEIF (I.EQ.2) THEN
      CX=1.
      CY=0.
      CZ=0.
      ELSEIF (I.EQ.3) THEN
      CX=0.
      CY=1.
      CZ=0.
      ELSEIF (I.EQ.4) THEN
      CX=-1.
      CY=0.
      CZ=0.
      ELSEIF (I.EQ.5) THEN
      CX=0.
      CY=0.
      CZ=-1.
      ELSEIF (I.EQ.6) THEN
      CX=0.
      CY=0.
      CZ=1.
      ENDIF
      DO 210 M=1,27
      DO 220 N=1,27
      COR=0.
      DO 230 K=1,9
      COR=COR+FACM(K)*DETJ(K)*PF(M,K)*(PFDX(N,K,1)*CX+PFDX(N,K,2)*CY+
     &PFDX(N,K,3)*CZ)
  230 CONTINUE
      COR=COR/RE/36.
      EK(1,1,M,N)=EK(1,1,M,N)-COR
      EK(2,2,M,N)=EK(2,2,M,N)-COR
      EK(3,3,M,N)=EK(3,3,M,N)-COR
  220 CONTINUE
  210 CONTINUE
      DO 240 N=1,27
      DO 250 J=1,8
      P1=0.
      P2=0.
      P3=0.
      DO 260 K=1,9
      P1=P1+FACM(K)*DETJ(K)*PS(J,K)*PF(N,K)*CX
      P2=P2+FACM(K)*DETJ(K)*PS(J,K)*PF(N,K)*CY
      P3=P3+FACM(K)*DETJ(K)*PS(J,K)*PF(N,K)*CZ
  260 CONTINUE
      P1=P1/RE/36.
      P2=P2/RE/36.
      P3=P3/RE/36.
      EP(1,N,J)=EP(1,N,J)+P1
      EP(2,N,J)=EP(2,N,J)+P2
      EP(3,N,J)=EP(3,N,J)+P3
  250 CONTINUE
  240 CONTINUE
```

```
  200 CONTINUE
      RETURN
      END
C
C  ***************************************************************
C
      SUBROUTINE BASIS(NEL)
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION RR(27),SS(27),TT(27),PSDR(8,8,3),NVTX(8),PFDR(27,27,3)
      COMMON/LOCAL/XE(27),YE(27),ZE(27),VXE(27),VYE(27),VZE(27),
     &NTE(27)
      COMMON/ELE/X(4000),Y(4000),Z(4000),NT(4000)
      COMMON/VELK/VX(4000),VY(4000),VZ(4000)
      COMMON/VELDX/ VXDX(27,3),VYDX(27,3),VZDX(27,3)
      COMMON/JOCOB/DETJ(27),TJ(3,3,27),TJNV(3,3,27)
      COMMON/ELNOD/NED(4000,27)
      COMMON/BASE/PF(27,27),PFDX(27,27,3),PS(8,27)
      DATA NVTX/1,2,3,4,10,11,12,13/
      DATA RR/0.D0,1.D0,1.D0,0.D0,0.5D0,1.D0,0.5D0,0.D0,0.5D0,
     &0.D0,1.D0,1.D0,0.D0,0.5D0,1.D0,0.5D0,0.D0,
     & 0.5D0,0.D0,1.D0,1.D0,0.D0,0.5D0,1.D0,0.5D0,0.D0,0.5D0/
      DATA TT/0.D0,0.D0,0.D0,0.D0,0.D0,0.D0,0.D0,0.D0,0.D0,
     &1.D0,1.D0,1.D0,1.D0,1.D0,1.D0,1.D0,1.D0,1.D0,
     & 0.5D0,0.5D0,0.5D0,0.5D0,0.5D0,0.5D0,0.5D0,0.5D0,0.5D0/
      DATA SS/0.D0,0.D0,1.D0,1.D0,0.D0,0.5D0,1.D0,0.5D0,
     &0.5D0,0.D0,0.D0,1.D0,1.D0,0.D0,0.5D0,1.D0,0.5D0,
     & 0.5D0,0.D0,0.D0,1.D0,1.D0,0.D0,0.5D0,1.D0,0.5D0,0.5D0/
      DO 5 I=1,3
      DO 5 J=1,3
      DO 5 K=1,27
   5  TJ(I,J,K)=0.
      DO 10 I=1,27
      DO 20 J=1,27
      IF (I.EQ.J) THEN
      PF(I,J)=1.
      ELSE
      PF(I,J)=0.
      ENDIF
   20 CONTINUE
   10 CONTINUE
C Calculate the derivatives of basis functions with respect to
C variables r,s,t at each node.
      DO 30 J=1,27
      R=RR(J)
      S=SS(J)
      T=TT(J)
      PFDR(1,J,1)=(4.D0*R-3.D0)*(1.D0-S)*(1.D0-2.*S)*(1.-T)*(1.-2.*T)
C PFDR(I,J,N) = Derivatives of basis function Gi with respect to
C r (N=1), s (N=2), t (N=3) at node Xj.
      PFDR(1,J,2)=(1.-R)*(1.-2.*R)*(4.*S-3.)*(1.-T)*(1.-2.*T)
      PFDR(1,J,3)=(1.-R)*(1.-2.*R)*(1.-S)*(1.-2.*S)*(4.*T-3.)
      PFDR(2,J,1)=(4.*R-1.)*(1.-S)*(1.-2.*S)*(1.-T)*(1.-2.*T)
      PFDR(2,J,2)=R*(2.D0*R-1.D0)*(4.D0*S-3.D0)*(1.D0-T)*(1.D0-2.*T)
      PFDR(2,J,3)=R*(2.*R-1.)*(1.-S)*(1.-2.*S)*(4.*T-3.)
      PFDR(3,J,1)=(4.*R-1.)*S*(2.*S-1.)*(1.-T)*(1.-2.*T)
```

