Pulsatile flow in curved elastic tubes

This item was submitted to Loughborough University's Institutional Repository by the/an author.

Additional Information:

- A doctoral thesis submitted in partial fulfilment of the requirements for the award of Doctor of Philosophy at Loughborough University.

Metadata Record: [https://dspace.lboro.ac.uk/2134/32000](https://dspace.lboro.ac.uk/2134/32000)

Publisher: © John Ascough

Rights: This work is made available according to the conditions of the Creative Commons Attribution-NonCommercial-NoDerivatives 2.5 Generic (CC BY-NC-ND 2.5) licence. Full details of this licence are available at: [http://creativecommons.org/licenses/by-nc-nd/2.5/](http://creativecommons.org/licenses/by-nc-nd/2.5/)

Please cite the published version.
This item was submitted to Loughborough University as a PhD thesis by the author and is made available in the Institutional Repository (https://dspace.lboro.ac.uk/) under the following Creative Commons Licence conditions.

**Attribution-NonCommercial-NoDerivs 2.5**

You are free:
- to copy, distribute, display, and perform the work

Under the following conditions:

**Attribution.** You must attribute the work in the manner specified by the author or licensor.

**Noncommercial.** You may not use this work for commercial purposes.

**No Derivative Works.** You may not alter, transform, or build upon this work.

- For any reuse or distribution, you must make clear to others the license terms of this work.
- Any of these conditions can be waived if you get permission from the copyright holder.

Your fair use and other rights are in no way affected by the above.

This is a human-readable summary of the [Legal Code (the full license)](http://creativecommons.org/licenses/by-nc-nd/2.5/).

For the full text of this licence, please go to:
http://creativecommons.org/licenses/by-nc-nd/2.5/
PULSATILE FLOW IN CURVED ELASTIC TUBES

by

John Ascough

A Doctoral Thesis

Submitted in partial fulfilment of the requirements for the award of

DOCTOR OF PHILOSOPHY

of Loughborough University

October 1996

© by John Ascough 1996
DECLARATION

This is to certify that I am responsible for the work submitted in this thesis, that the original work is my own, except as specified in acknowledgements, and that neither the thesis nor the work contained herein has been submitted to this or any other institution for a higher degree.
ACKNOWLEDGEMENTS

My thanks are due to

Dr Keith Peat, my supervisor, for his advice and for his time when the demands on a university lecturer's time have never been so great.

Mr Graham Gerrard, Computer Centre Operations Manager, for his invaluable help on computing matters.

Professor C G Caro and Dr Kim Parker, Physiological Flow Studies Unit, Imperial College, for the brief, but extremely useful, discussions held with them at the outset of the project.

My wife for her patience, support and encouragement when her time was also in great demand.

I am also indebted to Professor M W Collins and Dr X Y Xu, City University, for the provision of additional material for the study.
PULSATILE FLOW IN CURVED ELASTIC TUBES

Abstract

Wall shear stresses are thought to have an influence on the formation of deposits of blood fats on the linings of the arteries, in atherosclerosis. Measuring velocities close to an artery wall to determine wall shears is difficult in view of the thinness of the boundary layer. Analytical solutions are limited to simple geometries and numerical analyses of three-dimensional, unsteady blood flows are expensive in terms of computational time. In the present study, finite element analyses of blood flow in models representative of the human aorta are based on two-dimensional sections in order to reduce the computational requirement. The Galerkin method of weighted residuals is employed to produce the finite element analogue to the Navier-Stokes equations for viscous, pulsatile, incompressible, Newtonian flow in toroidal coordinates. A time-stepping scheme is used to linearise the pressure cycle and the non-linear convective terms in the Navier-Stokes equations are handled by Picard iteration. Vessel fluid-wall interaction is modelled using alternate fluid flow and wall displacement solutions at each time step, where the fluid pressures determine the wall loading and the corresponding wall displacements are then used to update the element mesh for the next time step. For fully-developed flow, the analysis is based on a model of the tube cross-section. For entry flow, a semi-analytical finite element technique is investigated for the modelling of the required longitudinal section. Within the range of flow and frequency parameters permitted by the available computer resource, good agreement is obtained between published experimental and numerical solutions and the results of the fully developed flow analysis, the semi-analytical solution for entry flow and for the prediction of fluid-wall interaction.
## CONTENTS

Acknowledgements

Abstract

1 Introduction
   1.1 Background  
   1.2 Fluid Flow in Curved Tubes  
   1.3 The Present Study  

2 The Arterial System
   2.1 Anatomy and Mechanical Properties of the Arteries  
   2.2 Fluid Properties of the Blood  
   2.3 Fluid Mechanics of the Blood
      2.3.1 Blood Pressure  
      2.3.2 Velocity and Turbulence  
      2.3.3 Wall Shear Stress  
      2.3.4 Impedance and Wave Speed  
      2.3.5 Vascular Geometry  

3 Fluid Flow in Curved Tubes
   3.1 Experimental Studies
   3.2 Flow Regimes
      3.2.1 Steady Flow
      3.2.2 Pulsatile and Oscillatory Flow
      3.2.3 Entry Flow
      3.2.4 Bifurcation
   3.3 Compliant Walls
   3.4 Recent Developments

4 Finite Element Analysis in Fluid Flow
   4.1 The Finite Element Method
      4.1.1 Galerkin's Method
      4.1.2 Numerical Integration
      4.1.3 Solution of Equations
   4.2 Fluid Flow
      4.2.1 Flow without Inertia
      4.2.2 Flow with Inertia
      4.2.3 Pulsatile Flow
   4.3 Solid-fluid Interaction
      4.3.1 Equations of Motion for Vessel Wall
      4.3.2 Damping
      4.3.3 Dynamic Response

5 Finite Element Analysis of Pulsatile Flow in a Curved Tube
   5.1 Coordinate System
   5.2 Equations of Motion in Toroidal Coordinates
5.2.1 Continuity Equation 39
5.2.2 Momentum Equations 39
5.3 Fully Developed Flow 42
  5.3.1 Finite Element Mesh 43
  5.3.2 Boundary Conditions 44
  5.3.3 Fluid-Wall Interaction 45
5.4 Entry Flow 46
  5.4.1 Semi-analytical Solution 47
  5.4.2 Choice of Fourier terms 48
5.5 Finite Element Equations 52
  5.5.1 Radial Direction Terms 55
  5.5.2 Tangential Direction terms 56
  5.5.3 Axial Direction terms 57
  5.5.4 Pressure Gradient Terms 58
  5.5.5 Continuity Equation Terms 58
  5.5.6 Mass Terms 59
  5.5.7 Forcing Vector Terms 59
  5.5.8 Natural Boundary Conditions 59
  5.5.9 Axial Boundary Conditions 61

6 Results 65
  6.1 Straight Tube Solution 65
  6.2 Structured vs Unstructured Finite Element Mesh 69
  6.3 Fully Developed Flow in a Curved Tube 85
    6.3.1 Velocity and Pressure Distributions 89
    6.3.2 Comparison between codes PFECTX and PFECTL 102
  6.4 Wall Shear Stresses 103
  6.5 Entry Flow in a Curved Tube 107
    6.5.1 Steady Flow 107
    6.5.2 Oscillating Flow 115
  6.6 Elastic Wall 115
    6.6.1 Curved Tube 115
    6.6.2 Straight Tube 120

7 Discussion 126
  7.1 Fully Developed Flow 126
  7.2 Semi-analytical Solution 128
  7.3 Elastic Walls 130
    7.3.1 Curved Tube 131
    7.3.2 Straight Tube 131
  7.4 Future Work 132

8 Conclusions 135
Nomenclature
References
Appendix A  Gartling-Becker Finite Element Formulation
Appendix B  Equations of Motion in Toroidal Coordinates
Appendix C  Finite Element Program Description and Listing
Appendix D  Mesh Generation Program Description and Listing
Appendix E  Numerical Integration
1 INTRODUCTION

1.1 Background

The study of blood flow in arteries has become a subject of major interest in the biomedical engineering field. This interest is due to the suggested link between blood fluid mechanical effects and the depositing of lipids on artery linings in the disease known as atherosclerosis (Fry, 1968, Caro et al., 1971, Nerem and Levesque, 1983). This disease, in causing strokes and heart attacks, is the major cause of death in the developed world. It is believed that changes in the large arteries are apparent even in the very young and may well be the early stages of the disease. The earliest changes in the arteries are the formation of fatty streaks, under the endothelium, the lining of the artery. These may then develop into raised plaques, or atheromas, which with further development may then cause a narrowing of the vessel, or stenosis. If the consequent reduction of blood flow to the brain or heart muscles, ischaemia, becomes serious this may lead to a stroke or heart failure.

While the exact mechanism is unknown, it is believed that in addition to biochemical and bio-physical effects, fluid dynamics effects, particularly blood pressure and wall shear rates, may have a significant effect either on the siting and initiation of atheroma, or on the development of the disease once initiated by some other cause. What is well known is that atheromas tend to occur in the larger arteries and appear on the inside of the bend in curved arteries and on the outer walls of arterial bifurcations. The blood vessels most affected are the highly curved aortic arch, the curved coronary arteries and the carotid, iliac and popliteal arterial bifurcations (Fig 1.1).

The involvement of such diverse groups as the medical profession, applied mathematicians, physicists and engineers in research into arterial blood flows reflects the wide range of analytical techniques needed to study the subject. Most fluid dynamic research into arterial flows is centred on the two main structures affected by atheroma; the main arterial bifurcations and the curved sections of artery, such as the aortic arch, of interest in the present study.

The coronary arteries and the aortic arch, which has a large curvature (Fig 1.2), are usually modelled as curved tubes, the branches of the aortic arch being neglected in the interests of achieving practicable solutions. Fluid flow patterns in curved tubes have been extensively investigated using a variety of analytical and numerical techniques and the many publications on this subject illustrate a continuing interest extending from the seminal papers by Dean (1927, 1928) to the present day.
Fig 1.1 Location of atheroma in man
(from Spain, D.M., 1966)
1.2 Fluid Flow in Curved Tubes

Fluid flow in a curved tube, although appearing at first sight to be a relatively simple flow regime is, in fact, extremely complex, involving a transverse, secondary flow system set up by the centrifugal forces due to the tube curvature. In steady, fully developed, laminar flow, the faster moving fluid near the tube centre is subjected to higher centrifugal forces than that near the wall and is drawn towards the outside of the curve. The slower moving fluid is forced to move inwards around the tube wall until it joins the fluid moving outwards at the centreline. This results in the secondary
flow consisting of two vortices symmetrical about the centre line of the tube, in the plane of curvature, and giving helical streamlines (Fig 1.3). Steady flow is characterised by the *Dean* number, which relates the Reynolds number to the *curvature ratio*, the ratio of tube radius to radius of curvature. Time-dependent flows are characterised by both the Dean and the *Womersley* number, which relates the kinematic viscosity to the frequency and tube radius (see Sections 3.1 and 3.2). Time-dependent flows are referred to either as oscillatory or pulsatile, depending on the absence or presence of a mean flow. Flow patterns also depend on whether or not the flow is fully-developed. The direction of rotation of the vortices changes, for example, in both entry and pulsatile flows. Furthermore, at high Dean numbers, it has been proposed that there is a solution branch, or *bifurcation*, where the two-vortex system may break down into a four-vortex system. The complexity of curved tube flow and the absence of theoretical solutions is reflected in the considerable research interest shown in curved tube flow and the diversity of approaches to its analysis, reviewed in Pedley (1980), Berger *et al.* (1983) and Ito (1987).

While earlier theoretical analyses were essentially based on mathematical approaches, finite difference and finite element methods currently dominate the research in this field. The interest in obtaining numerical results is due to the need to complement the experimental effort where there are difficulties both in making *in vivo* measurements on the one hand and in making realistic laboratory models of arteries on the other. The advantage of access to the model and the easier application of measurement techniques in the laboratory is outweighed by the difficulty in producing realistic models of arteries from both the geometry and material properties aspects, while *in vivo* measurements must be minimally invasive, precluding the use of many experimental techniques.
Considerable progress has been made with experimental techniques for measuring flow and pressure in artery models, but there are still difficulties in their use. The determination of wall shear stress requires accurate measurements of blood velocity in the thin boundary layer, the Stokes' layer, adjacent to the artery lining. The use of hot-wire or hot-film anemometers is difficult since the presence of the probe itself affects the flow (Pedley, 1980). Doppler ultrasound flowmeters have poor space resolution (Wells, 1990) and laser Doppler anemometry (LDA) requires a large amount of data to measure pulsatile flow (e.g., Agrawal et al., 1978). Flow visualisation techniques are also limited, but are popular for the immediate and full field flow patterns produced. Contrast-dye injection and seeding particles have been used in models of bifurcations and curved tubes to produce both qualitative and quantitative results (e.g., Karino et al., 1983, Liepsch et al., 1983, Sudo et al., 1992).

Newer methods such as magnetic resonance imaging (MRI) (Cunha et al., 1994), for in vivo measurements in particular (Caro et al., 1992), and developments in particle-image velocimetry (PIV) for in vitro measurements (Adrian, 1991) promise future progress on the experimental front, but up to the present the consequence of the difficulties in making flow measurements has resulted in considerable interest in the numerical analysis of blood flow to complement the experimental data obtained.

Progress in the numerical analysis of general blood flow problems has reached the stage where analyses are currently being carried out using finite difference and finite element models of arterial bifurcations with realistic geometries for pulsatile, Newtonian or non-Newtonian flow and with rigid or distensible vessel walls.

The continuing interest in curved tube flows for other than biomedical applications is reflected in the research by Sillekens (1995), who studied the mixed convection phenomenon using particle imaging and finite element techniques, and Jayanti (1990) who used the finite difference technique to study turbulent and two-phase flows in curved tubes. Both of these studies have application in chemical engineering, but are of direct interest to the study of curved tube flows in general.

If a numerical method is adopted the basic choice exists between a finite element or finite difference approach to the problem and arguments are presented for both approaches. In the past finite difference or finite volume methods have dominated the computational fluid dynamics field, mainly because of the predominance of commercial finite difference codes developed during the earlier days of computer use. There are however difficulties in generating meshes for finite difference models and the additional difficulty of generating a vessel wall mesh with changing geometry has probably inhibited progress in incorporating a distensible vessel wall into finite difference blood flow models. The mesh generation problem is much easier to accomplish with the finite element method and this method promises the easier approach to the distensible wall
problem.

The development of curved tube finite element analyses to full three-dimensional flow, non-Newtonian fluid flow and visco-elastic wall properties is feasible, in principle, but three-dimensional analyses for the simpler cases of Newtonian fluid and elastic walls already require the use of very high-powered computers. Perktold et al. (1991), for example, analysed a slightly-curved tube, representing the left main coronary artery, three-dimensionally, and it is significant that their analysis gives limited detail of the flow in the tube cross section. This gives some argument for using the available computational power in detailed two-dimensional analyses, rather than in relatively coarse three-dimensional models. Carrying out full three-dimensional fluid flow analyses with more modest computer facilities is impracticable and two-dimensional models are the only course in this situation. In addition the increasing use of personal computers and work stations in numerical analysis (Sedlár, 1993, Dvinsky and Ohja, 1994) depends on the availability of techniques which economise on computational power.

1.3 The Present Study

Reviews of the arterial system and the literature on curved tube flows and finite element analysis in fluid flow are carried out to provide a background to the main study. The main study covers the development of a two-dimensional finite element model for pulsatile, incompressible, laminar, Newtonian flow in a curved tube with an elastic wall. The primary objective of the study is to investigate whether results of adequate quality to usefully complement experimental results can be obtained without recourse to high-powered computers. The analysis is based on the two-dimensional flow in cross- and longitudinal-sections of the tube. Analysing cross-section flow alone has the obvious limitation of restricting applications to fully developed flows, unless some method of predicting downstream cross-section flows is employed as, for example, by Patankar and Spalding (1974). Entry flow, of considerable importance in the aorta, since the entry length is greater than the length of the aortic arch throughout most of the pressure cycle, could be analysed, however, if it were possible to model a longitudinal section of the tube. Part of the present study is devoted to an investigation of the feasibility of using a semi-analytical finite element analysis. This would enable the non-axisymmetric effects due to the centrifugal effects, arising from the curvature of the tube, to be incorporated into the analysis of a quasi-axisymmetric tube. In this type of analysis, in terms of computational effort, the three-dimensional problem is effectively reduced to a two-dimensional problem, where the finite element model is based on a radial plane through the fluid and tube wall. Although the use of the semi-analytical technique is well established in other fields,
such as stress analysis and heat transfer, its use in fluid dynamics problems appears to have been documented only rarely.

In both cross- and longitudinal sections, in addition to the reduced demands on computer capacity required through the reduction of the dimensions of the problem, the modelling of the tube wall is also much simplified. Where, in a three-dimensional model, a shell element would be required, in the two-dimensional case a plane stress or plane strain element can be used. In addition, a choice exists between a coupled solution, where equations involving both fluid variables and wall distension are present, or an uncoupled solution where the solutions for fluid variables and wall distension are separate. The present study, on the basis of the published literature, favours the use of an uncoupled solution.
2 THE ARTERIAL SYSTEM
The objective of the current study is to produce a finite element model of pulsatile fluid flow in a curved elastic tube and, by analogy, model blood flow in the human aorta. In order to produce a meaningful numerical model an understanding of the fluid mechanics of the systemic arterial system in man would be useful. The physical and fluid mechanics aspects of the system are reviewed by McDonald (1974) and Nichols and O'Rourke (1990), Caro et al. (1978), Pedley (1980) and Fung (1984). Nerem (1981, 1983) discusses the fluid dynamics of blood and the interaction of blood with the artery wall in the context of atherogenesis.

2.1 Anatomy and Mechanical Properties of the Arteries
The cardio-vascular system in man consists of the heart and the pulmonary and systemic arterial and venous systems. The systemic arterial system supplies blood to every part of the body, excepting the vessels concerned purely with the exchange of gases in the lungs which form the pulmonary system, and is the system of most interest in the study of atherosclerosis.

The systemic arterial system commences with the aorta which receives blood flow directly from the heart and ends, after considerable branching and sub-branching, in the capillaries. The aorta, which is typically about 2.5 cm in diameter in man, extends from the left ventricle of the heart, runs upwards for about 3 cm in the ascending aorta, then curves three-dimensionally in the aortic arch where it discharges into branches for the head and upper limbs. It then descends through the body, branching into the main arteries for the lower body en route, thence into the arterioles, 30-100 μm in diameter and finally into the capillaries which are typically 4-5 μm in diameter. The capillaries then connect with the smallest veins, the venules, which join together to form a system of increasingly larger veins, ending with the vena cava which discharges into the right atrium of the heart. The average time for blood to complete a full circuit of the system is about 60 sec.

Almost all the arteries are branched and curved in a complicated way and it is significant from a modelling point of view that there are apparently very few straight segments where the fluid mechanics of long, straight tubes can be applied. While the arteries taper individually, the area of cross-section of the arterial 'tree' increases with distance from the heart. The arterial branches are generally asymmetrical with branching angles of less than 90 degrees.

The artery walls are traditionally divided into three layers (Fig 2.1). The innermost, the tunica intima, consists of the endothelium, the continuous lining of all blood vessels and a surrounding layer of collagen fibres. The middle layer, the tunica media,
usually the thickest part of the vessel wall, consists mainly of *elastin* and smooth muscle in the larger arteries and spirally wound, smooth muscle fibres and collagen in the smaller. The outer layer, the *tunica adventitia*, which although relatively thick, consists mainly of connective tissue. The boundary between the adventitia and the surrounding tissue is usually ill-defined. Arteries are about 60% water and 40% tissue.

![Fig 2.1 The structure of an artery](image)

Collagen and smooth muscle are markedly *visco-elastic*, exhibiting creep, stress relaxation and hysteresis. Collagen has a Young's modulus of around 1 GPa. Smooth muscle has a modulus which varies from 0.1 MPa in the relaxed condition to about 1.2 MPa in the active condition. Elastin has a non-linear stress-strain relationship having a modulus of about 0.3 MPa up to 40% strain, thereafter increasing in stiffness. This increase in stiffness is necessary to avoid a 'blow out', where the increasing vessel radius leads in turn to increasing circumferential stress (Roach, 1957). The way in which the modulus of elasticity for a human aorta varies with the diameter, $d$, is illustrated qualitatively in Fig 2.2, where $d_o$ is the original diameter.

With such a complex arrangement of muscle, elastin and collagen in the artery wall, estimation of the properties of the wall as a whole from its constituent parts is virtually impossible and reliance must be placed on the available experimental data for wall properties. The experimental determination of artery properties for fluid mechanics calculations is made difficult by the problem of measurement *in vivo* as arteries typically shorten by about 40% on excision. The available *in vivo* data is obtained from
animals and little information is available for human arteries. Caro et al., (1978) give a table of canine experimental data. The problem of modelling blood vessel properties mathematically has been studied extensively, however, and Atabek (1968), Lighthill (1975), Cox (1982), Weizsächer and Pascale (1982), Bauer et al. (1982) and Fung (1993), among many others, derived expressions for the distensibility of arteries and arterial muscle under static and dynamic pressure loading. Iida (1989) used a mathematical model to investigate the effects of in vivo changes in the arterial wall due to flow-dependent, metabolic, and pressure-dependent, myogenic, effects. Numerous experiments have been carried out both in studying the properties of tissue and in simulating these properties for in vitro flow and wave speed measurements (Ling and Atabek, 1972, Papageorgiou and Jones, 1987a, 1987b).

Simplifying assumptions are necessary for numerical analyses, as in the present study, when the modelling of blood-artery interaction would be impracticable with other than a simple pressure-distension relationship for the artery properties. In view of the non-linearity of the stress-strain relationship for the artery wall, it is convenient to give the modulus of elasticity, $E$, as an incremental modulus for a uniform, cylindrical, homogeneous, isotropic tube wall, thickness $h$ and diameter $d$, in the form

$$D_i^{-1} = \frac{E}{1 - \nu^2} \left( \frac{h}{d} \right)$$

(2.1)

where the distensibility, $D_i$, is given by

$$D_i = \frac{1}{A} \left( \frac{\partial A}{\partial P_{tm}} \right)$$

(2.2)

$A$ is the vessel cross-sectional area and $P_{tm}$ is the transmural pressure.
Because the boundary of an artery is usually ill-defined it is understood that the surrounding tissues affect the inertia, stiffness and viscosity of the wall and all these are effectively increased (Patel and Fry, 1966). Because wall inertia plays no part in pulse-wave propagation the radial tethering of the arteries to the surrounding tissue may be ignored (Atabek, 1968).

### 2.2 Fluid Properties of the Blood

Blood is a colloidal solution, the red blood cells, the *erythrocytes*, and the white cells, the *leukocytes*, together with the platelets constituting the solid phase, about 46% of the volume, and the *plasma*, the fluid phase. The red cells are the most numerous, the proportion of red cells per unit volume, being normally about $5 \times 10^6$ $\text{mm}^3$ of whole blood, usually expressed as the *haematocrit*, or concentration of red cells per unit volume, and given as a percentage. The normal haematocrit is usually around 45%. The white cells are larger but about 0.1 % as numerous as red cells. The platelets are much smaller than red cells, 2-4 $\mu$ m in diameter, and these play a significant role in blood clotting and in the formation of blockages, or *stenoses* in atherosclerosis. Plasma is a solution of large molecules and is usually regarded as a homogeneous Newtonian fluid.

Blood viscosity is known to vary with flow rate; this non-Newtonian, *thixotropic* behaviour being attributed to the highly deformable nature of the red blood cells, which as the main component of the solid phase have the most effect on blood viscosity. Blood viscosity decreases with an increase in flow rate. It also varies with blood vessel diameter, so that in vessels above 0.5 mm in diameter there is no effect on viscosity.
the red blood cells being small compared with the vessel diameter, while below 0.5 mm diameter there is a fall in viscosity as vessel diameter is reduced. Below diameters of 15 μm viscosity again increases.

Blood viscosity is independent of the velocity gradient, or shear rate, for shear rates less than 100 /s. The assumption of blood being Newtonian for calculation purposes is therefore in question since in pulsatile flow the shear rate passes through zero twice per cycle, in separated flow in some regions the average shear rate will be small, and in the centre of straight tubes the shear rate will also be small. As a first approximation, however, blood is generally assumed to be a homogeneous, Newtonian fluid of density 1050 kg/m³ and kinematic viscosity $4 \times 10^{-6}$ m²/s.

2.3 Fluid Mechanics of the Blood

2.3.1 Blood Pressure

Blood pressure is composed of three parts; the atmospheric pressure $p_0$, taken as the pressure in the right atrium when its muscles are relaxed, the hydrostatic pressure, $-p \cdot gh$, where $h$ is measured positively, vertically from the right atrium, and the excess pressure, or blood pressure, $p$, (Lighthill, 1975) generated by the heart.

During the heart beat the pressure in the left ventricle increases as the muscles contract until it exceeds that of the aorta when the valve between them opens and ejection begins. About halfway through ejection an adverse pressure gradient closes the valve, decelerating the outflow. The ventricular and aortic pressures then fall rapidly as the heart muscles relax and blood flows out of the aorta. The closure of the aortic valve is indicated by a sudden reduction on a blood velocity plot known as the dicrotic notch.

The pulse is increasingly delayed with increasing distance from the heart, indicating that it is propagated as a wave. The shape of the pulse also changes so that the amplitude increases and the front becomes steeper, the mean pressure falls only 0.5 kN/m² along the whole aorta, but in the smaller arteries less than 1 mm diameter, the pressure falls more rapidly.

Pressure was thought initially to have an effect on atherogenesis in driving a blood-arterial wall lipid transport process. Data now available tends to reject this process as being of minor importance. Fry (1976), however, reported that high pressures in the aorta distort the geometry of the branches to the extent where the blood flow becomes much more disturbed, accentuating the problem of wall shears. Further arguments exist as to whether it is the mean or pulse pressure that is significant (Nerem, 1981).
2.3.2 Velocity and Turbulence

Blood is ejected from the heart for about a third to a half of the beat. The ejection phase is followed by backflow as the aortic valve closes and the amount of the backflow increases with distance from the heart, being accommodated by swelling of the aorta. Velocity waveforms downstream compared with that at the entrance to the aorta show that the velocity decreases with distance from the heart.

Most canine velocity measurements show smooth variation with time indicating that blood flow is predominantly laminar. Measurements of the aortic velocity occasionally show that high frequency disturbances are present and the frequency and randomness of these disturbances indicate that turbulence is present. The frequency spectrum is not the same, however, as that of fully developed straight pipe flow turbulence. A scaling analysis of the whole animal kingdom (Stahl, 1967) indicates that aortic turbulence is a function of body size, so that animals larger than dogs can be expected to show turbulent flow in the aorta. The indications are that the aortic flow in man may well be turbulent at times, particularly during exercise. Reynolds numbers in the aorta of 5700-8900, corresponding to velocities of 96-141 cm/s have been recorded by hot-film probe (Stein and Sabah, 1976).

Elsewhere in the system, local dilatation of the arteries is known to occur downstream of stenoses large enough to cause turbulence, but not so large as to reduce the flow to a laminar trickle (Roach, 1972). It has also been proposed that as the aorta is known to become stiffer and dilated with age, while other arteries become more flexible, this may be a physiological response to aortic turbulence. There is also some evidence that turbulence may affect the permeability of the artery wall and damage the endothelium (Caro et al., 1978).

2.3.3 Wall Shear Stress

It is increasingly accepted that mechanical influences contribute to the initiation of atherosclerosis and evidence exists that particular sites where fatty deposits have been found include the outer walls of arterial junctions and the inside walls of curves such as the aortic arch where shear rates are relatively low. Caro's evidence (Caro et al., 1971), later supported by Ku et al. (1985), indicates a positive correlation between plaque location and areas of low shear stress where low permeability of the endothelium is thought to inhibit the removal of lipids from the artery wall. Fry's results (Fry, 1968) suggest that the opposite is true and that, in causing damage to the endothelium, high wall shear rates contribute to the initiation of deposits. Nerem and Levesque (1983) concluded that high wall shear stress may be an influence on initiation of the disease in the very young, but that low wall shear stresses are an influence in adults.
The indications are that wall permeability and hence the ability for large molecules to pass through the artery wall and initiate deposits may well depend on pressure oscillations as well as wall shear stress. Cornhill and Roach (1976) have found that the sites where fatty deposits occur are initially extremely localised and the argument about which mechanical factors influence the formation of fatty deposits is made more complicated by the inability to measure pressures and wall shears accurately with the fine resolution required.

2.3.4 Impedance and Wave Speed
There is a great deal of interest in the way in which blood pressure and flow vary as the blood flows through the arterial tree. Impedance and pressure wave propagation and reflection occupy a significant proportion of the literature (McDonald, 1974. Caro et al., 1978, Pedley, 1980). The relationship between oscillating pressure and flow is given, by analogy with electrical alternating voltage and current, by the impedance, \( Z \), where

\[
Z = \frac{P}{Q}
\]  

(2.3)

where \( P \) and \( Q \) are the blood pressure and flow, respectively. There is interest in arterial impedance in particular as this may give valuable diagnostic information (Papageorgiou and Jones, 1989) in, for example, assessing the success of arterial bypass operations. The concept of an overall resistance of the arterial tree to blood flow appeals to the intuition of medical practitioners in assessing the general arterial health of a patient.

2.3.5 Vascular Geometry
Nerem and Levesque (1983) summarised the case for fluid dynamics as a factor in atherogenesis. They conclude that vascular geometry is an important factor in initiating the disease, whether inherited or influenced by a factor such as hypertension, and that local fluid flow patterns due to this and their influence on the endothelium, coupled with bio-chemical factors, determine the focal nature of the disease. Whether or not an individual had a predilection for the disease would depend not only on the bio-chemically related factors, but also on the branching pattern, degree of curvature and tortuosity of the arteries most at risk. With increased knowledge of these processes and advances in \textit{in vivo} imaging techniques it might ultimately be possible to identify individuals with abnormal risk at an early stage. The theoretical study of blood flows in complementing experimental studies has a leading role to play in determining the relationship between mechanical influences and the initiation of atherosclerosis.
3 FLUID FLOW IN CURVED TUBES

A considerable number of experimental, analytical and numerical studies have been carried out in the search for improved knowledge of the flow characteristics of the arterial system, in turn leading ultimately to a better understanding of the way in which atherosclerosis is initiated and develops. The analysis of blood flow for physiologically realistic problems is difficult, however, involving the pulsating flow of a non-Newtonian fluid through a visco-elastic tube of complex geometry. The diversity of curved tube flows is illustrated by Smith (1975) who studied ten types of flow for steady, pulsatile and oscillatory curved tube flow, while Sudo et al. (1992) have evidence of five types of flow pattern in fully-developed oscillatory flow alone (Fig. 6.14).

Progress in the field has been gradual and broadly-based among the available analytical, numerical and experimental techniques. The reviews of curved tube flow by Pedley (1980), Berger et al. (1983) and Ito (1987) demonstrate the complexity and variety of curved tube flows. Chandran (1993) describes the flow in the human aorta specifically.

3.1 Experimental Studies

Of the experimental techniques employed in arterial flows, hot-wire or hot-film anemometers have been used by Ling and Atabek (1972) and Pedley (1980), in vitro, and by Nerem and Seed (1972) and Stein and Sabah (1976), in vivo, Doppler ultrasound flowmeters by Agrawal et al. (1978) and laser Doppler anemometry by Muguercia et al. (1993), Swanson et al. (1993) and Khodadadi et al. (1988). Flow visualisation techniques are popular for the immediate and full field flow patterns produced. Contrast-dye injection (Lyne, 1970, Giddens et al., 1980) and seeding particles (Karino et al., 1979, Bertelsen and Thorsen, 1982, Sudo et al., 1992) have been used in models of bifurcations and curved tubes to produce both qualitative and quantitative results. Lobodzinski (1982), modelled an aortic arch complete with its branches for his dye injection experiments. Liepsch et al. (1983, 1986) measured steady and pulsating flows in both rigid polyester resin and elastic silicon rubber models of arterial bifurcations using injected-dye flow visualisation and LDA.

3.2 Flow regimes

3.2.1 Steady Flow

Analytical or mathematical studies of problems relevant to the arterial flow problem have tended to concentrate on flows in straight or curved tubes as the more geometrically complex problem of a bifurcation is less amenable to mathematical analysis. Since the aortic arch and coronary arteries in particular can be modelled as
curved tubes, and these are in any case important arteries from the atherosclerosis viewpoint, many mathematical analyses for curved tube flows problem appear in the literature. Although some curved tube analyses are motivated by the engineering applications of helical pipes, typically from a heat transfer and mixing viewpoint, currently the majority are concerned with blood flow. Consequently most of our current understanding of the complex flows occurring in curved tubes arises from the blood flow analyses.

Although the behaviour of fluids in curved ducts had been studied experimentally much earlier (Thomson, 1876, Eustice, 1910), Dean (1927, 1928) produced the seminal mathematical analysis for steady flow in curved tubes, introducing the Dean number, $K$, to relate inertial and centrifugal forces to viscous forces in curved tube flow. Dean's definition of this relationship, others use different versions also confusingly called Dean numbers, is

$$K = 2\left(\frac{a}{R}\right)^2 \left(\frac{W_o a}{v}\right)^2$$  \hspace{1cm} (3.1)

where $W_o$ is the maximum axial velocity in a straight tube of radius $a$, $R$ is the radius of curvature and $v$, the kinematic viscosity. McConalogue and Srivastava (1968) use

$$D = \left(\frac{2a^3}{v^2 R}\right)^{1/2} \frac{Ga^2}{\mu}$$  \hspace{1cm} (3.2)

where $G$ is the pressure gradient. Van Dyke (1978) and Berger et al. (1983), discuss the various forms of Dean number in use and Berger et al. propose that the preferred version should be

$$\kappa = \left(\frac{a}{R}\right)^{1/2} \left(\frac{2a W_o}{v}\right) = 2\left(\frac{a}{R}\right)^{1/2} Re_m$$  \hspace{1cm} (3.3)

where $W_o$ is the mean axial velocity, and, incidentally, easier to measure experimentally than Dean's maximum velocity.

Here, unless stated otherwise, the Dean number is given by

$$De = \left(\frac{a}{R}\right)^{1/2} \left(\frac{a W_o}{v}\right) = Re_m \left(\frac{a}{R}\right)^{1/2}$$ \hspace{1cm} (3.4)

Dean's analyses, using Fourier series for the solution in powers of the Dean number
Dean's analyses, using Fourier series for the solution in powers of the Dean number were confined to Dean numbers up to 96 with small curvatures and it was some time before McConalogue and Srivastava (1968) improved on this number, extending Dean's work to take account of Dean numbers up to 605 in steady flow, making use of computers to solve their equations. Before these, Barua (1963) used a boundary layer model to analyse a much larger range of Dean number flows. McConalogue and Srivastava, able to produce more comprehensive results than Barua, produced theoretical axial velocity contours and streamlines, showing the shift in peak axial velocity away from the centre of the tube towards the outside of the curve and the movement of the centres of the secondary flow vortices in the same direction. Since McConalogue and Srivastava published their results there have been a number of finite difference solutions. Greenspan (1973), for example, produced solutions for Dean numbers from 10 to 5000 and was able to show that the movement of the axial velocity peak moves further towards the outside of the curve as Dean number increases. At $D = 5000$, his contours appear to be developing oscillatory regions, consistent with the onset of turbulence. He showed that the movement of the core of the secondary flow streamlines is clockwise about the tube centreline up to $D = 500$ and then for increasing $D$ moves anticlockwise. Collins and Dennis (1975), using a more accurate finite difference scheme than Greenspan, explored the same range of Dean numbers and produced more detailed secondary flow contours. They also compared results obtained with three finite difference grid densities. Van Dyke (1978) used a computer to extend Dean's series solution to 24 terms and showed good agreement with experiment and other numerical solutions up to $K = 100$.

The approach adopted in analysing steady flows in curved tubes depends on the Dean number, $D$, so that at low Dean numbers ($D<600$), series solutions are prevalent while at high numbers, up to $D = 5000$, boundary layer models are prevalent. The intermediate range requires the solution of the non-linear Navier-Stokes equations and numerical methods are therefore prevalent in this range.

### 3.2.2 Pulsatile and Oscillating Flow

Womersley (1955, 1957), in representing the flow in a straight blood vessel, made the first mathematical analysis of pulsating flow in rigid and elastic tubes, introducing the Womersley parameter

$$\alpha = a \sqrt{\frac{\omega}{\nu}}$$

(3.5)

where $\omega$ is the angular frequency of the sinusoidally-varying pressure gradient. The Womersley number, $\alpha$, gives an indication of the extent that the flow differs from the
dominates the flow, the flow is mainly inviscid at the core and the viscous boundary layer, the Stokes layer, is relatively thin. At low numbers (< 1) the flow is dominated by viscous forces, inertia is low and the flow is parabolic.

Womersley's analyses were based on flow through a straight tube. The first study of pulsating flow in curved tubes was made by Lyne (1970), who introduced the parameter

\[
\beta = \sqrt{\frac{2\nu}{\omega a^2}} \tag{3.6}
\]

which represents the ratio of the thickness of the Stokes layer to the tube radius, and the Reynolds number for secondary flow

\[
R_s = \frac{W_0^2 a}{R \omega \nu} \tag{3.7}
\]

Lyne showed that for small values of \( \beta \), at certain points in the cycle the secondary flow direction in the core is opposite to that in fully-developed, steady flow (Fig.3.2). This was confirmed analytically by Zalosh and Nelson (1973) and later experimentally, by Chandran and Yearwood (1981).

Ling and Atabek (1972) used simplifying assumptions in the equations of motion for the arterial wall and in the fluid flow equations to enable them to carry out a finite difference analyses incorporating both pulsatile flow and a distensible artery wall for a model of an aorta. They confirmed their results with hot film velocity measurements of the flow through a compliant model artery, attempting to reproduce the non-linear behaviour of a canine aorta. Chandran et al. (1974, 1979) extended the work on
curved tubes in pulsatile flow to take account of a thin elastic tube wall. Finite
difference analyses have been carried out by O'Brien et al. (1976) on a two-
dimensional model of an arterial branch, using stream functions, Chakravarty and
Datta (1989) for an axisymmetric analysis of flow through a stenosis and Chang and
Tarbell (1985) for pulsatile flow in a model of a cross-section of the aorta, also using
a stream function approach. More recently, Hamakiotes and Berger (1988, 1990)
alysed steady and pulsating curved tube flow using a finite difference approach for
primitive variables, with a two-dimensional model of the tube cross-section. Swanson
et al. (1993) used LDA for a model based on Hamakiotes' and Berger's data, but
were unable to confirm the existence of period tripling, where the flow patterns
repeat over a three cycle period, predicted by them.

3.2.3 Entry Flow
Entry flow has been of continuing interest to fluid dynamics researchers, on account of
the pressure drop and friction losses occurring at the entrance to a tube. The problem
is difficult to model experimentally due to the difficulty in producing a uniform
entrance velocity profile and also of determining where the fully developed profile
occurs. There is interest in entry flow for aortic blood flow in particular, since, in the
aorta, the pulsating flow never becomes fully developed. The unsteady entry length, \( l_1 \),
for a frequency, \( \omega \), and core velocity, \( u \), is given by Caro et al. (1978) as

\[
l_1 \approx 3.4 \frac{u}{\omega}
\]

The steady entrance length, \( x_e \), for a vessel radius \( a \), is given by Lew and Fung
(1970) as

\[
x_e/a = 0.16 \text{ Re}
\]

and by Shah and London (1978) as

\[
x_e/a = \frac{0.6}{1 + 0.035 \text{ Re}} + 0.056 \text{ Re}
\]

Much interest has been shown in entry flow in tubes by researchers into arterial
flows. Ward-Smith (1980), summarised the experimental and analytical work carried
out on entry flow for straight tubes. Entry flow in curved tubes, however, is of more
interest to arterial blood flow researchers. Patankar et al. (1974) carried out flow and
heat transfer analyses for entry flows in large radius of curvature tubes using a two-
dimensional model of the tube cross-section with a pressure correction technique for
dimensional model of the tube cross-section with a pressure correction technique for predicting the flows at downstream sections. Yao and Berger (1975) analysed steady entry flow using boundary layer matching and Singh et al. (1978) extended Singh's work on a steady entry flow analysis (Singh, 1976) to show that in entry flow the secondary flow direction is again the reverse of that in steady, fully-developed flow. Pedley (1976) used an approximate method to analyse viscous boundary layer flow for reversing flow and predicted wall shear rates in entry flow for a straight tube. Rindt et al., (1991) carried out a finite element analysis for a 90°-curved tube, checking their results against laser Doppler anemometry measurements. Muguercia et al. (1993) used laser Doppler anemometry to study developing flow over the whole 180° of a curved tube. Agrawal et al. (1978) also used laser Doppler anemometry to study steady, developing flow in the entrance to a curved pipe. Snyder et al. (1985) studied the skew of the axial flow profile for steady entry flow, using a hot-wire anemometer to measure velocities. Chandran and Yearwood (1981) made an experimental study of aortic flow using a three-dimensional hot-film velocity probe to produce detailed velocity profiles for pulsatile entry flow and Talbot and Gong (1983) studied pulsatile entry flow using laser Doppler velocimetry.

3.2.4 Bifurcation

Researchers have suggested that at high Reynolds numbers two distinct secondary flow regimes can exist. Daskopoulos and Linhoff (1989) used a collocation method and Nandakumar and Masliyah (1982), Bara et al. (1992) and Kao (1992) used finite difference procedures to study the bifurcation structure in flows at high Reynolds numbers when the two vortex secondary flow may, or may not, break down into a four vortex flow. It is thought that these flows are theoretical and that in the real situation imperfections will mask any bifurcation. Experimental evidence does not as yet appear to have been produced to support the existence of bifurcation flows.