```
PFDR(3,J,2)=R*(2.*R-1.)*(4.*S-1.)*(1.-T)*(1.-2.*T)
PFDR(3,J,3)=R*(2.*R-1.)*S*(2.*S-1.)*(4.*T-3.)
PFDR(4,J,1)=(4.*R-3.)*S*(2.*S-1.)*(1.-T)*(1.-2.*T)
PFDR(4,J,2)=(1.-R)*(1.-2.*R)*(4.*S-1.)*(1.-T)*(1.-2.*T)
PFDR(4,J,3)=(1.-R)*(1.-2.*R)*S*(2.*S-1.)*(4.*T-3.)
PFDR(5,J,1)=(4.-8.*R)*(1.-S)*(1.-2.*S)*(1.-T)*(1.-2.*T)
PFDR(5,J,2)=4.*R*(1.-R)*(4.*S-3.)*(1.-T)*(1.-2.*T)
PFDR(5,J,3)=4.*R*(1.-R)*(1.-S)*(1.-2.*S)*(4.*T-3.)
PFDR(6,J,1)=(4.*R-1.)*4.*S*(1.-S)*(1.-T)*(1.-2.*T)
PFDR(6,J,2)=R*(2.*R-1.)*(4.-8.*S)*(1.-T)*(1.-2.*T)
PFDR(6,J,3)=R*(2.*R-1.)*4.*S*(1.-S)*(4.*T-3.)
PFDR(7,J,1)=(4.-8.*R)*S*(2.*S-1.)*(1.-T)*(1.-2.*T)
PFDR(7,J,2)=4.*R*(1.-R)*(4.*S-1.)*(1.-T)*(1.-2.*T)
PFDR(7,J,3)=4.*R*(1.-R)*S*(2.*S-1.)*(4.*T-3.)
PFDR(8,J,1)=(4.*R-3.)*4.*S*(1.-S)*(1.-T)*(1.-2.*T)
PFDR(8,J,2)=(1.-R)*(1.-2.*R)*(4.-8.*S)*(1.-T)*(1.-2.*T)
PFDR(8,J,3)=(1.-R)*(1.-2.*R)*4.*S*(1.-S)*(4.*T-3.)
PFDR(9,J,1)=(4.-8.*R)*4.*S*(1.-S)*(1.-T)*(1.-2.*T)
PFDR(9,J,2)=4.*R*(1.-R)*(4.-8.*S)*(1.-T)*(1.-2.*T)
PFDR(9,J,3)=4.*R*(1.-R)*4.*S*(1.-S)*(4.*T-3.)
PFDR(10,J,1)=(4.*R-3.)*(1.-S)*(1.-2.*S)*T*(2.*T-1.)
PFDR(10,J,2)=(1.-R)*(1.-2.*R)*(4.*S-3.)*T*(2.*T-1.)
PFDR(10,J,3)=(1.-R)*(1.-2.*R)*(1.-S)*(1.-2.*S)*(4.*T-1.)
PFDR(11,J,1)=(4.*R-1.)*(1.-S)*(1.-2.*S)*T*(2.*T-1.)
PFDR(11,J,2)=R*(2.*R-1.)*(4.*S-3.)*T*(2.*T-1.)
PFDR(11,J,3)=R*(2.*R-1.)*(1.-S)*(1.-2.*S)*(4.*T-1.)
PFDR(12,J,1)=(4.*R-1.)*S*(2.*S-1.)*T*(2.*T-1.)
PFDR(12,J,2)=R*(2.*R-1.)*(4.*S-1.)*T*(2.*T-1.)
PFDR(12,J,3)=R*(2.*R-1.)*S*(2.*S-1.)*(4.*T-1.)
PFDR(13,J,1)=(4.*R-3.)*S*(2.*S-1.)*T*(2.*T-1.)
PFDR(13,J,2)=(1.-R)*(1.-2.*R)*(4.*S-1.)*T*(2.*T-1.)
PFDR(13,J,3)=(1.-R)*(1.-2.*R)*S*(2.*S-1.)*(4.*T-1.)
PFDR(14,J,1)=(4.-8.*R)*(1.-S)*(1.-2.*S)*T*(2.*T-1.)
PFDR(14,J,2)=4.*R*(1.-R)*(4.*S-3.)*T*(2.*T-1.)
PFDR(14,J,3)=4.*R*(1.-R)*(1.-S)*(1.-2.*S)*(4.*T-1.)
PFDR(15,J,1)=(4.*R-1.)*4.*S*(1.-S)*T*(2.*T-1.)
PFDR(15,J,2)=R*(2.*R-1.)*(4.-8.*S)*T*(2.*T-1.)
PFDR(15,J,3)=R*(2.*R-1.)*4.*S*(1.-S)*(4.*T-1.)
PFDR(16,J,1)=(4.-8.*R)*S*(2.*S-1.)*T*(2.*T-1.)
PFDR(16,J,2)=4.*R*(1.-R)*(4.*S-1.)*T*(2.*T-1.)
PFDR(16,J,3)=4.*R*(1.-R)*S*(2.*S-1.)*(4.*T-1.)
PFDR(17,J,1)=(4.*R-3.)*4.*S*(1.-S)*T*(2.*T-1.)
PFDR(17,J,2)=(1.-R)*(1.-2.*R)*(4.-8.*S)*T*(2.*T-1.)
PFDR(17,J,3)=(1.-R)*(1.-2.*R)*4.*S*(1.-S)*(4.*T-1.)
PFDR(18,J,1)=(4.-8.*R)*4.*S*(1.-S)*T*(2.*T-1.)
PFDR(18,J,2)=4.*R*(1.-R)*(4.-8.*S)*T*(2.*T-1.)
PFDR(18,J,3)=4.*R*(1.-R)*4.*S*(1.-S)*(4.*T-1.)
PFDR(19,J,1)=(4.*R-3.)*(1.-S)*(1.-2.*S)*4.D0*T*(1.-T)
PFDR(19,J,2)=(1.-R)*(1.-2.*R)*(4.*S-3.)*4.D0*T*(1.-T)
PFDR(19,J,3)=(1.-R)*(1.-2.*R)*(1.-S)*(1.-2.*S)*(4.-8.*T)
PFDR(20,J,1)=(4.*R-1.)*(1.-S)*(1.-2.*S)*4.D0*T*(1.-T)
PFDR(20,J,2)=R*(2.*R-1.)*(4.*S-3.)*4.D0*T*(1.-T)
PFDR(20,J,3)=R*(2.*R-1.)*(1.-S)*(1.-2.*S)*(4.-8.*T)
PFDR(21,J,1)=(4.*R-1.)*S*(2.*S-1.)*4.D0*T*(1.-T)
PFDR(21,J,2)=R*(2.*R-1.)*(4.*S-1.)*4.D0*T*(1.-T)
```

```
      PFDR(21,J,3)=R*(2.*R-1.)*S*(2.*S-1.)*(4.-8.*T)
      PFDR(22,J,1)=(4.*R-3.)*S*(2.*S-1.)*4.D0*T*(1.-T)
      PFDR(22,J,2)=(1.-R)*(1.-2.*R)*(4.*S-1.)*4.D0*T*(1.-T)
      PFDR(22,J,3)=(1.-R)*(1.-2.*R)*S*(2.*S-1.)*(4.-8.*T)
      PFDR(23,J,1)=(4.-8.*R)*(1.-S)*(1.-2.*S)*4.D0*T*(1.-T)
      PFDR(23,J,2)=4.*R*(1.-R)*(4.*S-3.)*4.D0*T*(1.-T)
      PFDR(23,J,3)=4.*R*(1.-R)*(1.-S)*(1.-2.*S)*(4.-8.*T)
      PFDR(24,J,1)=(4.*R-1.)*4.*S*(1.-S)*4.D0*T*(1.-T)
      PFDR(24,J,2)=R*(2.*R-1.)*(4.-8.*S)*4.D0*T*(1.-T)
      PFDR(24,J,3)=R*(2.*R-1.)*4.*S*(1.-S)*(4.-8.*T)
      PFDR(25,J,1)=(4.-8.*R)*S*(2.*S-1.)*4.D0*T*(1.-T)
      PFDR(25,J,2)=4.*R*(1.-R)*(4.*S-1.)*4.D0*T*(1.-T)
      PFDR(25,J,3)=4.*R*(1.-R)*S*(2.*S-1.)*(4.-8.-T)
      PFDR(26,J,1)=(4.*R-3.)*4.*S*(1.-S)*4.D0*T*(1.-T)
      PFDR(26,J,2)=(1.-R)*(1.-2.*R)*(4.-8.*S)*4.D0*T*(1.-T)
      PFDR(26,J,3)=(1.-R)*(1.-2.*R)*4.*S*(1.-S)*(4.-8.*T)
      PFDR(27,J,1)=(4.-8.*R)*4.*S*(1.-S)*4.D0*T*(1.-T)
      PFDR(27,J,2)=4.D0*R*(1.-R)*(4.-8.*S)*4.D0*T*(1.-T)
      PFDR(27,J,3)=4.D0*R*(1.-R)*4.D0*S*(1.-S)*(4.-8.*T)
   30 CONTINUE
C Find the Jocobian matrix and its inverse.
      DO 40 I=1,27
      NTE(I)=NT(NED(NEL,I))
      VXE(I)=VX(NED(NEL,I))
      VYE(I)=VY(NED(NEL,I))
      VZE(I)=VZ(NED(NEL,I))
      XE(I)=X(NED(NEL,I))
      YE(I)=Y(NED(NEL,I))
      ZE(I)=Z(NED(NEL,I))
   40 CONTINUE
      DO 50 I=1,27
      DO 60 IR=1,3
      DO 70 ISUM=1,27
      TJ(IR,1,I)=TJ(IR,1,I)+PFDR(ISUM,I,IR)*XE(ISUM)
      TJ(IR,2,I)=TJ(IR,2,I)+PFDR(ISUM,I,IR)*YE(ISUM)
      TJ(IR,3,I)=TJ(IR,3,I)+PFDR(ISUM,I,IR)*ZE(ISUM)
   70 CONTINUE
C TJ(IR,J,I)=IR-th row and J-th column element of Jocobian matrix at
C            node I of element.
   60 CONTINUE
C Calculate the determinant of Jocobian matrix at node I and
C  the inverse matrix of Jocobian.
      DETJ(I)=TJ(1,1,I)*TJ(2,2,I)*TJ(3,3,I)+TJ(2,1,I)*TJ(3,2,I)*
     &  TJ(1,3,I)+TJ(3,1,I)*TJ(1,2,I)*TJ(2,3,I)-TJ(1,1,I)*TJ(3,2,I)*
     &  TJ(2,3,I)-TJ(2,1,I)*TJ(1,2,I)*TJ(3,3,I)-TJ(3,1,I)*TJ(2,2,I)*
     &  TJ(1,3,I)
C Inverse matrix of Jocobian.
      TJNV(1,1,I)=(TJ(2,2,I)*TJ(3,3,I)-TJ(3,2,I)*TJ(2,3,I))/DETJ(I)
      TJNV(2,1,I)=(TJ(2,3,I)*TJ(3,1,I)-TJ(2,1,I)*TJ(3,3,I))/DETJ(I)
      TJNV(3,1,I)=(TJ(2,1,I)*TJ(3,2,I)-TJ(3,1,I)*TJ(2,2,I))/DETJ(I)
      TJNV(1,2,I)=(TJ(3,2,I)*TJ(1,3,I)-TJ(1,2,I)*TJ(3,3,I))/DETJ(I)
      TJNV(2,2,I)=(TJ(1,1,I)*TJ(3,3,I)-TJ(3,1,I)*TJ(1,3,I))/DETJ(I)
      TJNV(3,2,I)=(TJ(1,2,I)*TJ(3,1,I)-TJ(1,1,I)*TJ(3,2,I))/DETJ(I)
      TJNV(1,3,I)=(TJ(1,2,I)*TJ(2,3,I)-TJ(2,2,I)*TJ(1,3,I))/DETJ(I)
      TJNV(2,3,I)=(TJ(2,1,I)*TJ(1,3,I)-TJ(1,1,I)*TJ(2,3,I))/DETJ(I)
```

```
      TJNV(3,3,I)=(TJ(1,1,I)*TJ(2,2,I)-TJ(1,2,I)*TJ(2,1,I))/DETJ(I)
   50 CONTINUE
C Find the derivatives of basis functions with respect to x,y,z.
      DO 80 I=1,27
      DO 90 J=1,27
      DO 100 K=1,3
C Here PFDX(I,J,K)=Gi(Xj) 's derivative with respect to x(K=1),y(K=2),
C  z(K=3).
      PFDX(I,J,K)=TJNV(K,1,J)*PFDR(I,J,1)+TJNV(K,2,J)*PFDR(I,J,2)+
     &TJNV(K,3,J)*PFDR(I,J,3)
  100 CONTINUE
   90 CONTINUE
   80 CONTINUE
C Evaluate the basis function for pressure stress.
      DO 110 I=1,8
      DO 120 J=1,8
      IF(I.EQ.J) THEN
       PS(I,J)=1.
      ELSE
       PS(I,J)=0.
      ENDIF
  120 CONTINUE
  110 CONTINUE
C Find the derivatives of velocity components with respect to x (N=1),
C  y (N=2), z (N=3) at node I.
      DO 200 I=1,27
      DO 200 N=1,3
      VXDX(I,N)=0.
      VYDX(I,N)=0.
      VZDX(I,N)=0.
  200 CONTINUE
      DO 210 I=1,27
      DO 220 N=1,3
      DO 230 J=1,27
      VXDX(I,N)=VXDX(I,N)+PFDX(J,I,N)*VXE(J)
      VYDX(I,N)=VYDX(I,N)+PFDX(J,I,N)*VYE(J)
      VZDX(I,N)=VZDX(I,N)+PFDX(J,I,N)*VZE(J)
  230 CONTINUE
  220 CONTINUE
  210 CONTINUE
      DO 300 I=1,27
      R=RR(I)
      S=SS(I)
      T=TT(I)
      PS(1,I)=(1.-R)*(1.-S)*(1.-T)
      PS(2,I)=R*(1.-S)*(1.-T)
      PS(3,I)=R*S*(1.-T)
      PS(4,I)=(1.-R)*S*(1.-T)
      PS(5,I)=(1.-R)*(1.-S)*T
      PS(6,I)=R*(1.-S)*T
      PS(7,I)=R*S*T
      PS(8,I)=(1.-R)*S*T
  300 CONTINUE
      RETURN
      END
```