3.3 Compliant Walls

Since the distensibility of the vessel wall has an influence on pulsatile flow characteristics and wave propagation speeds, any improvement on the three-dimensional rigid-wall analyses reported so far in the literature must involve the modelling of the interaction of blood flow and the deformation of the vessel wall. Olsen and Shapiro (1967) analysed viscous laminar and turbulent flow in an elastic tube using a perturbation technique for a one-dimensional model, supporting their results by displacement measurements on a latex tube. Reuderink (1991) and Reuderink et al. (1993) use a one-dimensional finite element model and an uncoupled fluid-wall interaction. Chandran et al. (1979), used a perturbation method to analyse
pulsatile flow in a curved elastic tube with a coupled fluid-wall interaction. Their interest was in the stresses in the vessel wall, however, and they made no comment on the effect of wall distensibility on the flow field. Perktold et al. (1994) have successfully produced analyses of flows in models of the carotid bifurcation with both rigid and distensible walls and non-Newtonian fluid properties and these analyses probably represent the state-of-the-art finite element analyses for arterial flows. Henry and Collins (1993a, 1993b), Xu and Collins (1995) and Lan et al., (1995) have produced coupled finite element and finite volume analyses of the interaction between fluid and compliant walls for both straight tubes and models of arterial bifurcations. With the exception of these fluid flow analyses and analyses carried out specifically to investigate wave propagation in tubes (Anliker et al., 1968, Atabek, 1968, Anderson and Johnson, 1990, Belardinelli and Cavalcanti, 1992, Dardel, 1987,1988, Horsten et al., 1989 and Lucey and Carpenter, 1992), however, many analysts still make the assumption of rigid vessel walls.

The biomedical importance of wall compliance is unclear. Perktold et al. (1994), for example, demonstrate that wall distensibility modifies the flow field in a bifurcation and that wall shear stresses are reduced, but the effect is minimal. Pedley (1976) argues that the flow field in the aorta is relatively unaffected by wall elasticity, due to the length of the pressure wave and that, similarly, taper is also unimportant.

3.4 Recent Developments

Since a complete study of real blood flow problems requires a fully three-dimensional analysis and analytical solutions are necessarily confined to relatively simple problems, numerical approaches using either finite element or finite difference methods and employing sophisticated software packages are becoming increasingly common. Advances in experimental techniques are also being increasingly exploited in this area. Collins (1995) reviews current developments and the progress made in the use of numerical and experimental techniques in biomedical research. Analytical solutions still appear in the literature, however, and the continuing interest in this approach for curved tube flow is demonstrated by Misra et al. (1994) who solved the boundary layer momentum integral equations for steady flow.

The finite difference method in the form of the finite volume method has been more frequently used in blood flow analyses than the finite element method (Xu et al., 1992, and Xu and Collins, 1994) chiefly on account of the historical predominance of finite volume codes in computational fluid dynamics (Patankar and Spalding, 1972). Xu and Collins (1990) review the numerical analyses carried out for arterial bifurcations and argue for finite difference methods on the basis that these are less expensive in computer storage and run times than finite element methods, although, as reported
above, wall and fluid interaction analyses tend to be finite element based. Zienkiewicz (1975) made the case for the finite element method before the developments in finite volume grid generation weakened the main argument in favour of the finite element method; that complex geometries and problem boundary conditions are much more easily handled. More recently Pepper and Baker (1988) discussed the relative merits of the two methods. There are still problems in modelling realistic geometries with the finite difference method, however, where spurious fluid motions are experienced due to the transformations required for body-fitted coordinates (Thompson, 1988). The continuing use of finite difference analyses in blood flow problems is ensured by the progress being made in grid generation and problems such as the modelling of a distensible vessel wall, with the attendant problem of a changing mesh geometry, will eventually be solved with this method, if this has not already been done.

While the earlier 'theoretical' studies were based on analytical approaches involving simplified problem geometries, blood fluid properties and vessel wall characteristics, the use of the finite difference and finite element techniques, capable of analysing much more realistic problems, has become the more usual approach today. At the same time experimental techniques have become progressively more sophisticated with the advent of the laser and developments in electronic instrumentation so that currently a whole variety of flow measurement techniques is available to the researcher. MRI now frequently used in in vivo measurements (Klipstein et al., 1987, Caro et al., 1992), is now being used in laboratory experiments. Chung et al. (1993) evaluated the technique for a steady flow analysis, comparing their results with those from a finite difference code and with Snyder et al. (1985). The finite element analysis of incompressible, Newtonian flow, with inertia, is a well-established procedure for steady and pulsatile flow and the number of references to applications in arterial flows is increasing annually. Earlier, two-dimensional problems were prevalent and Wille (1982) analysed pulsatile flow in a straight tube with a stenosis using an axisymmetric finite element model, while Chen and Fan (1984) analysed fully-developed, steady flow in a model of a tube cross-section. Pietrabissa et al. (1990) analysed a two-dimensional model of an aorto-coronary bypass under steady flow conditions. Wille (1984), Reuderink et al., (1989), Rindt et al. (1990) and Krijger et al. (1992) were able to model steady flows in representative three-dimensional models of carotid artery and aortic bifurcations. Perktold et al. (1986, 1989, 1991a, 1991b, 1994), modelled three-dimensional pulsatile flows through bifurcations of various branch angles and geometries. Tandon et al. (1994) analysed steady flow though a bifurcation using non-Newtonian, thixotropic properties for the blood, representing normal and diseased states.

The effect of torsion on curved tube flow has been studied by Germano (1982,
Liu and Masliyah (1993) and Chen and Jan (1992), who show that for high Dean numbers in steady flow and significant tube coil pitch the secondary flow pattern is both rotated and distorted. Since the aorta has significant torsion, the effect of this adds yet another factor to the complexity of aortic flow.

Some analyses focus on blood flow through the smaller arteries, involving the difficult representation of the blood at this level. Hogan and Henriksen (1989), for example, use the finite element method to analyse the flow through a stenosis using a micropolar model to include the effect of rotation of blood solids in an attempt to improve the representation of blood flow properties. Simon et al. (1993) review the finite element analyses carried out specifically to model arterial wall mechanics. More recently, Kumar and Naidu (1995) demonstrate the versatility of the finite element method in modelling blood flow through an aneurysm as non-linear, two-phase flow, to take account of the effect of the dynamics of the blood solids.

Parker (1977) studied transition in arterial flows experimentally, but there are few references to numerical turbulent flow analyses for curved tubes in the literature. It is clear that turbulence does occur in the aorta at peak velocities, e.g., during exercise. Nakamura et al. (1993) confirmed experimentally that turbulence occurred in a model of the aorta at Reynolds and Womersley numbers of 4000 and 10, respectively.
The finite element method has been developed considerably from its use for aircraft structural analysis in the late 1950's into a numerical method applicable to all branches of science and engineering. Of the large number of texts published on the method, Zienkiewicz (1977) and Zienkiewicz and Taylor (1989, 1991) wrote what are generally regarded as the standard texts on the general use of the finite element method. Huebner and Thornton (1982) and Stasa (1985) give accounts of the method and its applications from an engineer's viewpoint. Bathe and Wilson (1976), among many others, summarise the numerical aspects of the finite element method and Smith and Griffiths (1988) concentrate on the computer programming of the method. Baker (1985) and Girault and Raviart (1986), with a more rigorous mathematical treatment, concentrate on the fluid dynamics applications of the method.

4.1 The Finite Element Method

The finite element method is based on the discretisation of the problem domain into a finite number of sub-domains or elements. Elements are assumed to be connected to each other at points on their boundaries, called nodes (Fig 4.1). The value of the field variable(s) at any point within an element is then interpolated among the nodal values using shape functions as interpolating functions. Shape functions are chosen so that continuity of the variable, or its derivatives, depending on the type of problem, is maintained across the boundaries of the elements, throughout the domain. Typically the shape functions, and the distribution of the field variable within an element, will be linear, quadratic or cubic so that the generally complex distribution of the field variable throughout the domain is expressed locally within each element in a relatively simple form. In regions of the domain where there is a rapid change in a variable, a finer mesh of elements is required if the true distribution is to be reasonably well approximated by the model.

In general the transformation of the governing equations of a problem into finite element equations in matrix form for solution by computer, is carried out either by using the variational calculus, if a functional for the problem exists, or more commonly by means of a weighted residual method (Zienkiewicz and Taylor, 1970; Finlayson, 1975; Fletcher, 1984). Galerkin's method, in the form known as the Bubnov-Galerkin formulation, where the weighting functions are the shape functions, is frequently used in practice and this approach has been adopted for the present study.

4.1.1 Galerkin's Method

Consider the governing equation of a field problem in the form
\[ Lu = f \] \hspace{1cm} (4.1)

where \( u \) represents the unknown variable(s), \( L \) the (partial) differential operator and \( f \), the forcing function. Galerkin's method minimises the error, \( R \), or residual, arising from the use of the shape functions to describe the distribution of the variables within an element, by weighting the residual, using the shape function, integrating the result over the domain of the element and equating this to zero. This has the effect of minimising the error in some average way over the volume of the element.

\[ L\tilde{u} - f = R \] \hspace{1cm} (4.2)

\[ \int_{\Omega} (L\tilde{u} - f) \; N_j \; d\Omega = 0 \] \hspace{1cm} (4.3)

where

\[ \tilde{u} = N_i u_i^e \] \hspace{1cm} (4.4)

The order of the differential operator(s) is reduced, where necessary, using integration by parts; Green's Theorem in two- or Gauss's Theorem in three-dimensions.

4.1.2. Numerical Integration
Integration of the terms produced by Galerkin's method, involving the shape functions and derivatives of the shape functions, is normally impossible analytically and is usually carried out using numerical integration in the form of Gauss quadrature. The optimum number of sampling points, the Gauss points, for the integration has been subject to extensive investigation and the guidelines are now well established (Zienkiewicz, 1977).

4.1.3 Solution of Equations
Having established the finite element equations in matrix form and included the problem boundary conditions by modifying the appropriate equations, it only requires the use of a suitable equation solver to determine the unknown variables. In view of the non-linear convective terms in the matrix equations for fluid flow, the matrices are unsymmetrical and solution procedures based on Gauss elimination must therefore make use of full pivoting.

In order to make the most efficient use of the available computer storage, the frontal solution technique, proposed by Irons (1970) and revised by Hood (1976) to take account of unsymmetric matrices, has been widely adopted. In this technique only the
matrices for those elements surrounding a particular degree of freedom are assembled at any one time. The resulting equations are then partially solved and the coefficients produced can then be stored economically as a vector on a scratchfile, for example, as in the present study. The next degree of freedom is then treated in the same way, and so on, so that the effect is that of a front moving through the model. On completion of the assembly and partial solution phase, back-substitution using the coefficients recalled from the scratchfile, yields the solutions.

The solution of large numbers of equations in three-dimensional flow problems becomes prohibitive in terms of computer process time and storage requirements with direct methods such as the frontal solution technique. Direct methods have therefore generally been replaced on large computers by iterative methods, where their use has led to very efficient schemes (Hirsch, 1990).

4.2 Fluid Flow
4.2.1 Flow without Inertia
Many formulations for the finite element analysis of viscous, incompressible fluid flow have been proposed (Oden, 1973, Hood and Taylor, 1974, Kawahara et al., 1974, Baker, 1975, Hughes et al., 1975, Olson, 1975, Yamada et al., 1975, and Gartling and Becker, 1976). Hutton (1980) reviewed the approaches available at that time. The Gartling-Becker formulation been found by the author to be relatively problem-free and is the formulation used here. The formulation is summarised in Appendix A.

The choice exists between using stream function and vorticity or fluid velocities and pressure as the unknown variables in a two-dimensional problem in fluid dynamics. The popularity of primitive variables is due to the relative ease of defining problem boundary conditions and the direct utility of solutions in the form of pressure and velocity fields (Olson and Tuann, 1975).

To demonstrate the use of the Gartling-Becker formulation the Navier-Stokes equations plus the continuity equation, for incompressible, viscous fluid flow, with inertia, are first expressed in axisymmetric coordinates, \( x, r \)

\[
\rho \left( \frac{\partial u}{\partial t} + v \frac{\partial u}{\partial r} + u \frac{\partial u}{\partial x} \right) = \frac{1}{r} \frac{\partial}{\partial r} \left( r \tau_{rx} \right) + \frac{\partial \sigma_x}{\partial x} - \frac{\partial \rho}{\partial x} \tag{4.5}
\]

\[
\rho \left( \frac{\partial v}{\partial t} + v \frac{\partial v}{\partial r} + u \frac{\partial v}{\partial x} \right) = \frac{1}{r} \frac{\partial}{\partial r} \left( \sigma_r \right) + \frac{\partial \tau_{rx}}{\partial x} - \frac{\partial \rho}{\partial r} \tag{4.6}
\]
\[
\frac{1}{r} \frac{\partial (r v)}{\partial r} + \frac{\partial u}{\partial x} = 0
\]  
\[(4.7)\]

where

\[
\sigma_{sr} = \mu \left[ 2 \frac{\partial u}{\partial x} \right]
\]  
\[(4.8)\]

\[
\sigma_{rr} = \mu \left[ 2 \frac{\partial v}{\partial r} \right]
\]  
\[(4.9)\]

\[
\tau_{rx} = \tau_{xr} = \mu \left[ \frac{\partial u}{\partial r} + \frac{\partial v}{\partial x} \right]
\]  
\[(4.10)\]

The fluid domain is sub-divided into eight-noded, rectangular, isoparametric elements (Fig 4.1). Quadratic shape functions are used to interpolate the velocities \( u, v \) within each element, among the eight nodal values, and linear functions are used to interpolate the pressure \( p \) among the four corner nodal values.

\[
u = N_i^u (x, r) \quad u_i, \quad i = 1, 8
\]  
\[(4.11)\]

\[
v = N_i^v (x, r) \quad v_i, \quad i = 1, 8
\]  
\[(4.12)\]

\[
p = N_i^p (x, r) \quad p_i, \quad i = 1, 4
\]  
\[(4.13)\]
The use of *mixed-interpolation*, shape functions of different order for velocities and pressure, in primitive variable problems is generally accepted and is justified by Yamada et al. (1975), through examination of a variational formula, and by Hood and Taylor (1974), from consideration of the errors involved in the weighted residual formulation. Olson and Tuann (1975) demonstrated that rigid body motions can occur if this procedure is not followed and Bercovier and Pirrione (1979), justified the procedure on the basis of error studies for Stokes' Flow.

Applying Galerkin's method to the Navier-Stokes equations, Eqtns 4.5-4.7, yields the equations:

\[
\int_\Omega \left[ \rho \left( \frac{\partial u}{\partial t} + \nu \frac{\partial u}{\partial r} + u \frac{\partial u}{\partial x} \right) - \frac{1}{r} \frac{\partial}{\partial r}\left( r \tau_{rr} \right) - \frac{\partial \sigma_{x}}{\partial x} + \frac{\partial p}{\partial x} \right] N_i^x \, d\Omega = 0 \tag{4.14}
\]

\[
\int_\Omega \left[ \rho \left( \frac{\partial v}{\partial t} + \nu \frac{\partial v}{\partial r} + u \frac{\partial v}{\partial x} \right) - \frac{1}{r} \frac{\partial}{\partial r}\left( r \tau_{rv} \right) - \frac{\partial \sigma_{r}}{\partial r} + \frac{\partial p}{\partial r} \right] N_i^r \, d\Omega = 0 \tag{4.15}
\]

\[
\int_\Omega \left( \frac{1}{r} \frac{\partial}{\partial r} + \frac{\partial u}{\partial x} \right) N_i^x \, d\Omega = 0 \tag{4.16}
\]

Substitution of Eqtns 4.8-4.10 into Eqtns 4.14-4.16, application of Green's Theorem, and neglecting the mass and convective acceleration terms, leads to the finite element equations for viscous, incompressible fluid flow, without inertia. In matrix form these are

\[
\begin{bmatrix}
2K_{11} + K_{22} & K_{12} & L_1 & u \\
K_{21} & K_{11} + 2K_{22} & L_2 & v \\
L_1 & L_2 & 0 & p
\end{bmatrix}
\begin{bmatrix}
R^u \\
R^v \\
0
\end{bmatrix} = 0
\tag{4.17}
\]

where

\[
K_{11} = \mu \int_\Omega \frac{\partial N_i^x}{\partial x} \frac{\partial N_j^x}{\partial x} \, d\Omega \tag{4.18}
\]

\[
K_{12} = \mu \int_\Omega \frac{\partial N_i^x}{\partial x} \frac{\partial N_j^r}{\partial r} \, d\Omega \tag{4.19}
\]

(4.20)
\[
K_{ij} = \mu \int_{\Omega} \frac{\partial N^*_i}{\partial r} \frac{\partial N^*_j}{\partial r} \, d\Omega \\
L_1 = \int_{\Omega} N^*_i \frac{\partial N^*_i}{\partial r} \, d\Omega \\
\tag{4.21}
L_2 = \int_{\Omega} N^*_i \frac{\partial N^*_i}{\partial r} \, d\Omega \\
\tag{4.22}
R^u = \int_{\Gamma} N^*_i \bar{\sigma}_r \, dS \\
\tag{4.23}
R^r = \int_{\Gamma} N^*_i \bar{\sigma}_r \, dS \\
\tag{4.24}
\]

4.2.2 Flow with Inertia

In order to complete the finite element formulation Eqtns (4.17) require the addition of a mass matrix and acceleration terms. In the finite element method a choice exists between consistent and diagonalised or lumped mass matrices. Consistent mass matrices are derived on the basis of equal kinetic energies for the masses attributed to each degree of freedom and for the true distributed mass. Lumped mass matrices are based on the distribution of mass among the degrees of freedom on purely physical grounds, for example, on the basis of the volume of the element associated with a particular node. The mass matrices in finite difference and finite volume methods are lumped mass matrices.

In general consistent mass matrices are fully occupied whereas lumped mass matrices consist of diagonal terms only. The use of lumped masses therefore offers computational advantages. Ideally the optimum accuracy in structural dynamics and heat transfer problems is obtained using a consistent mass matrix (Gresho and Lee, 1971; Gresho et al., 1975) and there is no reason to doubt that this also true for fluid dynamics problems, although the mass matrix problem does not appear to have been discussed in any detail for fluid dynamics problems. Here, in view of the time-stepping scheme adopted, a single term is used for the mass in order to be able to use the same mass term on both sides of Eqtn (4.28).

There are several methods of achieving a diagonal mass matrix, Cook (1989), for example, discusses the use of HRZ (Hinton, Rock and Zienkiewicz, 1976) and optimal lumping schemes. The HRZ scheme uses the leading diagonal of the consistent
mass matrix, but with the terms scaled to preserve the total mass of the element. The optimal scheme is based on the fact that there are no off diagonal terms in the mass matrix if this is generated by choosing integration points to coincide with nodes with translational degrees of freedom only (Fix, 1976). Stasa (1985), Zienkiewicz and Taylor (1989) and Gresho (1975) discuss the HRZ scheme and the use of the alternative row-sum and column-sum schemes where either the rows or the column terms of the consistent mass matrix are summed and placed on the leading diagonal.

The use of the HRZ scheme is reported to give more accurate results than other mass lumping schemes and this is the scheme used here. Its use is justified on the basis of the accuracy of straight tube results obtained with this scheme, but this still leaves open the argument for its use in curved tube problems where the quality of the mass modelling must affect the fluid response to the centripetal accelerations due to tube curvature.

The matrices for viscous, incompressible flow with inertia are then

\[
\begin{bmatrix}
M_1 & 0 & 0 \\
0 & M_2 & 0 \\
0 & 0 & 0
\end{bmatrix}
\begin{bmatrix}
\dot{u} \\
\dot{v} \\
\dot{p}
\end{bmatrix} + 
\begin{bmatrix}
(C_1 \ddot{u} + C_2 \ddot{v}) & 0 & 0 \\
0 & (C_1 \ddot{u} + C_2 \ddot{v}) & 0 \\
0 & 0 & 0
\end{bmatrix}
\begin{bmatrix}
u \\
v \\
p
\end{bmatrix}
\]

\[
\begin{bmatrix}
(2K_{11} + K_{22}) & K_{12} & L_1 \\
K_{21} & (K_{11} + 2K_{22}) & L_2 \\
L_1 & L_2 & 0
\end{bmatrix}
\begin{bmatrix}
u \\
v \\
p
\end{bmatrix} = 
\begin{bmatrix}
R_1^* \\
R_2^* \\
0
\end{bmatrix}
\]

(4.25)

where the masses and convective acceleration terms are as follows

\[
M_1 = \rho \int_A \overline{N}_i^u \, \overline{N}_i^u \, dA
\]

(4.26)

\[
M_2 = \rho \int_A \overline{N}_i^v \, \overline{N}_i^v \, dA
\]

\[
C_1 = \rho \int_A \overline{N}_i^u \frac{\partial \overline{N}_i^u}{\partial r}
\]

(4.27)

\[
C_2 = \rho \int_A \overline{N}_i^v \frac{\partial \overline{N}_i^v}{\partial \theta}
\]
The non-linear convective terms $C_1 \bar{u}$, $C_2 \bar{v}$, can be handled by various methods, Rindt et al (1990), for example, use a Newton-Raphson formulation. The method used here, the so-called Picard or natural iteration, assumes that the velocities in the convective terms are initially zero, as in Stokes’ flow, then substitutes the solution found using this assumption in the convective terms for the next solution and so on, until convergence between successive solutions is obtained within acceptable limits.

4.2.3 Pulsatile Flow

Almost all of the numerical solutions to the pulsatile inlet pressure problem recorded in the literature use some form of finite difference scheme to discretise the pressure cycle with respect to time. The use of finite differences leads to two-point recurrence schemes of which the backward difference scheme has proved to be extremely popular with analysts, Wille (1982), for example. This approach gives unconditional stability of successive solutions, but solution accuracy decreases with increase in the size of the time step. Since the full set of simultaneous equations needs to be solved at each time step the solutions are said to be implicit (Stasa,1985).

Eqtns 4.25 are, in general form

$$M\ddot{u} + (C\ddot{u} + K)u = R \quad (4.28)$$

Expressing the acceleration term in difference form and rearranging gives the simple recurrence scheme

$$\left\{ \frac{M}{\Delta t} + (C\ddot{u} + K) \right\} u_i = \frac{Mu_{i-1}}{\Delta t} + R \quad (4.29)$$

where $u_i$, $u_{i-1}$ denote the velocities at the present and previous time steps, $\ddot{u}$ is the velocity to be iterated during each time step, and $\Delta t$ is the time interval.

4.3 Solid-Fluid Interaction

Many finite element solution schemes for the solid-fluid interaction problem have been proposed. Zienkiewicz (1984) classified the different types of problem and proposed the alternatives of the solution of a directly-coupled set of equations or of an iterative solution with each uncoupled component of the problem solved alternately in a staggered solution. Lucey and Carpenter (1992) proposed a coupled solution based on the use of a boundary element method for the interaction of a fluid with a compliant wall, Zarda et al. (1977) proposed a coupled scheme for investigating the interaction
between blood plasma and the compliant blood cells, using a variational principle in an axisymmetric problem. Perktold et al. (1994) and Lan et al. (1995) propose finite element methods which rely on the direct coupling of the solid and fluid equations while Reuderink et al. (1993), Wnag and Tarbell (1992) and Dutta et al. (1992) all used a staggered, weakly coupled, solution. Park (1988) discussed general solutions of solid-fluid interaction problems and confirmed that difficulties exist with the solution of coupled sets of equations. They also identified time step size problems with uncoupled solutions.

There are, however, distinct advantages in using uncoupled sets of equations, identified by Zienkiewicz (1977) as
(a) standard solution procedures can be used and the difficulties of solving some types of coupled equation systems avoided
(b) established programs exist for the solutions of the separate components of the problems and the introduction of the coupling variable as a forcing function is straightforward
(c) if the coupling is weak, convergence will be fast, whereas slow convergence indicates stronger coupling. The speed of convergence therefore gives an insight into the physical characteristics of the problem
(d) as iteration is generally involved in the solution due to non-linearity in at least one component of the problem, the finite element program is already amenably structured to incorporate the staggered solution iteration.

The particular case for decoupling the fluid and wall motions in arteries is made by Ling and Atabek (1972) on the basis of the assumptions:
(a) the radial motion of the artery wall is primarily dictated by the pressure wave
(b) the longitudinal motion is damped by the perivascular tethering
(c) velocity profiles are developed locally as the pressure wave propagates along the artery.

Lan et al. (1995) state, without further qualification, that there are limitations with the uncoupled approach to the solid-fluid interaction problem with regard to solution convergence and in applications problems with a non-uniform solid and arbitrary boundaries. Their coupled solution is based on the transformation of the current fluid region to the original state, with the solution being carried out on the original mesh.

In the present study the staggered solution procedure is adopted. The fluid pressures are calculated first, the wall displacements are then calculated using the fluid pressure as the forcing function in the wall equations. The finite element mesh geometry for the whole model, i.e., both fluid and solid components, is then updated according to the values of the wall displacements. This is known as the single pass approach.
4.3.1 Equations of Motion for Vessel Wall

The equations of motion for a point in the vessel wall are

\begin{align}
\rho_w \frac{\partial^2 u_w}{\partial t^2} + \mu_w \frac{\partial u_w}{\partial t} &= \frac{\partial \sigma_x}{\partial x} + \frac{\partial \tau_{xr}}{\partial r} \tag{4.30} \\
\rho_w \frac{\partial^2 v_w}{\partial t^2} + \mu_w \frac{\partial v_w}{\partial t} &= \frac{\partial \tau_{xr}}{\partial x} + \frac{\partial \sigma_r}{\partial r} \tag{4.31}
\end{align}

where \( u_w, v_w \) are the wall displacements in the \( x, r \) directions, \( \sigma_x, \sigma_r, \tau_{xr} \) are the wall stresses and \( \rho_w, \mu_w \) are the wall density and wall material damping coefficient, respectively.

Eqtns (4.30) and (4.31) may be written as

\[ A^T \sigma = \rho_w \frac{\partial^2 u}{\partial t^2} + \mu_w \frac{\partial u}{\partial t} \tag{4.32} \]

where \( A = \begin{bmatrix}
\frac{\partial}{\partial x} & 0 \\
0 & \frac{\partial}{\partial r} \\
\frac{\partial}{\partial r} \frac{\partial}{\partial x} & 0
\end{bmatrix} \), \( \sigma = \begin{bmatrix} \sigma_x \\ \sigma_r \\ \tau_{xr} \end{bmatrix} \), \( u = \begin{bmatrix} u_w \\ v_w \end{bmatrix} \)

The constitutive relation for an isotropic, linear elastic material in cylindrical coordinates is

\[ \begin{bmatrix}
\sigma_x \\
\sigma_r \\
\tau_{xr}
\end{bmatrix} = \frac{E(1-\nu)}{(1+\nu)(1-2\nu)} \begin{bmatrix}
1 & \frac{\nu}{1-\nu} & 0 \\
\frac{\nu}{1-\nu} & 1 & 0 \\
0 & 0 & \frac{1-2\nu}{2(1-\nu)}
\end{bmatrix} \begin{bmatrix}
\varepsilon_x \\
\varepsilon_r \\
\gamma_{xr}
\end{bmatrix} \tag{4.33a} \]

where \( E \) is the modulus of elasticity and \( \nu \), Poisson's ratio, or

\[ \{\sigma\} = [D]\{\varepsilon\} \tag{4.33b} \]
The strain-displacement relation is

\[
\begin{pmatrix}
\varepsilon_x \\
\varepsilon_r \\
\gamma_{sr}
\end{pmatrix} =
\begin{bmatrix}
\frac{\partial}{\partial x} & 0 \\
0 & \frac{\partial}{\partial r} \\
\frac{\partial}{\partial r} & \frac{\partial}{\partial x}
\end{bmatrix}
\begin{pmatrix}
u_w \\
v_w
\end{pmatrix}
\]  

(4.34a)

or \( \{\varepsilon\} = [A] \{u\} \)  

(4.34b)

Substituting Eqtns (4.33) and (4.34) in Eqtn (4.32), gives

\[
A^T D A \, u + F = \rho \frac{\partial^2 u}{\partial t^2} + \mu \frac{\partial u}{\partial t}
\]  

(4.35)

Using the same isoparametric shape functions as used in the fluid velocities, Eqtns (4.11) and (4.12) and applying Galerkin's method gives

\[
\begin{array}{l}
\int_{\Omega} N^T \left\{ A^T D A \, u - \rho \frac{\partial^2 u}{\partial t^2} - \mu \frac{\partial u}{\partial t} \right\} \, d\Omega \\
\end{array}
\]  

(4.36)

and the use of Eqtns 4.11 and 4.12 leads to the following equation, typical for all elements

\[
\left[ B^T D B \right] a + \mu w \left[ N^T N \frac{\partial a}{\partial t} + \rho w N^N \frac{\partial^2 a}{\partial t^2} \right] d\Omega^t
\]  

(4.37)

or

\[
Ka + C\dot{a} + M\ddot{a} = 0
\]  

(4.38)

where

\[
K_{ij} = \int_{\Omega} B_i^T D B_j \, d\Omega
\]
\[ M_{ij} = \int_\Omega \rho_w N_i^T N_j \, d\Omega \]

\[ C_y = \int_\Omega \mu_w N_i^T N_j \, d\Omega \]

### 4.3.2 Damping

The determination of \( C \) is difficult in practice as the damping coefficient, \( \mu_w \), is not easily found. Rayleigh damping (Bathe and Wilson, 1976) assumes that the damping matrix is a linear combination of the stiffness and mass matrices

\[
C = \alpha M + \beta K
\]

(4.39)

where \( \alpha \) and \( \beta \) are determined experimentally.

The wall mass matrix may also be based on consistent or lumped masses and here, as in the fluid equations, lumped masses are used.

### 4.3.3 Dynamic Response

Direct integration of Eqtns 4.35 in time is required to give the dynamic response of the vessel wall to the varying internal fluid pressure. There are many ways of carrying out the integration, Lan et al., (1994), for example, use Gresho’s predictor-corrector method (Gresho, 1979), but Newmark’s method (Newmark, 1959) is commonly used and is the method used here.

Using Rayleigh damping, the Newmark recurrence scheme is

\[
\left[ \left( \alpha + \frac{1}{\theta Dt} \right) M + (\beta + \theta \Delta t) K \right] a_1 = \theta \Delta t f_i + (1-\theta) \Delta t f_0
\]

\[
+ \left( \alpha + \frac{1}{\theta Dt} \right) M a_0 + \frac{1}{\theta} M \frac{\partial a_0}{\partial t} + [\beta - (1-\theta) \Delta t] K a_0
\]

(4.40)

where

\[
\frac{\partial a_1}{\partial t} = \frac{1}{\theta \Delta t} (a_1 - a_0) - \frac{1-\theta}{\theta} \frac{\partial a_0}{\partial t}
\]
\[
\frac{\partial^2 a_i}{\partial t^2} = \frac{1}{\theta \Delta t} \left( \frac{\partial a_i}{\partial t} - \frac{\partial a_0}{\partial t} \right) - \frac{1 - \theta}{\theta} \frac{\partial^2 a_0}{\partial t^2}
\]

and the suffices 0 and 1 refer to the present and previous values.

\( \theta \) may be varied to produce a number of different schemes, but unconditionally stable solutions are obtained when \( \theta = 0.5 \).
5 FINITE ELEMENT ANALYSIS OF PULSATILE FLOW IN A CURVED TUBE

5.1 Coordinate System

Three-dimensional fluid flow analyses for arterial bifurcations are typically based on a cartesian coordinate system (Xu and Collins, 1994) in view of the complex geometry and at least one case of curved tube analysis has also been based on this coordinate system (Perktold et al., 1991b). For two-dimensional analyses, where the curvature ratio is small (< 1/20) the centrifugal terms arising from the tube curvature can simply be added to the Navier-Stokes and continuity equations for cylindrical coordinates (Patankar et al., 1974; Jayanti, 1990) (Fig 5.1). Jayanti demonstrates that the straight tube approximation gives solutions with errors, compared with the full curved tube simulation, of up to 21.6% for a curvature ratio of 0.43 and up to 18% for a curvature ratio of 0.2, with a Reynolds number of 800, depending on the position within the bend. The straight tube approximation, however, not only omits the centrifugal effect on the axial velocity, but also the effect of curvature on the relative volumes of sections of fluid at different tube radii, subtended by the same angle. Jayanti used a straight tube approximation for laminar flow as the finite difference scheme he used could not accommodate toroidal coordinates. In addition the outlet boundary condition poses a problem in finite difference models for curved tube flow; a problem that does not arise with the present finite element formulation.

Where the curvature ratio is larger than 1/20, the solution becomes more dependent on the curvature ratio and two-dimensional numerical analyses are almost invariably based on a toroidal coordinate system (Chen and Fan, 1980; Kao, 1992; Hamakiotes and Berger, 1987 and 1990) (Fig 5.2) and this coordinate system has been used here. Derivation of the Navier-Stokes and continuity equations in toroidal coordinates, the natural choice of coordinate system for the curved tube problem, yields all of the appropriate terms (Eqtns (5.8-5.10)). For this reason, almost all of the analytical methods for curved tube flow referred to in Chapter 3 use a toroidal coordinate system.
Fig 5.1 Straight tube approximation for curved tube flow

In the toroidal coordinate system shown in Fig 5.2

\[ \bar{r} = r_0 + r \cos \theta \]  \hspace{1cm} (5.1)

\[ h = \frac{\bar{r}}{r_0} = 1 + b \cos \theta \]  \hspace{1cm} (5.2)

where \( b = \frac{r}{r_0} \)  \hspace{1cm} (5.3)
5.2 Equations of Motion in Toroidal Coordinates

5.2.1 Continuity Equation

\[ \nabla V = \frac{\partial u}{\partial r} + \frac{u}{r} + \frac{1}{r} \frac{\partial v}{\partial \theta} + \frac{1}{h} \frac{\partial w}{\partial z} \left( \frac{\cos \theta - \sin \theta}{r} \right) = 0 \]  

(5.4)

5.2.2 Momentum Equations

The Gartling-Becker formulation requires the momentum equations to be expressed in terms of the stress tensor, \( \tau \), so that the momentum equations in vector notation are
\[
\rho \left( \frac{\partial V}{\partial t} + (V \cdot \nabla)V \right) = -\nabla p + \nabla \cdot \tau + \rho g \tag{5.5}
\]

where \( \nabla \cdot \tau = \mu \left[ \nabla^2 V + \nabla (V \cdot \nabla) \right] = \mu \left[ 2 \nabla (V \cdot V) - \nabla \wedge (\nabla V \wedge V) \right] \tag{5.6} \]

giving
\[
\rho \left( \frac{\partial V}{\partial t} + (V \cdot \nabla)V \right) = -\nabla p + \mu \left[ 2 \nabla (V \cdot V) - \nabla \wedge (\nabla V \wedge V) \right] + \rho g \tag{5.7}
\]

This and other alternative forms of the equations of motion are discussed by Yamada et al. (1975), Hutton (1980) and Gresho (1991). Eqtns 5.4 and 5.7 are given in here in dimensional form. Invariably in mathematical analyses the continuity and Navier-Stokes equations are expressed in the non-dimensional forms

\[
\frac{\partial u^*}{\partial r^*} + \frac{u^*}{r^*} \frac{1}{r^*} \frac{\partial v^*}{\partial \theta^*} + \frac{1}{h} \frac{\partial w^*}{\partial z^*} + \frac{(u^* \cos \theta^* - v^* \sin \theta)}{r^*} = 0 \tag{5.8}
\]

\[
\frac{\partial V^*}{\partial t^*} + (V^* \cdot \nabla)V^* = -\nabla p^* + \frac{1}{\text{Re}} \left[ 2 \nabla (V^* \cdot V^*) - \nabla \wedge (\nabla V^* \wedge V^*) \right] + \rho g \tag{5.9}
\]

where

\[
V^* = \left\{ \frac{u}{W_o}, \frac{v}{W_o}, \frac{w}{W_o} \right\}, \quad p^* = \frac{p}{\rho W_o^2}, \quad r^* = \frac{r}{a}, \quad z^* = \frac{z}{a}, \quad t^* = \frac{t W_o}{a},
\]

\[
\text{Re} = \frac{W_o \frac{a \rho}{\mu}}, \quad \text{and} \quad W_o \quad \text{and} \quad a \quad \text{are the average axial velocity and tube radius, respectively. The ratio } a/W_o \text{ is the residence time and in periodic flows of frequency } \omega, \text{ the Strouhal number, } \omega a/W_o \text{ may also be incorporated into these equations (Rindt et al., 1991; Hamakiotes and Berger, 1990). In numerical solutions the use of non-dimensionalised equations is not consistently used and Chen and Fan (1988) and Wille (1980), for example, use dimensional equations in their finite element analyses. There appears to be no advantage in using the non-dimensional forms of the equations and the dimensional forms are therefore used in the present study. Evaluating Eqtn (5.7), yields the momentum equations, the Navier-Stokes equations, in the } r, \theta \text{ and } z \text{ directions, as}
\]


\[ \rho \left[ \frac{\partial u}{\partial t} + u \frac{\partial u}{\partial r} + \frac{v}{r} \frac{\partial u}{\partial \theta} + \frac{w}{\rho} \frac{\partial u}{\partial z} + \frac{w^2}{\rho} \cos \theta \right] = \frac{\partial p}{\partial r} + \mu \left[ 2 \frac{\partial^2 u}{\partial r^2} + \frac{1}{r^2} \frac{\partial^2 u}{\partial \theta^2} + 1 \frac{\partial^2 u}{\partial h^2} \frac{\partial z}{\partial z} + \frac{1}{h^2} \frac{\partial z}{\partial z} + \frac{1}{h^2} \frac{\partial v}{\partial \theta} + \frac{1}{h^2} \frac{\partial w}{\partial z} + \frac{2}{h^2} \frac{u}{r^2} + \frac{2}{h^2} \frac{u}{r} \frac{\partial u}{\partial r} + \frac{3}{h^2} \frac{\partial u}{\partial \theta} \right] \]

\[ \rho \left[ \frac{\partial v}{\partial t} + u \frac{\partial v}{\partial r} + \frac{v}{r} \frac{\partial v}{\partial \theta} + \frac{w}{\rho} \frac{\partial v}{\partial z} + uv + \frac{w^2}{\rho} (\sin \theta) \right] = \frac{1}{r} \frac{\partial p}{\partial \theta} + \mu \left[ \frac{\partial^2 v}{\partial r^2} + \frac{2}{r^2} \frac{\partial^2 v}{\partial \theta^2} + \frac{1}{r^2} \frac{\partial^2 v}{\partial z^2} + \frac{1}{r^2} \frac{\partial^2 z}{\partial z^2} + \frac{1}{r^2} \frac{\partial^2 u}{\partial \theta \partial \theta} + \frac{1}{r^2} \frac{\partial^2 w}{\partial \theta \partial z} + \frac{3}{r^2} \frac{\partial u}{\partial \theta} + \frac{\partial u}{\partial \theta} \right] \]

\[ \rho \left[ \frac{\partial w}{\partial t} + u \frac{\partial w}{\partial r} + \frac{v}{r} \frac{\partial w}{\partial \theta} + \frac{w}{\rho} \frac{\partial w}{\partial z} + w (u \cos \theta - \nu \sin \theta) \right] = \frac{1}{h^2} \frac{\partial p}{\partial z} + \mu \left[ \frac{\partial^3 w}{\partial r^3} + \frac{1}{r^2} \frac{\partial^3 w}{\partial \theta^2} + \frac{2}{r^2} \frac{\partial^3 w}{\partial z^2} + \frac{1}{r^2} \frac{\partial^3 u}{\partial \theta^2} + \frac{1}{r^2} \frac{\partial^3 w}{\partial \theta \partial z} + \frac{1}{r^2} \frac{\partial^3 v}{\partial \theta \partial z} + \frac{3}{r^2} \frac{\partial u}{\partial \theta} + \frac{w}{r^2} \right] \]

\[ \frac{\cos \theta}{r} \left[ \frac{3}{r} \frac{\partial u}{\partial \theta} + \frac{w}{r} \right] + \left( \frac{\sin \theta}{r} \right) \left[ \frac{3}{r^2} \frac{\partial v}{\partial z} - \frac{1}{r^2} \frac{\partial v}{\partial \theta} \right] \]

(5.10a)

(5.10b)

(5.10c)
The derivation of Eqtns (5.10) is given in Appendix B.

5.3 Fully Developed Flow

Since fully developed flow requires the flow at each point along the tube axis to be the same, the fully developed solution can be obtained from a two-dimensional model of the tube cross-section. The fully three-dimensional continuity and Navier-Stokes equations (Eqtns (5.4 and 5.10)) then need to be modified in order to eliminate all the z-direction gradient terms, with the exception of the pressure gradient, leading to the following equations for fully developed flow.

\[
\nabla \cdot \mathbf{V} = \frac{\partial u}{\partial r} + \frac{u}{r} \frac{\partial v}{\partial \theta} + \frac{1}{r} \frac{\partial v}{\partial \theta} + \frac{(u \cos \theta - v \sin \theta)}{r} = 0
\]  
(5.11)

\[
\rho \left[ \frac{\partial v}{\partial t} + u \frac{\partial v}{\partial r} + v \frac{\partial v}{\partial \theta} + \frac{\nu}{\partial r} + \frac{\nu}{\partial \theta} \right] = \frac{1}{r} \frac{\partial p}{\partial \theta} + \mu \left[ \frac{\partial^2 v}{\partial r^2} + \frac{2}{r^2} \frac{\partial^2 v}{\partial \theta^2} + \frac{1}{r} \frac{\partial^2 u}{\partial r \partial \theta} + \frac{1}{r} \frac{\partial v}{\partial r} + \frac{3}{r^2} \frac{\partial u}{\partial \theta} \right] 
\]

\[
+ \left( \frac{\cos \theta}{r} \right) \left( \frac{1}{r} \frac{\partial u}{\partial \theta} - \frac{\nu}{\partial \theta} + \frac{\partial u}{\partial \theta} \right) + \left( \frac{\sin \theta}{r} \right) \left( -\frac{2}{r} \frac{\partial v}{\partial \theta} + \frac{2}{r} \frac{\partial v}{\partial \theta} \right) + 2 \frac{\sin \theta (u \cos \theta - v \sin \theta)}{r^2}
\]

(5.12)

\[
\rho \left[ \frac{\partial u}{\partial t} + u \frac{\partial u}{\partial r} + v \frac{\partial u}{\partial \theta} - \frac{v^2}{r} - \frac{w^2 \cos \theta}{r} \right] = \frac{\partial p}{\partial r} + \mu \left[ \frac{\partial^2 u}{\partial r^2} + \frac{1}{r^2} \frac{\partial^2 u}{\partial \theta^2} + \frac{1}{r} \frac{\partial^2 v}{\partial r \partial \theta} - \frac{2}{r} \frac{\partial u}{\partial \theta} + \frac{2}{r} \frac{\partial u}{\partial \theta} + \frac{3}{r^2} \frac{\partial v}{\partial \theta} \right] 
\]

\[
+ \left( \frac{\cos \theta}{r} \right) \left( 2 \frac{\partial u}{\partial r} \right) + \left( \frac{\sin \theta}{r} \right) \left( -\frac{1}{r} \frac{\partial u}{\partial \theta} + \frac{v}{\partial \theta} - \frac{\partial v}{\partial \theta} \right) - 2 \frac{\cos \theta (u \cos \theta - v \sin \theta)}{r^2}
\]

(5.13)
\[
\begin{align*}
& r \left[ \frac{\partial w}{\partial t} + \frac{u}{r} \frac{\partial w}{\partial \theta} + \frac{v}{r} \frac{\partial w}{\partial \theta} + \frac{w}{r} \left( \frac{\cos \theta}{r} - \frac{\sin \theta}{r^2} \right) \right] \\
= & -\frac{1}{h} \frac{\partial p}{\partial z} + m \left[ \frac{\partial^3 w}{\partial r^2} + \frac{1}{r} \frac{\partial w}{\partial r} + \frac{1}{r^2} \frac{\partial^2 w}{\partial \theta^2} \right] \left[ \frac{\cos \theta}{r} - \frac{1}{r^2} \frac{\partial \theta}{\partial r} \right] - \left( \frac{\sin \theta}{r} \right) \left( \frac{1}{r} \frac{\partial \theta}{\partial r} \right)
\end{align*}
\]

(5.14)

5.3.1 Finite element mesh

Transformation of Eqtns (5.12-5.14) using the Galerkin method and incorporation of the appropriate boundary conditions leads to the solution for the fully developed flow.