```
C
C ***************************************************************
C
      SUBROUTINE GRID (R0,B,NX,NY,NZ,XX,YY,ZZ,NODT,NE,NR,NPR)
C  NR = number of TOTAL ROWS of the global matrix.
C   B = gap ratio (b = B*2*R0 )
C  R0 = 0.5
C  All of the parameters are in dimensionless form (L/D, u/U)
C X(I),Y(I),Z(I) are the coordinates of I-th node.
C NT(I) denotes the characteristic of node I.
C NT(I)= 1: i-TH NODE IS NOT ON BOUNDARY AND IS A VERTEX OF ELEMENT.
C NT(I)=-1: i-TH NODE IS NOT ON BOUNDARY AND IS NOT A VERTEX OF
C ELEMENT.
C NT(I)= 2: i-TH NODE IS ON FLEXIBLE BOUNDARY AND IS VERTEX OF
C ELEMENT.
C NT(I)=-2: i-TH NODE IS ON FLEXIBLE BOUNDARY ,IS NOT VERTEX OF
C ELEMENT.
C NT(I)= 3: i-TH NODE IS ON SOILD BOUNDARY AND IS VERTEX OF ELEMENT.
C NT(I)=-3: i-TH NODE IS ON SOLID BOUNDARY AND IS NOT VERTEX OF
C ELEMENT.
C NX = number of nodes in x-direction.  XX(i)= x-coordinates
C ***************************************************************
C
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION XX(20),YY(20),ZZ(20)
      COMMON /ASEMB/NROW(4000)
      COMMON/ELE/ X(4000),Y(4000),Z(4000),NT(4000)
      COMMON/ELNOD/NED(4000,27)
      COMMON/ELNOI/ NIJK(50,50,50)
      ZL=-1.0
      DO 5 L=1,50
      DO 5 M=1,50
      DO 5 N=1,50
      NIJK(L,M,N)=0
   5  CONTINUE
      NOD=0
      RB=-(R0+B)
      IF (ZL.LT.RB) ZL=RB
      DO 10 I=1,NX
      IF(XX(I).LE.-1.0.OR.XX(I).GE.1.0) THEN
      DO 20 J=1,NY
      DO 30 K=1,NZ
      NOD=NOD+1
      NIJK(I*2-1,J*2-1,K*2-1)=NOD
      X(NOD)=XX(I)
      Y(NOD)=YY(J)
      Z(NOD)=ZZ(K)
      NT(NOD)=1
      IF(I.EQ.1.OR.I.EQ.NX.OR.K.EQ.NZ) NT(NOD)=2
      IF(K.EQ.NZ) NT(NOD)=4
      IF(J.EQ.1.OR.J.EQ.NY) NT(NOD)=3
      IF(Z(NOD).LE.RB) NT(NOD)=3
   30 CONTINUE
   20 CONTINUE
      GOTO 10
```

```
      ENDIF
C In the domain x,y,z between [-1.2R,1.2R] ( a box),
1/SQRT(3)=0.288675
      IF(XX(I).EQ.-0.288675D0.OR.XX(I).EQ.0.288675D0) THEN
      DO 40 J=1,NY
      DO 50 K=1,NZ
      IF(YY(J).LE.-1.D0.OR.YY(J).GE.1.D0.OR.ZZ(K).LE.ZL.
     &OR.ZZ(K).GE.1.D0) THEN
      NOD=NOD+1
      NIJK(I*2-1,J*2-1,K*2-1)=NOD
      X(NOD)=XX(I)
      Y(NOD)=YY(J)
      Z(NOD)=ZZ(K)
      NT(NOD)=1
      IF(K.EQ.NZ) NT(NOD)=4
      IF(J.EQ.1.OR.J.EQ.NY) NT(NOD)=3
      IF(Z(NOD).LE.RB) NT(NOD)=3
      GOTO 50
      ENDIF
      IF(YY(J).EQ.-0.288675D0.OR.YY(J).EQ.0.288675D0) THEN
C Note: The possibility of ZZ(K) < or = -1.2R or ZZ(K) > or = 1.2R
C       has been covered at the beginning of DO 50.
      IF(ZZ(K).EQ.-0.288675D0.OR.ZZ(K).EQ.0.288675D0) THEN
      NOD=NOD+1
      NIJK(I*2-1,J*2-1,K*2-1)=NOD
      X(NOD)=XX(I)
      Y(NOD)=YY(J)
      Z(NOD)=ZZ(K)
      NT(NOD)=3
      GOTO 50
      ELSE
      NOD=NOD+1
      NIJK(I*2-1,J*2-1,K*2-1)=NOD
      X(NOD)=DSQRT(0.125D0-ZZ(K)*ZZ(K)*0.5D0)
      Y(NOD)=X(NOD)
      Z(NOD)=ZZ(K)
      IF(XX(I).LT.0.) X(NOD)=-X(NOD)
      IF(YY(J).LT.0.) Y(NOD)=-Y(NOD)
      NT(NOD)=3
      GOTO 50
      ENDIF
      ENDIF
      IF(YY(J).GT.-0.288675D0.AND.YY(J).LT.0.288675D0) THEN
      IF(ZZ(K).EQ.-0.288675D0.OR.ZZ(K).EQ.0.288675D0) THEN
      NOD=NOD+1
      NIJK(I*2-1,J*2-1,K*2-1)=NOD
      X(NOD)=DSQRT(0.125D0-YY(J)*YY(J)*0.5D0)
      Z(NOD)=X(NOD)
      Y(NOD)=YY(J)
      IF(XX(I).LT.0.) X(NOD)=-X(NOD)
      IF(ZZ(J).LT.0.) Z(NOD)=-Z(NOD)
      NT(NOD)=3
      GOTO 50
      ELSE
      NOD=NOD+1
```

```
      NIJK(I*2-1,J*2-1,K*2-1)=NOD
      X(NOD)=DSQRT(0.25D0-YY(J)*YY(J)-ZZ(K)*ZZ(K))
      Y(NOD)=YY(J)
      Z(NOD)=ZZ(K)
      IF(XX(I).LT.0.) X(NOD)=-X(NOD)
      NT(NOD)=3
      GOTO 50
      ENDIF
      ENDIF
   50 CONTINUE
   40 CONTINUE
      ENDIF
C
      IF(XX(I).GT.-0.288675D0.AND.XX(I).LT.0.288675D0) THEN
      DO 60 J=1,NY
      DO 70 K=1,NZ
      IF(YY(J).LE.-1.D0.OR.YY(J).GE.1.D0.OR.ZZ(K).LE.ZL.OR.
     &ZZ(K).GE.1.D0) THEN
      NOD=NOD+1
      NIJK(I*2-1,J*2-1,K*2-1)=NOD
      X(NOD)=XX(I)
      Y(NOD)=YY(J)
      Z(NOD)=ZZ(K)
      NT(NOD)=1
      IF(K.EQ.NZ) NT(NOD)=4
      IF(J.EQ.1.OR.J.EQ.NY) NT(NOD)=3
      IF(Z(NOD).LE.RB) NT(NOD)=3
      GOTO 70
      ENDIF
      IF(YY(J).EQ.-0.288675D0.OR.YY(J).EQ.0.288675D0) THEN
      IF(ZZ(K).EQ.-0.288675D0.OR.ZZ(K).EQ.0.288675D0) THEN
      NOD=NOD+1
      NIJK(I*2-1,J*2-1,K*2-1)=NOD
      X(NOD)=XX(I)
      Y(NOD)=DSQRT(0.125D0-XX(I)*XX(I)*0.5D0)
      Z(NOD)=Y(NOD)
      IF(YY(J).LT.0.) Y(NOD)=-Y(NOD)
      IF(ZZ(K).LT.0.) Z(NOD)=-Z(NOD)
      NT(NOD)=3
      GOTO 70
      ELSE
      NOD=NOD+1
      NIJK(I*2-1,J*2-1,K*2-1)=NOD
      X(NOD)=XX(I)
      Y(NOD)=DSQRT(0.25D0-XX(I)*XX(I)-ZZ(K)*ZZ(K))
      Z(NOD)=ZZ(K)
      IF(YY(J).LT.0.) Y(NOD)=-Y(NOD)
      NT(NOD)=3
      GOTO 70
      ENDIF
      ENDIF
      IF(YY(J).GT.-0.288675D0.AND.YY(J).LT.0.288675D0) THEN
      NOD=NOD+1
      NIJK(I*2-1,J*2-1,K*2-1)=NOD
      X(NOD)=XX(I)
```

```
      Y(NOD)=YY(J)
      Z(NOD)=DSQRT(0.25D0-XX(I)*XX(I)-YY(J)*YY(J))
      IF(ZZ(K).LT.0.) Z(NOD)=-Z(NOD)
      NT(NOD)=3
      GOTO 70
      ENDIF
   70 CONTINUE
   60 CONTINUE
      ENDIF
   10 CONTINUE
C NE= element number
      NE=0
      DO 80 I=1,NX-1
      IF(XX(I).LE.-1.D0.OR.XX(I).GE.0.288675D0) THEN
      DO 90 J=1,NY-1
      DO 100 K=1,NZ-1
      NE=NE+1
C Vertices of element:
      NED(NE,1)=NIJK(I*2-1,J*2-1,K*2-1)
      NED(NE,2)=NIJK(I*2+1,J*2-1,K*2-1)
      NED(NE,3)=NIJK(I*2+1,J*2-1,K*2+1)
      NED(NE,4)=NIJK(I*2-1,J*2-1,K*2+1)
      NED(NE,10)=NIJK(I*2-1,J*2+1,K*2-1)
      NED(NE,11)=NIJK(I*2+1,J*2+1,K*2-1)
      NED(NE,12)=NIJK(I*2+1,J*2+1,K*2+1)
      NED(NE,13)=NIJK(I*2-1,J*2+1,K*2+1)
C Interior nodes of element:
C         Order of sequence: 5,6,7,8,9,  14,15,16,17,18, 19,20,21,22,
C                            23,24,25,26,27
      I5=I*2
      J5=J*2-1
      K5=K*2-1
      CALL INTDA(I5,J5,K5,NE,5,1,2,NOD)
      I6=I*2+1
      J6=J*2-1
      K6=K*2
      CALL INTDA(I6,J6,K6,NE,6,2,3,NOD)
      I7=I*2
      J7=J*2-1
      K7=K*2+1
      CALL INTDA(I7,J7,K7,NE,7,3,4,NOD)
      I8=I*2-1
      J8=J*2-1
      K8=K*2
      CALL INTDA(I8,J8,K8,NE,8,1,4,NOD)
      I9=I*2
      J9=J*2-1
      K9=K*2
      CALL INTDB(I9,J9,K9,NE,9,1,2,3,4,NOD)
      I14=I*2
      J14=J*2+1
      K14=K*2-1
      CALL INTDA(I14,J14,K14,NE,14,10,11,NOD)
      I15=I*2+1
      J15=J*2+1
```

```
      K15=K*2
      CALL INTDA(I15,J15,K15,NE,15,11,12,NOD)
      I16=I*2
      J16=J*2+1
      K16=K*2+1
      CALL INTDA(I16,J16,K16,NE,16,12,13,NOD)
      I17=I*2-1
      J17=J*2+1
      K17=K*2
      CALL INTDA(I17,J17,K17,NE,17,10,13,NOD)
      I18=I*2
      J18=J*2+1
      K18=K*2
      CALL INTDB(I18,J18,K18,NE,18,10,11,12,13,NOD)
      I19=I*2-1
      J19=J*2
      K19=K*2-1
      CALL INTDA(I19,J19,K19,NE,19,1,10,NOD)
      I20=I*2+1
      J20=J*2
      K20=K*2-1
      CALL INTDA(I20,J20,K20,NE,20,2,11,NOD)
      I21=I*2+1
      J21=J*2
      K21=K*2+1
      CALL INTDA(I21,J21,K21,NE,21,3,12,NOD)
      I22=I*2-1
      J22=J*2
      K22=K*2+1
      CALL INTDA(I22,J22,K22,NE,22,4,13,NOD)
      I23=I*2
      J23=J*2
      K23=K*2-1
      CALL INTDA(I23,J23,K23,NE,23,19,20,NOD)
      I24=I*2+1
      J24=J*2
      K24=K*2
      CALL INTDA(I24,J24,K24,NE,24,20,21,NOD)
      I25=I*2
      J25=J*2
      K25=K*2+1
      CALL INTDA(I25,J25,K25,NE,25,21,22,NOD)
      I26=I*2-1
      J26=J*2
      K26=K*2
      CALL INTDA(I26,J26,K26,NE,26,19,22,NOD)
      I27=I*2
      J27=J*2
      K27=K*2
      CALL INTDB(I27,J27,K27,NE,27,19,20,21,22,NOD)
  100 CONTINUE
   90 CONTINUE
      ENDIF
      IF (XX(I).GT.-1.D0.AND.XX(I).LT.0.288675D0) THEN
      DO 110 J=1,NY-1
```