The finite element model (Fig 5.3) makes use of the mixed interpolation, isoparametric element in its rectangular form (Fig 4.1). Normally triangular elements would be used at the centre of the mesh to avoid degrading rectangular elements by the need to reduce the length of the inner side to zero. Although the physical problem is well-behaved at the centre, the triangular elements available with the NAGFE library (Greenhough and Robinson, 1981) have the Gauss points at the apices and their use would lead to mathematical singularity as \( r \) approaches zero.

While finite difference formulations can produce solutions which are adversely influenced by the form of the mesh, particularly whether 'structured' or 'unstructured', this form of mesh dependence is not generally believed to affect finite element formulations as severely. Structured meshes are those based on the coordinate system in use, so that the mesh points can be located by integers, e.g., in a cylindrical system, by radial \( i \), radius \( j \), and therefore easier to generate, while unstructured meshes are based on real numbers (Hirsch, 1988a).

A problem does arise, however, if a structured mesh based on a toroidal or cylindrical coordinate system is used, as in the present problem, because a singularity occurs at the centre of the finite element mesh due to the Navier-Stokes and continuity terms involving \( 1/r \). If elements can be used where the Gauss points are located away from the boundary of the element the problem can be avoided. The singularity can also be avoided by providing a special mesh at the tube centre, an approach used by Hamakiotes and Berger (1989) in their finite difference analysis, or, as proposed by Kao (1992) and Chen and Fan (1980), by simply leaving a very small area around the tube centre empty of elements. The alternatives of either using triangular elements with internal Gauss points or of setting up an unstructured mesh in cartesian coordinates are available (Hirsch, 1988a; Lan et al., 1994). The first of these,
however, requires the programming of special, untried elements and unavailable in the NAGFE library, while the latter involves a pre- and post-processing penalty.

It is argued here that the singularity can be avoided without adverse effects on the solution by simply incorporating a small radius at the centre of the model, effectively creating a small void in the fluid at this point, but with the advantage in generating the mesh of allowing the use of rectangular elements throughout. The question then arises as to what extent the solution is affected by the fluid void at the tube centre. Clearly, if the boundary conditions around the central void are not chosen carefully and, for example, a 'no-slip' condition is imposed, the resulting wake occurring in the transverse flow would extend an order of magnitude downstream and be 'convected' around the tube, with adverse effects on the velocity prediction. With boundary conditions around the central radius specified to allow free slip, however, the disturbance due to the void is of the order of the void diameter and hence the effect is probably no worse than that due to the relatively high aspect ratios produced in the elements near the centre.

Regarding the effect of the fluid void on the axial velocity distribution in the tube, the imposition of the 'no-slip' condition in this direction would change the problem entirely and the effect would be so obvious that there could be no question of accepting the results on face value. Imposition of no constraint on axial velocity around the void would simply lead to no solution within the void.

Whether ill-conditioning of the equations could result from the presence of the fluid void, or other adverse effects on the solution arise due to cross coupling between the transverse and axial effects of the presence of the fluid void, is not clear. An assessment of the error introduced by the fluid void would be instructive and could be easily carried out by comparing the results of the same flow problem from models with structured (polar) and unstructured (cartesian) finite element meshes for the tube cross-section.

5.3.2 Boundary conditions

In incompressible, viscous flow problems the boundary conditions are of two types (a) specified velocity components and (b) specified surface tractions. In the Gartling-Becker formulation the stress tensor appears explicitly as a boundary integral as a result of the integration by parts. In practical cases, at inflow or outflow boundaries, the viscous term vanishes, leaving only the pressure term, so that specifying the surface traction is equivalent to specifying the pressure. In the general situation, the distinction between pressure and viscous contributions to the boundary condition is not made. Furthermore, with the use of the total surface traction, the absolute pressure
level for the problem under consideration cannot be determined (Gartling, 1974). Gresho (1993) and Conca (1993) discuss the problem of defining boundary conditions for the solution of Navier-Stokes equations for incompressible and other flows and it is clear that potentially there can be difficulties with some formulations.

For the formulation used here, however, no problems are indicated in the literature and the velocity boundary conditions required are simply those due to the symmetry of the model, the 'no-slip' condition at the inner surface of the tube wall, and the 'no-slip' on and 'no-flow' through the circumference of the semi-circle at the centre.

5.3.3 Fluid-wall Interaction

The fluid-wall interaction problem requires the pressure distribution in the cross-section for the calculation of the wall loading and the absence of absolute pressure values in the finite element solution raises a problem. Patankar and Spalding (1972) and Selikens (1990) discuss the inconsistency in the Navier-Stokes equations (Eqtns (5.12-5.14)) where the pressure component in the z-direction is the single, average pressure driving the axial flow, while the pressure component in the cross-section, balancing the centrifugal forces, varies with r and \( \theta \). In order to determine the wall loading the average pressure in the z-direction needs to be modified to take account of the pressure component in the cross-section. Specifying, as a boundary condition, a single pressure value at an appropriate point in the model leads to a pressure distribution across the section relative to this value. The wall loading is then determined by adding the z and r, \( \theta \) pressure components on the basis of the distribution across the section. In the current problem the pressure is assumed to be zero at the node, \( n \), at the inner wall of the tube on the horizontal centreline, where the lowest pressure occurs.

The boundary conditions are then:

\[
\begin{align*}
    u &= v = w = 0 \quad \text{at the wall} \\
    v &= 0 \quad \text{on the horizontal centreline} \\
    u &= 0 \quad \text{on the central semi-circle} \\
    p_n &= 0 \quad \text{at a specified node}
\end{align*}
\]
The model (Fig 5.3) incorporates wall elements of the same type as the fluid element, but formulated as solid, stress type elements in Cartesian coordinates. The pressure loading is calculated from the fluid pressures, as stated above, and the boundary conditions for free expansion of the tube are those required merely to enforce symmetry. A description of the program for fully developed flow, PFECTX, is given in Appendix C.

5.4 Entry Flow
The determination of entry flow is an important requirement in any study of the aorta as the entry length for mean flow in the aorta is much longer than the aorta itself. Patankar et al., (1974) produced an entry flow analysis for curved tubes based on predicting and correcting flows in successive downstream cross-sections by a finite difference marching technique. They do not appear to include axial viscous effects in their analysis. Their steady flow analysis used cylindrical coordinates for radius of curvature/tube radius ratios, $R/a \geq 15$. While their use of a CDC 6600 computer produced solutions in minutes, the corresponding analysis carried out on an HP 9000/700 series computer would be prohibitive in terms of processing time. The current study therefore requires a complete entry flow solution for each iteration and this can only be achieved by modelling a longitudinal section of the tube.

Fig 5.3 Finite element model of tube cross-section

46
5.4.1 Semi-analytical Solution

The situation where a structure or component is axisymmetric, while the loading or material properties are non-axisymmetric, occurs frequently in problems in stress analysis, dynamics and heat transfer and in many other areas, but more rarely in fluid flow. Since the problem domain is axisymmetric, although three-dimensional, it can be modelled two-dimensionally and a method of analysis has been developed to exploit the domain symmetry (Wilson, 1965). A Fourier series representation of the variables in the circumferential direction is used together with a standard finite element approximation for the other two directions in what is usually referred to as a semi-analytical analysis. In many problems the use of this method can make the use of a full three-dimensional analysis, with its high computing cost, unnecessary.

If the loading and variables can be expressed in the form of a truncated Fourier series,

\[ u = \sum_{i=1}^{n} (a_i \sin i\theta + b_i \cos i\theta) \]

the corresponding shape functions are then given by

\[ u = \sum_{i=1}^{n} \left[ \overline{N}_k \sin i\theta + \overline{N}_k \cos i\theta \right] \{u_k \} \]

where

\[ \overline{N}_k = N_k(a_i), \quad \overline{N}_k = N_k(b_i) \]

and are the shape functions corresponding to the amplitudes \( a_i, b_i \) of the Fourier terms.

Application of the Galerkin, or any other method, to transform the governing equations into finite element form leads to the elimination of the resulting orthogonal terms, while other integrals involving \( \theta \) may be determined explicitly, leaving only terms in \( i, r \) and \( z \). The problem can then be treated as two-dimensional and modelled as axisymmetric in the normal way, requiring only the modelling of the radial plane and its rotation about the axis to form the three-dimensional fluid domain. The solution for each harmonic is the amplitude of the variable and the complete solution can be found by subsequent expansion. All the degrees of freedom are retained and although separate, uncoupled analyses are required for each harmonic, \( i \), the practicality of the
method relies on the fact that in suitable problems only three or four harmonics at most are required to obtain a satisfactory solution. In some cases only one harmonic is required (Smith, 1978). However, as the solution is given by the sum of the results for each harmonic, a problem requiring a large number of harmonics proves to be no more economical than the full three-dimensional analysis.

Many examples of this method of analysis appear in the literature, particularly for problems in shells (Wunderlich et al., 1989), dynamics (Noor and Peters, 1988) and heat transfer (Crose, 1972), among many others. Zienkiewicz and Taylor (1991) state that any problem governed by minimisation of a quadratic functional or by linear differential equations is amenable to solution by this technique, but it is shown that non-linear problems can also be solved in this way (Winnicki and Zienkiewicz, 1979, Sedaghat and Herrmann, 1983, Wunderlich et al. 1989).

Although there appears to be no reports of a semi-analytical analysis being carried out for a fluid flow problem there appears to be no prima facie reason why the method should not be applicable to fluid flow or indeed to any other problems in engineering science. In the current problem the non-linearity due to the convective acceleration terms in the Navier-Stokes equations requires a complete solution to be carried out at every iteration. The summing of solutions for each harmonic would probably not give an acceptable solution with the level of non-linearity involved and the only way forward, therefore, is to use only a single term for each of the variables. If an acceptable solution can be achieved in this way, there is not only a considerable saving in computer run time to be made, but there is the additional advantage of being able to model the tube wall simply as an extension of the radial plane. There is little to be lost in terms of accuracy in using this simple model of the wall since in practice axial and circumferential bending stresses will be insignificant compared with the direct, circumferential, stresses in an artery wall.

The technique described above requires an axisymmetric fluid domain with non-axisymmetric loading. The curved tube is not truly axisymmetric as the distance between any two cross-sections normal to the tube axis varies in the circumferential direction. It is proposed here, however, that this variation can be accounted for in the integration of the finite element terms and that there is negligible error in practice in adopting such an approach.

5.4.2 Choice of Fourier terms
The choice of terms in the Fourier series representation of the variables in the tangential direction is straightforward in problems in stress analysis, for example, where the relationship between force and material response is usually explicit. The choice of terms here, however, is less straightforward and requires some study of the
effects of various forms on the finite element formulation before a satisfactory form can be obtained.

The complete solution for the velocities, \( u, v \) and \( w \) and the pressure, \( p \), is given by the Fourier series

\[
\begin{align*}
\mathbf{u}(r,z,\theta) &= \mathbf{u}_{10}(r,z) + \sum_{m=1}^{\infty} u_{lm}(r,z) \cos m\theta + \sum_{n=1}^{\infty} \hat{u}_{ln}(r,z) \sin n\theta \\
\mathbf{v}(r,z,\theta) &= \mathbf{v}_{10}(r,z) + \sum_{m=1}^{\infty} v_{lm}(r,z) \sin m\theta + \sum_{n=1}^{\infty} \hat{v}_{ln}(r,z) \cos n\theta \\
\mathbf{w}(r,z,\theta) &= \mathbf{w}_{10}(r,z) + \sum_{m=1}^{\infty} w_{lm}(r,z) \cos m\theta + \sum_{n=1}^{\infty} \hat{w}_{ln}(r,z) \sin n\theta \\
\mathbf{p}(r,z,\theta) &= \mathbf{p}_{10}(r,z) + \sum_{m=1}^{\infty} p_{lm}(r,z) \cos m\theta + \sum_{n=1}^{\infty} \hat{p}_{ln}(r,z) \sin n\theta
\end{align*}
\]

(5.18)

Balachandran (1972) shows that the terms in \( m, n \) higher than 1 produce negligible improvement in the solution and Eqtns (5.18) may, therefore, conveniently be reduced to

\[
\begin{align*}
u &= \mathbf{u}_{10} + u_{11} \cos \theta \\
\mathbf{v} &= \mathbf{v}_{10} + v_{11} \sin \theta \\
\mathbf{w} &= \mathbf{w}_{10} + w_{11} \cos \theta \\
\mathbf{p} &= \mathbf{p}_{10} + p_{11} \cos \theta
\end{align*}
\]

(5.19)

For computational purposes the flow in curved tubes is usually considered to have an axial, straight tube component and a component due to tube curvature, and the terms on the right hand sides of Eqtns (5.19) are the straight and curved tube components respectively. The straight tube flow is said to be \textit{perturbed} by the curved tube flow. Since fully developed flow in the tube can be analysed using the cross-section model, the semi-analytical technique can be restricted to entry flow only. The semi-analytical technique requires that the variables take the same \textit{form} as the applied loading. In flow through a tube with a curved axis, the fluid experiences centripetal accelerations due to the curvature and the corresponding centrifugal forces produce the secondary flow in the fluid in cross-sectional planes. Here it is argued
that, as an approximation, the fluid velocities in the radial and tangential directions are distributed around the tube in the same way as the corresponding components of the centrifugal forces.

\[ -\frac{mw^2 \sin \theta}{\bar{r}}, \quad \frac{mw^2 \cos \theta}{\bar{r}} \]

Fig 5.4 Centrifugal loading

The components of centrifugal force in the radial and tangential direction are

\[ \frac{mw^2 \cos \theta}{\bar{r}}, \quad \frac{-mw^2 \sin \theta}{\bar{r}} \] (Fig 5.4)

These terms arise in the derivation of the equations of motion in toroidal coordinates (Eqtns (5.10)), together with the term

\[ \frac{mw(u \cos \theta - v \sin \theta)}{\bar{r}} \]

which represents the force in the axial direction due to the rotation of the centrifugal components of the radial and tangential velocities about the centre of curvature. The tangential velocity and the curved tube component of the radial velocity must therefore take the form

\[ -\frac{\sin \theta}{\bar{r}}, \quad \frac{\cos \theta}{\bar{r}}, \] respectively.
Recognising that from symmetry a straight tube component of tangential velocity does not exist

\[ v_{10} = 0 \]

then Eqtms (5.19) may be written as

\[
\begin{align*}
    u &= u_{10} + u_{11} \cos \theta \\
    v &= \frac{v_{11} \sin \theta}{r_0} \\
    w &= w_{10} + w_{11} \cos \theta \\
    p &= p_{10} + p_{11} \cos \theta
\end{align*}
\]

(5.20)

The relative magnitudes of the straight and curved tube components of the velocities and pressure are not known \textit{a priori}. Preliminary analyses were carried out on the assumption of a simple relationship between the straight and curved tube components such as

\[
\begin{align*}
    u &= u_{10} (1 + b' \cos \theta) \\
    v &= -b' v_{11} \sin \theta \\
    w &= w_{10} (1 + b' \cos \theta) \\
    p &= p_{10} (1 + b' \cos \theta)
\end{align*}
\]

(5.21)

where \( b' \), for example, is a function of \( a \) and \( r_O \), or \( r \) and \( r_O \), i.e., the curved tube components of velocity were assumed to depend on the curvature ratio, or on the position within the tube. This assumption produced inconsistent solutions. Eqtms 5.20 imply different values for the straight component and amplitude of the curved
component (e.g. \( u_{10} \) and \( u_{11} \)). Specifying a simple relationship between them could result in over constraining the solution.

The better approach is that, assuming that separate solutions can be found for the straight and curved tube components, these solutions can be summed to give the total curved tube velocities. The straight tube solution is simply that of a tube with very large radius of curvature and axial pressure gradient, while the curved tube solution is that of a tube with the prescribed radius of curvature and zero axial pressure gradient, but with centrifugal force terms based on the axial velocity distribution of the straight tube solution.

In practice the procedure is straightforward. The straight tube analysis is carried out first, the flow driven by the axial pressure gradient. The resulting axial and radial velocities are then used in calculating the centrifugal terms to drive the curved tube flow. The straight tube solutions are iterated until convergence is obtained, the centrifugal terms are then expressed in terms of the resulting axial velocities and the curved tube solution is then iterated until convergence is obtained. Two steps are then required for each time step.

### 5.5 Finite Element Equations

The shape functions corresponding to the curved tube components of the variables (Eqtns (5.20)) are then

\[
N^u = \overline{N}^u (\cos \theta) \\
N^v = \overline{N}^v (\sin \theta) \\
N^w = \overline{N}^w (\cos \theta) \\
N^p = \overline{N}^p (\cos \theta)
\]  

(5.23)

where \( \overline{N}^u, \overline{N}^v, \overline{N}^w, \overline{N}^p \) are the shape functions for the finite element shown in Fig 4.1.

Using the Galerkin formulation, a typical term of the equations of motion, Eqtns (5.10) is expressed in finite element form as follows. Taking the third term of Eqtn (5.10b), for example,
Substituting for the curved tube component shape function
\[ u = N_i u_i = \overline{N}_i (\cos \theta) u_i \text{, and differentiating, gives} \]
\[ \frac{v}{r} \frac{\partial v}{\partial \theta} = \frac{\overline{N}^*}{r} \overline{N}^* (\sin \theta) \frac{\partial \overline{N}^*}{\partial \theta} (\sin \theta) v_i = \frac{\overline{N}^*}{r} \overline{N}^* \overline{N}^* (\sin^2 \theta \cos \theta) v_i \]

Given that the width of the element varies with \( h \) (Fig 5.3)
\[ \int_0^1 (...) d\Omega = \int_{\xi}^{1} \int_{\eta}^{1} (...) hr d\theta |J| d\xi d\eta \]

Integrating:
\[ \int_{\xi}^{1} \int_{\eta}^{1} (...) \left( \frac{\overline{N}^*}{r} \overline{N}^* \overline{N}^* (\sin^2 \theta \cos \theta) \right) hr d\theta |J| d\xi d\eta v_i \]

and substituting for \( h = 1.0 + b \cos \theta \) and \( dA = |J| d\xi d\eta \)

the term becomes \( \pi \int_{\xi}^{1} 0.25b \frac{\overline{N}^*}{r} \overline{N}^* \overline{N}^* r dA \)

(5.24)

\[ h = \frac{\bar{r}}{r_0} \]
\[ d\bar{z} = dz \left( \frac{\bar{r}}{r_0} \right) = dz. h \]

Fig 5.5 Variation of element width with angle \( \theta \)
It is convenient to integrate the terms in $\theta$ beforehand and incorporate the resulting factors into the program coding. Integration with respect to the normalised coordinates $\xi, \eta$ is carried out using Gauss quadrature in the finite element program, in the normal way. The same process is applied to all the terms of the Navier-Stokes equations, Eqtns (5.10), and the continuity equation, Eqtn (5.4). All the non-zero terms are listed below (Eqtns (5.31-5.78)), together with the corresponding Galerkin integral and finite element matrix notation. The factor $\pi$ is common throughout and is therefore omitted. For convenience, both straight and curved tube components of the finite element matrix terms are included below and in the coding. Straight or curved tube analyses are selected either by setting $\alpha = 1$ and $\beta = 0$, for straight tube flow, or by setting $\alpha = 0$ and $\beta = 1$, for curved tube flow.

The finite element equations in matrix form are

$$[K][U] = [F]$$

(5.27)

where the system characteristic matrix, $K$, is

$$
\begin{bmatrix}
K_{11} + C_{11} \tilde{u}_i + \frac{M}{\Delta t} & K_{12} & K_{13} & L_1^p \\
K_{21} & K_{22} + C_{22} \tilde{u}_i + \frac{M}{\Delta t} & K_{23} & L_2^p \\
K_{31} & K_{11} & K_{33} + C_{33} \tilde{u}_i + \frac{M}{\Delta t} & L_3^p \\
L_1 & L_2 & L_3 & 0
\end{bmatrix}
$$

(5.28)

and

$$C_{ii} \tilde{u}_i = C_{i1}(\tilde{u}) + C_{i2}(\tilde{v}) + C_{i3}(\tilde{w}), \text{ etc.}$$

and where the vector of unknown velocity and pressure amplitudes, $U$, and the vector of forcing terms, $F$, are given by
\[
U = \begin{bmatrix}
    u_i \\
    v_i \\
    w_i \\
    F
\end{bmatrix} \quad (5.29)
\]
and
\[
F = \begin{bmatrix}
    \frac{M}{\Delta t} u_{i-1} \\
    \frac{M}{\Delta t} v_{i-1} \\
    R^w + \frac{M}{\Delta t} w_{i-1} \\
    0
\end{bmatrix} \quad (5.30)
\]

Eqtns (5.28) are assembled, the boundary conditions appropriate to the particular problem are inserted, and the equations are solved using the frontal technique in the FORTRAN program PFECTL. A full description of the program is given in Appendix C.

### 5.5.1 Radial direction terms

\[
u \frac{\partial u}{\partial r} + C_{11} = \int_A (2.0 \alpha + 0.75 \beta \beta) \, \bar{w} \, N^u \bar{N}^u \frac{\partial \bar{N}^u}{\partial r} \, r \, dA \quad (5.31)
\]

\[
u \frac{\partial u}{r \, \partial \theta} + C_{12} = \int_A \frac{\beta}{r} \, \bar{v} \, \bar{N}^v \bar{N}^u \, r \, dA \quad (5.32)
\]

\[-\frac{\nu^2}{r} + K_{12} = -\int_A \frac{0.25 \beta}{r} \, \bar{v} \, N^v N^u N^v \, r \, dA \quad (5.33)
\]

\[
\frac{w}{h} \frac{\partial u}{\partial z} + C_{13} = \int_A (2.0 \alpha) \bar{w} \, \bar{N}^w \bar{N}^u \frac{\partial \bar{N}^u}{\partial z} \, r \, dA \quad (5.34)
\]

\[-\frac{w^2 \cos \theta}{F} + K_{13} = -\int_A \frac{0.75 \beta}{r_0} \, \bar{w} \, \bar{N}^w \bar{N}^u \bar{N}^w \, r \, dA \quad (5.35)
\]

\[-2 \frac{\partial^2 u}{\partial r^2} + K_{11} = \int_A 2.0(2.0 \alpha + \beta) \frac{\partial \bar{N}^u}{\partial r} \frac{\partial \bar{N}^u}{\partial r} \, r \, dA \quad (5.36)
\]

\[-\frac{1}{r^2} \frac{\partial^2 u}{\partial \theta^2} + K_{11} = \int_A \frac{\beta}{r^2} \bar{N}^u \bar{N}^u \, r \, dA \quad (5.37)
\]
\[ \frac{1}{h^2} \frac{\partial^2 u}{\partial z^2} \quad K_{11} = |A| (2.0 \alpha + \beta) \frac{\partial N_u \partial N_v}{\partial z \partial z} r dA \quad (5.38) \]

\[ \frac{1}{r} \frac{\partial^2 v}{\partial r \partial \theta} \quad K_{12} = |A| \frac{\beta}{r} \frac{\partial N_u}{\partial r} r dA \quad (5.39) \]

\[ \frac{1}{r} \frac{\partial^2 w}{\partial r \partial \theta} \quad K_{13} = |A| \frac{\beta}{r} \frac{\partial N_w}{\partial z} r dA \quad (5.40) \]

\[ \frac{3}{r^2} \left( \frac{\partial^2 v}{\partial \theta^2} \right) \quad K_{12} = |A| 3 \frac{\beta}{r^2} N_u N_v r dA \quad (5.41) \]

\[ \frac{2}{r^2} \quad K_{11} = |A| 2.0 \frac{\partial (2.0 \alpha + \beta)}{r^2} N_u N_v \frac{\partial N_u}{\partial r} r dA \quad (5.42) \]

\[ \frac{2}{r} \frac{\partial u}{\partial r} \quad K_{11} = |A| 2.0 \frac{(2.0 \alpha + \beta)}{r} N_u \frac{\partial N_u}{\partial r} r dA \quad (5.43) \]

5.5.2 Tangential direction terms

\[ u \frac{\partial v}{\partial r} \quad C_{21} = |A| 0.25 b \beta \bar{u} N_u N_v \frac{\partial N_v}{\partial r} r dA \quad (5.44) \]

\[ \frac{v \partial v}{r \partial \theta} \quad C_{22} = |A| 0.25 b \beta \bar{v} N_v N_v N_v \frac{\partial N_v}{\partial \theta} r dA \quad (5.45) \]

\[ \frac{w \partial v}{h \partial z} \quad C_{23} = |A| 0.75 b \beta \bar{w} N_w N_v \frac{\partial N_v}{\partial z} r dA \quad (5.46) \]

\[ \frac{u v}{r} \quad K_{22} = |A| 0.25 b \beta \bar{u} N_u N_v \bar{N}_v \frac{\partial N_v}{\partial \theta} r dA \quad (5.47) \]

\[ \frac{w^2 \sin \theta}{r} \quad K_{23} = |A| 0.25 b \beta \bar{w} N_u N_u N_u \frac{\partial N_u}{\partial z} r dA \quad (5.48) \]

\[ \frac{2}{r^2} \frac{\partial^2 v}{\partial \theta^2} \quad K_{22} = |A| 2.0 b \beta \bar{v} N_v N_v \frac{\partial N_v}{\partial \theta} r dA \quad (5.49) \]

\[ \frac{1}{h^2} \frac{\partial^2 v}{\partial z^2} \quad K_{22} = |A| 2.0 b \beta \frac{\partial N_v}{\partial z} \frac{\partial N_v}{\partial z} r dA \quad (5.50) \]
\[
\frac{\partial^2 v}{\partial r^2} = K_{22} = \int_{A} (2.0\alpha + \beta) \frac{\partial \bar{N}_v}{\partial r} \frac{\partial \bar{N}_v}{\partial r} r \, dA \\
(5.51)
\]

\[
\frac{1}{r} \frac{\partial v}{\partial r} = K_{22} = -\int_{A} \beta \bar{N}_v \frac{\partial \bar{N}_v}{\partial r} r \, dA \\
(5.52)
\]

\[
\frac{v}{r^2} = K_{22} = \int_{A} \beta \bar{N}_v \bar{N}_v r \, dA \\
(5.53)
\]

\[
\frac{3}{r^2} \frac{\partial u}{\partial \theta} = K_{21} = \int_{A} \frac{3.0\beta}{r^2} \bar{N}_v \bar{N}_u \, dA \\
(5.54)
\]

\[
-\frac{1}{r} \frac{\partial^2 u}{\partial r \partial \theta} = K_{21} = \int_{A} (2.0\alpha + \beta) \bar{N}_v \frac{\partial \bar{N}_u}{\partial r} r \, dA \\
(5.55)
\]

\[
\frac{1}{r h} \frac{\partial^2 w}{\partial z \partial \theta} = K_{23} = \int_{A} \beta \bar{N}_v \frac{\partial \bar{N}_w}{\partial z} r \, dA \\
(5.56)
\]

### 5.5.3 Axial direction terms

\[
u \frac{\partial w}{\partial r} = C_{31} = \int_{A} (2.0\alpha + 0.75\beta) \bar{u} \bar{N}_u \bar{N}_w \frac{\partial \bar{N}_w}{\partial r} r \, dA \\
(5.57)
\]

\[
\frac{\nu}{r} \frac{\partial w}{\partial \theta} = C_{32} = -\int_{A} \frac{0.25\beta}{r} \bar{v} \bar{N}_v \bar{N}_w \bar{N}_w \, r \, dA \\
(5.58)
\]

\[
\frac{w}{h} \frac{\partial w}{\partial z} = C_{33} = \int_{A} (2.0\alpha + 0.75\beta) \bar{w} \bar{N}_w \bar{N}_w \frac{\partial \bar{N}_w}{\partial z} r \, dA \\
(5.59)
\]

\[
\frac{w}{r} (u \cos \theta - v \sin \theta) = \bar{K}_{3132} = \int_{A} \beta \bar{w} \bar{N}_w \bar{N}_w \left[ 0.75 \bar{N}_u - 0.25 \bar{N}_v \right] r \, dA \\
(5.60)
\]

\[
\frac{1}{r h} \frac{\partial u}{\partial z} = K_{31} = -\int_{A} \frac{(2.0\alpha + \beta)}{r} \bar{N}_u \frac{\partial \bar{N}_u}{\partial z} r \, dA \\
(5.61)
\]

\[
\frac{2}{h^2} \frac{\partial^2 w}{\partial z^2} = K_{33} = \int_{A} (2.0\alpha + \beta) \frac{\partial \bar{N}_w}{\partial z} \frac{\partial \bar{N}_w}{\partial z} r \, dA \\
(5.62)
\]

\[
\frac{\partial^2 w}{\partial r^2} = K_{33} = \int_{A} (2.0\alpha + \beta) \frac{\partial \bar{N}_w}{\partial r} \frac{\partial \bar{N}_w}{\partial r} r \, dA \\
(5.63)
\]
\[
\frac{1}{r^2} \frac{\partial^2 w}{\partial \theta^2} = K_{33} = \int_A \frac{(2.0 \alpha + \beta)}{r^2} \tilde{N}^* \tilde{N}^* \, r \, dA 
\] (5.64)

\[
\frac{1}{r} \frac{\partial w}{\partial r} = K_{33} = -\int_A \frac{(2.0 \alpha + \beta)}{r} \tilde{N}^w \frac{\partial \tilde{N}^w}{\partial r} \, r \, dA 
\] (5.66)

\[
\frac{1}{r h} \frac{\partial^2 v}{\partial \theta \partial z} = K_{32} = \int_A \frac{\beta}{r} \tilde{N}^v \frac{\partial \tilde{N}^w}{\partial z} \, r \, dA 
\] (5.67)

\[
\frac{1}{h} \frac{\partial^2 u}{\partial r \partial z} = K_{31} = \int_A (2.0 \alpha + \beta) \frac{\partial \tilde{N}^u}{\partial r} \frac{\partial \tilde{N}^w}{\partial z} \, r \, dA 
\] (5.68)

5.5.4 Pressure gradient terms

\[
\frac{\partial p}{\partial r} = L_1^p = \int_A (2.0 \alpha + \beta) \tilde{N}^p \frac{\partial \tilde{N}^p}{\partial r} \, r \, dA 
\] (5.69)

\[
\frac{1}{r} \frac{\partial p}{\partial \theta} = L_2^p = \int_A \frac{\beta}{r} \tilde{N}^p \tilde{N}^r \, r \, dA 
\] (5.70)

\[
\frac{1}{h} \frac{\partial p}{\partial z} = L_3^p = \int_A (2.0 \alpha + \beta) \tilde{N}^u \frac{\partial \tilde{N}^p}{\partial z} \, r \, dA 
\] (5.71)

5.5.5 Continuity equation terms

\[
\frac{\partial u}{\partial r} = L_1^u = \int_A (2.0 \alpha + \beta) \tilde{N}^u \frac{\partial \tilde{N}^u}{\partial r} \, r \, dA 
\] (5.72)

\[
\frac{u}{r} = L_1^u = \int_A \frac{(2.0 \alpha + \beta)}{r} \tilde{N}^p \tilde{N}^u \, r \, dA 
\] (5.73)

\[
\frac{1}{r} \frac{\partial v}{\partial \theta} = L_2^v = \int_A \frac{\beta}{r} \tilde{N}^p \tilde{N}^r \, r \, dA 
\] (5.74)

\[
\frac{1}{h} \frac{\partial w}{\partial z} = L_3^w = \int_A (2.0 \alpha + \beta) \tilde{N}^w \frac{\partial \tilde{N}^p}{\partial z} \, r \, dA 
\] (5.75)
\[
\frac{(u \cos \theta - v \sin \theta)}{\vec{p}} L_{\gamma/2} = \int_{\partial\Omega} \frac{1}{r} \overrightarrow{N} \left[ (2.0 \alpha + 0.75 \beta) \overrightarrow{N} - 0.25 \beta \overrightarrow{N} \right] r dA \quad (5.76)
\]

5.5.6 Mass Terms

\[
m^u = \int_{\partial\Omega} (2.0 \alpha + \beta) \overrightarrow{N} \overrightarrow{N} r dA
\]
\[
m^v = \int_{\partial\Omega} \beta \overrightarrow{N} \overrightarrow{N} r dA \tag{5.77}
\]
\[
m^w = \int_{\partial\Omega} (2.0 \alpha + \beta) \overrightarrow{N} \overrightarrow{N} r dA
\]

5.5.7 Forcing vector term

\[
R^v = \vec{p} \int_{z} 2.0 \overrightarrow{N} \overrightarrow{N} r dS \tag{5.78}
\]

5.5.8 Natural Boundary Conditions

The terms on the right hand sides of Eqtns 5.31-5.76 arise from integration by parts, or Green's theorem, of the weighted terms of the Navier-Stokes and continuity equations (Eqtns 5.4 -5.10). The natural boundary condition terms, that also arise from integration by parts, cancel on internal element boundaries and are therefore omitted for the purpose of deriving the terms required for the finite element code. However, for completeness, the natural boundary condition terms for the fully developed flow case are presented below.

Integration by parts over a two- or three-dimensional domain, \( \Omega \), with boundary, \( \Gamma \), is given by the expression

\[
j_{\Omega} u (\nabla \cdot v) d \Omega = j_{\partial\Omega} u (v \cdot \hat{n}) d \Gamma - j_{\partial\Omega} v \cdot (\nabla u) d \Omega \quad (5.79)
\]

where \( u \) is the weighting function, \( N \), \( \nabla = \frac{\partial}{\partial r} \hat{i} + \frac{\partial}{r \partial \theta} \hat{j} + \frac{\partial}{h \partial z} \hat{k} \) and 
\( \hat{n} \) is the unit normal vector to the boundary, \( \hat{n} = n_r \hat{i} + n_\theta \hat{j} + n_z \hat{k} \)

(Huebner and Thornton, 1982)

More specifically, in two-dimensions, integration by parts with differentiation in the
\[ r \text{-} \text{and} \theta \text{-directions, respectively, gives the expressions} \]
\[
\int_a^b u \frac{\partial v}{\partial r} \, r \, d\theta \, dr = -\int_a^b \frac{v}{\partial \theta} \, r \, d\theta \, dr + \frac{1}{\rho} \, u v r \, n_r \, d\Gamma \tag{5.80}
\]
\[
\int_a^b u \frac{\partial v}{\partial \theta} \, d\theta \, dr = -\int_a^b \frac{v}{\partial \theta} \, d\theta \, dr + \frac{1}{\rho} \, u v n_\theta \, d\Gamma \tag{5.81}
\]

where \( n_r \) and \( n_\theta \) are the cosines between the outward normal and the \( r \) and \( \theta \) directions, respectively.

The natural boundary conditions for the relevant \( r, \theta \) differential terms from Eqtns 5.31-5.76 are as follows

5.5. 8.1 Radial direction terms

\[
u \frac{\partial u}{\partial r} = \frac{1}{\rho} \, \overline{N}_u \, \overline{N}_u \, u_i \, n_r \, d\Gamma \tag{5.82}
\]
\[
\frac{v}{r} \frac{\partial v}{\partial \theta} = \frac{1}{\rho} \, \overline{N}_v \, \overline{N}_v \, v_i \, n_\theta \, d\Gamma \tag{5.83}
\]
\[
-2 \frac{\partial^2 u}{\partial r^2} = -2 \frac{1}{\rho} \, \overline{N}_u \, \partial \overline{N}_u \, u_i \, n_r \, d\Gamma \tag{5.84}
\]
\[
\frac{1}{r^2} \frac{\partial^2 u}{\partial \theta^2} = \frac{1}{\rho} \, \overline{N}_u \, \partial \overline{N}_u \, u_i \, n_\theta \, d\Gamma \tag{5.85}
\]
\[
\frac{1}{r} \frac{\partial^2 v}{\partial r \partial \theta} = -\frac{1}{\rho} \, \overline{N}_v \, \partial \overline{N}_v \, v_i \, n_r \, d\Gamma \tag{5.86}
\]
\[
\frac{3}{r^2} \left( \frac{\partial v}{\partial \theta} \right) = -3 \frac{1}{\rho} \, \overline{N}_v \, \partial \overline{N}_v \, v_i \, n_\theta \, d\Gamma \tag{5.87}
\]
\[
\frac{2}{r} \frac{\partial u}{\partial r} = -2 \frac{1}{\rho} \, \overline{N}_u \, u_i \, n_r \, d\Gamma \tag{5.88}
\]
5.5.8.2 Tangential direction terms

\[ \frac{u}{\partial r} \]  
\[ \frac{v}{r \partial \theta} \]  
\[ \frac{2}{r^2 \partial \theta^2} \]  
\[ \frac{\partial^2 v}{\partial r^2} \]  
\[ \frac{1}{r \partial r} \]  
\[ \frac{3}{r^2 \partial \theta} \]  
\[ \frac{1}{r \partial r \partial \theta} \]

\[ \frac{\partial p}{\partial r} \]  
\[ \frac{1}{r \partial \theta} \]

5.5.9 Axial Boundary Conditions

Experimental evidence (Agrawal et al., 1978) indicates that whatever the flow regime upstream of the entrance, the flow very rapidly takes on a pattern corresponding to a
potential vortex before developing further due to the tube curvature. The entry conditions can therefore be assumed to be

\[ u = v = 0 \]

\[ w = \frac{-w}{(1 + b \cos \theta)} \]  \hspace{1cm} (5.100)

\[ p = -\frac{\bar{p}}{2(1 + b \cos \theta)^2} \]

Fig 5.6(a) shows a coarse mesh version of the longitudinal section model with 48 fluid and 12 wall elements. The element widths are scaled by \( h \) to represent the correct curved tube proportions at any angle \( \theta \), so that in the two-dimensional mesh shown here in the \( \theta = 90^\circ \) position, for example, the elements have the nominal widths.

![Wall Elements](image)

![Fluid Elements](image)

**Fig 5.6(a) Finite element model**

The boundary conditions additional to Eqtn (5.100), to impose the 'no slip' condition and fix the wall in the longitudinal direction are shown in Fig 5.6(b).
5.6 Wall Elasticity

In order to be able to express the wall stiffness in the program in terms of the distensibility, $D$, where

$$ D^{-1} = \frac{E}{1-\nu^2} \left( \frac{h}{d} \right) $$

(2.1)

$$ D = \frac{1}{A} \left( \frac{\partial A}{\partial \rho_{om}} \right) $$

(2.2)

Eqtns (2.1) and (2.2) must be expressed in difference form.

Writing Eqtn (2.2) in terms of the tube radius gives

$$ D = \frac{2}{r} \left( \frac{\partial r}{\partial \rho_{om}} \right) $$

(5.101)

Writing Eqtn (5.101) in backward difference form
$$D_i = \frac{2}{r_{i-1}} \left( \frac{r_i - r_{i-1}}{p_i - p_{i-1}} \right)$$ (5.102)

where \( r_i, p_i \) are the current values of tube radius and pressure and \( r_{i-1}, p_{i-1} \) are the values at the end of the previous time step.

The current value of incremental modulus, \( E_i \), is then given by

$$\frac{E_i}{1 - \nu^2} = D_i^{-1} \left( \frac{2r_i}{h} \right) = \frac{r_i r_{i-1}}{h} \left( \frac{p_i - p_{i-1}}{r_i - r_{i-1}} \right)$$ (5.103)

Providing the time steps are small, \( r_i \to r_{i-1} \) and \( p_i \to p_{i-1} \), so that

$$\frac{E_i}{E_{i-1}} \approx \left( \frac{r_i}{r_{i-1}} \right)^2$$ (5.104)

The modulus can then be easily updated at the end of each time step. To cater for any out-of-roundness in the tube the (circumferential) mean tube radii, \( \bar{r}_i, \bar{r}_{i-1} \), can be used, so that Eqtn 5.83 becomes

$$\frac{E_i}{E_{i-1}} = \left( \frac{\bar{r}_i}{\bar{r}_{i-1}} \right)^2$$ (5.105)

no rigour is claimed for the above derivation. The result is found, however, to give a reasonable approximation to the behaviour of elastin and provides an expedient method of incorporating increasing stiffness with increasing pressure into the fluid-wall analysis.
6 RESULTS

In the absence of theoretical solutions for curved tube flow problems, validation of the computer codes PFECTL (longitudinal tube sections) and PFECTX (tube cross-sections) for curved tube cases can only be achieved by comparison of program results with those from suitable numerical and experimental results in the literature. Although there are no absolute values for the solutions, certain trends in the patterns of curved tube flows are confirmed by many researchers and it is on the ability to reproduce these that the curved tube performance of the programs is based.

Analytical solutions for the straight tube case are available, however, (Womersley, 1955) and validation of the codes for this case confirms the time-stepping and Picard iteration algorithms in addition to such code details as the method of mass lumping and application of surface tractions.

The finite element and pre-processing programs were run on a Hewlett-Packard 9000/700 Series system. The output data was downloaded to a 386 PC for post-processing.