```
      DO 120 K=1,NZ-1
      IF((YY(J).GT.-1.D0.AND.YY(J).LT.0.288675D0).AND.
     &(ZZ(K).GT.ZL.AND.ZZ(K).LT.0.288675D0)) GOTO 120
      NE=NE+1
C Vertices of element:
      NED(NE,1)=NIJK(I*2-1,J*2-1,K*2-1)
      NED(NE,2)=NIJK(I*2+1,J*2-1,K*2-1)
      NED(NE,3)=NIJK(I*2+1,J*2-1,K*2+1)
      NED(NE,4)=NIJK(I*2-1,J*2-1,K*2+1)
      NED(NE,10)=NIJK(I*2-1,J*2+1,K*2-1)
      NED(NE,11)=NIJK(I*2+1,J*2+1,K*2-1)
      NED(NE,12)=NIJK(I*2+1,J*2+1,K*2+1)
      NED(NE,13)=NIJK(I*2-1,J*2+1,K*2+1)
C Interior nodes of element:
C         Order of sequence: 5,6,7,8,9,  14,15,16,17,18, 19,20,21,22,
C                            23,24,25,26,27
      I5=I*2
      J5=J*2-1
      K5=K*2-1
      CALL INTDA(I5,J5,K5,NE,5,1,2,NOD)
      I6=I*2+1
      J6=J*2-1
      K6=K*2
      CALL INTDA(I6,J6,K6,NE,6,2,3,NOD)
      I7=I*2
      J7=J*2-1
      K7=K*2+1
      CALL INTDA(I7,J7,K7,NE,7,3,4,NOD)
      I8=I*2-1
      J8=J*2-1
      K8=K*2
      CALL INTDA(I8,J8,K8,NE,8,1,4,NOD)
      I9=I*2
      J9=J*2-1
      K9=K*2
      CALL INTDB(I9,J9,K9,NE,9,1,2,3,4,NOD)
      I14=I*2
      J14=J*2+1
      K14=K*2-1
      CALL INTDA(I14,J14,K14,NE,14,10,11,NOD)
      I15=I*2+1
      J15=J*2+1
      K15=K*2
      CALL INTDA(I15,J15,K15,NE,15,11,12,NOD)
      I16=I*2
      J16=J*2+1
      K16=K*2+1
      CALL INTDA(I16,J16,K16,NE,16,12,13,NOD)
      I17=I*2-1
      J17=J*2+1
      K17=K*2
      CALL INTDA(I17,J17,K17,NE,17,10,13,NOD)
      I18=I*2
      J18=J*2+1
      K18=K*2
```

```
      CALL INTDB(I18,J18,K18,NE,18,10,11,12,13,NOD)
      I19=I*2-1
      J19=J*2
      K19=K*2-1
      CALL INTDA(I19,J19,K19,NE,19,1,10,NOD)
      I20=I*2+1
      J20=J*2
      K20=K*2-1
      CALL INTDA(I20,J20,K20,NE,20,2,11,NOD)
      I21=I*2+1
      J21=J*2
      K21=K*2+1
      CALL INTDA(I21,J21,K21,NE,21,3,12,NOD)
      I22=I*2-1
      J22=J*2
      K22=K*2+1
      CALL INTDA(I22,J22,K22,NE,22,4,13,NOD)
      I23=I*2
      J23=J*2
      K23=K*2-1
      CALL INTDA(I23,J23,K23,NE,23,19,20,NOD)
      I24=I*2+1
      J24=J*2
      K24=K*2
      CALL INTDA(I24,J24,K24,NE,24,20,21,NOD)
      I25=I*2
      J25=J*2
      K25=K*2+1
      CALL INTDA(I25,J25,K25,NE,25,21,22,NOD)
      I26=I*2-1
      J26=J*2
      K26=K*2
      CALL INTDA(I26,J26,K26,NE,26,19,22,NOD)
      I27=I*2
      J27=J*2
      K27=K*2
      CALL INTDB(I27,J27,K27,NE,27,19,20,21,22,NOD)
  120 CONTINUE
  110 CONTINUE
      ENDIF
   80 CONTINUE
      NODT=NOD
C Determine the position of velocity components and pressure stresses
C  of nodal point in the global matrix.
      NR=0
      DO 200 I=1,NE
      DO 210 J=1,27
C For nodal points at the vertices of elements, there are four
C unknowns:
C  Vx,Vy,Vz and P.  (For these nodes, NT(I) > 0 )
C For nodes not at the vertices of elements, there are three unknowns:
C  Vx,Vy,Vz.  (For these nodes, NT(I) < 0 )
C For nodes at boundary: Move the values to the right hand side and
C  delete the corresponding rows and columns. (NROW(I)=0)
      ND=NED(I,J)
```

```
      IF(NROW(ND).GT.0) GOTO 210
      IF (NT(ND).EQ.1) THEN
       NR=NR+4
       NPR=NR
C Note: NROW(I)= the row number of the last unknown at the current
C  node.
C       The first unknown will start at row/column number (NROW(ND)-3)
C       or (NROW(ND)-2).
      NROW(ND)=NR
      ELSEIF (NT(ND).EQ.-1) THEN
        NR=NR+3
        NROW(ND)=NR
      ELSEIF (NT(I).EQ.2.OR.NT(I).EQ.3) THEN
      NR=NR+1
      NROW(I)=NR
      ELSEIF (NT(I).EQ.4.OR.NT(I).EQ.-4) THEN
      NR=NR+3
      NROW(I)=NR
C FOR NT(I)=-2 OR -3, THERE IS NO PRESSURE TERM.  THE VELOCITIES ARE
C  KNOWN.
C FOR NT(I)=+4 OR -4, P=0 (+4 ONLY), Vx, Vy, VZ ARE UNKNOWN.
      ENDIF
  210 CONTINUE
  200 CONTINUE
      RETURN
      END
C
C  ---------------------------------------------------------------------
C
      SUBROUTINE INTDA(I,J,K,NE,N,NU,NL,NOD)
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON/ELE/ X(4000),Y(4000),Z(4000),NT(4000)
      COMMON/ELNOD/NED(4000,27)
      COMMON/ELNOI/ NIJK(50,50,50)
      IF (NIJK(I,J,K).EQ.0) THEN
      NOD=NOD+1
      X(NOD)=(X(NED(NE,NU))+X(NED(NE,NL)))*0.5D0
      Y(NOD)=(Y(NED(NE,NU))+Y(NED(NE,NL)))*0.5D0
      Z(NOD)=(Z(NED(NE,NU))+Z(NED(NE,NL)))*0.5D0
      T=X(NOD)*X(NOD)+Y(NOD)*Y(NOD)+Z(NOD)*Z(NOD)
      IF (T.LT.0.25) THEN
       IF(X(NOD).LT.0.288675D0.AND.X(NOD).GT.-0.288675D0) THEN
        IF(Y(NOD).LT.0.288675D0.AND.Y(NOD).GT.-0.288675D0) THEN
         SGN=1.
        IF (Z(NOD).LT.0.) SGN=-1.
        Z(NOD)=DSQRT(0.25D0-X(NOD)*X(NOD)-Y(NOD)*Y(NOD))
        IF(SGN.LT.0.) Z(NOD)=-Z(NOD)
       ELSE
        SGN=1.
        IF (Y(NOD).LT.0.) SGN=-1.
        Y(NOD)=DSQRT(0.25D0-X(NOD)*X(NOD)-Z(NOD)*Z(NOD))
        IF(SGN.LT.0.) Y(NOD)=-Y(NOD)
        ENDIF
       ELSE
        SGN=1.
```

```
          IF (X(NOD).LT.0.) SGN=-1.
          X(NOD)=DSQRT(0.25D0-Y(NOD)*Y(NOD)-Z(NOD)*Z(NOD))
          IF(SGN.LT.0.) X(NOD)=-X(NOD)
         ENDIF
        ENDIF
        NT(NOD)=-1
        IF(ABS(NT(NED(NE,NU))).EQ.2.AND.ABS(NT(NED(NE,NL))).EQ.2)
       & NT(NOD)=-2
        IF(ABS(NT(NED(NE,NU))).EQ.3.AND.ABS(NT(NED(NE,NL))).EQ.3)
       & NT(NOD)=-3
        IF(ABS(NT(NED(NE,NU))).EQ.3.AND.ABS(NT(NED(NE,NL))).EQ.2)
       & NT(NOD)=-2
        IF(ABS(NT(NED(NE,NU))).EQ.2.AND.ABS(NT(NED(NE,NL))).EQ.3)
       & NT(NOD)=-2
        IF(ABS(NT(NED(NE,NU))).EQ.4.AND.ABS(NT(NED(NE,NL))).EQ.4)
       & NT(NOD)=-4
        IF(ABS(NT(NED(NE,NU))).EQ.2.AND.ABS(NT(NED(NE,NL))).EQ.4)
       & NT(NOD)=-2
        IF(ABS(NT(NED(NE,NU))).EQ.3.AND.ABS(NT(NED(NE,NL))).EQ.4)
       & NT(NOD)=-4
        IF(ABS(NT(NED(NE,NU))).EQ.4.AND.ABS(NT(NED(NE,NL))).EQ.2)
       & NT(NOD)=-2
        IF(ABS(NT(NED(NE,NU))).EQ.4.AND.ABS(NT(NED(NE,NL))).EQ.3)
       & NT(NOD)=-4
        NIJK(I,J,K)=NOD
        ENDIF
        NED(NE,N)=NIJK(I,J,K)
        RETURN
        END
C
C  ---------------------------------------------------------------------
C
        SUBROUTINE INTDB(I,J,K,NE,N,N1,N2,N3,N4,NOD)
        IMPLICIT REAL*8(A-H,O-Z)
        COMMON/ELE/ X(4000),Y(4000),Z(4000),NT(4000)
        COMMON/ELNOD/NED(4000,27)
        COMMON/ELNOI/ NIJK(50,50,50)
        IF (NIJK(I,J,K).EQ.0) THEN
        NOD=NOD+1
        X(NOD)=(X(NED(NE,N1))+X(NED(NE,N2))+X(NED(NE,N3))+
       & X(NED(NE,N4)))*0.25D0
        Y(NOD)=(Y(NED(NE,N1))+Y(NED(NE,N2))+Y(NED(NE,N3))+
       & Y(NED(NE,N4)))*0.25D0
        Z(NOD)=(Z(NED(NE,N1))+Z(NED(NE,N2))+Z(NED(NE,N3))+
       & Z(NED(NE,N4)))*0.25D0
        NT(NOD)=-1
        IF(ABS(NT(NED(NE,N1))).GE.2.AND.ABS(NT(NED(NE,N2))).GE.2.AND.
       & ABS(NT(NED(NE,N3))).GE.2.AND.ABS(NT(NED(NE,N4))).GE.2)
        NT(NOD)=-2
        IF(ABS(NT(NED(NE,N1))).GE.3.AND.ABS(NT(NED(NE,N2))).GE.3.AND.
       & ABS(NT(NED(NE,N3))).GE.3.AND.ABS(NT(NED(NE,N4))).GE.3)
        NT(NOD)=-3
        NIJK(I,J,K)=NOD
        ENDIF
        NED(NE,N)=NIJK(I,J,K)
```

```
      RETURN
      END
C
C ***************************************************************
C This subroutine reads in the initial guessed for the velocities at
C all nodes and the velocities and pressures for the boundary nodes.
C For the boundary Conditions: the non-solid boundary at the
C downstream side has no known values for velocities and pressure.
C Variab|es: H = fLow depth, VIS=viscosity, VELU=uniform veLocity
C  HYR=hydraulic radius, SLP=channel slope, B=gap, DIA=sphere diameter
      SUBROUTINE INITBD (NOD)
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON /V2YCV/ YT(30),VP(30),NCURV
      COMMON/ELE/ X(4000),Y(4000),Z(4000),NT(4000)
      COMMON /VELK/VX(4000),VY(4000),VZ(4000)
C
      RI=0.5D0
      DO 10 I=1,NOD
      IF(ABS(NT(I)).NE.2) GOTO 10
      XD=X(I)
      YD=Y(I)
      ZD=Z(I)
      CALL BSPLN(NCURV,YT,VP,YD,V)
       R=DSQRT(XD*XD+YD*YD+ZD*ZD)
       TH=DACOS(XD/R)
       PS=DATAN2(YD,ZD)
       UR=V*DCOS(TH)*(1.D0+0.5*(RI/R)**3.-1.5*RI/R)
       UT=V*DSIN(TH)*(-1.D0+0.25*(RI/R)**3.+0.75*RI/R)
       VX(I)=UR*DCOS(TH)-UT*DSIN(TH)
       VY(I)=(UR*DSIN(TH)+UT*DCOS(TH))*DSIN(PS)
       VZ(I)=(UR*DSIN(TH)+UT*DCOS(TH))*DCOS(PS)
   10 CONTINUE
      RETURN
      END
C
C ***************************************************************
C
C In the main routine, add this line:
C     IF (PARAM.GT.0.) CALL READCV
      SUBROUTINE READCV(VELU1,DIA1)
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON /V2YCV/ YT(30),VP(30),NCURV
      READ(19,1000) NCURV
 1000 FORMAT(I5)
      DO 3 I=1,NCURV
      READ(19,*) YT(I),VP(I)
   3  CONTINUE
      DO 4 I=1,NCURV
      YT(I)=YT(I)/DIA1
      VP(I)=VP(I)/VELU1
    4 CONTINUE
      RETURN
      END
C
C ***************************************************************
```