6.1 Straight Tube Solution

Since the performance of the programs for any viscous, time-dependent flow is dependent on the iteration and time-stepping algorithm, common to both programs, it is crucial that this is validated at the outset. The simplest pulsatile flow, curved tube problem is that of a tube with large radius of curvature and rigid walls, subjected to a sinusoidally varying pressure cycle. Starting from the equation of motion of the oscillatory flow in a straight, rigid tube

\[
\frac{\partial^2 w}{\partial t^2} + \frac{1}{r} \frac{\partial w}{\partial r} - \frac{1}{\nu} \frac{\partial w}{\partial t} = \frac{A e^j \omega t}{\mu}
\]

(6.1)

where \( w \) is the axial fluid velocity, \( r \) the distance from the tube axis, \( A \) the amplitude of the pressure gradient, oscillating with angular frequency \( \omega \), and \( \mu \) and \( \nu \) are the dynamic and kinematic viscosities respectively, Womersley (1955) derived the expression for the axial velocity distribution as

\[
w = \frac{M}{\rho \omega} \left\{ \sin(\omega t + \phi) - \frac{M_0(y)}{M_0} \sin(\omega t + \phi - \delta_0) \right\}
\]

(6.2)
where $y = \frac{r}{a}$, $M$ is the real part of $Ae^{i\omega t}$, $M_0$ and $M_0(y)$ are Bessel functions of order zero, $(\omega t + \phi)$ is the phase angle, and $\delta_0 = \theta_0 - \theta_0(y)$, the phase change from the tube centreline phase, at $y$.

McDonald (1974) predicted the velocity distribution in a canine femoral artery using Womersley's equation (Eqtn 6.2), representing the pressure cycle as the sum of four harmonic components of frequency parameters, $\alpha = 3.34, 4.72, 5.78$ and 6.67, respectively, together with a Poiseuille distribution representing steady forward flow. Agreement between the results obtained with the finite element code and an analytical solution such as McDonald's provides very good indication of the validity of the finite element code, excepting the effects of curvature.

A simple axisymmetrical finite element model was set up (Fig 6.1). The use of 10 eight-noded elements was thought to be the minimum that could be reasonably used and also comparison with the analytical solution with a mesh of this coarseness would be a good test of the program performance. The pressure cycle was idealised into 24 discrete steps, again thought to be the minimum necessary to achieve comparable results and again providing a good test of the program performance. The curvature ratio was set at $10^{-6}$ to simulate a straight tube.

![Fig 6.1 Axisymmetric finite element model](image)

Axial velocity profiles are plotted for 15° intervals of pressure cycle for McDonald's and the finite element solution in Figs 6.2 and 6.3. It can be seen that,
notwithstanding the coarseness of the finite element mesh and the relatively large time steps, good agreement was obtained between the analytical and finite element solutions. It can therefore be assumed that the performance of time-stepping and iteration algorithms as well as all other features of the programs not dependent on tube curvature are satisfactory.

Fig 6.2 Comparison between McDonald's solution and finite element solution for canine femoral artery pressure cycle. Variation of axial velocity over the pressure cycle at tube locations $r/a = 0.0$, $0.5$, $0.75$, and $0.875$.

Fig 6.3 (a) Comparison between McDonald's solution and finite element solution for canine femoral artery pressure cycle. Velocity profiles at the tube horizontal centreline for phase angles $0$ to $75^\circ$. 

67
Fig 6.3(b) Comparison between McDonald's solution and finite element solution for canine femoral artery pressure cycle. Velocity profiles at the tube horizontal centreline for phase angles 105 to 180°.

Fig 6.3(c) Comparison between McDonald's solution and finite element solution for canine femoral artery pressure cycle. Velocity profiles at the tube horizontal centreline for phase angles 195 to 270°.
Fig 6.3(d) Comparison between McDonald's solution and finite element solution for canine femoral artery pressure cycle. Velocity profiles at the tube horizontal centreline for phase angles 300° to 360°.

6.2 Structured vs unstructured finite element mesh

It was argued in Section 5.3.1 that the boundary conditions around the void can be selected to allow free slip and that providing the void is very small, intuitively the disturbance to the transverse flow will also be very small. In the current structured mesh (Mesh 1) the radius of the central void is only 0.1% of the tube radius.

In order to investigate the effect of the fluid void on the flow, the solution given by the structured mesh (Fig 6.4) is compared with the solutions obtained with three other meshes with a similar number of elements; Mesh A, mixture of both unstructured and structured areas (Fig 6.5), Mesh B, a predominantly unstructured mesh (Fig 6.6) and Mesh C, a structured mesh consisting of Mesh 1 with the addition of an unstructured core (Fig 6.7) to cater for the singularity at the tube centre.

In Meshes A, B and C the nodes in the unstructured sections of the mesh are set out in rectangular arrays, but the nodal geometry remains in toroidal (cylindrical in the plane) coordinates. This arrangement allows the use of rectangular elements with internal Gauss points at the centre of the tube, avoiding the use of triangular elements and the consequent problem of a Gauss point located at the origin of the coordinates. With a void at the origin, as in Mesh 1, the problem of defining the orientation of the velocities for a node at the origin, where the θ coordinate is unspecified, is avoided.
With elements arranged around the centre of the tube in the unstructured sections of mesh a node is placed on the origin. The problem of defining the velocity orientation there is overcome by arbitrarily specifying the angular coordinate for this node as 0°. This effectively aligns the radial velocity at the node with the horizontal centreline so that the symmetric boundary condition, \( v = 0 \) along the centreline, can be applied at the centre node. An alternative approach would have been to displace the centre node from the origin by a small amount, relying on smoothing from the shape functions and Gauss quadrature.

The analyses for all four meshes; Mesh 1, Mesh A, Mesh B and Mesh C are carried out using the PFECTX code, i.e., in toroidal coordinates with fully developed flow. The analyses were carried out for the pressure cycle in Fig 6.8, for a mean Reynolds number, \( Re_m = 350 \) and frequency parameter, \( \alpha = 11.4 \). Axial flows predicted by the analyses for three of the four meshes are plotted along the tube horizontal axis of symmetry in Figs 6.9 and the corresponding transverse flows are plotted in Figs 6.10. As it was found that, outside the immediate core area at the centre of the tube, there was no appreciable difference between the results obtained for Mesh 1 and Mesh C and the graphs plotted for Mesh 1 are equally applicable to Mesh C.

Figs 6.9 show that in general there are very little differences between the solutions for axial velocity obtained with the four meshes, except for the velocities in the reverse flow regions at the tube wall for phase angles \( \pi \) and \( 2\pi \) which show some scatter. Scatter in regions of low, reverse flow appears to be a common feature of the results obtained with the present finite element codes. Where absolute velocities are small, the differences between solutions become more apparent, but there also appears to be tendency for solutions to vary when the flow is low and reversing. This is probably a numerical problem attributable to the size of time step.
Fig 6.5 Mixed structured-unstructured mesh - Mesh A

Fig 6.6 Unstructured finite element mesh - Mesh B
While the axial velocities for each mesh are very similar, Figs 6.10 show that there are significant differences between the unstructured and structured mesh solutions for the transverse velocities. These differences are highlighted in the velocity contour and streamline plots in Figs 6.11 and 6.12. The near wall velocities in the region of the primary vortex are similar for each mesh and the velocity contours in this region indicate similar velocity gradients, but in the central region of the tube, although the magnitudes of the velocities are again similar, the directions of flow are markedly different. The increases in velocity contour density from Mesh I to Mesh A and from Mesh A to Mesh B in Fig 6.11 appear to be due to the contour plot software handling of the data in the various mesh configurations.

The velocity vector plots in Fig 6.13, all enlarged to the same scale to show the local flow at the centreline, indicate that there is no significant disturbance to the flow across the centre of the tube due to the unstructured mesh and the arbitrary specification of the velocities at the central node for Meshes A and B. The much further enlarged vector plot for the centre of the structured mesh, Mesh 1, shows that the applied boundary condition is effective in allowing only tangential velocity around the central fluid void. It is clear that whichever device is used to attempt to overcome

---

Fig 6.7 Unstructured finite element core for Mesh C. (Remainder of mesh as Fig 6.4)
the problem of the zero radius at the centre of the tube, the effect is very localised and has little influence on the general secondary flow in the tube.

With the parameters, $Re_m = 350$ and $\alpha = 11.4$, Sudo et al. (1992) predict that the flow regime will include a second, Lyne-type vortex over part of the cycle (Fig 6.14). While this vortex is clearly present in the solutions for the structured mesh, it is absent from the others.

![Fig 6.8 Pressure cycle](image)

![Fig 6.9(a) Axial velocities for structured and unstructured meshes.](image)

$Re_m = 350$, $\alpha = 11.4$, $\delta = 1/7$, $\theta = \pi/2$
Fig 6.9(b) Axial velocities for structured and unstructured meshes.

\[ Re_m = 350, \ \alpha = 11.4, \ \delta = 1/7, \ \theta = \pi \]

Fig 6.9(c) Axial velocities for structured and unstructured meshes.

\[ Re_m = 350, \ \alpha = 11.4, \ \delta = 1/7, \ \theta = 3\pi/2 \]

Fig 6.9(d) Axial velocities for structured and unstructured meshes.

\[ Re_m = 350, \ \alpha = 11.4, \ \delta = 1/7, \ \theta = 2\pi \]
The results for the structured mesh, using parameters $Re_m = 375$, $\alpha = 15$, and summarised in Section 6.3 also show no second vortex, whereas the finite difference results of Hamakiotes and Berger (1988, 1990), with which those results are compared, show this vortex.

There appear to be two possible reasons for the non-appearance of the second vortex. The finite difference solutions of Hamakiotes and Berger are based on three cycles of 400 time steps per cycle, compared with the single cycle, 24 and 48 time steps of the finite element solutions. The second vortex for the range of parameters $350 < Re_m < 375$ and $11.25 < \alpha < 15.0$ is reported by Sudo et al. (1992) and Lyne (1971), among others, to be short-lived, the vortex becoming established for the duration of the cycle, with these frequency parameters, only at higher Reynolds numbers. The vortex may therefore fail to appear if relatively large time steps are used.

Secondly there appears to be a mesh dependence of the solutions for Mesh A and Mesh B in Figs 6.10. The transition from structured to unstructured mesh would intuitively have some minor effect on the solution, but more difficult to explain is that the solution for the completely unstructured mesh also fails to produce a second vortex. It is reported (Hirsch, 1988a) that while finite difference solutions may be adversely affected by the form of the mesh, this does not apply to finite element solutions, except in regions where there is a major change in element size or type. The only inference that can be drawn is that the structured mesh, at least for the Reynolds number and frequency parameter used here, is more likely to produce the second vortex than the unstructured mesh.

While it may be argued that the mesh dependence demonstrated could be reduced considerably with improved mesh refinement, the present study is concerned with evaluating the quality of finite element solutions for blood flow problems within the limitations of the available computer resources. Within this constraint the structured mesh provides the best solution and its use in the present study is therefore vindicated.
Fig 6.10 (a) Comparison between transverse velocity solutions for structured and unstructured finite element meshes, (i) Mesh 1 (structured) (ii) Mesh A (unstructured) (iii) Mesh B (unstructured) Phase angle = 90°
Fig 6.10 (b) Comparison between transverse velocity solutions for structured and unstructured finite element meshes, (i) Mesh 1 (structured) (ii) Mesh A (unstructured) (iii) Mesh B (unstructured), Phase angle = 180°
Fig 6.10 (c) Comparison between transverse velocity solutions for structured and unstructured finite element meshes, (i) Mesh 1 (structured) (ii) Mesh A (unstructured) (iii) Mesh B (unstructured). Phase angle = 225°
Fig 6.10 (d) Comparison between transverse velocity solutions for structured and unstructured finite element meshes, (i) Mesh 1 (structured) (ii) MeshA (unstructured) (iii) Mesh B (unstructured), Phase angle = 270°
Fig 6.10 (e) Comparison between transverse velocity solutions for structured and unstructured finite element meshes, (i) Mesh 1 (structured) (ii) Mesh A (unstructured) (iii) Mesh B (unstructured), Phase angle = 360°
Fig 6.11 Comparison between transverse velocity contours for structured and unstructured finite element meshes, (i) Mesh 1 (structured) (ii) Mesh A (unstructured) (iii) Mesh B (unstructured), Phase angle = 360°
Fig 6.12 Comparison between transverse velocity streamlines for structured and unstructured finite element meshes, (i) Mesh 1 (structured) (ii) Mesh A (unstructured) (iii) Mesh B (unstructured), Phase angle = 360°
Fig 6.13(a) Comparison between transverse velocity solutions for structured and unstructured finite element meshes, (i) Mesh 1(structured) (ii) Mesh A (unstructured) (iii) Mesh B (unstructured), Enlarged views around centreline of tube, to same scale. Phase angle = 360°
Fig 6.13(b) Transverse velocity solution for structured finite element mesh. Enlarged view around tube centreline, Mesh 1. Phase angle = 360°
6.3 Fully Developed Flow in a Curved Tube

Since Hamakiotes and Berger (1988, 1990) carried out a recent finite difference analysis for a tube cross-section, using a high-powered computer, their results are appropriate for comparison with those from the program PFECTX, the code for the analysis of a transverse section of the tube. It must be pointed out, however, that their results were subsequently evaluated by Swanson et al. (1993), who investigated the second problem analysed by them using laser Doppler anemometry (Hamakiotes and Berger, 1990) and found to be inaccurate in some areas. However, of the few reported in the literature, their solutions are probably the most relevant for comparison with the results from PFECTX.

In the input data for program PFECTX the pressure cycle, Reynolds number, frequency parameter and curvature ratio were all chosen to be the same as those used by Hamakiotes and Berger, i.e., mean Reynolds numbers, $Re_m = 200$ and 375, frequency parameter $\alpha = 15.0$ and curvature ratio $\delta = 1/7$.

$Re_m$ is defined as $W_{DC}\left(\frac{a}{v}\right)$, where $W_{DC}$ is the time-mean value of axial velocity.

The pressure cycle and mean average axial velocity for $Re_m = 200$ are shown in Figs 6.15 and 6.16. The pressure cycle was chosen to give a volumetric flow rate with an alternating component of 98% of the mean component. The 11 x 12 finite element mesh is shown in Fig 6.17. The tube walls were assumed to be rigid. 24 time steps were used for the $Re_m = 200$ problems and 48 steps for the $Re_m = 375$ problem.

To put the flow patterns obtained in context, the diagram showing the flow patterns predicted by Sudo et al., 1992, for ranges of the Dean and Womersley numbers are reproduced in Fig 6.14. The flow regimes between lines A and F on Fig 6.14 are relevant.
Equi-kinetic energy lines of the secondary flow. ——, numerical results. ——, A–F distinguish the flow patterns.
Fig 6.15 Mean axial velocity, $Re_m = 200, \alpha = 15, \delta = 1/7$

Fig 6.16 Pressure Cycle, $Re_m = 200, \alpha = 15, \delta = 1/7$
Fig 6.17(a) Finite element mesh

Fig 6.17 (b) Finite element mesh - enlarged central section
6.3.1 Velocity and Pressure Distributions

The distributions of axial velocity across the tube horizontal diameter are shown in Figs 6.18 and Fig 6.19 for the two cases, $Re_m = 200$ and $Re_m = 375$, respectively, for every other time step. The distributions of the pressure across the tube cross-section for $Re_m = 200$ ($D = 76$) are shown in Fig 6.20. The distributions for $Re_m = 375$ ($D = 142$) are similar.

The resultants of the finite element nodal tangential and radial velocities are plotted as secondary flow vectors in Figs 6.21 and those for the finite difference solution of Hamakiotes and Berger (1988, 1990) are plotted in Fig 6.22, for phase angles of $\pi/4$, $\pi/2$, $3\pi/4$, $\pi$, $5\pi/4$, $3\pi/2$, $7\pi/4$ and $2\pi$, for the two cases, $Re_m = 200$ and $Re_m = 375$, respectively. The flow vectors are normalised with respect to the maximum tangential velocities.

Comparing the secondary flow vectors in Figs 6.21, for the $Re_m = 200$, $\alpha = 15$ cases, it can be seen that there is very good qualitative agreement between the finite element solution and that of Hamakiotes and Berger (1988), the flow patterns agreeing at every region of the pressure cycle. The flow patterns correspond to the Dean type flow in Fig 6.14., as expected for $D < 100$.

The flow patterns for the $Re_m = 375$ case compare less well, however. Although the finite element results show the beginning of the Lyne type flow reversal in the earlier stages of the cycle, the second vortex, completing the flow reversal across the horizontal centreline, is absent. The flows around the tube wall, however, agree well.

Lyne's results (Lyne, 1970) predicted that the secondary flow in unsteady flow is in the opposite direction to that in steady flow. This was thought to be due to 'centrifuging' generating motion entirely confined to the Stokes layer. In the Stokes layer the pressure gradient is no longer balanced by the centrifugal force due to flow along the tube. The pressure gradient then drives fluid along the wall from the outside of the bend to the inside and returns on the inside of the Stokes layer, dragging the interior fluid with it. At the same time the peak axial velocity moves toward the inside of the bend. Zalosh and Nelson (1973) confirmed Lyne's prediction and concluded that the onset of the reversal of the secondary flow occurs at a frequency parameter, $\alpha = 9$. Below this value the flow is similar to that in steady flow.

The analysis was repeated for a mesh of 260 elements, corresponding to that of Hamakiotes and Berger but no significant change in secondary flow patterns was obtained.
Fig 6.18 Axial velocity distribution across tube axis of symmetry
\( \text{Re}_m = 200 \)
Fig 6.19 Axial velocity distribution across tube axis of symmetry

Re_m = 375
Fig 6.20 Pressure distribution across section, $Re_m=200$
Fig 6.21 (a-d) Secondary velocity vectors - finite element analysis
Phase angles (a) $\pi/4$  (b) $\pi/2$  (c) $3\pi/4$  (d) $\pi$. $Re_m = 200$
Fig 6.21(e-h) Secondary velocity vectors - finite element analysis.

Phase angles (e) $\frac{5\pi}{4}$ (f) $\frac{3\pi}{2}$ (g) $\frac{7\pi}{4}$ (h) $2\pi$, $Re_m = 200$
Fig 6.21 (i - l)  Secondary velocity vectors - finite element analysis

Phase angle (i) $\pi/4$ (j) $\pi/2$ (k) $3\pi/4$ (l) $\pi$ , $Re_m = 375$
Fig 6.21 (m - p) Secondary velocity vectors - finite element analysis
Phase angle (m) $\pi/4$ (n) $\pi/2$ (o) $3\pi/4$ (p) $\pi$, $Re_m = 375$
Fig 6.22 (a-e) Secondary velocity vectors - finite difference analysis
(Hamakiotes and Berger, 1988) Phase angle (a) $\pi/4$  (b) $\pi/2$  (c) $3\pi/4$
(d) $3\pi/2$  (e) $\pi$ , $Re_m = 200$
Fig 6.22 (f-k)  Secondary velocity vectors - finite difference analysis
(Hamakiotes and Berger, 1990) Phase angle (a) $\pi/4$  (b) $\pi/2$  (c) $3\pi/4$
(d) $\pi$  (e) $3\pi/2$  (f) $2\pi$, $Re_m=375$
Sudo et al. (1992), confirm the flow reversal predicted by Lyne, but comment that the second vortex is short-lived. In addition, Fig 6.14 shows that the flow pattern for $D = 142$ and $\alpha = 15$ lies almost exactly on the border between Dean and Lyne type flows.

Comparing the axial velocity distributions in Fig 6.18 and 6.19 with those predicted by Hamakiotes and Berger (1988), it can be seen that generally there is agreement between the two. The small increase in velocity at the centreline is due to the form of the mesh at the centre of the tube. The profile of the axial velocity for the $Re_m = 375$ case (Fig 6.19) can be compared directly with the plots produced by Swanson et al. (1993) and here it can be seen that positions of the higher parts of the velocity profile are reversed, the finite element analysis predicting the higher axial velocity near the inside of the bend. The quantitative differences between the finite element and those of Swanson et al., however, are small.

The pressure distributions across the tube (Fig 6.20) are also as expected, the pressure gradient on the tube horizontal centreline, being directed towards the inside of the bend to balance the centrifugal force.

A comparison is made between the finite difference solutions for axial velocity of Hamakiotes and Berger (1988, 1990) and the finite element code PFECTX in Fig 6.23 and Fig 6.24. It can be seen that good agreement is obtained for the $Re = 200$ case, at every phase angle, while agreement for the $Re = 375$ case less consistent. The profiles at phase angles showing reversed flow in particular show poor agreement.

![Diagram showing comparison between finite difference analysis and finite element analysis](image)

**Fig 6.23** Comparison between finite difference analysis (Hamakiotes and Berger, 1988) and finite element analysis (PFECTX). Axial velocity distribution $Re_m = 200$
Fig 6.24 Comparison between finite difference analysis (Hamakiotes and Berger, 1990) and finite element analysis (PFECTX).

Axial velocity distribution \( \text{Re}_m = 375 \)

Fig 6.25 Comparison between results from the two codes. Axial velocity \( \text{Re} = 200 \)

PFECTL = longitudinal section analysis  PFECTX = transverse section analysis
6.3.2 Comparison between codes PFECTX and PFECTL
Solutions obtained with the semi-analytical code, for tube longitudinal sections, PFECTL, and the code for the fully developed tube cross sections, PFECTX, are compared in Fig 6.25 for the Re = 200 case. Good agreement is obtained between the codes throughout the pressure cycle, except in the flow reversal phase where the semi-analytical solution seems unable to follow the curvature of the velocity profile there. The velocity vectors for the transverse flow obtained from PFECTL are shown in Fig 6.26. The figure shows typical velocity patterns, only the magnitude of the vectors changes throughout the cycle, the centre of the vortex appearing at the 90° position.

Fig 6.26 Transverse velocity vectors - PFECTL code
Fully developed flow
6.4 Wall Shear Stresses

Many methods exist for estimating wall shear rates from velocity data points near to the wall. Lou et al. (1993) assessed the errors involved in predicting wall shear rates from curve fitting velocity profiles. Perktold et al., 1991 used a cubic curve fit. Quadratic (Duncan et al., 1990) and least squares curve fits (Ku et al., 1985) are found to be superior to the linear method for pulsatile flows. The quadratic curve fit is used here in the form

\[
\text{shear rate} = \frac{u_2 dr_1^2 - u_1 dr_2^2}{dr_2^2 dr_1 - dr_1^2 dr_2} \tag{6.3}
\]

where \(u_1, u_2\) are the velocities at two finite element nodes and \(dr_1\) and \(dr_2\) are the nodal coordinates in the radial direction (Fig 6.27).

![Fig 6.27 Velocity data points](image)

The axial and tangential shear rates for positions around the tube circumference corresponding to the finite element corner nodes are plotted in Figs 6.28 and 6.29 for \(Re_m = 200\) and 375, respectively, using the PFECTX code over one cycle, together with the values obtained by Hamakiotes and Berger for their third cycle. It can be seen that in both flow cases very good qualitative agreement is obtained for the tangential
wall shear stresses, while the axial shear stresses compare less well qualitatively. The opposite is true for the peak values where the axial stresses compare well. The method of calculating the wall shears is the same for both directions for the finite element results and presumably this is also the case for Hamakiotes and Berger, although their method is not described. Fig 6.28(a) shows a step in the axial velocity shear distributions for the phase angles of 225° and above, which do not appear in the Fig 6.29(a). As the steps occur at the end of the cycle this may be due to the effect of flow reversal.

![Diagram](Fig 6.28(a) Axial wall shear rate - finite element solution vs finite difference solution. \( \text{Re}_m = 200, \ \alpha = 15, \ \delta = 1/7 \))
Fig 6.28(b)  Tangential wall shear rate - finite element solution vs finite difference solution. $Re_m = 200$, $\alpha = 15$, $\delta = 1/7$

Fig 6.29(a)  Axial wall shear rate - finite element solution vs finite difference solution. $Re_m = 375$, $\alpha = 15$, $\delta = 1/7$
Fig 6.29(b)  Tangential wall shear rate - finite element analysis vs finite difference solutions. $Re_m = 375$, $\alpha = 15$, $\delta = 1/7$
6.5 Entry Flow in a Curved Tube
Program PFECTL was run with a rectangular mesh of 10 x 20 elements for the steady flow problem and 10 x 10 elements for the pulsatile flow problem, giving tube lengths of 4a and 2a, respectively. The boundary conditions corresponded to entry flow, with the entry axial velocity distributed uniformly and the tangential and radial velocities set to zero. The program was run for both steady flow in straight and curved tubes and for oscillatory flows in a curved tube.

6.5.1 Steady Flow

6.5.1.1 Straight tube
Typical velocity profiles for the straight tube, steady flow case are shown in Fig 6.30 for a Reynolds number of 50. Two finite element models were used with uniform and graded meshes (Fig 6.31). The uniform mesh model was run for Reynolds numbers of 50, 100, 150, 200 and 250, but the graded mesh became too short to model fully developed entry flows above Reynolds numbers above 150, if a reasonable aspect ratio was to be maintained for the elements. It was found that aspect ratios larger than 5 caused severe oscillation between the midside and corner node velocities. The predicted entry lengths are plotted in Fig 6.32 against the results of Lew and Fung (1970) and Shah and London (1978). It can be seen that there is a tendency to underestimate the entry length, taken as the mesh line where the maximum velocity reached 99% of the Poiseuille value, but the results are much improved over those obtained previously. Results obtained with the graded mesh show some further improvement over the uniform mesh results, indicating that mesh refinement is a factor in the underestimation of entry lengths.

6.5.1.2 Curved tube
The finite element input data was chosen to match that of Snyder et al. (1985), so that a direct comparison could be made with their hot-wire anemometry results. The input data was chosen to give a curvature ratio, δ = 1/8 and Rem = 540. This Reynolds number was not achievable with the 10 x 20 mesh, but a value of Rem = 490 was obtained with a uniform mesh. The axial velocity profiles determined by hot-wire are compared with the finite element solutions in Fig 6.33. The finite element model length corresponds to a cross-section position of 40°, so that a comparison could be made with the first four profiles of the experimental results. It can be seen that the development of the finite element axial velocity profile follows the experimental profile closely up to the 20° section and then begins to become flatter.
The entry length for a curved tube is quoted by Snyder *et al.*, as

\[ \frac{L_c}{L_s} \approx \frac{8e_1}{\kappa^{1/2}}, \]

where \( L_s = 0.125a\text{Re} \)

\( L_s, L_c \) are the straight and curved tube entry lengths and \( e_1 \) is a parameter, depending weakly on \( \delta \), so that \( 2 \leq e_1 \leq 4 \). With a \( Re_m = 490 \) \( (\kappa = 200) \), the finite element solution should give a developed length of \( 50a \), well beyond the length of the model.

The trends in the axial velocity profiles compare reasonably well for the range achieved, showing the changeover from entrance flow where the maximum velocity is on the inside of the bend, moving to the outside, downstream. Snyder *et al.* (1985), discuss developing flow in curved tubes and conclude in part that there is no single dimensionless parameter for the characterisation of axial velocity skew. Figs 6.34 show plots of secondary flow vectors for cross-sections of the finite element model at 0.1\( a \), 0.5\( a \) and \( a \), roughly corresponding to the positions of the cross-sections for Singh's streamlines (Fig 6.35). It can be seen that the general trend of the secondary flow vectors at the two downstream positions is in agreement with Singh's prediction. Perktold *et al.* (1991), show diagrams of secondary flow for a curved tube representing a coronary artery. Their secondary flow vectors show a very similar pattern to Fig 6.34(a), which, as would be expected, shows a strong radial flow towards the centre of the tube. The length of tube where this radial velocity dominates the flow is equivalent to about one diameter.

![Fig 6.30 Steady entry flow for straight tube
Re = 50](image)
Fig 6.31 Uniform and graded finite element meshes
Fig 6.29 Entrance length vs Reynolds number

Fig 6.33 Entry velocity profiles for steady flow in a curved tube compared with Snyder et al (1985) Re = 230, $\delta = 1/8$
Fig 6.34  Steady entry flow - secondary velocity vectors
finite element analysis
Fig 6.35 Steady entry flow - secondary velocity vectors - Singh (1974)
Fig 6.36 (a) Unsteady entry flow - finite element results compared with LDA results of Rindt et al. (1991). Mean Re = 500, $\alpha = 7.8$, $\delta = 1/6$, $\theta = 0^\circ$

Fig 6.36 (b) $\theta = 90^\circ$
Fig 6.36 (c) $\theta = 180^\circ$

Fig 6.36 (d) $\theta = 270^\circ$
6.5.2 Oscillating Flow

The problem data for determining oscillating flow entry lengths for curved tubes was chosen to match that of Rindt et al. (1991) who carried out a three-dimensional finite element analysis on a 90° bend and compared their results with LDA measurements from a perspex model. Again the Reynolds number achieved was less than the value of $Re = 500 \pm 300$ value used by them for oscillating flow and the cycle actually used was $Re = 330 \pm 200$. The results of the finite element analysis are plotted in Fig 6.36 together with LDA measurements. Since it is the developing profiles that are of interest here, the velocities were normalised.

The graphs show that, notwithstanding the difference in Reynolds numbers between the finite element and experimental results, quite reasonable agreement has been obtained. As in the steady flow case mentioned above, there is some flattening of the finite element profiles relative to the experimental profiles and there is a maximum discrepancy of about 15%. The flattening of the profile can be explained in part by the fact that, due to the use of a single harmonic in the semi-analytical formulation, the axial velocity is made up of a straight tube component, symmetrical about the tube axis, and an anti-symmetrical curved tube component. This constrains the finite element solution to follow lower order curves than the measured values produce.

6.6 Elastic Wall

6.6.1 Curved Tube

As wall elasticity is not thought to have an influence on blood flow in the aorta at the entrance, where the wall deflections are constrained by the adjacent structures (Pedley, 1976), the fluid-wall interaction was only investigated for the curved tube case for fully-developed flows. Program PFECTX was therefore appropriate and was run for the velocity cycle shown in Fig 6.37. The cycles for both axial and maximum radial pressures are shown in Fig 6.38. Analyses were carried out, firstly with a rigid-walled model (Young's modulus, $E = 10^2$) and then with an elastic wall modulus calculated to give a maximum deflection in the tube wall at the peak pressure of about 10%. The wall displacements normal to the horizontal axis of symmetry were constrained, together with the horizontal displacements on the vertical axis. No damping was applied. The wall was modelled with either one or two elements in the radial direction. Fluid properties appropriate to blood were used, i.e., $\nu = 0.004 \, \text{m}^2/\text{s}$, $\rho = 1050 \, \text{kg/m}^3$. The mean Reynolds number, $Re_m = 37$ and $\alpha = 15$. The tube internal radius was taken as 1.20 cm and the wall thickness 0.15 cm.
Fig 6.37  Curved elastic tube - mean axial velocity

$Re_m = 37, \quad \alpha = 15$

Fig 6.38  Curved elastic tube - axial and maximum radial pressures
Fig 6.40 Curved elastic tube - secondary velocities
Phase angles (a) $\pi/2$  (b) $\pi$  (c) $3\pi/2$  (d) $2\pi$
Fig 6.41 Curved elastic tube - deflected shape of tube
The axial velocities for the elastic wall model are plotted in Figs 6.39. Secondary velocities are plotted in Fig 6.40. There was found to be no appreciable difference in the axial and secondary velocities predicted for the rigid and elastic wall cases, nor was there any difference for the one- and two-element wall models. The deflections at the inside of the tube wall are shown in Fig 6.41. The sharp bends in the tube wall at the horizontal axis are spurious and are due to the coarse modelling of the wall.

Referring to the pressure plots (Fig 6.38), it can be seen that the maximum radial pressure is of the same order as, and two time steps out of phase with the axial pressure, corresponding exactly to $\pi/4$ radians. It can be seen that there is an appreciable out-of-roundness in the tube during the whole of the cycle, the maximum deflection moving from the outside to the inside of the bend at around step 7, i.e., at the same time as the radial pressure peak. The wall deflects outwards on the high-pressure side initially, at the outside of the bend, the bulge then moves around to the inside of the bend as the radial pressure falls. The interaction between fluid and wall is, as expected, very complex, involving the pressure, fluid density, wall stiffness and frequency of the pressure cycle. The results obtained were repeatable and the deflections predicted were found to be dependent on all of these parameters.

### 6.6.2 Straight Tube

As there were found to be apparently no results from the literature with which to compare the curved tube wall deflections, in order to make some assessment of the performance of the fluid-wall interaction algorithm, further analyses were carried out with a model of a straight tube. Since it is the radial displacement of the tube that is of interest a simple axisymmetric model was used with 20 fluid and four wall elements in a single radial column. The use of a simple model also gives the advantage that much larger numbers of time steps can be used than would have been possible with larger models.

Published results are available for the analysis of flow through tubes with elastic walls, based on a wide variety of flow parameters, tube geometries and tube wall properties, making comparisons very difficult. Most published work in this area is concerned primarily with wave propagation and few publications deal specifically with tube wall distension. Xu and Collins (1995) however, published details of their coupled fluid-wall finite element analysis and their comprehensive code validation study and this provides very suitable data for validating the fluid-wall interaction code. Their analysis was based on a three-dimensional model using eight fluid and four wall twenty-noded brick elements in
the radial direction. The three-dimensional model and the axisymmetric model used here therefore have the same number of wall elements and interpolation order.

The code validation study here is based on (a) the comparison of static wall displacements produced by program PFECTL with the theoretical values, (b) a comparison with Womersley's analytical solution for a sinusoidal pressure cycle and the three-dimensional finite element solution mentioned above and (c) a comparison with the time history of wall displacements produced by a finite volume code using compliance data obtained in vivo from a porcine common carotid artery.

6.6.2.1 Wall displacements

The displacement of a thick cylinder wall can be determined using the Lamé equations for radial and circumferential stress, \( \sigma_r \) and \( \sigma_\theta \), respectively,

\[
\sigma_\theta = \frac{a^2 p}{(b^2 - a^2)} \left( 1 + \frac{b^2}{r^2} \right), \quad \sigma_r = \frac{a^2 p}{(b^2 - a^2)} \left( 1 - \frac{b^2}{r^2} \right)
\]

(6.5)

where the constants \( a \) and \( b \) are the inner and outer cylinder radii, respectively, \( p \) is the pressure differential across the cylinder wall and \( r \) is the general radius. The displacements in the radial and axial directions, \( u \) and \( w \), may then be found from the stresses using

\[
u = \frac{r}{E} \left( \sigma_\theta - \nu (\sigma_r + \sigma_x) \right)
\]

(6.6)

\[
w = \frac{z}{E} \left( \sigma_z - \nu (\sigma_r + \sigma_\theta) \right)
\]

The theoretical and finite element displacements are plotted in Fig 6.42 based on the wall properties quoted in Xu and Collins (1995) for the cases of free and fixed tube ends, i.e., tube bore 10 mm, thickness 1.0 mm, Young's modulus, \( E = 0.5 \) MPa, Pressure \( P = 2668 \) Pa.
Fig 6.42 Comparison between theoretical thick cylinder radial displacement and finite element solutions

It can be seen that good agreement is obtained for both free and fixed end cases and the results are comparable with those obtained by Xu and Collins. The underestimate of displacement is characteristic of the finite element method which tends to produce overstiff models. The quadratic interpolation used in the finite element is only capable of representing a linear stress distribution while the circumferential stress is a function of $1/r^2$. The use of four elements in the radial direction is shown by the results to be justified.
6.6.2.2 Sinusoidal flow in an elastic cylinder

Womersley (1957) gives the axial and radial fluid velocities in an elastic tube as

\[
\begin{align*}
  w &= \frac{A}{\rho} \frac{1}{\im} \left\{ 1 + \eta \frac{J_1}{J_0} \left( \frac{\alpha y i^{3/2}}{\alpha i^{3/2}} \right) \right\} e^{\im t} \\
  u &= \frac{A R}{\rho} \frac{2J_0}{2c} \left\{ \eta \frac{2J_1}{\alpha i^{3/2} J_0} \left( \frac{\alpha y i^{3/2}}{\alpha i^{3/2}} \right) + y \right\} e^{\im t}
\end{align*}
\]

where \( A \) is the amplitude of pressure gradient, \( y = r/R \), \( J_0 \) and \( J_1 \) are Bessel functions of the first kind, order zero and one, \( c \) is the wave speed, \( n \) is the circular frequency and \( \eta \) is a constant determined by the problem boundary conditions and material properties.

The same cycle was used as by Xu and Collins (1995), i.e.,

\[
P = 100 \cos \left( \frac{2\pi t}{0.75} \right)
\]

100 time steps were taken and a single cycle took 7 hours to run. Since the motion of the tube wall is effectively damped by the fluid, the Rayleigh damping coefficients, \( \alpha \) and \( \beta \), in the code were chosen to give full damping.

The results are plotted in Fig 6.43 for phase angles 0°, 90°, 180° and 270°.
It can be seen that the agreement between the finite element and Womersley solutions is not as good as that obtained by Xu and Collins, where the differences are symmetrical about the axis. It could therefore be argued in their case that a phase difference between the theoretical and finite element solutions contributes to the differences with the theory and could be due to the length of time step. Here, however, there is a difference between forward and reverse flow velocities which must be attributed to the finite element formulation used for the wall-fluid interaction.

6.6.2.3 Finite element predictions vs in vivo displacement measurements.
Xu and Collins (1993) compare their finite volume predictions for radial wall displacements in a porcine common carotid artery with values measured in vivo ultrasonically. Changes in the compliance of the porcine artery were obtained by infusion of a vasodilator as part of a study on the link between arterial wall shear stress and the release of a locally active vasodilator agent from the endothelium. The finite volume analysis was carried out to predict wall displacements on the basis of measured inlet flows, downstream pressures and wall compliance.
In the present study, in order to make a comparison between the measured values of tube diameter and the finite element prediction, the compliance was incorporated into the code by expressing the modulus of elasticity at each time step in terms of the compliance, an assumed tube thickness and the current tube diameter. Thin-walled tube behaviour was assumed. The pressures and compliances at each time step were read into the program with the input data. A simple axisymmetrical finite element model with a single radial column of 20 fluid elements and four wall elements was used and ten second time steps were used to cover the 300 second period.

The finite element solution is plotted against the \textit{in vivo} measurements in Fig 6.44. The finite element solution is less accurate than the finite volume solution, underestimating the displacement during the earlier time steps, while overestimating displacement after the large change in compliance at 200 seconds. This behaviour is shared with that of the finite volume solution which also overestimates displacement during the later part of the solution. In view of the relatively simple wall-fluid interaction method used and the potential of either instability or divergence occurring over such a long time period, the differences between the prediction and the measured diameters is considered to be reasonable.

![Graph](image)

\textbf{Fig 6.44} Comparison between tube diameters measured \textit{in vivo}, (Xu And Collins, 1993) and PFECTL code finite element predictions.
7 DISCUSSION

7.1 Fully Developed Flow

Comparison between the results from the finite difference analysis of Hamakiotes and Berger (1988, 1990) and the results from the finite element program, PFECTX, show good agreement, both quantitatively and qualitatively, for the fully developed flow analysis for $Re_m = 200$ and $\alpha = 15$. There are, however, significant differences in the results for $Re_m = 375$, and $\alpha = 15$. In particular the four vortex, Lyne type flow predicted by Hamakiotes and Berger was not predicted by the finite element program. The results of Sudo et al. (1992) (Fig 6.10) indicate that at $Re_m = 375 (D = 142)$ and $\alpha = 15$, the flow pattern lies on the borderline between Dean flow and Lyne flow. Increasing the finite element mesh refinement to give approximately the same number of computational points as that of Hamakiotes and Berger and increasing the number of time steps to 48 produced no appreciable change in the flow pattern. Furthermore, increasing the frequency to give a value of $\alpha = 18$ to ensure that the flow was well within the Lyne region of Fig 6.14 also achieved very little improvement.

The most significant difference between the two analyses, apart from the obvious difference between the finite difference and finite element techniques, is in the number of time steps taken, Hamakiotes and Berger taking between 240 and 400 steps for the pressure cycle. The absence of the second vortex in the half-section is then most probably due to the length of time step taken by the finite element program. Since the development of the second vortex is apparently short-lived, although Sudo et al. (1992) have recorded such a vortex with their flow visualisation technique, it may be that the time step in the finite element analysis is too large and masks the formation of the second vortex. The physical implications of this are that in real flows the development of these and other short-lived flow regimes, particularly the complex changes in direction of secondary flow described by Chandran et al. (1974), may not develop, due to naturally occurring disturbances in the flows.

It is interesting in view of the above that for the solutions carried out to compare the results for unstructured and structured meshes, for Reynolds and Womersley numbers of 325 and 11.4, respectively, only the structured mesh produced a second vortex. The other meshes produced results similar to those obtained beforehand. This implies that the form of the mesh may also have some influence on the formation of the second vortex. This implication, however, is contrary to the general assumption that finite element meshes, unlike finite difference meshes, have little influence, providing the mesh is properly constructed, i.e., with no sudden changes in element size or shape, and element size is adequately small locally, on the solution. Furthermore the
fully unstructured mesh used has no preferred orientation or areas of refinement and would seem to be more suitable for the secondary flow problem where no area of the tube cross section more important in terms of complexity of flow than any other.

Ideally the mesh refinement should have been increased and the time step reduced to investigate the problem further, but the number of time steps required (> 200) could not be achieved with the current finite element analysis because of the computer operational constraints produced in an open access system. Sharing use of the computer could mean, for example, that either the ratio of real time to processing time for a given job run could increase considerably, or that the storage required by the users at any one time could exceed the available storage space. If that occurs the necessity of 'size swapping', or repetitively downloading files to ancillary storage between operations, would then result in prohibitive run times.

It is instructive to put the quality of the results obtained with the finite element analysis in context with the relative computer power, in terms of computational times, available to the author and to Hamakiotes and Berger. If it is assumed, as suggested above, that the differences between the results of the two analyses are primarily due to the number of time steps used, then using the same number of computation points and time steps should achieve comparable results. Hamakiotes and Berger used non-uniform finite difference grids of 266 points (14 radially and 19 circumferentially) with 400 time steps per cycle of 1 second period, requiring 3 min of CPU time per cycle. In comparison, with a mesh of 260 nodes, the finite element analysis took 65 min per time step, also of 1 second period. To achieve only 200 time steps with this cycle time would take 9 days of CPU time, almost 4500 times as long as the finite difference solution.

As an exercise, both codes were run on a Fujitsu VPX vector processor, with potentially orders of magnitude more power than the Hewlett-Packard 9000/700 used in the study. It was found that, because of the use of the frontal solution and the necessity of reading from and writing to disk at every solution step, the input-output time became the dominant factor and the total run times were no shorter.

It is very clear that the frontal solution, at least in the form used here, is far too time-consuming and puts a limit of about 250 on the number of elements that can be used. To avoid ill-conditioning of the finite element equations it was necessary to specify large front widths, i.e. to assemble a large number of equations before eliminating, any increase in front width resulting in disproportionately longer run times. This indicates that the pivot choice was very critical and that a large choice of pivots was therefore essential. At the same time convergence, with differences in solutions of the fourth significant figure, was always obtained within four or five iterations.
7.2 Semi-analytical Solution

The axial velocity profiles for steady flow agree well qualitatively with hot-wire anemometry, but changes in the entrance flow occur slightly more quickly than predicted, although the results are markedly improved on the results obtained earlier in the present study (not reported here). The secondary velocity vectors for the cross-sections close to the entrance also agree qualitatively with analytical predictions by Singh (1974) and the finite element model of Perktold et al. (1991). The oscillating flow analysis shows good agreement with measured results, but with some smoothing of the velocity profiles. This is attributed to the inability of the semi-analytical solution to produce the high orders of curvature shown in the measured profiles. This is in contradiction with the Balachandran (1972) analysis which showed that the higher harmonics were unnecessary.

It can be seen from Fig 6.23 that the secondary flows obtained with the semi-analytical solution show the vortex centre on the vertical centreline, whereas the experimental evidence is that the vortex appears at around the 135° position. There then seems to be some doubt about the ability of the semi-analytical solution to model the transverse flows correctly.

It is a relatively simple exercise to determine the effect of the use of only one curved tube term in the analysis on the position of the vortex. At the vortex centre the radial and tangential velocities will be zero. i.e.,

\[ u_{10} + u_{11} \cos \theta = 0 \quad \text{and} \quad -v_{11} \sin \theta = 0 \]  

(7.1)

Since, in developing flow, there is no reason why the curved tube component of the radial velocity, \( u_{11} \), may not be of opposite sign to the straight tube component, \( u_{10} \), then if \( v_{11} = 0 \), there is no restriction on the position of the vortex centre for this flow regime. The position will depend only on the relative magnitudes of \( u_{10} \) and \( u_{11} \). In developed flow, however, the straight tube component of radial velocity will be zero, so that at the vortex centre both \( u_{11} \cos \theta \) and \( -v_{11} \sin \theta \) must be zero. Discounting the option \( \theta = 0 \), because symmetry precludes vortices with centres on the horizontal centreline, either

(a) \( u_{11} \) and \( v_{11} \) are both zero and the value of \( \theta \) is indeterminate,

or,

(b) \( v_{11} \) is zero, \( \theta \) is 90° and \( u_{11} \) can have any value.

Alternatively, it can be assumed that the resultant velocity will be a minimum at the centre of a numerically modelled vortex; not necessarily zero, because there may not
be a node located at the vortex centre. In this case the resultant velocity, $v_r$, given by
\[ v_r^2 = u_{11}^2 \cos^2 \theta + v_{11}^2 \sin^2 \theta \] (7.2)
may be minimised, so that
\[
\frac{d v_r^2}{d \theta} = u_{11}^2 (-2 \sin \theta \cos \theta) + v_{11}^2 (2 \sin \theta \cos \theta) = 2 \left( v_{11}^2 - u_{11}^2 \right) \sin \theta \cos \theta = 0
\]
and either $v_{11}^2 = u_{11}^2$ and $\theta$ is indeterminate, or $\sin \theta \cos \theta = 0$ and $\theta = 90^\circ$.

The occurrence of the secondary vortex in developed flows with Womersley numbers of around 15 and above requires $u_{11} \cos \theta$ and $-v_{11} \sin \theta$ to be zero at two different values of $r$ and $\theta$, so that
\[
\begin{align*}
  u_{11}(r_1) \cos \theta_1 &= 0, & -v_{11}(r_1) \sin \theta_1 &= 0 \\
  u_{11}(r_2) \cos \theta_2 &= 0, & -v_{11}(r_2) \sin \theta_2 &= 0
\end{align*}
\] (7.3)
and again $\theta$ is either indeterminate or $90^\circ$ for both vortices.

The semi-analytical solution with one term in the curved tube component is therefore only capable of modelling the observed vortex position in developing flow, although it may be capable of modelling developed flow at Womersley numbers above 20 when the observed vortex position is at $90^\circ$. Although in principle the semi-analytical solution can model the position of the vortex in developing flow correctly, there is a serious question of the validity of the analysis if it is unable to model the fully developed condition.

The question arises of how useful the solution is if the vortex is offset from the true position. In the fully developed solution (PFECTX), the presence of the second vortex has little influence on the velocity gradients at the wall, but the position of the first vortex is crucial in positioning the high wall shear areas. In the developing flow solution the secondary vortex probably does not occur, certainly Singh (1974) did not consider this, in which case the semi-analytical solution may well be useful.

Using the same approach as in stress analysis by the semi-analytical solution, the answer to the present limitations of the analysis, due to the use of only one term in the curved tube component, would simply be to use more harmonics. At present one term leads to a straightforward linear solution where straight tube solution axial velocities are used to drive the curved tube solution. If more terms are incorporated, however, the amplitudes of the additional terms are unknown and the linearity is lost. A more complex approach would then be needed to deal with the unknown amplitudes and the
computational advantage achieved here, at least in principle, of producing a two-dimensional solution to a three-dimensional problem would also be lost.

The determination of entry flow numerically is not easily achieved, as demonstrated by the number of analyses carried out for this particular problem (Ward-Smith, 1980). The mesh for both finite element models was chosen to maintain reasonable aspect ratios for the finite elements, at the expense of the length of tube modelled. It would be expected that a more finely graded mesh adjacent to the entrance would improve the results and this was shown.

The most difficult aspect of the earlier semi-analytical formulation attempted was to justify the relationship between the axial velocity and the radial and tangential velocities. When it was assumed that a simple relationship held the solutions were generally unreliable. This problem does not occur in stress analysis, where the relationships between the variables have a simpler form. The separation of the straight and curved tube flows fortunately removed the necessity for finding such a relation.

7.3 Elastic Walls
7.3.1 Curved Tube
The curved tube elastic wall analysis was carried out for a range of wall stiffness and its performance was found to be dependent, not unexpectedly, on the length of time step used. The use of large time steps with physiologically realistic wall stiffness caused failures due to ill-conditioning, while at the same time the limitations on the number of time steps that could be used made large time steps essential. In practice the modulus of elasticity was chosen to produce a given maximum wall deflection and this proved to be a reliable technique.

The results of the curved tube elastic wall analyses are interesting in that the radial pressure is apparently high enough to cause distortion of the cross section of the tube. While, intuitively, mercury flowing through a thin rubber tube would be expected to produce large distortions of the tube cross-section, blood might not be expected to produce appreciable distortions in arteries. Whether or not the results are realistic, in view of the coarse modelling of the tube wall, the large time steps used and the effect on fluid velocities and local pressure gradients of large distortions of the tube, is still to be resolved. The lack of suitable published data on wall distensions to compare with the finite element results leaves this question open.

It is reported (McDonald, 1974) that radial distensions of artery walls probably have little effect on flow patterns compared with movements of the artery involving changes in the flow path. The tube distortions predicted by the finite element analysis
would have just this effect, since the distortions effectively move the tube centre line sideways.

7.3.2 Straight Tube

The straight tube analyses for linear and non-linear wall behaviour are very useful in confirming that the fluid-wall interaction algorithm gives results of acceptable accuracy. Static deflections of the tube wall proved to be in close agreement with theoretical predictions, indicating that the wall model, with four quadratic elements in the radial direction, was adequate. This would be expected since the problem is a relatively trivial one in finite element stress analysis. Increasing the number of wall elements in the radial direction merely increased the aspect ratio of the elements, cancelling out any improvements made by increasing the degrees of freedom through the tube wall.

The solution for the sinusoidal pressure cycle proved to be less accurate than the solution obtained with the coupled solution, but it is encouraging that comparable results can be obtained with the simpler, uncoupled approach to the problem. The incorporation of measured compliances into the code was relatively straightforward and again produced encouraging results when compared with those obtained with a coupled analysis.

It is instructive to examine the assumptions made in the analysis with a view to discovering the limitations of the uncoupled approach to the fluid-wall interaction problem. The basic assumption made in the analysis is that fluid-wall interaction can be predicted with an uncoupled approach, i.e., the fluid and wall responses to an applied fluid (axial) pressure cycle are calculated using two separate, alternating solutions. The analysis proceeds by alternately calculating fluid velocities and pressures and wall responses. No account is taken of fluid dynamic pressure as this is assumed to be negligible.

The response of the wall is based on the compliance and fluid pressure at the beginning of the wall solution step; the fluid flow solution is then based on the wall configuration at the end of the step. The wall properties are linearised over each step and in the present analysis a tangent stiffness approach is used, where the non-linear wall stiffness is assumed to be linear over an increment of displacement or pressure and the pressure is applied incrementally. The wall response is then an incremental change in the current diameter. The size of the pressure increment in a sinusoidal pressure cycle is dependant on the length of time step chosen and the two are therefore linked.

The tangent stiffness approach is satisfactory only if the wall movement during each step is small, i.e., the differences between the calculated and theoretical value at the beginning of each step are small enough not to cause divergence. The corresponding
changes in fluid flow will then also be small and the accuracy of the solutions during each step should be satisfactory. This approach tends to produce overstiff solutions in cases where the slope of the pressure:wall stiffness curve is decreasing with pressure and understiff solutions where the slope is increasing. As a check on the behaviour of the linearised wall stiffness and the *smallness* of the step, a plot of the stiffness used at the beginning of each step against the theoretical value could be made.

The static response of the vessel wall is based on thick-cylinder theory, which is a well established theory in stress analysis and, as shown, the finite element results can easily be checked against the theoretical values. This aspect of the analysis is therefore soundly based. The dynamic response of the vessel wall is based on the assumption of a constant, average wall acceleration over a time step in the Newmark method. The accuracy of the Newmark method is therefore dependent on the size of the time step. The method itself, however, provides a well-established approach to dynamic response and it can be stated that, providing the time step is adequately small, this aspect of the analysis is also soundly based.

The problems arising from a restricted number of time steps are exemplified by the wall response shown in Fig 6.43 where it can be seen that not only are there errors in the finite element solution, compared with the Womersley prediction of wall displacement, but the response to the positive and negative pressure half-cycles is unsymmetric, implying that the calculated wall stiffness is different during these half cycles.

With adequate computer capacity to use small time steps the performance of the analysis then depends on the ability of the fluid flow analysis to produce accurate pressures. In the straight tube problem this is straightforward and the wall pressures follow the applied pressure. In the curved tube, however, the difficulty of calculating fluid pressures becomes the main source of error.

The results of the two fluid-wall analyses carried out show that, in a qualitative sense, wall-response solutions are possible with the present uncoupled method. Qualitative, here, means that although the results from the two analyses currently show the same trends as the theoretical and experimental results, respectively, the numerical correlation has limitations. The indications are that an improvement in time step size will improve the accuracy of the solution, however, which infers that the uncoupled solution is a valid approach to the fluid-wall interaction problem.

### 7.4 Future Work

Sedlář's work (Sedlář, 1993) demonstrates that a quasi-three-dimensional approach can be very efficient in computer usage. His use of a more sophisticated scheme
than the Gauss elimination-based frontal solution use here, however, is crucial to his success in being able to run complex fluid flow problems on a personal computer. The use of such solution techniques as the conjugent gradient technique is well established in fluid dynamics and provides the obvious way forward for the present study. If this technique were to be implemented then the future development of both the current fully-developed flow and semi-analytical programs can take three directions:

(a) to remain with the current computing facility, but improve the solution technique to enable much shorter run times with more degrees of freedom

(b) to run the current program on a faster computer. This would be a particularly interesting exercise, enabling the full potential of the two-dimensional cross-section solution to be realised, or

(c) to recast the program in three-dimensional coordinates and run this on a vector processor or similar advanced computer system.

More specifically, it would be useful to develop the fluid-wall algorithm further, to take into account a more realistic model of the tube wall. The inclusion of the three layers of the artery with their different properties would be feasible and the ability to represent material properties in the form of a look-up table has been proved. This promises more success than the difficult problem of producing formulae to represent real material properties. It would also be interesting to repeat the analyses of Hamakiotes and Berger, and others. If it is assumed that some problems are specific to the solution technique, the use of another technique would give a useful comparison. In particular the development of the Lyne flow vortex and the bifurcation of flow regimes could be further explored.

Regarding the semi-analytical technique, the prediction of realistic flows has been obtained, but there is some doubt about the ability of the present code to model complex flows where velocity profiles, for example, have greater curvature than can be modelled with one harmonic. The use of another harmonic in the solution would involve an additional solution at each time step. The use of a semi-analytical technique, where a three-dimensional problem can be solved by a two-dimensional analysis is extremely efficient in use of computer power, but clearly this advantage would be lost if the solution required several solutions at each time step.

On the experimental front, work is in progress to develop a particle-tracking technique suitable for analysing general curved tube flow and aortic flow in particular. This technique promises an improvement on the current flow visualisation techniques reported in the literature on curved tube flow, to the extent where it should be possible to obtain more reliable wall shear rate data. The need to support and complement this requires a reliable numerical technique and future use of the programs developed in the present study will be closely related to this experimental work.
The importance of a capability to predict changing flow patterns due to the movement of curved tube axes under pressure is suggested and, together with the lack of suitable wall distension data for elastic tubes in the literature, indicates another area where an experimental approach could be used to support numerical analyses.

The aorta, with its curvature, torsion, taper and arterial branches, presents a complex problem to model either physically or numerically. The dimensions and wall mechanical properties vary significantly from person to person and also during the cardiac cycle. In view of the changes that occur when any part is removed from the body, size and shape must be determined \textit{in vivo}. The wall mechanical properties are particularly difficult to determine. In addition to the problem of modelling the aorta itself, the blood flow through it is pulsatile, sometimes turbulent, non-Newtonian and, throughout most of the cardiac cycle, entry flow. A complete solution is therefore difficult to obtain without access to powerful computer resources, unless simplifications are made. The complexity of the aortic blood flow problem is reflected in the computer power required to carry out analyses, as in the present study, involving even quite gross simplifications.
CONCLUSIONS

The intention at the outset was to determine whether or not it is feasible to model the complex three-dimensional curved tube problem, of general utility in arterial flows, and specifically in aortic flows, using two-dimensional finite element analyses with limited computer power.

Computer programs have been developed for incompressible, viscous, laminar, Newtonian flow in curved tubes, based on finite element models of two-dimensional sections through the tube. Solutions have been obtained for all three areas of interest considered: fully-developed flow, entry flow and fluid-wall interaction. The solutions generally compare well with experimental data and other numerical results, but within the limited range of application available in view of the computational constraints. The semi-analytical formulation adopted, however, appears to over-constrain the solutions for transverse velocities.

The basic algorithm for unsteady, incompressible, viscous flow has, however, been validated by comparison with an analytical solution. Development of the programs to provide a full three-dimensional capability for the analysis of general unsteady, incompressible, viscous flows would require no further development of the basic algorithm.

The results have demonstrated the potential of a two-dimensional approach to solving difficult flow problems. Interesting results have been obtained and the study has provided the incentive, as well as a very useful basis, for future work in arterial flows.
NOMENCLATURE

All units are S I standard units unless stated otherwise

\( a \)  
Tube radius

\( a_0 \)  
Tube wall displacement

\( b \)  
Non-dimensional tube radial coordinate, \( b = \frac{r}{R_o} \)

\( \{C\} \)  
Matrix of convective terms

\( De \)  
Dean Number, \( De = \text{Re} \sqrt{\frac{a}{R_o}} \)

\( Di \)  
Tube wall distensibility

\( \{D\} \)  
Tube wall stress-strain matrix

\( E \)  
Modulus of elasticity

\( h \)  
Non-dimensional radius of curvature to a point, \( h = \frac{\bar{r}}{R_o} \)

\( \{K\} \)  
Finite element characteristic matrix

\( \{L\} \)  
Matrix of divergence terms

\( \{M\} \)  
Mass matrix

\( N^u, N^v, N^w, N^p \)  
Velocity and pressure finite element shape functions

\( p \)  
Transverse component of fluid pressure

\( \bar{p} \)  
Axial (average) component of fluid pressure

\( r \)  
Tube radial coordinate

\( \bar{r} \)  
Tube radius of curvature to a point, \( \bar{r} = R_o + r \cos \theta \)

\( Re \)  
Reynolds number, \( Re = \frac{2 W_o a}{v} \)
<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \text{Re}_m )</td>
<td>mean Reynolds number, ( \frac{a W_o}{v} )</td>
</tr>
<tr>
<td>( R_s )</td>
<td>secondary Reynolds number, ( \frac{W_o^2}{2 \omega v} )</td>
</tr>
<tr>
<td>( R_O )</td>
<td>tube radius of curvature</td>
</tr>
<tr>
<td>( u, v, w )</td>
<td>radial, tangential and axial velocities</td>
</tr>
<tr>
<td>( W_o )</td>
<td>mean axial velocity</td>
</tr>
</tbody>
</table>

**Greek letters**

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \alpha )</td>
<td>Womersley frequency parameter, ( \alpha \sqrt{\frac{\omega}{\nu}} ), Rayleigh damping coefficient</td>
</tr>
<tr>
<td>( \beta )</td>
<td>Lyne frequency parameter, ( \sqrt{\frac{2 \nu}{\omega a^2}} ), Rayleigh damping coefficient</td>
</tr>
<tr>
<td>( \delta )</td>
<td>Curvature ratio, ( \frac{a}{R_o} )</td>
</tr>
<tr>
<td>( \varepsilon )</td>
<td>tube wall strain</td>
</tr>
<tr>
<td>( \kappa )</td>
<td>Alternative Dean number, ( 2 \cdot De )</td>
</tr>
<tr>
<td>( \mu )</td>
<td>dynamic viscosity,</td>
</tr>
<tr>
<td>( \nu )</td>
<td>kinematic viscosity</td>
</tr>
<tr>
<td>( \nu_w )</td>
<td>tube wall Poisson's ratio</td>
</tr>
<tr>
<td>( \theta )</td>
<td>toroidal coordinate</td>
</tr>
<tr>
<td>( \rho )</td>
<td>fluid density</td>
</tr>
<tr>
<td>( \rho_w )</td>
<td>wall density</td>
</tr>
<tr>
<td>( \sigma )</td>
<td>stress</td>
</tr>
<tr>
<td>( \bar{\sigma} )</td>
<td>fluid surface traction</td>
</tr>
</tbody>
</table>
\( \tau \) fluid stress tensor

\( \omega \) pressure cycle angular frequency

\( \xi, \eta \) normalised finite element coordinates

\( \eta \) Womersley velocity coefficient
REFERENCES


Ref 1


Ref2


Collins, M. W. (1995) It looks very nice Holmes, but is it any use? The Walmesley Lecture, Research in Conventional and Biomedical Engineering, City University


Ref 3


Ref 4


Ref 5


Ref 6


Ref 7


Ref 8


Ref 10


Ref 12


Ref 13


Ref 14


Ref 15


Womersley, J. R. (1955) Method for the Calculation of Velocity, Rate of Flow and Viscous Drag in Arteries when the Pressure Gradient is Known, *J. Physiol.*, v 127, pp 553-563

Ref 16


Ref 17


APPENDIX A
Gartling-Becker Finite Element Formulation

The fluid motion is assumed to be laminar, steady, isothermal and incompressible. The following derivation is based on a two-dimensional domain, the axisymmetric case follows by analogy.

A.1 Basic Equations

The equations of motion are

\[ \rho u_j \frac{\partial u_i}{\partial x_j} = \rho f_i + \frac{\partial}{\partial x_j} \tau_{ij} \]  \hspace{1cm} (A.1)

where

\[ \tau_{ij} = -p \delta_{ij} + \mu \left( \frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) \]  \hspace{1cm} (A.2)

\( u_i \) is the velocity component in the \( x_i \) coordinate direction, \( \rho \) is the fluid density, \( \mu \), the dynamic viscosity, \( f_i \), the body force vector, \( p \), the pressure, \( \tau_{ij} \), the stress tensor and \( \delta_{ij} \), the Kronecker delta.

with the incompressibility constraint

\[ \frac{\partial u_i}{\partial x_i} = 0 \]  \hspace{1cm} (A.3)

and boundary conditions given by the surface traction vector \( \tau_{ij} = \tau_{ij} \hat{n} \) and/or \( u_i \) specified on the boundary.

The equation of mechanical energy is formed by the scalar product of velocity with Eqtn A.1 and integrating over the volume,

\[ \int_V \rho u_i u_j \frac{\partial u_i}{\partial x_j} dV = \int_V \rho f_i u_i dV + \int_V u_i \frac{\partial \tau_{ij}}{\partial x_j} dV \]  \hspace{1cm} (A.4)
which relates the change of kinetic energy to the rate of work done by the body forces and stress fields. Integrating by parts and using the divergence theorem on the last term in Eqtn A.4 gives the equation

\[ \int_{V} \rho u_{i} \frac{\partial u_{i}}{\partial x_{j}} \, dV + \int_{V} \tau_{ij} \frac{\partial u_{i}}{\partial x_{j}} \, dV = \int_{V} f_{i} \, u_{i} \, dV + \int_{A} \tau_{ij} u_{i} \, n_{j} \, dA \]  

\[ (A.5) \]

The work done by the pressure resulting from a change in fluid volume must be zero, for an incompressible fluid so that

\[ \int_{V} p \frac{\partial u_{i}}{\partial x_{j}} \, dV = 0 \]

\[ (A.6) \]

The above equations are similar to those obtained from the Galerkin formulation with weight functions given by \( u_{i} \) and \( p \).

**A.2 Finite Element Formulation**

Within each finite element the values of the unknown variables are interpolated among the nodal values by the shape functions so that

\[ u_{i}(x_{i}) = N^{u} u_{i}^{e}, \quad p(x_{i}) = N^{p} p_{i}^{e} \]

\[ (A.7) \]

where \( u_{i}^{e}, \, p_{i}^{e} \) are the nodal values and \( N^{u} \) and \( N^{p} \) are the shape functions.

Substituting Eqtns A.7 into Eqtns A.5 and A.6, leads to

\[ \int_{V} \rho u_{i}^{T} N^{u} N^{u^{T}} u_{i} \frac{\partial N^{u}}{\partial x_{j}} \, dV - \int_{V} u_{i}^{T} \frac{\partial N^{u}}{\partial x_{j}} N^{p} \, p \, dV + \int_{V} \mu \left( u_{i}^{T} \frac{\partial N^{u}}{\partial x_{j}} \frac{\partial N^{u^{T}}}{\partial x_{j}} + u_{i}^{T} \frac{\partial N^{u}}{\partial x_{j}} \frac{\partial N^{u^{T}}}{\partial x_{j}} u_{j} \right) \, dV \]

\[ = \int_{V} \rho u_{i}^{T} N^{u} f_{i} \, dV + \int_{A} u_{i}^{T} N^{u} \tau_{ij} n_{j} \, dA \]

\[ (A.8) \]

\[ \int_{V} p^{T} N^{p} \frac{\partial N^{u}}{\partial x_{i}} u_{i} \, dV = 0 \]

\[ (A.9) \]
\[
\begin{bmatrix}
\int_{V} \rho N^u N^{uT} u_i \frac{\partial N^u}{\partial x_j} \, dV
\end{bmatrix}
\begin{bmatrix}
\frac{\partial N^u}{\partial x_j} dV
\end{bmatrix}
\begin{bmatrix}
\int_{V} \rho \frac{\partial N^u}{\partial x_j} \, dV
\end{bmatrix}
\begin{bmatrix}
\frac{\partial N^{uT}}{\partial x_j} dV
\end{bmatrix}
\begin{bmatrix}
0
\int_{A} N^u \tau_y n_j \, dA
\end{bmatrix}
\]
\begin{equation}
(A.10)
\end{equation}

\[
\begin{bmatrix}
\int_{V} N^p \frac{\partial N^p}{\partial x_j} \, dV
\end{bmatrix}
\begin{bmatrix}
u_i
\end{bmatrix}
= 0
\]
\begin{equation}
(A.11)
\end{equation}

Eqtns A.10 and A.11 are the finite element analogues to the equations of motion and the incompressibility condition, respectively. Specification of the shape functions \(N^u\) and \(N^p\) leads to the calculation of the terms in brackets, but additional interpolation is required for the right hand side terms. Typically the body forces are assumed constant over the each element and the boundary stress is approximated as

\[
\tau_y(s) = N^{rT} \tau'_y,
\]
where \(s\) is the coordinate along the element boundary, \(N^r\) is the shape function and \(\tau_{ij}\) the nodal values of stress.

Eqtns A.10 and A.11 are conveniently written in matrix form as

\[
\begin{bmatrix}
C_1(u_1) + C_2(u_2) & 0 & 0 & u_1 \\
0 & C_1(u_1) + C_2(u_2) & 0 & u_2 \\
0 & 0 & 0 & p
\end{bmatrix}
\begin{bmatrix}
u_1 \\
u_2 \\
p
\end{bmatrix}
= \begin{bmatrix}
2K_{11} + K_{22} & K_{12} \\
K_{21} & 2K_{22} + K_{11}
\end{bmatrix}
\begin{bmatrix}
u_1 \\
u_2 \\
p
\end{bmatrix}
= \begin{bmatrix}
R_1 \\
F_2
\end{bmatrix}
\]
\begin{equation}
(A.12)
\end{equation}

where

\[
C_i(u_i) = \int_{V} \rho N^u N^{uT} u_i \frac{\partial N^{uT}}{\partial x_i} \, dV
\]
\[
L_i = \int_{V} \frac{\partial N^u}{\partial x_i} N^p \, dV
\]
\[
K_y = \int_{V} \mu \frac{\partial N^u}{\partial x_i} \frac{\partial N^{uT}}{\partial x_j} \, dV
\]
\[
F_i = \int_{V} \rho N^u \, dV f_i + \int_{V} N^u N^{uT} n_j \, dA \tau_y
\]

A.3
APPENDIX B
Derivation of Equations of Motion in Toroidal Coordinates

In toroidal coordinates the element of arc length, $ds$, is given by (Fig B1)

$$ds^2 = dr^2 + r^2 \, d\theta^2 + h^2 \, dz^2$$  \hspace{1cm} (B.1)

so that the *scale factors* are

$$h_1 = 1$$
$$h_2 = r$$
$$h_3 = h$$  \hspace{1cm} (B.2)

Fig B1 Arc lengths
B.1 Continuity Equation

The Continuity Equation is given by

\[ \nabla \cdot \mathbf{V} = 0 \]

\[ = \frac{1}{h_1 h_2 h_3} \left\{ \frac{\partial}{\partial r} \left( h_1 h_2 u \right) + \frac{\partial}{\partial \theta} \left( h_1 h_3 v \right) + \frac{\partial}{\partial z} \left( h_2 w \right) \right\} \]

\[ = \frac{1}{rh} \left[ \frac{\partial}{\partial r} (rhu) + \frac{\partial}{\partial \theta} (hv) + \frac{\partial}{\partial z} (rw) \right] \]

\[ = \frac{1}{rh} \left[ hu + r \frac{\partial (hu)}{\partial r} + h \frac{\partial v}{\partial \theta} + v \frac{\partial h}{\partial \theta} + r \frac{\partial w}{\partial z} \right] \]

\[ \frac{u}{r} + \frac{1}{h} \frac{\partial (hu)}{\partial r} + \frac{1}{r} \frac{\partial v}{\partial \theta} + \frac{1}{h} \frac{\partial w}{\partial z} \]

\[ \frac{u}{r} + \frac{1}{h} \left( \frac{\cos \theta}{r_0} \right) u + \frac{1}{r} \frac{\partial u}{\partial \theta} + \frac{1}{r} \frac{\partial v}{\partial \theta} + \frac{v}{r} \left( \frac{\sin \theta}{r_0} \right) + \frac{1}{h} \frac{\partial w}{\partial z} \]

\[ \frac{u}{r} + \frac{\partial u}{\partial r} + \frac{1}{h} \frac{\partial w}{\partial z} + \frac{1}{r} \frac{\partial v}{\partial \theta} + \frac{\left( u \cos \theta - v \sin \theta \right)}{r} \]  

(B.3)

B.2 Momentum Equations

Neglecting gravitational body forces, the momentum equations are

\[ \rho \left[ \frac{\partial \mathbf{V}}{\partial t} + (\mathbf{V} \cdot \nabla) \mathbf{V} \right] = -\nabla p + \mu \left[ 2\nabla (\nabla \cdot \mathbf{V}) - \nabla \times (\nabla \times \mathbf{V}) \right] \]

(B.4)

Convective terms Diffusion terms
B.2.1 Convection Terms

Taking the first two terms of Eqtn B.4 and expanding:

\[ \frac{\partial V}{\partial t} = e_1 \frac{\partial u}{\partial t} + e_2 \frac{\partial v}{\partial t} + e_3 \frac{\partial w}{\partial t} \quad \text{(B.5)} \]

and

\[ (V \cdot \nabla) V = \nabla \left( \frac{V^2}{2} \right) - V \land \nabla \land V \quad \text{(B.6)} \]

where

\[ \left( \frac{V^2}{2} \right) = \frac{1}{2} (u^2 + v^2 + w^2) \quad \text{(B.7)} \]

hence

\[ \nabla \left( \frac{V^2}{2} \right) = e_1 \left( u \frac{\partial u}{\partial r} + v \frac{\partial v}{\partial r} + w \frac{\partial w}{\partial r} \right) + e_2 \left( u \frac{\partial u}{\partial \theta} + v \frac{\partial v}{\partial \theta} + w \frac{\partial w}{\partial \theta} \right) \]
\[ + \frac{e_3}{h} \left( u \frac{\partial u}{\partial z} + v \frac{\partial v}{\partial z} + w \frac{\partial w}{\partial z} \right) \quad \text{(B.8)} \]

From Eqtn B.6

\[ \nabla \land V = \frac{1}{h_1 h_2 h_3} \begin{vmatrix} h_1 e_1 & h_2 e_2 & h_3 e_3 \\ \frac{\partial}{\partial r} & \frac{\partial}{\partial \theta} & \frac{\partial}{\partial z} \\ h_1 u & h_2 v & h_3 w \end{vmatrix} \quad \text{(B.9)} \]
expanding the determinant leads to

$$
e = \frac{e_1}{r h} \left[ \frac{\partial(h w)}{\partial \theta} - \frac{\partial(r v)}{\partial z} \right] + \frac{e_2}{h} \left[ \frac{\partial u}{\partial z} - \frac{\partial(h w)}{\partial r} \right] + \frac{e_3}{r} \left[ \frac{\partial(r v)}{\partial r} - \frac{\partial u}{\partial \theta} \right]
$$

$$= e_1 \left( \frac{1}{r \partial \theta} \frac{\partial w}{\partial \theta} - \frac{1}{r \partial \theta} \frac{\partial v}{\partial \theta} \right) + e_2 \left( \frac{1}{h \partial z} \frac{\partial w}{\partial z} - \frac{w \partial h}{h \partial r} \right) + e_3 \left( \frac{\partial v}{\partial r} + \frac{v}{r} \frac{\partial u}{\partial \theta} \right)
$$

$$= e_1 A + e_2 B + e_3 C$$

(B.10)

hence

$$V \Lambda (\nabla \wedge V) = \begin{vmatrix} e_1 & e_2 & e_3 \\ u & v & w \\ A & B & C \end{vmatrix}
$$

$$= e_1 \left[ \frac{\partial v}{\partial r} + \frac{v^2}{r} \frac{\partial u}{\partial \theta} - \frac{w \partial u}{h \partial z} + \frac{w \partial w}{r \partial r} + \frac{w^2 \partial h}{h \partial r} \right]
$$

$$+ e_2 \left[ \frac{w \partial w}{r \partial \theta} + \frac{u^2 \partial h}{r \partial \theta} + \frac{w \partial v}{h \partial z} - \frac{u \partial v}{r \partial r} + \frac{uv \partial u}{r \partial \theta} \right]
$$

$$+ e_3 \left[ \frac{u \partial u}{h \partial z} - \frac{u \partial w}{h \partial r} + \frac{uw \partial h}{h \partial r} - \frac{v \partial w}{r \partial \theta} + \frac{vw \partial h}{h \partial r} + \frac{v \partial v}{h \partial z} \right]
$$

(B.11)

Subtracting Eqtn B.11 from Eqtn B.8 gives the convective terms

\[ (V \cdot \nabla) V = e_1 \left[ \frac{\partial u}{\partial r} + \frac{v \partial u}{r \partial \theta} - \frac{\partial \partial u}{r} + \frac{w \partial u}{h \partial z} - \frac{w^2 \partial \partial}{r \partial \theta} \cos \theta \right] \]

$$+ e_2 \left[ \frac{\partial v}{\partial r} + \frac{v \partial \partial v}{r \partial \theta} + \frac{w \partial v}{h \partial z} + \frac{uv \partial \partial}{r} \sin \theta \right]$$

$$+ e_3 \left[ \frac{w \partial w}{h \partial z} + \frac{u \partial w}{h \partial r} + \frac{v \partial w}{r \partial \theta} + \frac{w}{r} (u \cos \theta - v \sin \theta) \right]$$

(B.12)
B.2.2 Diffusion Terms

From the first term on the right hand side of Eqtn B.4

\[ \nabla \cdot \mathbf{V} = \frac{\partial u}{\partial r} + \frac{u}{r} + \frac{1}{r} \frac{\partial v}{\partial \theta} + \frac{1}{h} \frac{\partial w}{\partial z} + \frac{(ucos \theta - vsin \theta)}{r} \]  \hspace{1cm} (B.13)

Differentiating Eqtn B.13

\[ \vdash (\nabla \cdot \mathbf{V}) = \]

\[ = e_1 \left( \frac{\partial}{\partial r} \left( \frac{u}{r} + \frac{\partial u}{\partial r} + \frac{1}{h} \frac{\partial w}{\partial z} + \frac{1}{r} \frac{\partial v}{\partial \theta} + \frac{ucos \theta - vsin \theta}{r} \right) \right) \]

\[ + e_2 \left( \frac{\partial}{\partial \theta} \left( \frac{u}{r} + \frac{\partial u}{\partial r} + \frac{1}{h} \frac{\partial w}{\partial z} + \frac{1}{r} \frac{\partial v}{\partial \theta} + \frac{ucos \theta - vsin \theta}{r} \right) \right) \]

\[ + e_3 \left( \frac{\partial}{\partial z} \left( \frac{u}{r} + \frac{\partial u}{\partial r} + \frac{1}{h} \frac{\partial w}{\partial z} + \frac{1}{r} \frac{\partial v}{\partial \theta} + \frac{ucos \theta - vsin \theta}{r} \right) \right) \]
\[ e_1 \left[ \frac{u}{r^2} + \frac{1}{r} \frac{\partial u}{\partial r} + \frac{\partial^2 u}{\partial r^2} + \frac{1}{h} \frac{1}{\partial r} \frac{\partial w}{\partial z} + \frac{1}{r^2} \frac{\partial^2 w}{\partial \theta^2} - \frac{\cos \theta (u \cos \theta - v \sin \theta)}{r^2} + \frac{\cos \theta}{r} \left( \frac{1}{r} \frac{\partial u}{\partial \theta} - \frac{\sin \theta}{r} \frac{\partial v}{\partial \theta} \right) \right] \]

\[ + e_2 \left[ \frac{1}{r^2} \frac{\partial u}{\partial \theta} + \frac{1}{r} \frac{\partial^2 u}{\partial r \partial \theta} + \frac{1}{h} \frac{1}{\partial r} \frac{\partial w}{\partial z} + \frac{1}{r^2} \frac{\partial^2 w}{\partial \theta^2} + \frac{\sin \theta (u \cos \theta - v \sin \theta)}{r^2} + \frac{\cos \theta}{r} \left( \frac{1}{r} \frac{\partial u}{\partial \theta} - \frac{\sin \theta}{r} \frac{\partial v}{\partial \theta} \right) \right] \]

\[ + e_3 \left[ \frac{1}{h} \frac{\partial u}{\partial z} + \frac{1}{h} \frac{\partial^2 u}{\partial r \partial z} + \frac{1}{h} \frac{1}{\partial r} \frac{\partial w}{\partial z} + \frac{1}{h^2} \frac{\partial^2 w}{\partial \theta^2} + \frac{\partial u (\cos \theta) - \partial v (\sin \theta)}{r h} \right] \]

(B.14)

The second term on the right hand side of Eqtn B.4

\[ \nabla \wedge (\nabla \wedge V) = \frac{1}{h_1 h_2 h_3} \begin{vmatrix} h_1 e_1 & h_2 e_2 & h_3 e_3 \\ \frac{\partial}{\partial r} & \frac{\partial}{\partial \theta} & \frac{\partial}{\partial z} \\ h_1 A & h_2 B & h_3 C \end{vmatrix} \]

B.6
Expanding the determinant

\[ \nabla \wedge (\nabla \Lambda y') = e_1 \left( \frac{\partial}{\partial \theta} \left( \frac{\partial v}{\partial r} + \frac{\partial u}{\partial \theta} - \frac{1}{r} \frac{\partial u}{\partial \theta} \right) - \frac{\partial}{\partial z} \left( \frac{1}{h} \frac{\partial u}{\partial z} - \frac{\partial w}{\partial r} - \frac{w}{r} \frac{\partial h}{\partial r} \right) \right) \]

\[ -e_2 \left[ \frac{\partial}{\partial r} \left( \frac{\partial v}{\partial r} + \frac{\partial u}{\partial \theta} - \frac{1}{r} \frac{\partial u}{\partial \theta} \right) - \frac{\partial}{\partial z} \left( \frac{1}{r} \frac{\partial w}{\partial \theta} + \frac{w}{r} \frac{\partial h}{\partial \theta} - \frac{1}{h} \frac{\partial v}{\partial r} \right) \right] \]

\[ + e_3 \left[ \frac{\partial}{\partial r} \left( \frac{1}{h} \frac{\partial u}{\partial z} - \frac{w}{r} \frac{\partial h}{\partial r} \right) - \frac{\partial}{\partial z} \left( \frac{1}{r} \frac{\partial w}{\partial r} + \frac{w}{r} \frac{\partial h}{\partial \theta} - \frac{1}{h} \frac{\partial v}{\partial r} \right) \right] \]

\[ = e_1 \left[ \frac{1}{r} \frac{\partial^2 v}{\partial \theta \partial r} + \frac{1}{r^2} \frac{\partial v}{\partial \theta} - \frac{1}{r^2} \frac{\partial^2 u}{\partial \theta^2} - \frac{1}{h^2} \frac{\partial^2 u}{\partial z^2} + \frac{1}{h} \frac{\partial^2 w}{\partial r \partial z} \right. \]

\[ + \frac{\sin \theta}{r} \left( \frac{1}{r} \frac{\partial u}{\partial \theta} - \frac{\partial v}{\partial r} \right) + \frac{\cos \theta}{r} \left( \frac{1}{h} \frac{\partial v}{\partial z} \right) \]

\[ - e_2 \left[ \frac{\partial^2 v}{\partial r^2} - \frac{v}{r^2} + \frac{1}{r} \frac{\partial v}{\partial r} - \frac{1}{r} \frac{\partial^2 u}{\partial \theta r} + \frac{1}{r^2} \frac{\partial u}{\partial \theta} - \frac{1}{h} \frac{\partial^2 w}{\partial r \partial z} + \frac{1}{h^2} \frac{\partial^2 v}{\partial z^2} \right. \]

\[ + \frac{\cos \theta}{r} \left( - \frac{1}{r} \frac{\partial u}{\partial \theta} - \frac{\partial v}{\partial r} + \frac{\partial v}{\partial z} \right) + \frac{\sin \theta}{r} \left( \frac{1}{h} \frac{\partial w}{\partial z} \right) \]

\[ + e_3 \left[ \frac{1}{h} \frac{\partial u}{\partial r} - \frac{\partial^2 w}{\partial r \partial z} - \frac{1}{r} \frac{\partial w}{\partial r} + \frac{1}{r} \frac{\partial u}{\partial \theta} + \frac{1}{r} \frac{\partial^2 w}{\partial \theta^2} + \frac{1}{h^2} \frac{\partial^2 v}{\partial z^2} + \frac{w}{r} \right. \]

\[ - \frac{\cos \theta}{r} \left( \frac{1}{h} \frac{\partial u}{\partial z} + \frac{\partial w}{\partial r} - \frac{w}{r} \right) + \frac{\sin \theta}{r} \left( \frac{1}{h} \frac{\partial v}{\partial z} + \frac{\partial v}{\partial \theta} \right) \]

\[ \right] \]

(B.15)

Subtracting Eqn B.15 from twice Eqn B.14 yields the diffusion terms which, together with Eqtns B.5 and B.12, gives the momentum equations for each of the r, θ and z directions.