```
C
      SUBROUTINE STORE(NROW,NPR)
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON/COEF/A(4000,473),KA(4000,473),F(4000)
C     OPEN(17,FILE='RSIDE.COE',STATUS='NEW')
C     OPEN(18,FILE='LOW.COE',STATUS='NEW',FORM='UNFORMATTED')
      MSIZE=NROW
      MXD=1
      DO 10 I=1,MSIZE
      DO 110 L=1,473
      IF(KA(I,L).LE.0) GOTO 115
      MXD=MAX0(MXD,ABS(KA(I,L)-I))
  110 CONTINUE
  115 NA1=L-1
      WRITE(18) I,NA1
      IF(NA1.LE.64) THEN
      WRITE(18) (KA(I,K),K=1,NA1)
      WRITE(18) (A(I,K),K=1,NA1)
      ELSE
      DO 35 K=1,NA1,64
      KE=K+63
      IF(KE.GT.NA1) KE=NA1
   35 WRITE(18)(KA(I,KW),KW=K,KE)
      DO 45 K=1,NA1,64
      KE=K+63
      IF(KE.GT.NA1) KE=NA1
   45 WRITE(18)(A(I,KW),KW=K,KE)
      ENDIF
   10 CONTINUE
      WRITE(17,1000) MSIZE,MXD,NPR
 1000 FORMAT(3I10)
      WRITE(17,1100) (F(K),K=1,MSIZE)
 1100 FORMAT(15D15.8)
C     CLOSE(17)
C     CLOSE(18)
      RETURN
      END
```

```
      PROGRAM SOLVER
C A BAND MATRIX SOLVER
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON/COEF/T(3505,775),F(3505)
C     OPEN(17,FILE='RSIDE.COE',STATUS='OLD')
C     OPEN(18,FILE='LOW.COE',STATUS='OLD',FORM='UNFORMATTED')
      READ(17,'(3I10)') N,NBD,NPR
      READ(17,'(15D15.8)') (F(K),K=1,N)
      DO 10 I=1,3505
      DO 10 J=1,775
   10 T(I,J)=0.
      CALL BAND(N,NBD,NPR)
      REWIND 17
      WRITE(17,'(3I10)') N,NBD,NPR
      WRITE(17,'(15D15.8)') (F(K),K=1,N)
      STOP
      END
C
      SUBROUTINE BAND(N,NBD,NPR)
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON/COEF/T(3505,775),F(3505)
      DIMENSION NA(473),A(473),TA(3505)
C For i<= NBD+1, row i runs from column 1 to column (i+NBD)
      DO 15 J=1,3505
      TA(J)=0.
   15 CONTINUE
      NBD1=NBD+1
      DO 5 J=1,473
      NA(J)=0
      A(J)=0.
    5 CONTINUE
C For case I=1.
      READ(18) IR,NR
      IF(NR.LE.64) THEN
      READ(18) (NA(K),K=1,NR)
      READ(18) (A(K),K=1,NR)
      ELSE
      DO 10 IP=1,NR,64
      IP2=MIN0(IP+63,NR)
      READ(18) (NA(K),K=IP,IP2)
   10 CONTINUE
      DO 20 IP=1,NR,64
      IP2=MIN0(IP+63,NR)
      READ(18) (A(K),K=IP,IP2)
   20 CONTINUE
      ENDIF
      DO 30 IP=1,NR
   30 TA(NA(IP))=A(IP)
      DV=TA(1)
      DO 40 IP=1,NBD1
      T(1,IP)=TA(IP)/DV
   40 CONTINUE
      F(1)=F(1)/DV
C
C Forward elimination
```

```
C
C For  2 < i < (N-NBD) and (N-NBD) <= i <= N
      DO 50 I=2,N
      DO 55 J=1,3505
   55 TA(J)=0.
      READ(18) IR,NR
      IF(NR.LE.64) THEN
      READ(18) (NA(K),K=1,NR)
      READ(18) (A(K),K=1,NR)
      ELSE
      DO 60 IP=1,NR,64
      IP2=MIN0(IP+63,NR)
      READ(18) (NA(K),K=IP,IP2)
  60  CONTINUE
      DO 70 IP=1,NR,64
      IP2=MIN0(IP+63,NR)
      READ(18) (A(K),K=IP,IP2)
  70  CONTINUE
      ENDIF
      DO 80 IP=1,NR
  80  TA(NA(IP)-I+NBD1)=A(IP)
      IC1=MAX0(1,I-NBD)
      DO 90 IC=IC1,I-1
      IPC=IC-I+NBD1
      IF(TA(IPC).EQ.0.) GOTO 90
      DV=TA(IPC)
      IR2=MIN0(N,IC+NBD)
      DO 100 IR=IC,IR2
      IRP=IR-I+NBD1
      TA(IRP)=TA(IRP)-T(IC,IR-IC+1)*DV
  100 CONTINUE
      F(I)=F(I)-F(IC)*DV
   90 CONTINUE
      DV=TA(NBD1)
      DO 110 IP=I,I+NBD
      T(I,IP-I+1)=TA(IP-I+NBD1)/DV
  110 CONTINUE
      F(I)=F(I)/DV
  50  CONTINUE
C
C Back substitution
C
C Note: N-N+NBD+1 = NBD+1  (for the above argument in T(N,?))
C     F(N)=1.
      DO 120 I=N-1,1,-1
      SUM=0.
      J2=MIN0(I+NBD,N)
      IM=-I+1
      DO 125 J=I+1,J2
 125  SUM=SUM+F(J)*T(I,J+IM)
      F(I)=F(I)-SUM
 120  CONTINUE
      RETURN
      END
```

Vita

The author was born on April 4, 1957 in Yeelan, Taiwan. He graduated from the National Chen Kung University in 1979 with the B. S. degree in Civil Engineering. After working for two years in a construction firm and the government, he went to the University of Iowa in Iowa City for the graduate study. He received the M. S. degree in Civil Engineering in May, 1983 and the M. S. degree in Mathematics in May, 1985. In the same year, he enrolled at the Virginia Tech for the graduate work of the Ph.D. degree in Civil Engineering.