B.7
In the $r$-direction

\[
\rho \left[ \frac{\partial u}{\partial t} + u \frac{\partial u}{\partial r} + v \frac{\partial u}{\partial \theta} + w \frac{\partial u}{\partial z} + \frac{w^2}{r} \cos \theta \right]
\]

\[
= -\frac{\partial p}{\partial r} + \mu \left[ 2 \frac{\partial^2 u}{\partial r^2} + \frac{1}{r^2} \frac{\partial^2 u}{\partial \theta^2} + \frac{1}{r^2} \frac{\partial^2 u}{\partial z^2} - \frac{2}{r^2} \frac{\partial u}{\partial \theta} + \frac{2}{r^2} \frac{\partial u}{\partial z} \right. \\
+ \left. \frac{3}{r^2} \frac{\partial v}{\partial r} + \frac{1}{r^2} \frac{\partial^2 v}{\partial \theta^2} + \frac{1}{r^2} \frac{\partial^2 v}{\partial z^2} + \frac{1}{r^2} \frac{\partial^2 w}{\partial \theta^2} + \frac{1}{r^2} \frac{\partial^2 w}{\partial z^2} \right]
\]

\[
+ \left( \frac{\cos \theta}{r} \right) \left( \frac{2}{\partial r} \frac{\partial u}{\partial z} \right) + \left( \frac{\sin \theta}{r} \right) \left( -\frac{1}{r} \frac{\partial u}{\partial \theta} + \frac{v}{r} - \frac{\partial v}{\partial r} \right) - 2 \frac{\cos \theta (u \cos \theta - v \sin \theta)}{r^2}
\]

(B.16)

In the $\theta$-direction

\[
\rho \left[ \frac{\partial v}{\partial t} + u \frac{\partial v}{\partial r} + v \frac{\partial v}{\partial \theta} + w \frac{\partial v}{\partial z} + \frac{w^2}{r} \sin \theta \right]
\]

\[
= -\frac{1}{r} \frac{\partial p}{\partial \theta} + \mu \left[ \frac{\partial^2 v}{\partial r^2} + \frac{2}{r^2} \frac{\partial^2 v}{\partial \theta^2} + \frac{1}{r^2} \frac{\partial^2 v}{\partial z^2} + \frac{1}{r^2} \frac{\partial^2 v}{\partial \theta^2} + \frac{3}{r^2} \frac{\partial u}{\partial r} + \frac{1}{r^2} \frac{\partial^2 u}{\partial \theta^2} + \frac{1}{r^2} \frac{\partial^2 w}{\partial \theta^2} + \frac{1}{r^2} \frac{\partial^2 w}{\partial z^2} \right]
\]

\[
+ \left( \frac{\cos \theta}{r} \right) \left( \frac{1}{r} \frac{\partial u}{\partial \theta} + \frac{v}{r} - \frac{\partial v}{\partial r} \right) + \left( \frac{\sin \theta}{r} \right) \left( -\frac{1}{r} \frac{\partial u}{\partial \theta} + \frac{v}{r} - \frac{\partial v}{\partial r} \right) + 2 \frac{\sin \theta (u \cos \theta - v \sin \theta)}{r^2}
\]

(B.17)

In the $z$-direction

\[
\rho \left[ \frac{\partial w}{\partial t} + u \frac{\partial w}{\partial r} + v \frac{\partial w}{\partial \theta} + w \frac{\partial w}{\partial z} + \frac{w^2}{r} (u \cos \theta - v \sin \theta) \right]
\]

\[
= \frac{1}{h} \frac{\partial p}{\partial z} + \mu \left[ \frac{\partial^2 w}{\partial r^2} + \frac{1}{r} \frac{\partial^2 w}{\partial \theta^2} + \frac{1}{r} \frac{\partial^2 w}{\partial z^2} + \frac{2}{r^2} \frac{\partial^2 w}{\partial \theta^2} + \frac{1}{r^2} \frac{\partial^2 u}{\partial \theta^2} + \frac{1}{r^2} \frac{\partial^2 u}{\partial z^2} \right]
\]

\[
+ \frac{1}{r} \frac{\partial^2 v}{\partial \theta \partial z} + \left( \frac{\cos \theta}{r} \right) \left( 3 \frac{\partial u}{h \partial z} + \frac{\partial w}{h \partial r} \right) + \left( \frac{\sin \theta}{r} \right) \left( \frac{1}{r} \frac{\partial w}{\partial \theta} + \frac{3}{r} \frac{\partial v}{h \partial z} \right)
\]

(B.18)
APPENDIX C
Finite Element Program Listings

C.1 Program Description
The basic algorithm and this description are generally common to both the semi-analytical analysis, PFECTL, and the fully-developed flow analysis, PFECTX, but line numbers below are for PFECTX. (Note that input and output file names for PFECTX include a final X). The programs are written in FORTRAN 75. NAGFE subroutines (Greenough and Robinson, 1981) are included where appropriate to handle general vector and matrix operations, calculate shape functions and derivatives and carry out Gauss quadrature.

The subroutine FRONT controls the element assembly and carries out the frontal solution. A full description of the frontal solution routine is given in Hood (1973) and is not repeated here.

1 Following the specification of the input and output files, variable types, array sizes, data and common blocks the problem data consisting of nodal coordinates, element topology, boundary conditions, fluid and wall properties, tube curvature, iteration and time step parameters are all read in from MESH.DAT (lines 1-183).

2 The time step loop is initiated (line 202), calls subroutine FRONT (line 216) which, in turn, calls subroutine FLUIDL (line 479), the fluid element subroutine, assembles the required number of elements and then carries out the first solution (Stokes' Flow). This solution then provides the velocities for the convective terms in the next solution (lines 227-229).

3 The iteration loop (lines 209-246) continues until either the convergence criterion is met (line 243) or the specified maximum number of iterations (NITNS) has been carried out, when IEL is set to 1. The velocities and pressures are written to VELOUT.DAT (line 284).

The convergence criterion is

\[ \left| \frac{1}{n} \sum_{i=1}^{n} \left( \frac{sk(i) - chk(i)}{chk(i)} \right) \right| \leq 0.001 \]

where \( n \) = total number of fluid degrees of freedom, and \( sk(i) \) and \( chk(i) \) are the fluid variables for the current and previous iterations, respectively. This criterion has proved to be successful in that solutions for successive iterations changing only by the fourth significant figure are achieved within five iterations. This confirms Gartling's prediction (Gartling, 1976) that convergence within four to five iterations is usually obtained using the Picard scheme.

4 Subroutine FRONT calls subroutine SOLIDL (line 482), the wall element subroutine, and assembles the required number of wall elements. Pressures from the
fluid solution are used to determine the forcing vector in SOLIDL for the wall displacement solution (lines 1428-1458).

5 Subroutine FRONT then carries out the wall displacement solution. The wall displacements, velocities and accelerations are written to DEFLOUT.DAT (lines 326-328).

6 The coordinates for the wall in subroutine SOLIDL and the fluid in subroutine FLUIDL are updated using the wall displacements (lines 306-325) and IEL is set to 0.

7 The next time step is taken and steps 2 to 6 are repeated until the specified number of time steps (NSTEPS, NS) has been achieved.
Fig C.1 Flow chart for basic algorithm
### C.2 List of Variables

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>a0, da0, d2a0</td>
<td>current wall displacement, velocity and acceleration</td>
</tr>
<tr>
<td>a1, da1, d2a1</td>
<td>previous wall displacement, velocity and acceleration</td>
</tr>
<tr>
<td>amu</td>
<td>fluid dynamic viscosity</td>
</tr>
<tr>
<td>chk</td>
<td>solution for previous iteration</td>
</tr>
<tr>
<td>coord</td>
<td>node coordinates</td>
</tr>
<tr>
<td>deltim</td>
<td>time increment</td>
</tr>
<tr>
<td>dens</td>
<td>wall density</td>
</tr>
<tr>
<td>dtpd</td>
<td>element characteristic matrix velocity coefficient</td>
</tr>
<tr>
<td>elk</td>
<td>system characteristic matrix velocity coefficient</td>
</tr>
<tr>
<td>ell1,ell2,ell3</td>
<td>system characteristic matrix pressure coefficient</td>
</tr>
<tr>
<td>ellt1,ellt2,ellt3</td>
<td>system characteristic matrix pressure coefficient</td>
</tr>
<tr>
<td>elm</td>
<td>system mass matrix coefficient</td>
</tr>
<tr>
<td>eltop</td>
<td>element topology</td>
</tr>
<tr>
<td>emod</td>
<td>wall modulus of elasticity</td>
</tr>
<tr>
<td>fdof</td>
<td>total number of fluid dof</td>
</tr>
<tr>
<td>fels</td>
<td>total number of fluid elements</td>
</tr>
<tr>
<td>fun</td>
<td>shape function</td>
</tr>
<tr>
<td>funp, funv</td>
<td>shape function for pressure and velocity interpolations</td>
</tr>
<tr>
<td>gder</td>
<td>global derivative of shape function</td>
</tr>
<tr>
<td>gderp, gderv</td>
<td>global derivative of shape function for pressure and velocity</td>
</tr>
<tr>
<td>geom, geomp</td>
<td>element nodal coordinate vector</td>
</tr>
<tr>
<td>jac, jacin</td>
<td>Jacobian terms</td>
</tr>
<tr>
<td>gaus</td>
<td>number of Gauss quadrature points</td>
</tr>
<tr>
<td>abss</td>
<td>Gauss point location</td>
</tr>
<tr>
<td>wgght</td>
<td>Gauss point weighting</td>
</tr>
<tr>
<td>xi, eta</td>
<td>local coordinates</td>
</tr>
<tr>
<td>lder</td>
<td>local derivative of shape function</td>
</tr>
<tr>
<td>lderp, lderv</td>
<td>local derivative of shape function for pressure and velocity</td>
</tr>
<tr>
<td>velocity</td>
<td>nodes on wall surfaces</td>
</tr>
<tr>
<td>intnod, outnod</td>
<td>element mass</td>
</tr>
<tr>
<td>mass, mm1, mm2</td>
<td>boundary condition code</td>
</tr>
<tr>
<td>ncod</td>
<td>boundary condition code - fluid</td>
</tr>
<tr>
<td>ncodf</td>
<td>boundary condition code - solid</td>
</tr>
<tr>
<td>ndof</td>
<td>total number of dof</td>
</tr>
<tr>
<td>nel</td>
<td>element number</td>
</tr>
<tr>
<td>nits</td>
<td>number of (Picard) iterations</td>
</tr>
<tr>
<td>nff</td>
<td>dof number - fluid</td>
</tr>
<tr>
<td>nfs</td>
<td>dof number - solid</td>
</tr>
<tr>
<td>nodcod</td>
<td>corner node identifier</td>
</tr>
<tr>
<td>ns, nsteps</td>
<td>number of pressure steps</td>
</tr>
<tr>
<td>numint</td>
<td>number of interface nodes</td>
</tr>
<tr>
<td>numnod</td>
<td>node number</td>
</tr>
<tr>
<td>pinc</td>
<td>step pressure</td>
</tr>
<tr>
<td>poisrt</td>
<td>wall Poisson's ratio</td>
</tr>
</tbody>
</table>
rg
rbar
r0
r1
w0, wrad
p0, p1, p3
pocord
rho
sdo
sk
totels
totime	
totnod
uinc
vec1
vec2

Gauss point radius from tube centre
Gauss point radius from centre of curvature
tube radius of curvature
forcing vector term
wall polar coordinates
wall pressure loading
node polar coordinates
fluid density
number of dof - solid
solution for current iteration
number of elements
pressure cycle time
number of nodes
solutions identified for each timestep
dynamic contribution to forcing vector
pressure contribution to forcing vector
C.3 Program Listings

C.3.1 PFECTX

Program PFECTX - Pulsatile Flow in Elastic Curved Tubes - Cross Section
- Isoviscous, incompressible flow with inertia
- Newtonian fluid
- Toroidal coordinate system
- Curved Tube Cross-section

```
program pfectx
implicit double precision(a-h,o-z)
integer dofel,totdof,totels,fels,totncs, ij,jk,jl,lm,nwall, wallbc,hband
integer numint,intnod,outnod,centnod
integer bdcnd,blist,btype,dofnod,eltop,dimen, * dirich,totnod,bncode,bnode,ndof,nodsid,ichk,ll,ivect,ir1
* nelr,nelt,nf,nff,nfs,nodnum,istep,nsteps,icoord
* bncodf,bnodef,bncods,bnodes
* idof,fdof,sdof,neuman
* ,nout,nra,mwga
* ,iel,ln
* ,inf,irhv,bdcnd,jbdcd
* ,nin,ieltop,jeftop,nopp,nopf,nops,nk,
* nel,nbel,nebs,nharm,nodref,nodel,mdf,nop,nodf,ncodf,ncods,itype,
* itest,blist,jblist
* ,numnod,num,nodcod,elnum,ns
* ,iijj,kk
* ,i,j,k,l,m
character *200
integer ios
dimension nodcod(6000)
1,convge(7500),bb1(1000)
common/dofnum/nf(7500,4),nff(7500,4),nfs(7500,2)
common/coaord(3000,2),wcoaord(500,2),wang(500)
common/press/pinc(50),delt(50),uinc(50,7500)
common/time!totime,istep,nsteps,deltim
common/bounds/btype(5),bdcd(5,200),blist(200,5),numnod(200)
1,neuman,nbel(20),nbels,nharm
common/pfact/pfact(3)
common/props/amu,ro,re,r0,rmax
common/reslets/sk(7500),chk(7500)
common/fron1/nodref,totdof,totnod,totels,fels,dofel,mwga,
* ntra,nout,dimen,nodel,idof,fdof,sdof,hband,nelr,neilt
common/fron4/mdf(2000),nopp(7500),nopf(7500),nops(7500)
common/fron2/eltop(500,10),bval(500),bvalf(500),bvals(500),
* elq(70),bncode(500),bnode(500),bc(7500)
* ,bncodf(500),bnodef(500),bcdn(7500),bclf(7500)
* ,bncods(500),bnodes(500),bcs(7500)
```
Problem size-dependent data statements

data ibdcnd /1/, iblist /50/, icoord /3000/, ieltop /500/,
*         inf /500/, irhv/7500/, jbdcond /200/,
*itype/1/,
*         jblist /5/, jeltop /10/, nin/1/,nodsid/3/,
*         ivec7/7500/,ir1/7500/

open(file='meshx.dat',unit=nin,form='formatted',
+        status='old',iostat=ios,err=199)
open(file='fluidoutx',unit=10,form='formatted')
open(file='meshoutx.dat',unit=11,form='formatted')
open(file='walloutx',unit=18,form='formatted')
open(file='veloutx.dat',unit=13,form='formatted')
open(file='presoutx.dat',unit=14,form='formatted')
open(file='defloutx.dat',unit=15,form='formatted')
open(file='dynoutx.dat',unit=16,form='formatted')
open(file='diag',unit=17, form='formatted')
open(unit=4,form='unformatted',status='scratch')

set itest for full checking
itest = 0

***************
*          *
* Input data section  *
*          *
***************

Nodal geometry
read(nin,8010) totnod,dimen
dofnod=2
do 1010 i=1,totnod
read(nin,8020) nodnum, (coord(nodnum,j),j=1,dimen)
write(11,8020) nodnum, (coord(nodnum,j),j=1,dimen)
1010 continue
c  

c Topology

c
70  read(nin,8010) totels,fels
71  write(11,8010) totels,fels
72  elnum=0
73  do 1020 i=1,totels
74  elnum=elnum+1
75  read(nin,8010) num,nodel,nodel,(eltop(elnum,j+2),j=1,nodel)
76  eltop(elnum,1) = nodel
77  eltop(elnum,2)=nodel
78  write(11,8010) num,nodel,nodel,(eltop(elnum,j+2),j=1,nodel)
79  1020 continue

c Identify corner nodes

c
80  do 1019 elnum=1,totels
81  if(eltop(elnum,2).eq.6) then
82    nodel=6  
83  else
84    nodel=8  
85  end if
86  do 1019 j=3,nodel+2
87  9 continue
88  do 1021 elnum=1,totels
89  nodcod(eltop(elnum,3))=1
90  nodcod(eltop(elnum,5))=1
91  nodcod(eltop(elnum,7))=1
92  if(nodel.eq.6) go to 1021
93  nodcod(eltop(elnum,9))=1
94  1021 continue
95  do 1112 i=1,totels
96  jj=eltop(i,2)+2
97  do 1112 j=3,jj
98  mdf(eltop(i,j))=nodcod(eltop(i,j))+3
99  nop(i,j-2)=eltop(i,j)
100  1112 continue

c Set up element freedom number array

c
101  read(nin,8011) r0
102  read(nin,8011) amu
103  read(nin,8011) rho
104  read(nin,8011) dens
105  read(nin,8011) totime
106  read(nin,8010) ns
107  deltim=totime/ns
108  read(nin,8009) pinc(i)
109  1051 continue
110  read(nin,8010) numint

C.9
do 6565 k=1,numint
read(nin,8010) num,intnod(k),outnod(k)
6565 continue

centnod(1)=1

do 300 i=2,numint
centnod(i)=1+outnod(i-1)
300 continue

totdof=0
fdof=0
sdof=0
isdof=0

k=1
do 2000 i=1,totnod
if(i.le.intnod(k).or.intnod(k).eq.outnod(k)) then
  do 1125 j=1,mdf(i)
  fdof=fdof+1
  nff(i,j)=fdof
1125 continue
end if
if(i.eq.outnod(k)) then
  k=k+1
end if
2000 continue
isdof=fdof

k=1
do 2001 i=1,totnod
if(i.ge.intnod(k).and.i.le.outnod(k)) then
  do 1126 m=1,2
  isdof=isdof+1
  nfs(i,m)=isdof
1126 continue
end if
if(i.eq.outnod(k).or.intnod(k).eq.outnod(k)) then
  k=k+1
end if
2001 continue
totdof=isdof

sdof=isdof-fdof
write(*,9021) sdof,fdof
9021 format('Degrees of freedom - Solid',i5,' Fluid',i5)
do 4551 i=1,fels
do 4552 j=1,node!
k=iabs(nop(i,j))
nopf(k)=nff(k,1)
4552 continue
4551 continue
do 4553 i=fels+1,totels
do 4554 j=1,node!
k=iabs(nop(i,j))
nops(k)=nfs(k,1)
continue

call vecnul(ncodf,inf,inf,itest)

Dirichlet boundary conditions

read(nin,8010) dirich
if(dirich.eq.0) go to 9874
do 1030 i=1,dirich
read(nin,8012)bnodes(i),bncods(i),bvals(i)
dof=nopf(bnodes(i))-1+bncods(i)
codf(dof)=1
c(i)=bvals(i)
cont;
9874 continue

Wall boundary conditions

read(nin,8010) wallbc
call vecnul(ncods,inf,inf,itest)
if(wallbc.eq.0) go to 9877
do 1033 i=1,wallbc
read(nin,8012)bnodes(i),bncods(i),bvals(i)
cods(dof)=1
c(i)=bvals(i)
cont;
9877 continue

do 1049 ii=1,toldof
uinc(1,ii)=0.0d0
cont;
1049 continue

do 1048 ii=fldof+1,totof
a0(ii)=0.0d0
da0(ii)=0.0d0
d2a0(ii)=0.0d0
a1(ii)=0.0d0
da1(ii)=0.0d0
d2a1(ii)=0.0d0
cont;
1048 continue

alpha=1.0d0
beta=1.0d0
gamma=0.0d0
ichk=toldof
write(*,1001)
format(' Calling Front')
cont;
1001
C.11
** ***************
* Time Step Loop *
* ***************

time=0.0d0
do 4000 istep=2,nsteps
ntra=1
ei=0
time=time+deltim
write(*,1041) istep-1, time
write(*,9056) pinc(istep)
!
1041 format(' Time Step No. ',i2,2x,'Elapsed Time =',f7.4, ' secs')

***************
* Velocity Iteration Loop *
* ***************
do 4750 itn=1,ntns
if(iel.eq.1) then
go to 4760
endif
write(*,9055) itn,istep-1
nnax=coord(iabs(nop(fels,3)),1)
re=rho/amu
call front(itn,iel)
if(iel.eq.1) go to 5600
!
4760 call front(itn,iel)
!
convvg=0.0d0
do 3600 i=1,f dof
if(chk(i).le.1.0d-012) go to 3600
convge(i)=(sk(i)-chk(i))/chk(i)
convvg=convvg+convge(i)
3333 format(/ Average Convergence = ',f10.5/) continue
3600 convvg=convvg/fdof
write(*,3333) convvg
!
do 3610 i=1,f dof
chk(i)=sk(i)
!
3610 continue
chksum=0.0d0
do 3611 i=1,totnod
if(nff(i,1).eq.0) then
go to 3611
end if
chksum=chksum+chk(nff(i,1))
3611 continue
reyn=chksum/totnod*rmax*2.0d0*rho/amu
dean=2.0d0*rey*(rmax/r0)**0.5d0
omn=2.0d0*3.14159/totime*rmax**2*rho/amu
write(*,3612) abs(reyn),omn,dean
3612 format(' Reynolds Number = ',f5.0,3x,'Omega* = ',f5.0
1.3x,'Dean Number = ',f5.0/
if(abs(convvg).lt.1.0e-03) then
go to 4500
endif
5000 continue
c
c
go to 2121
c
pcorr=0.0d0
do 1045 i=1,fels
do 1046 j=1,node
do 1047 k=1,mdf(i)
if(k.eq.4) then
l=nff(iabs(nop(i,j)),4)
if(uinc(istep,l).lt.pcorr) pcorr=uinc(istep,l)
end if
1047 continue
1046 continue
1045 continue
adjust pressure level
go to 2121
pcorr=0.0d0
do 1045 i=1,fels
do 1046 j=1,node
do 1047 k=1,mdf(j)
if(k.eq.4) then
l=nff(iabs(nop(i,j)),4)
if(uinc(istep,l).lt.pcorr) pcorr=uinc(istep,l)
end if
1047 continue
1046 continue
1045 continue
pcorr=dabs(pcorr)
do 1052 i=1,fels
do 1053 j=1,node
do 1054 k=1,mdf(j)
if(k.eq.4) then
l=nff(iabs(nop(i,j)),4)
if(uinc(istep,l).lt.pcorr) uinc(istep,l)=uinc(istep,l)+pcorr
1054 continue
1053 continue
1052 continue
1054 continue
1053 continue
1052 continue
2121 continue
nra=1
iel=1
do 301 i=1,totnod
write(10,9035) i,(sk(nff(i,j)),j=2,3)
write(13,8021) i,sk(nff(i,1)),sk(nff(i,2)),sk(nff(i,3))
write(14,9019) i,sk(nff(i,4))
continue
5600 continue
call vecnul(r1,7500,7500,itest)
do 295 i=fels+1,totels
do 294 j=1,node!
do 293 k=nfs(iabs(nop(i,j)),l)
a1(k)=sk(k)
da1(k)=c5*(a1(k)-a0(k))-c6*da0(k)
d2a1(k)=c5*(da1(k)-da0(k))-c6*d2a0(k)
continue
continue
continue
continue
do 296 ii=fels+1,totels
ii=1,nodel
jk=nfs(ii,2)
write(18,21) i,a1(ii),a1Gk)
continue
Adapt fluid mesh to wall profile

do 298 jj=1,numint
jk=intnod(jj)
jl=nfs(intnod(jj),1)
jm=nfs(intnod(jj),2)
do 299 ii=centnod(jj),jk-1
aa=a1(jl)*dcos(wang(jk))+a1(jm)*dsin(wang(jk))
321 \[ bb = \text{coord}(jk,1)*(-a_1(jl)\times \text{dsin}(\text{wang}(jk))+a_1(jm)\times \text{dcos}(\text{wang}(jk))) \]
322 \[ \text{coord}(ii,1) = \text{coord}(ii,1) + aa*\text{coord}(ii,1)/\text{coord}(jk,1) \]
323 \[ \text{coord}(ii,2) = \text{coord}(ii,2) + bb*\text{coord}(ii,1)/\text{coord}(jk,1) \]
324 continue
325 continue
326 do 304 i = 1, totnod
327 write(15,8020) i, \text{coord}(i,1), \text{coord}(i,2)
328 304 continue

c

c Zero wall displacements, velocities and accelerations

c
329 call veccop(a1,2000,a0,2000,2000,itest)
330 call veccop(da1,2000,da0,2000,2000,itest)
331 call veccop(d2a1,2000,d2a0,2000,2000,itest)
332 call vecnul(a1,2000,2000,itest)
333 call vecnul(da1,2000,2000,itest)
334 call vecnul(d2a1,2000,2000,itest)
335 continue
336 print *, linfo
337 199 write(*,3456)
338 3456 format(' Stopping at end of increment loop')
339 stop
340 format(2i5,3f10.5)
341 format(5f10.5)
342 format(18x,'Pressure=',f10.3)
343 format(2i5,3f10.5)
344 format(3f10.4)
345 format(4i5,2f10.5)
346 format(4i5,2f10.5)
347 format(20iS)
348 format(16iS)
349 format(2d10.2)
350 format(2d10.2)
351 format(2d10.2)
352 format(2d10.2)
353 format(2d10.2)
354 format(2d10.2)
355 format(2d10.2)
356 format(2d10.2)
357 format(2d10.2)
358 format(2d10.2)
359 format(2d10.2)
360 format(2d10.2)
361 format(2d10.2)
362 format(2d10.2)
363 format(2d10.2)
364 format(2d10.2)
365 format(2d10.2)
366 format(2d10.2)

c

C.15
subroutine front(itn,iel)

implicit double precision (a-h,o-z)
integer totdof,totels,dofel, fels,idof,fdof,sdof,hband,nelr,nelt
integer eltop,dim, totnod,bncode,bnode,
bncodf,bnodef,bncods,bnodes,
*i, iel, itn, nopp,nopf,nops,nk, nodref,nodel,mdf,nop,ncod,ncodf,ncods,
*itest, ii,kk, i,j,k,l,m,n,inf,jmod,irr,kl,th,kpiro,lpivo,lpivo,lpivco,kpiro,lco,
*outnod,centnod,ir1,nmax,ncri,nb,nlast,nnrot,lerror,l1,
*icol,krow,nell,kc,nj,ntk,khed,lhed,idf,nn, node,
*ldest,kdest,le,lk,ll,piv,ir,kr,kt,kpi,kr,ko,
*krow,nf,nff,nfs,mwga,ntra,nout,
*numint,ntnod,inf
common/dofnum/nf(7500,4),nff(7500,4),nfs(7500,2)
common/props/amu,rho,re,r0,rmax
common/reslts/sk(7500),chk(7500)
common/fron /nodref,totdof,totels,fels,dofel,mwga,
*ntra,nout,dim, nodel,idof,fdof,sdof,hband,nelr,nelt
common/fron2/eltop(500,10),bval(500),bvalf(500),bvals(500),
*elq(70),bncode(500),bnodes(500),bcf(7500),
*bncodf(500),bnoddef(500),bcf(7500),bcf1(7500),
*bncods(500),bnodes(500),bcs(7500)
common/fron4/mdf(2000),nopp(7500),nopf(7500),nops(7500)
common/fron5/nop(500,8),nk(8,3),elk(28,28)
common/fron6/ncodf(7500),ncodf(7500),ncods(7500)
common/factrs/alpha,beta,gamma
common/epi/ numint,ntnod,inf(200),outnod(200),centriod(200)
common/rhr/v1(7500),vec(7500),vecp(7500),vec1(7500),vec2(7500)
common/pfact/pfact(3)
dimension
eq(1500,1500),lhed(1500),khed(1500),lpiv(1500),kpi(1500)
405 *jmod(1500), qq(1500), nj(1500)
406 data inf /1500/jnf/4/, lirln500/
407 nmax=900
408
409 write(12,400)
410
c
411 if(iel.eq.0) then
412 nell=0
413 idof=fdof
414 na=1
415 nb=fels
416 do 1040 i=1,totnod
417 nopp(i)=0
418 nopp(i)=nopf(i)
419 do 1040 j=1,mdf(i)
420 nf(i,j)=0
421 nf(i,j)=nff(i,j)
422 1040 continue
423 do 1042 i=fdof+1,totdof
424 ncod(i)=0
425 bc(i)=0.0d0
426 bc(i)=bcs(i)
427 ncod(i)=ncodf(i)
428 1042 continue
429 else
430 nell=fels
431 -idof=sdof
432 na=fels+1
433 nb=totels
434 do 1041 i=1,totnod
435 nopp(i)=0
436 nopp(i)=nops(i)
437 do 1041 j=1,2
438 nf(i,j)=0
439 nf(i,j)=nfs(i,j)
440 1041 continue
441 do 1043 i=fdof+1,totdof
442 ncod(i)=0
443 bc(i)=0.0d0
444 bc(i)=bcs(i)
445 ncod(i)=ncods(i)
446 1043 continue
447 end if
448 if(ntra.eq.0) go to 14
find last appearance of each node

```
449  nlast=0
450  do 12 i=1,nodref
451  do 8 n=na,nb
452     do 4 l=1,eltop(n,2)
453     if(nop(n,l).ne.i)go to 4
454     nlastl=n
455     if(nlast.ne.nlastl)go to 3
456     nerror=1
457     write(* ,416)nerror,n
458     write(*,* )"Stopping"
459     stop
460   3 continue
461  nlast=n
462     11=1
463   4 continue
464   8 continue
465   if(nlast.eq.0) go to 12
466     nop(nlast,11)=-nop(nlast,11)
467  nlast=0
468  12 continue
469  ntra=0
```

assembly

```
470   14 continue
471     lcol=0
472     krow=0
473     do 16 i=1,nmax
474     do 16 j=1,nmax
475        eq(i,j)=0.
476   16 continue
477   18 nell=nell+1
478     if(iel.eq.0) then
479        call fluidl(nell,itn)
480     c write(*,1002) nell
481     else
482        call solidl(nell,itn)
483     c write(*,1003) nell
484     end if
485     n=nell
486     kc=0
487     mwga=0
488     if(mwga.eq.0)go to 21
489     do 20 i=1,eltop(n,2)
490     nj(i)=nop(n,i)
491   20 continue
```
492 \[ kc = \text{eltop}(n,2) \] 493 \[ \text{go to 23} \] 494 \[ \text{continue} \] 495 \[ \text{do 22 } j = 1, \text{eltop}(n,2) \] 496 \[ nn = \text{nop}(n,j) \] 497 \[ m = \text{iabs}(nn) \] 498 \[ k = \text{nopp}(m) \] 499 \[ idf = \text{mdf}(m) \] 500 \[ \text{if}(\text{iel} .eq. 1) \text{idf} = 2 \] 501 \[ \text{do 22 } i = 1, \text{idf} \] 502 \[ kc = kc + 1 \] 503 \[ ii = k + 1 \] 504 \[ \text{if}(nn .lt. 0) \text{ii} = -ii \] 505 \[ nj(kc) = ii \] 506 \[ \text{continue} \] 507 \[ \text{continue} \] 508 \[ \text{set up heading vectors} \] 509 \[ ntr = 0 \] 510 \[ \text{if}(\text{ntr} .eq. 1) \text{write}(12,420) \text{krow,lcol} \] 511 \[ \text{if}(\text{ntr} .eq. 1) \text{write}(12,424) \] 512 \[ \text{if}(\text{ntr} .eq. 1) \text{write}(12,428) \text{(khed(k),lhed(k),k=1,lcol)} \] 513 \[ \text{if}(\text{ntr} .eq. 1) \text{write}(12,432) \] 514 \[ \text{if}(\text{ntr} .eq. 1) \text{write}(12,438) \text{(kdest(k),ldest(k),k=1,kc)} \] 515 \[ \text{do 52 } lk = 1, \text{kc} \] 516 \[ \text{node} = \text{nj(lk)} \] 517 \[ \text{if}(\text{lcol} .eq. 0) \text{go to 28} \] 518 \[ \text{do 24 } l = 1, \text{lcol} \] 519 \[ ll = l \] 520 \[ \text{if}(\text{iabs(node).eq.iabs(lhed(l)))go to 32} \] 521 \[ \text{continue} \] 522 \[ \text{continue} \] 523 \[ \text{ldest(lk)} = \text{lcol} \] 524 \[ \text{lhed(lcol)} = \text{node} \] 525 \[ \text{go to 36} \] 526 \[ \text{ldest(lk)} = ll \] 527 \[ \text{lhed(ll)} = \text{node} \] 528 \[ \text{if}(\text{krow}.eq.0) \text{go to 44} \] 529 \[ \text{do 42 } k = 1, \text{krow} \] 530 \[ \text{kk} = k \] 531 \[ \text{if}(\text{iabs(node).eq.iabs(khed(k)))go to 48} \] 532 \[ \text{continue} \] 533 \[ \text{continue} \] 534 \[ \text{krow} = \text{krow} + 1 \] 535 \[ \text{kdest(lk)} = \text{krow} \] 536 \[ \text{khed(krow)} = \text{node} \] 537 \[ \text{go to 52} \]
538 48 continue
539 kdest(lk)=kk
540 khed(kk)=node
541 52 continue
542 c if(ntr.eq.1) write(12,420)krow,lcol
543 c if(ntr.eq.1) write(12,424)
544 c if(ntr.eq.1) write(12,428) (khed(k),lhed(k),k=1,nmax)
545 c if(ntr.eq.1) write(12,432)
546 c if(ntr.eq.1) write(12,428) (kdest(k),ldest(k),k=1,kc)
547 c if(ntr.eq.1) write(12,436) nell
548 c if(ntr.eq.1) write(12,436)nell
549 c if(ntr.eq.1) write(12,436)nell
550 if(ntr.eq.1) write(12,436)nell
551 54 continue
552 do 56 l=1,kc
553 ll=ldest(l)
554 do 56 k=1,kc
555 kk=kdest(k)
556 eq(kk,ll)=eq(kk,ll)+elk(k,l)
557 if(nell.gt.fels) then
558 write(l2,417)nerror
559 c write(12,*) nell,( eq(i,i),elk(i,i),i=1,16)
560 c write(12,436)nell
561 c if(nrow.lt.ncrit.and.nell.lt.nb)go to 18
562 c find out which matrix elements are fully summed
563 60 lc=0
564 do 64 l=1,lcol
565 if(lhed(l).ge.0)go to 64
566 lc=lc+1
567 lpiv(lc)=l
568 64 continue
569 ir=0
570 kr=0
571 do 68 k=1,krow
572 kt=khed(k)
573 if(kt.ge.0)go to 68
574 kr=kr+1
575 kpiv(kr)=k
576 kro=iabs(kt)
577 if(ncod(kro).ne.1)go to 68
578 ir=ir+1
579 jmod(ir)=k
580 ncod(kro)=2
581 r1(kro)=bc(kro)
582 68 continue
modify equations with applied boundary conditions

```fortran
write(12,448) lc, kr
write(12,428)(lpiv(k), kpi(k), k=1,nmax)
if(ir.eq.0) go to 71
write(12,456)
do 70  
irr=1,ir
k=jmod(irr)
c write(12,428) k
kh=iabs(khed(k))
do 691=1,icol
eq(k,l)=0.
lh=iabs(lhed(l))
if(lh.eq.kh) eq(k,l)=1.
69 continue
70 continue
71 continue
if(kr.gt.0.and.lc.gt.0) go to 72
error=3
write(12,418) error
stop
72 continue
write(12,460)
search for absolute pivot

pivot=0.0d0
do 76 l=1, ic
lpivc=lpiv(l)
kpivr=lpivc
piva=eq(kpivr,lpivc)
if(abs(piva).lt.(pivot) go to 76
pivot=piva
lpivco=lpivc
kpivro=kpivr
76 continue
1989 format(/e20.10/)
normalise pivotal row

kro=iabs(khed(kpivro))
lco=iabs(lhed(lpivco))
if(ntr.eq.1) write(12,452) kro, lco, pivot, nell
if(abs(pivot).lt.1e-20) then
write(*,476)
write(*,8787) pivot
8787 format(/'Pivot = ', e10.5)
stop
```

C.21
end if
625 do 80 l=1,lcol
626 qq(l)=eq(k pivro,l)/pivot
627 80 continue
628 rhs=r1(kro)/pivot
629 r1(kro)=rhs
630 c write(12,468)
631 c write(12,440)(qq(l),l=1,lcol)
c c eliminate then delete pivotal row and column
632 if(k pivro.eq.1)go to 104
633 kpivr=k pivro-1
634 do 100 k=1,k pivr
635 krw=iabs(k hed(k))
636 fac=eq(k,lpivco)
637 c write(12,480)fac
638 if(lpivco.eq.1.or.dabs(fac).lt.1.0d-20) go to 88
639 lpivc=lpivco-1
640 do 84 l=1,lpivc
641 eq(k,l)=eq(k,l)-fac*qq(l)
642 84 continue
643 88 if(lpivco.eq.lcol)go to 96
644 lpivc=lpivco+1
645 do 92 l=lpivc,lcol
646 eq(k,l-1)=eq(k,l-1)-fac*qq(l)
647 92 continue
648 96 r1(krw)=r1(krw)-fac*rhs
649 100 continue
650 104 if(k pivro.eq.krow)go to 128
651 kpivr=k pivro+1
652 do 124 k=k pivr,krow
653 krw=iabs(k hed(k))
654 fac=eq(k,lpivco)
655 c write(12,480)fac
656 if(lpivco.eq.1)go to 112
657 lpivc=lpivco-1
658 do 108 l=1,lpivc
659 eq(k-1,l)=eq(k-1,l)-fac*qq(l)
660 108 continue
661 112 if(lpivco.eq.lcol)go to 120
662 lpivc=lpivco+1
663 do 116 l=lpivc,lcol
664 eq(k-1,l-1)=eq(k-1,l-1)-fac*qq(l)
665 116 continue
666 120 r1(krw)=r1(krw)-fac*rhs
667 124 continue
668 128 continue

C.22
c write pivotal equation on disc

c write(4) kro,lcol,lpivco,(lhed(l),qq(l)),l=1,lcol)
670 *krow,pivot,kpivro,(khed(k),k=1,krow)
671 do 129 k=1,krow
672 eq(k,lcol)=0.
673 129 continue
674 do 130 l=1,lcol
675 eq(krow,l)=0.
676 130 continue
677 c write(12,436)nell
678 c write(12,440)((eq(ij)j=1,nmax),i=1,nmax)
679 c write(12,460)
c rearrange heading vectors
c
680 lcol=lcol-1
681 if(lpivco.eq.lcol+1)go to 136
682 do 132 l=lpivco,lcol
683 lhed(l)=lhed(l+1)
684 132 continue
685 136 krow=krow-1
686 if(kpivro.eq.krow+1)go to 144
687 do 140 k=kpivro,krow
688 khed(k)=khed(k+1)
689 140 continue
690 144 continue
691 c write(12,420)krow,lcol
692 c write(12,424)
693 c write(12,428) (khed(k),lhed(k),k=1,nmax)
c
c determine whether to assemble, eliminate, or backsubstitute
c
694 if(krow.gt.ncrit)go to 60
695 if(nell.lt.nb) go to 18
696 if(krow.gt.1)go to 60
697 lco=iabs(lhed(1))
698 kpivro=1
699 pivot=eq(1,1)
700 kro=iabs(khed(1))
701 lpivco=1
702 qq(1)=1.
703 if(ntr.eq.1)write(*,452)lco,kro,pivot,nell
704 if(abs(pivot).lt.1e-20) then
705 write(*,*) pivot
706 write(*,476)
707 stop
708 end if
709 r1(kro)=r1(kro)/pivot
write(4) kro, lcol, lpivco, lhed(1), qq(1), krow, pivot, kpivro, khed(1)
write(*,1004)
format(' Assembling FLUID Element No. ',i3)
format(' Assembling SOLID Element No. ',i3)
format(' Writing Back Substitution ')
call bacs
format(i10)
format(5h node, 6h niast)
format(1x,2i5)
format(/16h nodal numbering/)
format(9i5)
format(/8h nerror=,i5/i5, 62h th. element has more than one node with
1/62h same nodal number 1/)
format(/8h nerror=i5//
1 62h the difference nmax-ncrit is not sufficiently large
1/62h to permit the assembly of the next element--
1/62h either increase nmax or lower ncrit
1/)
format(/8h nerror=i5//
1 62h there are no more rows fully summed, this may be due to---
1/62h (1)incorrect coding of iabs(nop or nk arrays
1/62h (2)incorrect value of ncrit. increase ncrit to permit
1/62h whole front to be assembled
1/)
format(/6h krow=,i5,6h lcol=,i5/)
format(/5h khed,5h lhed/)
format(2i6)
format(/6h kdest,6h ldest/)
format(/22h eq matrix element no=,i3/)
format(/6h l, n=,i5)
format(/5h jmod/)
format(/18h right hand vector/)
format(15,e20.10)
format(/12h pivotal row=,i4,16h pivotal column=,i4,7h pivot=,e20.10
1,'nell=' ,i5)
format(/18h right hand vector/)
format(i5,e20.10)
format(/12h pivotal row/)
format(43h warning-matrix singular or ill conditioned)
format(5h fac=,e20.10)
return
end

implicit double precision (a-h,o-z)
integer totdof,totnod,totels,eltop, dofel, hband, fels, dimen,
*idof,fdof,sdof,nopp,nopf,nops,nk,
*nodref,node1,mf, ndof,ncodf,ncods, itest
integer bncodf,bnodef,bncods,bnodes,bncode,khed,l
* ,nf,nff,nfs,mwga,ntra,nout,iv,lhed
*i,k,krow,lcol,l pivco, krow, k pivro, bnode
common/dofnum/nf(7500,4),nff(7500,4),nfs(7500,2)
common/props/amu,rho,re,r0,max
common/reslts/sk(7500),chk(7500)
common/fron1/nodref,totdof,totnod,totels,fels,dofel,mwga,
*ntra,nout,dimen,nodel, idof,fdof, sdo, hband, nel, nelt
common/fron2/eltop(500,10),bval(500),bvalf(500),bvals(500),
elq(70),bncode(500),bnode(500),bc(7500)
*,bncodf(500),bncof(500),bcf(7500),bcfl(7500)
*,bncofs(500),bncoes(500),bcso(7500)
common/fron4/mdf(2000),nopp(7500),nopf(7500),nops(7500)
common/fron5/nop(500,8),nk(8,3),elk(28,28)
common/fron6/ncod(7500),ncodf(7500),ncods(7500)
common/rhv/rl(7500),vec(7500),vecp(7500),vec1(7500),vec2(7500)
dimension
llhed(1500),khed(l500)
2,qq(1500)
do 4 i=1,totdof
4 sk(i)=b(i)
do 32 iv=1,totdof
backspace 4
read(4) kro,lcol,l pivco,(lhed(l),qq(l),l=1,lcol)
l,krow, pivot,k pivro,(khed(k),k=1,krow)
backspace 4
c write(12,404)
c write(12,408) kro,lcol,l pivco
c write(12,408) (lhed(l),l=1,lcol)
c write(12,412) (qq(l),l=1,lcol)
if(ncod(kro).gt.0)then
write(12,* ) kro,ncod(kro)
go to 24
end if
if (ncod(kro).gt.0) then
write(12,* ) kro,ncod(kro)
go to 24
end if
r1=r1+gash
24 continue
ncod(kro)=1
continue
format(14h disk contents)
format(10i5)
format(5e20.10)
call vecnul(r1,7500,7500,itest)

C.25
rewind(4)
return
end

***************
* Element generation  *
***************

Fluid Elements

subroutine fluidl(nell,itn)
implicit double precision(a-h,o-z)
double precision jac,jacin,ldevr,ldevr, mass, m1, mm1, mm2
integer totdof, totnod, totelel, dofe, bnode, eltop, dimen,
*fels, nf, nff, nfs, istep, hband, outnod, centnod,
nn, nend, m, btype, bdcnd, blist, nsteps, icood, itn,
nout, ntra, mwga, jjac, jjac, igderv, jgderv, jgdert, jgdert, jgdert,
 Jdpar, Jdpar, nfpar, ncod, idof, fof, sdo, sf,
nopp, nops, nk, numnod, ilabss, ieltop, jeltop,
test, ielk, jek, iropiv, nbelt, nbels, nharm, nodref
,nodcode, ial, ial, idum2, idum2, il, ilarg, ilwght
,ister, iquad, kl, km, ilabss, ial, jek, neln, nodel, md
d, i, scvec, jcoord, ildev, jeldev, i, k, jlabss
integer
bncodf, bnodedf, bncodbs, bnodes, nell, neuman, numint, inod, isize, jn
* idmvec, ir1, nquad, ngaus, iwght, nosd, nd, ifun, ifunp, ifunv, ifunv, bnode, bnode, bnode, jgeom,
 * jgeom, jjac
 dimension abss(2,9),
dtptd(24,24), fun(24), funv(8), dummy(28,28), gdervr(2,8),
gdert(8,2),
geom(8,2), jac(2,2), geom(4,2), jacin(2,2), Iderv(2,8),
lder(2,4), gder(2,4),
* wght(9), ell1(4,8), ell2(4,8), ell3(4,8), ell1(8,4), ell2(8,4), ell3(8,4)
*, dmvce(28), dum2(28,28), lder(2,8), c1(24,24), c2(24,24), c3(24,24),
* mass(24,24), m1(7500), mm1(24,24), mm2(24,24), xm(8), ym(8), r (8)
* f88(8), f99(8)
common/ props/amu, rho, re, r0, rmax
common/dofnum/nf(7500,4), nff(7500,4), nfsw(7500,2)
common/press/pinc(50), delt(50), uinc(50, 7500)
common/time/totim, istep, nsteps, deltime
common/res/ sk(7500), chk(7500)
common/coord/coord(3000,2), wcoord(500,2), wang(500)
common/nod/nodcode(200), ncod(200)
common/fron1/nodref, totdof, totnod, totelel, fof, sdo, sdo, sdo, hnode, nelt
*ntra, nout, dimen, nodref, idof, fof, sdo, hnode, nelt
common/lfron2/eltop(500,10),bval(500),bvalf(500),bvals(500),
  *elq(70),bnode(500),bc(7500),bncodef(500),bnodef(500)
  *,bcf(7500),bclf1(7500)
  *,bnodes(500),bncode(500),bcs(7500)
  common/lfron4/mdf(2000),nopp(7500),nopf(7500),nops(7500)
  common/lfron5/nop(500,8),nk(8,3),elk(28,28)
  common/factrs/alpha,beta,gamma
  common/epi/numint,intnod(200),outnod(200),centnod(200)
  common/rhv/v1(7500),vect(7500),vecp(7500),vec1(7500),vec2(7500)
  common/bounds/btype(5),bdcnd(5,200),blstf(200,5),numnod(200)
  l,neuman,nel,nelb(20),nbels,nharm
  common/lpfact/lfact(3)
  data isize/28,inff/7500,inf1f/4,ilabss/3,ieltop/500/jeltop/10/
  ielk/28/jelk/28/
  *lwght/3,iropiv/200,ial/200,ial/200,ial/261,ielq/28,idum2/28,jdum2/28/
  *iscvec/24,ldmvec/24/
  ,icoord/3000,jcoord/3000,jlnder/3,jlnder/8,ir1/7500/
  ,nodsid/3/nqp/3,nqd/2/
  call vecnul(elq,ielq,isize,itest)
  if(eltop(nell,2).eq.6) then
  igeom=6
  jgeom=2
  nodel=6
  ngaus=7
  dofel=12
  ifun=6
  ifunv=6
  ifunp=3
  idtpd=12
  jdtbd=12
  jgderv=6
  igdert=6
  else
  igeom=8
  jgeom=2
  igeom=4
  jgeom=2
  nodel=8
  ngaus=4
  dofel=24
  ifun=8
  ifunv=8
  ifunp=4
  idtpd=12
  jdtbd=12
  jgderv=8
  igdert=8
  igdertp=dimen
  end if
iwght=ngaus
iabss=dimen
jabss=ngaus
ilderv=dimen
ilderp=dimen
ijac=dimen
jjac=dimen
ijacin=dimen
jjacin=dimen
igderv=dimen
jgdert=dimen
isteer=dofel
if (eltop(nell,2).eq.6) then
  call qtri7(wght,7,abss,2,7,ngaus,itest)
else
  call qqua4(wght, iwght, abss, iabss, jabss, ngaus, itest)
end if

Call matnul(dtpd,idtpd,jdtpd,idtpd,jdtpd,itest)
call matnul(dum2,idum2,jdum2,idum2,jdum2,itest)
call matnul(eil,k,jel,k,jel,k,jel,k,itest)
call matnul(mass,dofel,dofel,dofel,dofel,itest)
call matnul(mm1,dofel,dofel,dofel,dofel,itest)
call matnul(mm2,dofel,dofel,dofel,dofel,itest)
call vecnul(m1,5ob,500,itest)
call vecnul(xm,ifun,ifun,itest)
call vecnul(ym,ifun,ifun,itest)
call vecnul(rgp,ifun,ifun,itest)
call vecnul(vect,7500,7500,itest)
do 1122 i=1,nodel
geom(i,1)=coord(eltop(nell,i+2),1)
geom(i,2)=coord(eltop(nell,i+2),2)
k=1
1122 continue
1123 continue

Integration loop for element matrices

do 1090 iquad=1,ngaus
  i = abss(1,iquad)
ta = abss(2,iquad)
f(eltop(nell,2).eq.6) then

fun=6
lderv=2
lderv=6
geom=6
all trim6(fun,6,lderv,2,6,xi,eta,itest)
lse
fun=8
lderv=2
lderv=8
lderp=4
geom=8
all quam8(fun,ifun,lderv,ilderv,xi,eta,itest)
nd if
all matmul(lderv,jlderv,geom,igeom,jgeom,jac,
*jac,jjac,jgeom,jlderv,ilderv,itest)
n=igeom
all scaprd(geom(1,1),igeom,fun,ifun,n,rad,itest)
all scaprd(geom(1,2),igeom,fun,ifun,n,theta,itest)
gr=rad
ang=theta
gr2=rg**2
r02=r0**2
rbar=r0+rg*dcos(ang)
h=rbar/r0
rbar2=rbar**2
call matinv(jac,ijac,jjac,jacin,ijacin,jjacn,jgeom,jlderv,ilderv,lderv,
*igderv,jgderv,dimen,dimen,jlderv,itest)
call matran(gderv,igderv,jgderv,gdert,igderv,jgderv,itest)
quot = dabs(det)*wght(iquad)
c
Form element matrix for velocity terms
k=1
l=1
do 200 i=1,nodel
kk=nff(iabs(nop(nell,i)),1)
kl=nff(iabs(nop(nell,i)),2)
km=nff(iabs(nop(nell,i)),3)
do 101 j=1,nodel
dtpd(k,l)=gdert(i,1)*gderv(1,j)
1+gdert(i,2)*gderv(2,j)/rg2
2-1.0d0/rg*fun(i)*gderv(1,j)
3+1.0d0/rbar2*fun(i)*fun(j)
4-dcos(ang)/rbar*fun(i)*gderv(1,j)
5+dsin(ang)/rbar*fun(i)*gderv(2,j)/rg
dtpd(k,l)=dtpd(k,l)/re

c2( k,l)=fun(i)*fun(i)*chk(kl)*gderv(1,j)

C.29
c3(k,l)=fun(i)*fun(i)*chk(km)*gderv(2,j)/rg
if(l.eq.k) then
mass(k,l)=1.5d0*fun(i)*fun(j)/deltim
mm1(k,l)=mass(k,l)
end if
dtpd(k,l)=dtpd(k,l)+c1(k,l)+c2(k,l)+c3(k,l)
l=l+3
c3(k,l)=fun(i)*fun(i)*chk(km)*gderv(2,j)/rg
if(l.eq.k) then
mass(k,l)=1.5d0*fun(i)*fun(j)/deltim
mm1(k,l)=mass(k,l)
end if
dtpd(k,l)=dtpd(k,l)
l=l+3
c3(k,l)=fun(i)*fun(i)*chk(km)*gderv(2,j)/rg
if(l.eq.k) then
mass(k,l)=1.5d0*fun(i)*fun(j)/deltim
mm1(k,l)=mass(k,l)
end if
dtpd(k,l)=dtpd(k,l)
l=l+3
c2(k,l)=fun(i)*fun(i)*chk(km)*gderv(1,j)
c3(k,l)=fun(i)*fun(i)*chk(km)*gderv(2,j)/rg
if(l.eq.k) then
mass(k,l)=1.5d0*fun(i)*fun(j)/deltim
mm1(k,l)=mass(k,l)
end if
dtpd(k,l)=dtpd(k,l)+c1(k,l)+c2(k,l)+c3(k,l)
l=l+3
c2(k,l)=fun(i)*fun(i)*chk(km)*gderv(1,j)
c3(k,l)=fun(i)*fun(i)*chk(km)*gderv(2,j)/rg
if(l.eq.k) then
mass(k,l)=1.5d0*fun(i)*fun(j)/deltim
mm1(k,l)=mass(k,l)
end if
dtpd(k,l)=dtpd(k,l)
l=l+3
c2(k,l)=fun(i)*fun(i)*chk(km)*gderv(1,j)
c3(k,l)=fun(i)*fun(i)*chk(km)*gderv(2,j)/rg
if(l.eq.k) then
mass(k,l)=1.5d0*fun(i)*fun(j)/deltim
mm1(k,l)=mass(k,l)
end if
dtpd(k,l)=dtpd(k,l)+c1(k,l)+c2(k,l)+c3(k,l)
l=l+3
I +d*cos(ang)*rbar*chk(kk)*fun(i)*fun(i)*fun(j)

dtpd(k,l)=dtpd(k,l)/re

dtpd(l,k)=

1-d*cos(ang)*rbar*chk(kk)*fun(i)*fun(i)*fun(j)

dtpd(l,k)=dtpd(l,k)/re

l=l+3

1038 continue

1039 l=2

1040 k=k+3

1041 continue

1042 k=2

1043 l=3

1044 do 119 i=1, nodel

1045 kk=nff(iabs(nop(nell,i)),1)

1046 kl=nff(iabs(nop(nell,i)),2)

1047 km=nff(iabs(nop(nell,i)),3)

1048 do 120 j=1, nodel

1049 dtpd(k,l)=

1050 1-1.0*d0/rg*fun(i)*fun(i)*chk(km)*fun(j)

1051 2+1.0*d0/rg2*fun(i)*gderv(2,j)

1052 3+1.0*d0/rg*gdert(i,1)*gderv(2,j)

1053 4-d*sin(ang)/rbar*(fun(i)*fun(j)/rg+fun(i)*gderv(1,j))

1054 5-2.0*d0/rg2*dsin(ang)*dcos(ang)*fun(i)*fun(j)

1055 dtpd(k,l)=dtpd(k,l)/re

1056 dtpd(l,k)=

1057 1-2.0*d0/rg2*fun(i)*gderv(2,j)

1058 2+1.0*d0/rg*gdert(i,1)*gderv(2,j)

1059 3+d*cos(ang)/rbar*(fun(i)*gderv(1,j))

1060 4-2.0*d0*dsin(ang)*dcos(ang)/rbar2*fun(i)*fun(j)

1061 dtpd(l,k)=dtpd(l,k)/re

1062 l=l+3

1063 continue

1064 l=3

1065 k=k+3

1066 continue

1067 k=1

1068 l=3

1069 do 169 i=1, nodel

1070 kk=nff(iabs(nop(nell,i)),1)

1071 kl=nff(iabs(nop(nell,i)),2)

1072 km=nff(iabs(nop(nell,i)),3)

1073 do 170 j=1, nodel

1074 dtpd(k,l)=

1075 1-d*sin(ang)*rbar*fun(i)*fun(i)*chk(kk)*fun(j)

1076 dtpd(k,l)=dtpd(k,l)/re

1077 dtpd(l,k)=

1078 1+d*sin(ang)*rbar*fun(i)*fun(i)*chk(kk)*fun(j)

1079 dtpd(l,k)=dtpd(l,k)/re

1080 l=l+3

C.31
1081    170  continue
1082         l=3
1083         k=k+3
1084    169  continue
1085         k=3
1086         l=3
1087     do 104  i=1,nodel
1088         kk=nff(iabs(nop(nell,i)),1)
1089         kl=nff(iabs(nop(nell,i)),2)
1090         km=nff(iabs(nop(nell,i)),3)
1091         f88(i)=f88(i)+fun(i)
1092     do 105  j=1,nodel
1093         dtpd(k,1)=2.0d0/rg2*gdert(i,2)*gderv(2,j)+gdert(i,1)
1094             1*gderv(1,j).
1095         2-1.0d0/rg*fun(i)*gderv(1,j)
1096         3+fun(i)*fun(j)/rg2
1097             4-dcos(ang)/rbar*(fun(i)*fun(j)/rg+fun(i)*gderv(1,j))
1098         5+2.0d0*(dsin(ang))*2*fun(i)*fun(j)/rbar2
1099         dtpd(k,1)=dtpd(k,1)/re
1100         c2(k,1)=fun(i)*fun(i)*chk(kl)*gderv(1,j)
1101         c3(k,1)=1.0d0/rg*fun(i)*fun(i)*chk(km)*gderv(2,j)
1102         1+fun(i)*fun(i)*chk(kl)*funG)/rg
1103             if(l.eq.k) then
1104         mass(k,1)=1.5d0*fun(i)*fun(j)/deltim
1105         mm1(k,1)=mass(k,1)
1106             end if
1107         dtpd(k,1)=dtpd(k,1)+c1(k,1)+c2(k,1)+c3(k,1)
1108         1+mass(k,1)
1109         dtpd(k,1)=dtpd(k,1)
1110         l=l+3
1111    105  continue
1112         l=3
1113         k=k+3
1114    104  continue
1115     do 1080  i=1,dofel
1116     do 1070  j=1,dofel
1117         dtpd(i,j)=dtpd(i,j)*quot
1118         mm1(i,j)=mm1(i,j)*quot
1119    1070  continue
1120    1080  continue
1121    4004  continue
1122         k=1
1123         m=0
1124     do 1050  nn=1,8
1125     do 1040  nd=1,3
1126         nk(nn,nd)=0
1127         nk(nn,nd)=k+m
1128         k=k+1
1129    1040  continue

C.32
1130  nend=iabs(nop(nell,nn))
1131  if(mdf(nend).eq.4) m=m+1
1132  1050  continue
  c
  c           Assemble velocity terms
  c
1133     k=0
1134     kk=0
1135   do 1082 i=1,nodel
1136   do 1083 j=1,3
1137     k=k+1
1138     l=nk(i,j)
1139   do 1072 ii=1,nodel
1140   do 1073 jj=1,3
1141     kk=kk+1
1142     ll=nk(ii,jj)
1143     dum2(l,ll)=dtpd(k,kk)
1144  1073  continue
1145  1072  continue
1146     kk=0
1147  1083  continue
1148  1082  continue
1149  call vecadd(elq,dofel,dmvec,idmvec,dofel,itest)
1150  call matadd(elk,ie!kJelk,dum2,idum2jdum2,idum2,idum2,itest)
1151   do 68 i=1,24
1152     mm2(i,i)=mm2(i,i)+mm1(i,i)
1153 68   continue
1154  1090  continue
  c
  c           Assemble pressure terms
  c
1155  iabss=dimen
1156  if(eltop(nell,2).eq.6) then
1157    ngaus=4
1158    iwght=ngaus
1159    iabss=dimen
1160    jabss=ngaus
1161    igeom=6
1162    call qtri4(wght,iwght,iabss,jabss,ngaus,itest)
1163  else
1164    ngaus=4
1165    iwght=ngaus
1166    iabss=dimen
1167    jabss=ngaus
1168    igeom=8
1169  end if
  c
  c           Quadrature for pressure terms
  C.33
do 1095 iquad=1,ngaus
   xi=abss(1,iquad)
   eta=abss(2,iquad)
   if(e1top(nell,2).eq.6) then
      call trim6(funv,6,ldeir,2,6,xi,eta,itest)
      igeom=6
   else
      call quam8(funv,8,ldeir,2,8,xi,eta,itest)
      igeom=8
   end if
   call matmul(ldeir,lderv,jdeir,geom,igeom,jac, * ijac,ijac,jgeom,ldeir,ldeir,itest)
   call matinv(jac,ijac,ijac,jjac,ijjac,jjac,dimem,* det,itest)
   call matmul(jjac,ijjac,ijjac,jdeir,ldeir,jdeir,geomp, * igdeir,jgdere,dimen,0,igeom,jdeir,lderv,ldeir,itest)
   Calculate (xi,eta) in global (x,y)
   n=igeom
   call scaprd(geom(1,1),igeom,funv,ifun,n,rad,itest)
   call scaprd(geom(1,2),igeom,funv,ifun,n,theta,itest)
   rg=rad
   ang=theta
   rbar=rad*dcos(ang)
   h=rbar/0
   call matran(gderv,igderv,jgderv,geomp,igeom,jac, * igderv,jgderv,itest)
   quot = dabs(det)*wght(iquad)
   call quam4(funp,4,ldeir,2,4,xi,eta,itest)
   call matmul(ldeir,lderv,jdeir,geomp,igeom,jgeomp,jac, * ijac,ijac,jgeomp,ldeir,ldeir,itest)
   call matinv(jac,ijac,ijac,jjac,ijjac,jjac,dimem,* det,itest)
   call matmul(jjac,ijjac,ijjac,ldeir,ldeir,jdeir,geomp, * igderv,jgderv,dimen,0,igeom,jdeir,lderv,ldeir,itest)
   do 1074 i=1,node1
      do 1074 j=l,nodel/2
         1074  1074 continue
         1074  k=0
   1074  continue
if(eltop(nell,2).eq.6) then
    jj=l1
    kk=l8
    iarg=21
else if( eltop(nell,2).eq.8) then
    kk=25
    iarg=28
end if

do 1085 i=1,nodel
    ii=nk(i,1)
    l=0
    k=k+1
    do 1076 j=4,kk,7
        dumm(iij)=elltl (k,1)*quot/rho
        dumm(ii+ lj)=ellt2(k,l)*quot/rho
        dumm(ii+2j)=ellt3(k,l)*quot/rho
        dumm(i,j)=-ell2(1,k)*quot
        dumm(i+1j)=ell1(l,k)*quot
        dumm(i+2)=-ell3(l,k)*quot
    1076 continue
    1085 continue
    call matadd(e1k,ielkjelk,dumm,idum2jdum2,iarg,iarg,itest)
c
    do 1075 i=1,8
        kk=nff(iabs(nop(nell,i)),1)
        vecp(kk)=vecp(kk)+pinc(istep)/rho*funv(i)*quot/lh
    1075 continue
    1095 continue

    do 1076 i=1,8
        kk=nff(iabs(nop(nell,i)),1)
        kl=nff(iabs(nop(nell,i)),2)
        km=nff(iabs(nop(nell,i)),3)
        vect(kk)=mm2(k,k)*uinc(istep-1,kk)
        vect(kl)=mm2(k,k)*uinc(istep-1,kl)
        vect(km)=mm2(k,k)*uinc(istep-1,km)
    999 k=k+3
    69 continue
    do 176 i=1,fdof
        r1(i)=r1(i)+vecp(i)+vect(i)
    176 continue
vecp(i)=0.0d0
vecr(i)=0.0d0
continue
return
dend

subroutine solid1(nell,itn)
implicit double precision(a-h,o-z)
double precision jac,jacin
integer dimen, dofel, dofnod, eltop,
*ielw,ink,hband,jabbs,k,nelt,nelr,nodref,mwga,ntra,mdf,nopp,
*irl,ia0,ia1,iabss,ib,ibt,ibtdb,
*nell,nsteps,nopf,nops,itn,istep,nf,nff,nfs,
*lwght,labbs,isteer,lwgth ,icoord, id, id2a0, id2a1, ida0, ida1, idb, ielk,
*ielm,ieltop,ifun,igder,igeom,ijac,ijacin
integer ilder,inf,infv,infv,itest,ishp,ki,kj,
*j, jb,jdb, jdbdb, jcoord, jdb, jd, jelk,
*jelm, jeltop, jgder, jgeom, jjac, jjacm, jlder, jntn, jshp, jshp,
*kk,kl,km,ildof,mdo,fdof,dofof,nodl,notl, ngauss, ivect, nqd, numss,
*totdof,totels,totnod,fels
integer bncodf,bnodef,bncods,bnodes,bncode,bnode
*outnod,centnod,ilderv,ifunp,ifunv,itest,ishp,ki,kj,
*ilwght,ilabss,isteer,lwgth ,icoord, id, id2a0, id2a1, ida0, ida1, idb, ielk,
*ielm,ieltop,ifun,igder,igeom,ijac,ijacin
common/cord/coord(3000,2),wcoord(500,2),wang(500)
common/dofnum/nf(7500,4),nf(7500,4),nfs(7500,2)
common/press/pinc(50),delt(50),uinc(50,7500)
common/wprops/emod,poisrt,dens
common/ress/sk(7500),chk(7500)
common/fron1/nodref,totdof,totnod,totels,fels,dofel,mwga,
*ntra,nout,dimen,nodel,ildof,mdo,fdof,dofof,hband,nelr,nelt
common/fron2/eltop(500,10),bval(500),bvalf(500),bval(500),
*elq(70),bncode(500),bnode(500),bc(7500)
*bncodef(500),bnodef(500),bcf(7500),bcf1(7500)
*bncode(500),bnodes(500),bcs(7500)
common/fron4/mdf(2000),nopp(7500),nopp(7500),nopp(7500)
common/fron5/nopf(500,8),nk(8,3),eik(28,28)
common/factr/alpha,beta,gamma
common/cpi/nodint,intnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnod,coutnode
common/epig/nodint,intnod(200),outnod(200),centnod(200)

Wall Elements
common/const/cc1,cc2,cc3,c4,c5,c6,c7
common/time/time,istep,nsteps,deltim
d2a1(2000)
common/rhvr1(7500),vecp(7500),vec1(7500),vec2(7500)
common/pfact/pfact(3)

Problem size-dependent arrays

data iabss/2,ib/3,iibt/16/iibt/db/16/id/3/idb/3,ielk/28/
*ilderv/2,ifunp/4,ifunv/8/
dum1/16,idum2/16,ielw/16/ink/8/jdum1/16/jdum2/16/
jelw/16/jnk/2, ilwght/2/ilabss/2/
*intn/16/ishp/2/isteer/16/itshp/16/ielk/28/jelm/16/jgder/8/
*jgeom/2/jjac/2/jjacin/2/jlderv/8/jntn/16/jshp/16/jshp/2/
nin/1/dofnod/2/dimen/2/ngauss/4/numss/3/nqd/2/
*,ivect/7500,ir1/7500/

Problem size dependent data statements

data ia0/2000,ia1/2000,icoord/2000/id2a0/2000/id2a1/2000/
*ida0/2000/ida1/2000,ieltop/500/infs/7500/
*jcoord/2,ieltop/10/jnfs/2/
do 1102 i=1,nodel
ij=eltop(nell,i+2)
wrad(ij)=coord(ij,1)
wang(ij)=coord(ij,2)
wcoord(ij,1)=wrad(ij)*dcos(wang(ij))
wcoord(ij,2)=wrad(ij)*dsin(wang(ij))
geom(i,1)=wcoord(ij,1)
geom(i,2)=wcoord(ij,2)
1102 continue

Stress-strain matrix D

d(1,1)=emod/(1.0d0-poisrt**2)
d(1,2)=d(1,1)*poisrt
d(1,3)=0.0d0
d(2,1)=d(1,2)
d(2,2)=d(1,1)
d(2,3)=0.0d0
d(3,1)=0.0d0
d(3,2)=0.0d0
d(3,3)=emod/(2.0d0*(1.0d0+poisrt))
dofel=16
callaqua4(wght, iwght, abss, iabss, jabss, ngauss, 'itest)
Integration loop for element stiffness

call matnul( elk, ielk, jelk, ielk, jelk, itest)
call matnul(elm, ielm, jelm, ielm, jelm, itest)
call matnul(elw, ielw, jelw, ielw, jelw, itest)
call matnul(duml,16,16,16,16,itest)
call matnul(durn2,16,16,16,16,itest)

c Gauss Quadrature
c Form linear shape function and space
derivatives in local coordinates.
c Transform local derivatives to global
c coordinate system
xi = abss(1,iquad)
eta = abss(2,iquad)
c call quam8(fun, ifun, lder,ilder,jlder, xi, eta, itest)
c call matmul(lder,ilder,jlder,geom,igeom,jgeom,jac,
* ijac, jjac, dimen, nodel, dimen, itest)
c call matinv(jac, ijac, jjac, jacin, ijacin, jjacin, dimen,
* det, itest)
c call matmul(jacin, ijacin, jjacin, lder,ilder,jlder, gder,
*igder, jgder, dimen, dimen, nodel, itest)

Strain-displacement matrix B
and formation of integrand
for element stiffness matrix ELK

call b2c2(b,ib,jb,gder,igder,jgder,nodel,itest)
call matmul(d, id, jd, b, ib, jb, db, idb, jdb, id,
* jd, jdb, itest)
c call matran(b, ib, jb, bt, ibt, jbt, ib, jb, itest)
c call matmul(bt, ibt, jbt, db, idb, jdb, btdb, ibtdb, jbt db,
* dofel, numss, dof el, itest)

Formation of integrand for
element mass matrix ELM

call shapfn(shp, ishp, jshp, fun, ifun, nodel, dofnod, itest)
call matran(shp, ishp, jshp, tshp, itshp, jtshp, dofnod,
* dof el, itest)
c call matmul(tshp, itshp, jtshp, shp, ishp, jshp, ntn, intn,
* jtn, dof el, dofnod, dofel, itest)
quot=dabs(det)*wght(iquad)
do 580 i=1,dofel
do 570 j=1,dofel
if(i.eq.j) then
ntn(i,j) = 1.5d0*ntn(i,j)*quot*dens
ntn(i,j) = 0.0d0
else
ttn(i,j) = 0.0d0
end if
btdb(i,j) = btdb(i,j)*quot
continue
continue
Assembly of system stiffness matrix
ELK, + system mass matrix ELM

do 303 i=1,16
do 304 j=1,16
elk(i,j) = elk(i,j) + btdb(i,j)
continue
continue
call matadd(elm,16,16,ntn,16,16,dofel,dofel,itest)
continue
continue
cc 1 = alpha + 1.0d0/(gamma*deltim)
cc 1 = 0.0d0
cc 2 = beta + gamma*deltim
cc 2 = 1.0d0
cc 3 = (1.0d0-gamma)*deltim
cc 3 = 0.0d0
cc 4 = beta - (1.0d0-gamma)*deltim
cc 4 = 1.0d0
cc 5 = 1.0d0/(gamma*deltim)
cc 5 = 0.0d0
cc 6 = (1.0d0-gamma)/gamma
cc 6 = 0.0d0
cc 7 = gamma*deltim
cc 7 = 0.0d0
do 512 i=1,dofel
do 513 j=1,dofel
elw(i,j) = cc1*elm(i,j) + c4*elk(i,j)
elk(i,j) = cc1*elm(i,j) + cc2*elk(i,j)
elm(i,j) = elm(i,j)/gamma
continue
continue
c
9075 format(2i5,f10.5)
c
Assemble element stiffness matrix
c
k=1
do 173 i=1,nodel
ki=nfs(iabs(nop(nell,i)),1)
kp=nfs(iabs(nop(nell,i)),2)
l=1

do 174 j=1,nodel

kk=nfs(iabs(nop(nell,j)),1)
kl=nfs(iabs(nop(nell,j)),2)
vec1(ki)=vec1(ki)+elm(k,l)*da0(kk)
1+elw(k,l)*a0(kk)

vec1(kj)=vec1(kj)+elm(k+1,l)*da0(kk)
1+elw(k+1,l)*a0(kk)

l=l+2

k=k+2

174 continue

173 continue

call vecnul(vec1,7500,7500,itest)
s1=coord(iabs(nop(nell,1)),2)
s3=coord(iabs(nop(nell,7)),2)
r2=coord(iabs(nop(nell,8)),1)
ds=r2*(s1-s3)
ij=nff(iabs(nop(nell,1)),4)

ik=nff(iabs(nop(nell,7)),4)
p0=(uinc(istep)-uinc(istep-1))/6.0d0
p1=(uinc(istep,ij)-uinc(istep-1,ij))/6.0d0
p3=(uinc(istep,ik)-uinc(istep-1,ik))/6.0d0
kk=nfs(iabs(nop(nell,1)),1)

kl=nfs(iabs(nop(nell,8)),1)

krn=nfs(iabs(nop(nell,7)),1)

jk=nfs(iabs(nop(nell,1)),2)

jl=nfs(iabs(nop(nell,8)),2)

jm=nfs(iabs(nop(nell,7)),2)

ll=iabs(nop(nell,8))

lm=iabs(nop(nell,7))

ln=iabs(nop(nell,1))

vec2(kk)=(p1+p0)*ds*cos(wang(ll))

vec2(kl)=2.0d0*(p0+p1+p3)*ds*cos(wang(lm))

vec2(km)=(p0+p3)*ds*cos(wang(ln))

vec2(jl)=2.0d0*(p0+p1+p3)*ds*sin(wang(lm))

vec2(jm)=(p0+p3)*ds*sin(wang(ln))

r1(kk)=r1(kk)+vec1(kk)+vec2(kk)

r1(kl)=r1(kl)+vec1(kl)+vec2(kl)

r1(km)=r1(km)+vec1(km)+vec2(km)

r1(jk)=r1(jk)+vec1(jk)+vec2(jk)

r1(jl)=r1(jl)+vec1(jl)+vec2(jl)

r1(jm)=r1(jm)+vec1(jm)+vec2(jm)

return

dend
C.3.2 PFECTL

Program PFECTL - Pulsatile Flow in Elastic Curved Tubes - Longitudinal Section

- Isoviscous, incompressible flow with inertia
- Newtonian fluid
- Toroidal coordinate system
- Curved Tube Longitudinal section

```fortran
1 program pfectl
2 implicit double precision(a-h,o-z)
3 integer dofel,totdof,totels,fels,tnodes,ij,jk,jl,im,nwall,wallbc,hband
4 integer numint,intnod,outnod,centnod
5 integer bdcnd,blist,btype,dofnod,eltop,dimen,
6 * dirich,totnod,bnode,ndof,nodsid,ichk,il,ivect,ir1
7 *,nelr,neln,nf,nfs,nodnum,istep,nsteps,icoord
8 *,bncodf,bnodef,bncods,bnodes
9 *,idof,fdof,sdof,neuman
10 *,nout,ntra,mwga
11 *,iel,itin
12 *,inf,irhv,ibdcnd,jbdcnd
13 *,nin,eltop,eltop,nopp,nof,nops,nk,
14 *neln,nelb,nbel,nharm,nodref,nodef,mdf,nop,ncodf,ncods,ncods,itype,
15 *itest,jblist,jblist
16 *,numnod,nodcod,elnum,ns
17 *,ii,ij,ik,k
18 *Character linfo*200
19 integer ios
20 dimension nodcod(6000)
21 1,convge(7500),bb1(1000)
22 common/dofnum/nf(7500,4),nff(7500,4),nfs(7500,2)
23 common/cord/coord(3000,2),wcoord(500,2),wang(500)
24 common/press/pinc(50),delt(50),uinc(50,7500)
25 common/time/totime,istep,nsteps,deltim
26 common/bounds/btype(5),bdcnd(5,200),blist(200,5),numnod(200)
27 1,neuman,nl,nel(20),nbel,nharm
28 common/pfact/pfact(3)
29 common/props/amu,rho,relr,0,remax
30 common/results/sk(7500),chk(7500)
31 common/fron1/nodref,eltop,eltop,totels,fels,dofel,mwga,
32 *ntra,nout,dimref,nodef,fdof,sdof,hband,nelt
33 common/fron4/ndof(2000),nopp(7500),nopf(7500),nops(7500)
34 common/fron2/eltop(500,10),bval(500),bvalf(500),bvals(500),
35 *elq(70),bnode(500),bnodef(500),bc(2500)
36 *bncods(500),bnodef(500),bcf(2500),bcfl(2500)
37 *,bncodf(500),bnodef(500),bnode(500),bncod(500),bncodf(500),
38 common/fron5/nopf(500,8),nk(8,3),elk(28,28)
39 common/fron6/nodc(2500),ncodf(2500),ncodf(2500),ncodf(2500).

C.42
common/factrs/alpha,beta,theta
id *d2a1(2000)
common/rhvect/r1(2500),vect(2500)
common/wprops/emod,poisrt,dens
common/epi/nunit,intnod(200),outnod(200),centnod(200)
common/const/cc1,cc2,cc3,c4,c5,c6,c7

Problem size dependent data statements

data ibdcnd /5/, iblist /50/, icoord /2500/, ieltop /500/,
* inf /2500/, irhs/2500/, jbdcnd /200/, itype/1/,
* jblist /5/, jeltop /10/, nin/1/,nodsid/3/, ivect/2500/,ir1/2500/

open(file='mesh.dat',unit=nin,form='formatted',
+ status='old',iostat=ios,err=199)
open(file='fluidout',unit=10,form='formatted')
open(file='meshout.dat',unit=11,form='formatted')
open(file='wallout',unit=18,form='formatted')
open(file='velout.dat',unit=13,form='formatted')
open(file='presout.dat',unit=14,form='formatted')
open(file='deflout.dat',unit=15,form='formatted')
open(file='dynout.dat',unit=16,form='formatted')
open(file='diag',unit=17, form='formatted')
open(unit=4,form='unformatted',status='scratch')

Set itest for full checking

itest = 0

******************************
*   *
* Input data section  *
*   *
******************************

Input of nodal geometry

xmax=0.0d0
ymax=0.0d0
read(nin,8010) totnod,dimen
defnod=2
read(nin,8010) nrows,ncols,nwall
do 1010 i=1,totnod
read(nin,8020) nodnum, (coord(nodnum,j),j=1,dimen)
write(11,8020) nodnum, (coord(nodnum,j),j=1,dimen)
if(coord(nodnum,1).gt.xmax) then
xmax=coord(nodnum,1)  
end if  
if(coord(nodnum,2).gt.ymax) then  
ynmax=coord(nodnum,2)  
end if  
1010 continue  
Set up wall coordinates  
do 9023 i=1,totnod  
cords(i,1)=coord(i,2)  
cords(i,2)=coord(i,1)  
9023 continue  
Topology  
read(nin,8010) totels,fels  
do 1020 i=1,totels  
elnum=elnum+1  
read(nin,8010) num,nodel,(eltop(elnum,j+2),j=1,nodel)  
eltop(elnum,1) = nodel  
eltop(elnum,2) = nodel  
1020 continue  
Identify corner nodes  
do 1019 elnum=1,totels  
do 1019 j=3,10  
nodcod(eltop(elnum,j))=0  
1019 continue  
do 1021 elnum=1,totels  
nodcod(eltop(elnum,3))=1  
nodcod(eltop(elnum,5))=1  
nodcod(eltop(elnum,7))=1  
nodcod(eltop(elnum,9))=1  
1021 continue  
do 1112 i=1,totels  
jj=eltop(i,2)+2  
do 1112 j=3,jj  
mdf(eltop(i,j))=nodcod(eltop(i,j))+3  
nop(i,j-2)=eltop(i,j)  
1112 continue  
nodref=ncols*nrows-totels  
Set up element freedom number array  
read(nin,8011) r0  
read(nin,8011) amu
108       read(nin,8011) rho
109       read(nin,8081) emod,poisrt
110       read(nin,8081) dens
111       read(nin,8010) totncs
112       read(nin,8011) totime
113       read(nin,8010) ns
114       nsteps=ns+1
115       deltim=totime/ns
116       pinc(1)=0.0000d0
117       pmax=0.0d0
118       do 1 1051
119          i=2,nsteps
120          read(nin,9037) pinc(i)
121          if(pinc(i).gt.pmax) then
122             pmax=pinc(i)
123          end if
124          continue
125       re=rho/amu
126       ii=fe1s+1
127       jj=ii+ncols/2-1
128       kk=totels-ncols/2+1
129       numint=ncols
130       k=1
131       intnod(k)=iabs(nop(ii,1))
132       outnod(k)=iabs(nop(kk,3))
133       do 1100
134          i=ii,jj
135          k=k+1
136       end do
137       intnod(k)=iabs(nop(i,8))
138       outnod(k)=iabs(nop(kk,4))
139       k=k+1
140       intnod(k)=iabs(nop(i,7))
141       outnod(k)=iabs(nop(kk,5))
142       kk=kk+1
143       1100 continue
144       write(11,8010) (intnod(k),outnod(k),k=1,numint)
145       totdof=0
146       fdof=0
147       sdof=0
148       k=0
149       do 2000 i=1,nodref
150          if(coord(i,2).lt.0.0001d0) k=k+1
151          if(i-intnod(k)) 1102,1103,1104
152          1102 i=1
153          1103 do 1125 j=1,mdf(i)
154             fdof=fdof+1
155             nff(i,j)=fdof
156             1125 continue
157          1104 do 1126 m=1,2
158             sdof=sdof+1
159          1126 continue
160          1102 continue
161       2000 continue
nfs(i,m)=sdof
continue
ii=0
2000 continue
totdof=fdo+sdof
write(*,9021) sdof,fdo
9021 format(9'Degrees of freedom - Solid',i5,' Fluid',i5/)
do 4551 i=1,fels
do 4552 j=1,nodel
k=abs(nop(i,j))
nopf(k)=nff(k,1)
do 4552 continue
do 4551 continue
do 4553 i=fels+1,totels
do 4554 j=1,nodel
k=abs(nop(i,j))
nops(k)=nfs(k,1)
do 4554 continue
do 4553 continue
c Dirichlet boundary conditions
c
call vecnul(ncodf,inf,inf,itest)
read(nin,8010) dirich
if(dirich.eq.0) go to 9874
do 1030 i=1,dirich
read(nin,8012) bnodef(i),bncodf(i),bvalf(i)
ndof=nopf(bnodef(i))-1+bncodf(i)
ncodf(ndof)=1
bcs(ndof)=bvalf(i)
do 1030 continue
9874 continue
nharm=0

c Wall boundary conditions
c
call vecnul(ncods,inf,inf,itest)
read(nin,8010) wallbc
if(wallbc.eq.0) go to 9877
do 1033 i=1,wallbc
read(nin,8012) bnodes(i),bnods(i),bnods(i),bvals(i)
ndof=nops(bnodes(i))-1+bnods(i)
ncods(ndof)=1
bcs(ndof)=bvals(i)
do 1033 continue
9877 continue

C.46
Neumann boundary conditions

196 read(nin,8010) neuman
197 if(neuman.eq.0) then
198 go to 9876
199 end if
200 do 1035 i=1,neuman
201 read(nin,8009) pfact(i)
202 read(nin,8010) btype(i),numnod(i),(bdcnd(i,j+3),j=1,numnod(i))
203 bdcnd(i,2)=nodsid
204 bdcnd(i,3)=numnod(i)
205 1035 continue
206 9876 continue
207 do 1049 ii=1,totdof
208 uinc(1,ii)=0.0d0
209 1049 continue
210 do 1048 ii=1,sdof
211 a0(ii)=0.0d0
212 da0(ii)=0.0d0
213 d2a0(ii)=0.0d0
214 a1(ii)=0.0d0
215 da1(ii)=0.0d0
216 d2a1(ii)=0.0d0
217 1048 continue
218 alpha=1.0d0
219 beta=1.0d0
220 theta=0.5d0
221 ichk=totdof
222 write(*,1001)
223 1001 format(' Calling Front ')
224 do 5555 ii=1,ichk
225 chk(ii)=0.00d0
226 5555 continue
227 3208 continue

c c Time Step Loop
c
c228 time=0.0d0
229 do 4000 istep=2,nsteps
230 do 4555 i=1,dirich
231 ndof=nopf(bnodef(i))-1+bncodf(i)
232 if(neuman.eq.0) then
233 bcf(ndof)=bcfl(ndof)*pinc(istep)
234 else
235 bcf(ndof)=bcfl(ndof)
236 end if
237 4555 continue
238 ntra=1
239 iel=0

C.47
time=time+deltim
write(*,1041) istep-1, time
write(*,9056) pinc(istep)

1041 format(' Time Step No.,i2,2x,Elapsed Time =',f7.4,' secs')
c
Velocity Iteration Loop
c
4750 do 5000 ninc=1,totncs
totncs
if(iel.eq.1) then
go to 4760
end if
write(*,9055) ninc,istep-1

4760 call front(ninc,iel)
if(iel.eq.1) go to 5600

Check convergence

conavg=0.0
do 3600 i=1,f dof
if(chk(i).le.1.0d-012) go to 3600
convge(i)=(sk(i)-chk(i))/chk(i)
convvg=convvg+convge(i)
3333 format(' Average Convergence = ',f10.5/)
3600 continue
dof
write(*,3333) convvg
do 3610 i=1,f dof
chk(i)=sk(i)
3610 continue
dof
dchksum=0.0d0
do 3611 i=1,totnod
if(nf(i,1).eq.0) then
go to 3611
end if
chksum=chksum+chk(nf(i,1))
3611 continue
dof
rmax=coord(iabs(nop(fels,3)),2)
rey=chksum/totnod*rmax*2.0d0*rho/amu
dean=2.0d0*rey*(rmax/ro)**0.5d0
omstar=2.0d0*3.14159/totime*rmax**2*rho/amu
write(*,3612) abs(rey),omstar,dean
3612 format(' Reynolds Number = ',f5.0,3x,'Omega* = ',f5.0
1,3x,'Dean Number = ',f5.0/)
if(abs(convvg).lt.1.0e-03) then
go to 4500
end if
5000 continue
4500 do 1042 i=1,fels
        do 1043 j=1,node!
        do 1044 k=1,mdf(iabs(nop(i,j)))
            l=nff(iabs(nop(i,j)),k)
            uinc(istep,l)=chk(l)
            continue
        continue
    continue
    continue
    continue
    ntra=1
    iel=1
    do 301 i=1,totnod
        write(10,9035) i,(sk(nff(ij))j=2,3)
        write(13,8021) i,sk(nff(i,1)),sk(nff(i,2)),sk(nff(i,3))
        write(14,9019) i,sk(nff(i,4))
        write(18,20) i,sk(nfs(i,1)),sk(nfs(i,2))
        go to 4750
    continue
    5600 continue
    do 295 i=fels+1,totel
    do 294 j=1,node!
    do 293 k=1,2
        a1(k)=sk(k)
        da1(k)=c5*(a1(k)-a0(k))-c6*da0(k)
        d2a1(k)=c5*(da1(k)-da0(k))-c6*d2a0(k)
        continue
    continue
    continue
    c Adapt fluid mesh to wall profile
    c
    j=1
    do 399 i=1,totnod
        if(coord(i,2).eq.0) then
            centnod(j)=i
            j=j+1
        end if
    399 continue
    m=1
    do 399 ij=1,numint
        jk=intnod(ij)
        jm=centnod(ij)
        do 302 jl=jm,jk
        do 300 k=1,2
            if(k.eq.1) then
                kk=2
            else
                kk=1
            end if
        300 continue
end if
l=nfs(jl,k)
bbl(l)=a1(nfs(jk,kk))*coord(jl,2)/coord(jk,2)
coord(jl,k)=coord(jl,k)+bbl(l)
300 continue
302 continue
299 continue
336 call vecnul(sk,inf,inf,itest)
do 298 ij=1,numint
m=0
jk=intnod(ij)
do 290 jl=1,nwall
do 291 i=1,2
if(l.eq.1) then
11=2
else
11=1
end if
lm=nfs(jk+m,11)
coord(jk+m,1)=coord(jk+m,1)+a1(lm)
291 continue
m=m+1
290 continue
344 continue
345 ll=1
346 end if
347 ll=2
348 coord(jk+m,l)=coord(jk+m,l)+a1(lm)
291 continue
350 m=m+1
351 290 continue
352 298 continue
353 do 304 i=1,totnod
write(15,8020) i,coord(i,1),coord(i,2)
354 304 continue
355 304 continue
356 call vecnul(vect,ivect,ivect,itest)
call vecnul(r1,ir1,ir1,itest)
call veccop(a1,2000,a0,2000,2000,itest)
call veccop(da1,2000,da0,2000,2000,itest)
call veccop(d2a1,2000,d2a0,2000,2000,itest)
4000 continue
print *,linfo
199 write(*,3456)
3456 format( Stopping at end of increment loop/)
stop
20 format(i5,2f10.5)
8009 format(f10.5)
8010 format(200i5)
8011 format(2f10.3)
8081 format(2d10.3)
8012 format(2i5,f10.3)
8015 format(4i5,2f10.5)
8020 format(i5,2f10.6)
8021 format(i5,3f10.5)
9010 format//:25h **** Nodal Geometry ****//1h
9012 format(2i5,f10.5)
9019 format(i5,f10.5)
subroutine front(ninc,iel)
  implicit double precision (a-h,o-z)
  integer totdof,totels,dofel,
  fels,idof,fdof,sdof,hband,nrows,ncols
  integer eltop,dimem,
  *totnod,bncode,bnode
  *bncodf,bnodef,bncods,bnodes
  *iel,ninc
  *nopp,nopf,nops,nk,
  *nodref,nodel,mof,nop,ncod,ncodf,ncods,
  *itest
  *ii,kk
  *i,j,k,l,m,n
  *inf,immk,ir,kl,kpi,kpivc,kpivco,kpivro,kco
  *outnod,centnod,ir1,nmax,ncrit,na,nb,nlast,nnl1,nerror,11,
  *locj,krow,nell,kc,nj,ntr,khed,ihed,idi,nn,nnode
  *ldest,kdest,le,kl,kpiv,ir,kl,kpiv,kro
  *krw,kf,nff,nfs,mwga,ntra,nout,
  *numint,intnod,inf
  common/dofnum/nf(2500,4),nff(2500,4),nfs(2500,4)
  common/props/amu,ro,rec0,rmnax
common/reslt5/sk(5000),chk(5000)
common/fro1/nodref,totdof,totnod,totels,fels,dofel,mwga,
common/fro2/eltop(500,10),bval(5000),bvalf(5000),bvals(5000),
common/fro2/eltop(500,100),bval(500),bvalf(500),bvals(500),
common/fro2/eltop(500,100),bval(500),bvalf(500),bvals(500),
common/fro2/eltop(500,100),bval(500),bvalf(500),bvals(500),
common/fro2/eltop(500,100),bval(500),bvalf(500),bvals(500),
common/fro4/mdf(2500),nopf(2500),nops(2500)
common/fro5/nopf(2500),nk(8,3),elk(28,28)
common/fro6/ncod(2500),ncodf(2500),ncods(2500)
common/factrs/alpha,beta,theta
common/epi/numint,noutnod(200),centnod(200)
common/rhvect/r1(2500),vect(2500)
common/pfact/pfact(3)
dimension
11dest(1000),kdest(1000)
2.eq(1500,1500),lhed(1500),khed(1500),lpiv(1500),kpiv(1500)
3,jmod(1500),qq(1500)
4,nj(1500)
data inf/2500/jinf/4/
lirl/2500/
nmax=900
call vecnul(r1,lrl,r1,rl1,ltest)
c                Prefront
ncrit=nmax-28
write(12,4000)
if(iel.eq.0) then
    nell=0
    idof=fdo1
    na=1
    nb=fels
    do 1040 i=1,inf
        nopp(i)=0
        nopp(i)=nopf(i)
    do 1040 j=1,jinf
        nf(i,j)=0
        nf(i,j)=nff(i,j)
    1040 continue
    do 1042 i=1,inf
        ncod(i)=0
        bc(i)=0.0d0
        bc(i)=bcf(i)
        ncod(i)=ncodf(i)
    1042 continue
else
nell = fels
idof = sdof
na = fels + 1
nb = totals

\[\text{do 1041 } i=1,\text{inf} \]
\[nopp(i) = 0 \]
\[nopp(i) = \text{nops}(i) \]
\[\text{do 1041 } j=1,\text{inf} \]
\[nf(i,j) = 0 \]
\[nf(i,j) = \text{nfs}(i,j) \]
\[1041 \text{ continue} \]
\[\text{do 1043 } i=1,\text{inf} \]
\[ncod(i) = 0 \]
\[bc(i) = 0.0 \text{d0} \]
\[bc(i) = \text{bcs}(i) \]
\[ncod(i) = \text{ncods}(i) \]
\[1043 \text{ continue} \]
\[\text{end if} \]
\[\text{if(ntra.eq.0) go to 14} \]
\[\text{Find last appearance of each node} \]
\[\text{c} \]
\[\text{nlast} = 0 \]
\[\text{do 12 } i=1,\text{nodref} \]
\[\text{do 8 } n=na,nb \]
\[\text{do 4 } l=1,\text{eltop}(n,2) \]
\[\text{if(nop}(n,l).ne.i)\text{go to 4} \]
\[\text{nlast} = n \]
\[\text{if(nlast.ne.nlast1)go to 3} \]
\[\text{error} = 1 \]
\[\text{write}(12,416)\text{error,n} \]
\[\text{stop} \]
\[\text{3 continue} \]
\[\text{nlast} = n \]
\[\text{l1} = l \]
\[\text{4 continue} \]
\[\text{8 continue} \]
\[\text{if(nlast.eq.0) go to 12} \]
\[\text{nop}(\text{nlast},l1) = -\text{nop}(\text{nlast},l1) \]
\[\text{nlast} = 0 \]
\[\text{12 continue} \]
\[\text{ntra} = 0 \]
\[\text{Assembly} \]
\[\text{c} \]
\[\text{14 continue} \]
\[\text{lcol} = 0 \]
\[\text{krow} = 0 \]
\[\text{do 16 } i=1,nmax \]
do 16 j=1,nmax
eq(j,i)=0.
16 continue

18 nell=nell+1
if(iel.eq.0) then
call fluidl(nell,ninc)
write(*,1002) nell
else
call solidl(nell,ninc)
write(*,1003) nell
end if

n=nell
kc=0
mwp=0
if(mwp.eq.0)go to 21
do 20 i=1,eltop(n,2)
nj(i)=nop(n,i)
20 continue
kc=eltop(n,2)
go to 23

21 continue
do 22 j=1,eltop(n,2)
nn=nop(n,j)
m=iabs(nn)
k=nopp(m)
idf=mdf(m)
if(iel.eq.1) idf=2
do 221=1,idf
kc=kc+1
ii=k+l-1
if(nn.lt0)ii=-ii
nj(kc)=ii
22 continue
23 continue

Set up heading vectors

ntr=0
if(ntr.eq.1) write(12,420) krow,lcol
if(ntr.eq.1) write(12,424)
if(ntr.eq.1) write(12,428) (khed(k),lhed(k),k=1,lcol)
if(ntr.eq.1) write(12,432)
if(ntr.eq.1) write(12,428) (kdest(k),ldest(k),k=1,kc)
do 52 lk=1,kc
node=nj(lk)
if(lcol.eq.0)go to 28
do 24 l=1,lcol
ll=l
if(iabs(node).eq.iabs(lhed(l)))go to 32
556  24 continue
557  28 lcol=lcol+1
558     ldest(lk)=lcol
559     lhed(lcol)=node
560     go to 36
561  32 ldest(lk)=ll
562     lhed(ll)=node
563  36 if(krow.eq.0)go to 44
564     do 42 k=1,krow
565     kk=k
566     if(iabs(node).eq.iabs(khed(k)))go to 48
567     42 continue
568     44 continue
569     krow=krow+1
570     kdest(lk)=krow
571     khed(krow)=node
572     go to 52
573     48 continue
574     kdest(lk)=kk
575     khed(kk)=node
576     52 continue
577     cc if(ntr.eq.1) write(12,420)krow,lcol
578     cc if(ntr.eq.1) write(12,424)
579     cc if(ntr.eq.1) write(12,428) (khed(k),lhed(k),k=1,nmax)
580     cc if(ntr.eq.1) write(12,432)
581     cc if(ntr.eq.1) write(12,428) (kdest(k),ldest(k),k=1,kc)
582     if(krow.lt.nmax.and.lcol.le.nmax)go to 54
583     nerror=2
584     write(12,417)nerror
585     stop
586     54 continue
587     do 56 l=1,kc
588     ll=ldest(l)
589     do 56 k=1,kc
590     kk=kdest(k)
591     eq(kk,ll)=eq(kk,ll)+elk(k,l)
592     56 continue
593     c write(12,*) nell,(eq(i,i),elk(i,i),i=1,16)
594     cc write(12,436)nell
595     if(krow.lt.ncrit.and.nell.lt.nb)go to 18
596     c
597     c Find out which matrix elements are fully summed
598     c
599     60 lc=0
600     do 64 l=1,lcol
601     lc=lc+1
602     64 continue
 Modify equations with applied boundary conditions

 Search for absolute pivot

1988 format(5e20.10)
pivot=0.0d0
do 76 l=1,lc
lpivc=lpiv(l)
kpivr=lpivc
piva=eq(kpivr,lpivc)
if(abs(piva).lt.(pivot)) go to 76
pivot=piva
lpivco=lpivc
kpivro=kpiivr
76 continue
1989 format(//e20.10/)
c
Normalise pivotal row
c
kro=iabs(khed(kpivro))
lco=iabs(khed(lpivco))
if(ntr.eq.1)write(12,452)kro,lco,pivot,nell
write(*,476)
write(*,8787) pivot
8787 format(/'Pivot = ',e10.5)
stop
end if
do 80 l=l,lcol
qq(l)=eq(kpivro,l)/pivot
80 continue
rhs=rl(kro)/pivot
rl(kro)=rhs
c write(12,468)
c write(12,440)(qq(l),l=l,Icol)
c
Eliminate then delete pivotal row and column
c
if(kpivro.eq.1)go to 104
kpivr=kpivro-1
do 100 k=1,kpivr
krw=iabs(khed(k))
fac=eq(k,lpivco)
c write(12,480)fac
if(lpivco.eq.1.or.dabs(fac).lt.1.0d-20) go to 88
lpivc=lpivco-1
do 84 l=1,lpive
eq(k,l)=eq(k,l)-fac*qq(l)
84 continue
88 if(lpivco.eq.lcol)go to 96
lpivc=lpivco+1
do 92 l=lpivc,lcol
eq(k,l-1)=eq(k,l)-fac*qq(l)
92 continue
96 rl(krw)=rl(krw)-fac*rhs
100 continue
104 if(kpivro.eq.krow)go to 128
kpivr=kpiivr+l
do 124 k=kpiivr,krow
krw=iabs(khed(k))
fac=eq(k,lpivco)
 If(lpivco.eq.1) go to 112

lpivc=lpivco-1

do 108 l=1,lpivc

eq(k-1,l)=eq(k,l)-fac*qq(l)

108 continue

if(lpivco.eq.lcol) go to 120

lpivc=lpivco+1

do 116 l=lpivc,lcol

eq(k-1,l-1)=eq(k,l)-fac*qq(l)

116 continue

r1(krw)=r1(krw)-fac*rhs

120 continue

Write pivotal equation on disc

write(4)
kro,lcol,lpivco,(lhed(l),qq(l),l=1,lcol)
2,krow,pivot,kpivro,(khed(k),k=1,krow)
do 129 k=1,krow

eq(k,lcol)=0.

129 continue

do 130 l=1,lcol

eq(krow,l)=0.

130 continue

Write(12,436) nell

write(12,440)((eq(i,j),j=1,nmax),i=1,nmax)

write(12,460)

Rearrange heading vectors

lcol=lcol-1

if(lpivco.eq.lcol+1) go to 136

do 132 l=lpivco,lcol

lhed(l)=lhed(l+1)

132 continue

krow=krow-1

if(kpivro.eq.krow+1) go to 144

do 140 k=kpivro,krow

khed(k)=khed(k+1)

140 continue

144 continue

write(12,420)krow,lcol

write(12,424)

write(12,428) (khed(k),lhed(k),k=1,nmax)

C.58
Determine whether to assemble, eliminate or backsubstitute

if(krow.gt.ncrit) go to 60
if(nell.lt.nb) go to 18
if(krow.gt.1) go to 60
lco=iaabs(lhed(1))
kpio=1
pivot=eq(1,1)
kro=iaabs(khed(1))
lpivco=1

if(ntr.eq.I) write(*,452) lco, kro, pivot, nell
if(abs(pivot).lt.1e-20) then
write(*,476) stop
end if
rl(kro)=ri(kro)/pivot
write(4)
1 kro, lcol, lpivco, lhed(1), qq(1)
2, krow, pivot, kpio, khed(1)
write(*,1004)
1002 format(' Assembling FLUID Element No. ', i3)
1003 format(' Assembling SOLID Element No. ', i3)
1004 format(' Calling Back Substitution ')
call bacsub
9999 format(i10)
400 format(5h node, 6h nlast)
404 format(1x, 2i5)
408 format(/16h nodal numbering/)
412 format(9i5)
416 format(/8h nerror=, i5//)
1 62h the element has more than one node with the
1/62h same nodal number
1/62h the difference nmax-ncrit is not sufficiently large
1/62h to permit the assembly of the next element---
1/62h either increase nmax or lower ncrit
1/62h there are no more rows fully summed, this may be due to---
1/62h (1) incorrect coding of nop or nk arrays
1/62h (2) incorrect value of ncrit. Increase ncrit to permit
1/62h whole front to be assembled
1/
417 format(/8h nerror=, i5//)
1/62h there are no more rows fully summed, this may be due to---
1/62h (1) incorrect coding of nop or nk arrays
1/62h (2) incorrect value of ncrit. Increase ncrit to permit
1/62h whole front to be assembled
1/
418 format(/8h nerror=, i5//)
420 format(/6h krow=, i5, 6h lcol=, i5/)
424 format(/5h khed, 5h lhed/)
subroutine bacsub
  implicit double precision (a-h,o-z)
  integer totdof,totnod,totels,eltop,dofel,
  *hband,fels,dimen
  * ,idof,fdof,sdof
  *,nopp,nopf,nops,nk,
  *nodref,nodel,mfd,nop,ncod,ncodf,ncods,
  *itest
  integer bncodf,bnodef,bncods,bnodes,bncode,khed,l
  *,nf,nff,nfs,mwga,ntra,nout,iv ,lhed
  *,i,k,kro,lcol,lpivco,krow,kpivro,bnode
  common/dofnurnlnf(2500,4),nff(2500,4),nfs(2500,4)
  commonlprops/amu,rho,re,rO,rmax
  commonlreslts/sk(5000),chk(5000)
  commonlfronllnodref,totdof,totnod,totels,fels,dofel,mwga,
  *ntra,nout,dimen,nodel,idof,fdof,sdof,hband,nrows,ncols
  common/lfron2/eltop(500,10),bval(500),bvalf(500),bvals(500),
  *elq(70),bncode(500),bnodes(500),bc(2500)
  * ,bncof(500),bnodef(500),bcof(2500),bcfl(2500)
  * ,bncods(500),bnodes(500),bcs(2500)
  common/lfron4/mdf(2500),nopp(2500),nopf(2500),nops(2500)
  common/lfron5/nop(500,8),nk(8,3),elk(28,28)
  common/lfron6/ncod(2500),ncodf(2500),ncods(2500)
  common/lrhvect/r1(2500),vect(2500)
  dimension
  lhed(1500),khed(1500)
  2,qq(1500)
  
  c * * * * * * * * * * * * * 
  c * 
  c * Back-substitution 
  c * 
  c * * * * * * * * * * * * * 
  c subroutine bacsub
Back substitution

do 4 i=1,totdof
4 sk(i)=bc(i)
do 32 iv=1,totdof
backspace 4
read(4)
1 kro,lcol,lpivco,(lhed(l),qq(l),l=1,lcol)
2,krow,pivot,kpivro,(khed(k),k=1,krow)
backspace 4
cc write(12,404)
cc write(12,408) kro,lcol,lpivco
cc write(12,408) (lhed(l),l=1,lcol)
cc write(12,412) (qq(l),l=1,lcol)
if(ncod(kro).gt.0)then
write(12,* ) kro,ncod(kro)
go to 24
end if
gash=0.
qq(lpivco)=0.
do 16 l=1,lcol
gash=gash-qq(l)*sk(iabs(lhed(l)))
16 continue
sk(kro)=rl(kro)+gash
go to(24)
write(12,* ) kro,sk(kro)
go to 32
24 continue
ncod(kro)=1
32 continue
404 format(14h disc contents)
408 format(10i5)
412 format(5e20.10)
call vecnul(r1,2500,2500,itest)
rewind(4)
return
end

**** ****** ** **
* *
***** Element generation *****
* *
*******************************

subroutine fluidl(nell,ninc)
implicit double precision(a-h,o-z)
double precision jac,jacin,lder,labss,lderv,lwght(3)
*,mass,m1,mm1,lderp,funp
*,mm2
integer totdof,totnod,totels,dofel,bncode,bnode,eltop,ldim
*fels, nf, nff, nfs, istep, hband, outnod, centnod
*nn, nd, nend, m,
*btype, bcond, blist
*nsteps, icoord, mn, mm, kkk, ninc,
nout, ntra, mwga, jjac, ijacin, ijderv, ijderv, igderv, igderv, igderv, jgderv,
*idtpd, jdtpd
*inf, nopp, ncod, idof, fdof, sdof
*nopp, nopp, nopp, kn, numnod, ilabss, ieltop, jeltop
*test, ielk, jelk, iropiv
*nbel, nbels, nharm, nodref
*nodcod, ilal, jal, ielq, idum2, jdum2
*ll, ilarg, nbnd, ilwght
*isteer, iquad, kl, km, nlabss
*ii, jj, kk, neln, nodel, mdf
*iscvec, jcoord, ilder, jlder
*i, j, k, l, jlabss
integer bncodf, bnodef, bncods, bnodes
*nell, neuman, numint, intnod, isize, jnf
*ldmvce, ir1, nq, ngaus, lwght, nodsid, nq, np, nfac
*ifun, ifunv, ifunp, ilvder, jlder, jlder, jlder, jlder
*lder, lderp, jlder, jlder, jlder, jlder, jlder, jlder, jlder
*wdht, wght
*ell1(4,8), ell2(4,8), ell3(4,8)
*elt1(8,4), eltt2(8,4), eltt3(8,4)
*, dmvce(28), dum2(28,28)
*, lder(2,8)
*, wght(9),
*, ell1(4,8), ell2(4,8), ell3(4,8)
*, elt1(8,4), elt2(8,4), elt3(8,4)
*, dmvce(28), dum2(28,28)
*, lder(2,8)
*, c1(24,24), c2(24,24), c3(24,24),
*, mass(24,24), m1(2500)
*, mm1(24,24)
*, mm2(24,24)
*, f1(8), f2(8), f3(8), f4(8), f5(8),
*, b1(8), b2(8), b3(8), b4(8)
common/props/amu, rho, re, r0, rmax
common/dofnum/nf(2500,4), nff(2500,4), nfs(2500,4)
common/press/pinc(50), delt(50), uinc(50,3500)
common/time/totime, istep, nsteps, deltim
common/reslts/sk(5000), chk(5000)
common/cord/coord(2500,2), cords(2500,2)
common/nod/nodcod(200), ncod(200)
common/fref/nodref, totdof, totnod, totels, fels, dofel, mwga,
C.63
if(eltop(nel,2).eq.6) then
    call qtri7(wght,iwght,abss,iabss,jabss,ngaus,itest)
else if(eltop(nell,2).eq.8) then
    call qqua4(wght, iwght, abss, iabss, jabss, ngaus, itest)
end if

do 1122 i=1,nodel
    geom(i, 1)=coord( eltop(nell,i+2), 1)
    geom(i,2)=coord( eltop(nell,i+ 2),2)
1122 continue

integration loop for element matrices

do 1090 iquad=1,ngaus

form shape function and space
derivatives in the local coordinates.
transform local derivatives to global
coordinate system

\[ \xi = \text{abss}(1, \text{iquad}) \]
\[ \eta = \text{abss}(2, \text{iquad}) \]

if(elttop(nell,2).eq.6) then
  call trim6(fun,ifun,lderv,ilderv,jlderv,xi,eta,itest)
else if(elttop(nell,2).eq.8) then
  ifun=8
  ilderv=2
  jlderv=8
  jlderp=4
  call quarn8(fun, ifun,lderv, ilderv, jlderv, xi, eta, itest)
end if

call matmul(1derv, ilderv, jlderv, geom, igeom, jgeom, jac,
  * ijac,ijac,jgeom,jlderv,ilderv,itest)

n=igeom

call scaprd(geom(1,1), igeom, fun, ifun, n, x, itest)
call scaprd(geom(l,2), igeom, fun, ifun, n, y, itest)

rg=y
rg2=rg**2
call matinv(jac, ijac, jjac, jacin, jjacin, jjacin, dimen,
  * det, itest)
call matmul(jacin, jjacin, jjacin, ijac, jlderv, jlderv, gderv,
  * igderv, jgderv, dimen, dimen, jlderv, itest)
call matran(gderv, igderv, jgderv, gdert, igdert, jgdert,
  * jgderv, jgderv, itest)

quot = dabs(det)*wght(iquad)

Form element matrix for velocity terms

k=1
l=1
do 200 i=1,nodel
  mod=coord(jabs(nop(nell,i)),2)
  b1(i)=mod/ro
  b2(i)=b1(i)**2
  b3(i)=b1(i)**3
  b4(i)=b1(i)**4
  f1(i)=1.0d0+0.75d0*b2(i)
  f2(i)=1.0d0+0.25d0*b2(i)
  f3(i)=2.0d0+b2(i)
  f4(i)=2.0d0+3.0d0*b2(i)
  f5(i)=2.0d0+6.0d0*b2(i)
  kk=nff(jabs(nop(nell,i)),1)
  kl=nff(jabs(nop(nell,i)),2)
  km=nff(jabs(nop(nell,i)),3)
do 101 j=1,nodel
dtpd(k,l)=4.0d0*gdert(i,1)*gderv(1,j)
  l+f4(i)*gdert(i,2)*gderv(2,j)

C.65
2-f4(i)/rg*fun(i)*gderv(2,j)
3+b2(i)/rg2*fun(i)*fun(j)
c 4-f4(i)/rg2*fun(j)*fun(j)
dtpd(k,l)=dtpd(k,l)/re
cl(k,l)=f4(i)*fun(i)*fun(i)*chk(kk)*gderv(1,j)
c2(k,l)=f5(i)*fun(i)*fun(i)*chk(kl)*gderv(2,j)
c3(k,l)=b2(i)/rg*fun(i)*fun(i)*chk(km)*fun(j)
if(l.eq.k) then
mass(k,l)=f4(i)*1.5d0*fun(i)*fun(j)/deltim
mm1(k,l)=mass(k,l)
end if
dtpd(k,l)=dtpd(k,l)+c 1 (k,1)+c2(k,1)+c3(k,1)
im1(k,1)=mass(k,1)
1+mass(k,1)
l=l+3
continue
k=k+3
continue
k=2
l=2
102 do 102 i=1,nodel
103 kk=nff(iabs(nop(nell,i)),1)
104 kl=nff(iabs(nop(nell,i)),2)
105 km=nff(iabs(nop(nell,i)),3)
106 do 103 j=1,nodel
107 dtpd(k,1)=2.0d0*f4(i)*gdert(i,2)*gderv(2,j)
108 +2.0d0*gdert(i,1)*gderv(1,j)
109 2.0d0*(f4(i)/rg)*fun(i)*gderv(2,j)
110 3+b2(i)/rg*fun(i)*fun(j)
dtpd(k,1)=dtpd(k,1)+c 1 (k,1)+c2(k,1)+c3(k,1)
im1(k,1)=mass(k,1)
end if
111 dtpd(k,1)=dtpd(k,1)+c 1 (k,1)+c2(k,1)+c3(k,1)
l=l+3
continue
k=k+3
continue
k=1
l=2
109 do 109 i=1,nodel
kk=nff(iabs(nop(nell,i)),1)
k1=nff(iabs(nop(nell,i)),2)
k3=nff(iabs(nop(nell,i)),3)
do 110 j=1,node1
  dtpd(k,l)=-f3(i)/rg*fun(i)*gderv(1,j)
  1+f3(i)*gdert(i,1)*gderv(2,j)
  2+3.0d0*b1(i)/rO*re*fun(i)*chk(kk)*fun(i)*fun(j)
dtpd(k,l)=dtpd(k,l)/re
dtpd(1,k)=-3.0d0*b2(i)/rg2*fun(i)*funG)
  1-re*b2(i)/rg*chk(km)*fun(i)*fun(i)*funQ)
  2.0d0*b2(i)/rg2*fun(i)*gderv(1,j)
dtpd(1,k)=dtpd(1,k)/re
l=l+3
110 continue
l=2
k=k+3
109 continue
k=2
l=3
do 119 i=1,node1
  kk=nff(iabs(nop(nell,i)),1)
k1=nff(iabs(nop(nell,i)),2)
k3=nff(iabs(nop(nell,i)),3)
do 120 j=1,node1
  dtpd(k,1)=-6.0d0*b2(i)/rg2*fun(i)*fun(j)
  1-re*b1(i)/rO*fun(i)*fun(i)*chk(kk)*funG)
  3-2.0d0*b2(i)/rg*fun(i)*gderv(2,j)
dtpd(k,l)=dtpd(k,l)/re
dtpd(1,k)=-3.0d0*b2(i)/rg2*fun(i)*fun(j)
122 1-2.0d0*b2(i)/rg*fun(i)*gderv(2,j)
dtpd(1,k)=dtpd(1,k)/re
l=l+3
120 continue
l=3
k=k+3
119 continue
k=1
l=3
do 169 i=1,node1
  kk=nff(iabs(nop(nell,i)),1)
k1=nff(iabs(nop(nell,i)),2)
k3=nff(iabs(nop(nell,i)),3)
do 170 j=1,node1
  dtpd(k,l)=-b2(i)/rg*fun(i)*gderv(1,j)
  1-re*b1(i)/rO*fun(i)*fun(i)*chk(kk)*fun(j)
dtpd(k,l)=dtpd(k,l)/re
dtpd(1,k)=-b2(i)/rg*fun(i)*gderv(1,j)
139 1-re*b1(i)/rO*fun(i)*fun(i)*chk(kk)*fun(j)
dtpd(l,k) = dtpd(l,k)/re

l = l + 3

170 continue
l = 3
k = k + 3
169 continue
k = 3
l = 3

do 104 i = 1, node1
kk = nff(iabs(nop(nell,i)), 1)
k1 = nff(iabs(nop(nell,i)), 2)
km = nff(iabs(nop(nell,i)), 3)
do 105 j = 1, node1
dtpd(k, l) = 2.0d0*gdert(i, l)*gderv(1, j) + f4(i)*gdert(i, 2)*gderv(2, j)
3+3.0d0*b2(i)/rg*fun(i)*gderv(2,j)
3+re*b2(i)/rg*fun(i)*chk(kl)*fun(i)*fun(j)
dtpd(k,1) = dtpd(k,1)/re

c1(k, l) = b2(i)*fun(i)*fun(i)*chk(kk)*gderv(1, j)
c2(k, l) = b2(i)*fun(i)*fun(i)*chk(kl)*gderv(2, j)
c3(k, l) = -0.25d0*b4(i)/rg*fun(i)*fun(i)*chk(km)*fun(j)
if(l .eq. k) then
mass(k, l) = b2(i)*1.5d0*fun(i)*fun(j)/deltim
mm1(k, l) = mass(k, l)
end if
dtpd(k,l) = dtpd(k,l) + c1(k,l) + c2(k,l) + c3(k,l)
1 + mass(k, l)
dtpd(k, l) = dtpd(k, l)
l = l + 3
105 continue
l = 3
k = k + 3
104 continue

1080 do 1070 j = 1, dofel
dtpd(i,j) = dtpd(i,j)*quot
mm1(i,j) = mm1(i,j)*quot
1070 continue
1080 continue
4004 continue
112 do 111 i = 1, dofel
111 continue
k = 1
m = 0
do 1050 nn = 1, 8
do 1040 nd = 1, 3
nk(nn, nd) = 0
nk(nn, nd) = k + m
k=k+1
continue
nend=iaabs(nop(nell,nn))
if(mdf(nend).eq.4) m=m+1
continue

Assemble velocity terms

k=0
kk=0
do 1082 i=1,nodel
do 1083 j=1,3
k=k+1
l=nk(i,j)
do 1072 ii=1,node!
do 1073 jj=1,3
kk=kk+l
ll=nk(ii,jj)
dum2(ll,ll)=dtpd(k,kk)
1073 continue
1072 continue
1083 continue
1082 continue
call vecadd(elq,dofel,dmvec,idmvec,dofel,itest)
call matadd(ekl,ielk,elk,dum2,idum2,jdum2,idum2,idum2,itest)
if(ekl(i,i).eq.0.0d0) then
call vecadd(ekl,ielk,elk,dum2,idum2,jdum2,idum2,itest)
call matadd(ekl,ielk,elk,dum2,idum2,jdum2,idum2,itest)
end if
do 68 i=1,24
mm2(i,i)=mm2(i,i)+mm1(i,i)
68 continue
c
Assemble pressure terms

ngaus=4
iwght=ngaus
jabss=dimen
jabss=ngaus
ilder=2
jlder=4
call qqua4(wght,iwght,abss,iabss,jabss,ngaus,itest)
Quadrature for pressure terms

do 1095 iquad=1,ngaus
ifun=8
igeom=8
C.69
ilderv=2
jlderv=8
xi=abss(1,iquad)
et=abss(2,iquad)
call quam8(funv, ifun, lderiv, jlderv, xi, eta, itest)
call matmul(ilderv, jlderv, geom, igeom, jgeom, jac,
* ijac,ijac,jgeom,jlderv,ilderv,itest)
call matinv(jac, ijac, Jacin, Jacin, Jacin, Jacin, dimen,
* det, itest)
call matmul(jacin, Jacin, Jacin, lderiv, jlderv, gderiv,
* igderv,jgderv,dimem,dimem,jlderv,itest)
c calculate (xi,eta) in global (x,y)
n=igeom
call scaprd(geom(l, 1), igeom, funv, ifun, n, x, itest)
call scaprd(geom(l,2), igeom, funv, ifun, n, y, itest)
rg=y
call matran(gderv, igderv, jgderv, gdert, igdert, jgdert,
* igderv,jgderv,itest)
quot = dabs(det)*wght(iquad)
call quam4(funp,4,lderp,2,4,xi,eta,itest)
call matmul(lderp,ilderpj1derp,geomp,igeompjgeomp,jac,
* ijacjjacjgeomjlderv,ilderp,itest)
call matinvGac,ijacjjacjacin,ijacinjjacin,dimen,
* det,itest)
call matmulGacin,ijacinjjacin,lderp,ilderp,jlderp,gderp,
ligderpjgderp,dimen,demen,jlderv,itest)
ilder=2
jlder=4
do 1074 i=1,nodel
do 1074 j=1,nodel/2
ellt1(i,j)=-f3(i)*gderv(1,i)*funp(j)
ellt2(i,j)=-f4(i)*funv(i)*gderp(2,j)
ellt3(i,j)=-b2(i)/rg*funv(i)*funpQ)
elltG,i)=f3(i)*funpG)*gderv(1,i)
elltQ,i)=f4(i)*gderv(2,i)*funpQ)
1 +f4(i)/rg*funpQ)*funv(i)
e113Q,i)=-2.0d0*b2(i)/rg
1 *funv(i)*funpQ)
1074 continue
c c c Addition of ell1 and ell2 matrices to dumm
c
call matmul(dumm,idum2,jdum2,idum2,jdum2,itest)
k=0
c if(eltop(nell,2).eq.6) then
jj=11
kk=13
iarg=15

else if(eltop(nell,2).eq.8) then
  kk=25
  iarg=28
end if

do 1085 i=1,nodel
  ii=nk(i,1)
l=0
  k=k+1
do 1076
    j=4,kk,7
  continue
1076 continue
  dumm(ii,j)=ellt1(k,l)*quot/rho
  dumm(ii+1,j)=ellt2(k,l)*quot/rho
  dumm(ii+2,j)=ellt3(k,l)*quot/rho
  dumm(j,ii)=ell1(1,k)*quot
  dumm(j,ii+1)=ell2(1,k)*quot
  dumm(j,ii+2)=ell3(1,k)*quot
1085 continue
  call matadd(elk,ielkjelk,dumm,idurn2,jdurn2,iarg,iarg,itest)
1095 continue
if(nell.eq.2) then
end if

Formation of right hand side

Assemble mass terms

do 69 i=1,nodel
  vect(kk)=mm2(k,k)*uincc(istep-1,kk)
  vect(kl)=mm2(k,k)*uincc(istep-1,kl)
  vect(km)=mm2(k,k)*uincc(istep-1,km)
  k=k+3
69 continue

Determine whether element has boundary conditions

do 395 nbnd=1,neuman
  do 394 j=4,numnod(nbnd)+3
    do 393 k=1,7,2
      if(iabs(nop(nell,k)).eq.bdcnd(nbnd,j)) then
        go to 396
      else
        go to 393
      end if
    393 continue
  394 continue
395 continue

1316          go to 395
1317  396      call qlin2(lwght,ilwght,labss,ilabss,nqp,itest)
1318          mn=0
1319          if(btype(nbnd).eq.1) then
1320          neln=1
1321          mm=neln
1322          xi=-1.0d0
1323          else if(btype(nbnd).eq.2) then
1324          neln=3
1325          mm=2
1326          mn=1
1327          eta=1.0d0
1328          else if(btype(nbnd).eq.3) then
1329          neln=5
1330          mm=3
1331          xi=1.0d0
1332          end if
1333          do 401 iquad=1,nqd
1334          if(btype(nbnd).eq.2) then
1335          xi=labss(iquad)
1336          else
1337          eta=labss(iquad)
1338          end if
1339          call quam4(funp,ifunp,lder,ilder,ifunp,xi,eta,itest)
1340          call quam8(funv,ifunv,lderv,ilderv,ifunv,xi,eta,itest)
1341          call scaprd(geom(1,2),igeom,funv,ifunv,igeom,y,itest)
1342          rg=y
1343          kk=neln
1344          coeff=0.0d0
1345          do 402 i=1,3
1346          ll=2-mn
1347          coeff=coeff+lder(ll,kk)*coord(iabs(nop(nell,kk)),ll)
1348          kk=kk+1
1349          402 continue
1350          kkk=neln
1351          do 403 nn=1,3
1352          ll=iabs(nop(nell,kkk))
1353          jj=iabs(nopf(ii))+mn
1354          vect(jj)=vect(jj)+4.0d0*wght(iquad)*funv(kkk)*(funp(mm)
1355          1+funp(mm+1))*coeff*pinc(istep)/rho*pfact(nbnd)
1356          8899 format(2f10.5)
1357          kkk=kk+1
1358          403 continue
1359          401 continue
1360          395 continue
1361          do 176 i=1,totdof
1362          r1(i)=r1(i)+vect(i)
1363          vect(i)=0.0d0
1364          176 continue
Wall Elements

subroutine solidl(nell,ninc)
  implicit double precision(a-h,o-z)
  double precision jac,jacin
  1,l,der,ntn
  integer dimen, dofel, dofnod, eltop,
  *ielw,ink,hband,jabss,k,
  *ncols,nrows,nodref,mwga,ntra,mdf,nopp,
  *ir1, ia0, iai, iabss, ib, ibt, ibtdb,
  *nell,nsteps,nopf,nops,
  *ninc,istep,nf,nff,nfs,
  *ilwght,ilabss,isteer,ilwght
  * ,icoord, id, id2a0, id2a1, ida0, ida1, idb, ielk,
  * ielm, ieltop, ifun, igder, jgeom, ijac, ijacin
  integer ilder, infs, intn, iquad, ishp,jnfs,i,
  *nop,nk,numint,intnod,
  *ilderv,ifunp,ifunv,
  * itest, itshp,l,ki,kj,
  * j, jb, jbt, jtdb, jcoord, jd, jdb, jelk,
  * jelm, jeltop, jgder, jgeom, jjac, jjacin, jlder,
  * jntn, jshp, jtshp,
  *kk,kl,km,idof,fdof,sdof
  * ,nin, nodel, nout, ngauss,
  *ievect,
  * nq, numss,
  * totdof, totels, totnod,fels
  integer bncodf,bnodf,bncods,bnodes,bncode,bnode
  *,outnod,centnod,idum1,idum2,jdum1,jdum2,je1w,jnk
  dimension b(6,24), bt(24,6), btdb(24,24),
  * d(6,6), db(6,24),
  *jac(3,3),jacin(3,3),gder(3,8),
  *geom(8,3),lder(3,8) , ntn(24,24),
  * shp(3,24),
  * abss(3,9),fun(20),tshp(24,3)
 1400 * ,wght(9),dum1(16,16),dum2(16,16),elm(16,16),elw(16,16)

Problem size dependent arrays

dimension
* rad(8)
*,vec1(2500),vec2(2500)

c
common/cord/coord(2500,2),cords(2500,2)
common/dofnum/nf(2500,4),nff(2500,4),nfs(2500,4)
common/press/pinc(50),delt(50),uinc(50,3500)
1408  common/wprops/emod,poisrt,dens
1409  common/resuts/sk(5000),chk(5000)
1410  common/fron1/nodref,totdof,totnod,totels,fels,doef,mwga,
1411  *ntra,nout,dimen,nodel,idof,fdoef,sdoef,hband,nrows,ncols
1412  common/fron2/eltop(500,10),bval(500),bvalf(500),bvals(500),
1413  *elq(70),bncode(500),bnode(500),bcode(2500)
1414  *,bncoedf(500),bnodedf(500),bcf(2500),bcfl(2500)
1415  *,bncoeds(500),bnodes(500),bcs(2500)
1416  common/fron4/mdf(2500),nop(2500),nopf(2500),nops(2500)
1417  common/fron5/nop(500,8),nk(8,3),el(28,28)
1418  common/facts/alpha,beta,theta
1419  common/epi/numint,intnod(200),outnod(200),centnod(200)
1420  common/const/cc1,cc2,cc3,c4,c5,c6,c7
1421  common/time/time,istep,nstamps,deltim
1423  *d2a1(2000)
1424  common/rhvect/r1(2500),vector(2500)
1425  common/pfact/pfact(3)

Problem size dependent arrays

data iabss/3/,ib/4/,itb/24/,itbtdb/24/,id/4/,idb/4/,iem/16/,
1426  *ielm/16/,ifun/8/,jgder/3/,ijac/3/,ijsat/3/,iels/4/,
1427  *jder/2/,jfun/4/,jstep,nstamps,deltim
1428  *jelm1/16/,jelm2/16/,jelm1/16/,jelm2/16/,
1429  *jelm/16/,jelm2/16/,jelm1/16/,jelm2/16/,
1430  *jelm/16/,jelm2/16/,
1431  *jelm/16/,jelm2/16/
1432  *jelm/16/,jelm2/16/,jelm1/16/,jelm2/16/,
1433  *jelm/16/,jelm2/16/,jelm1/16/,jelm2/16/,
1434  *jelm/16/,jelm2/16/,jelm1/16/,jelm2/16/,
1435  *jelm/16/,jelm2/16/,jelm1/16/,jelm2/16/,
1436  *jelm/16/,jelm2/16/,jelm1/16/,jelm2/16/,
1437  *jelm/16/,jelm2/16/,jelm1/16/,jelm2/16/,
1438  *jelm/16/,jelm2/16/,jelm1/16/,jelm2/16/,

Problem size dependent data statements

1439  *ida0/2000/,ida1/2000/,ieltop/500/,infs/2500/,
1440  *jcoop/2/,
1441  *jelm/10/,jnsfs/3/
1442  call vecnul(vec1,ive,ivec,istest)
1443  call vecnul(vec2,ive,ivec,istest)

Construction of stress-strain matrix d

1446  1212 continue
1447  call daxi(d,idi,ide,emod,poisrt,numss,istest)

C.74
Integration loop for element stiffness and element lumped mass using ngauss quadrature points

call matnul(elk, ielk, jelk, ielk, jelk, itest)
call matnul(elm, ielm, jelm, ielm, jelm, itest)
call matnul(elw, ielw, jelw, ielw, jelw, itest)
call matnul(duml, i6, j6, i6, j6, itest)
call matnul(dum2, i6, j6, i6, j6, itest)
do 590
iquad=1,ngauss

Form linear shape function and space derivatives in the local coordinates.
transform local derivatives to global coordinate system

xi = abss(2,iquad)
eta = abss(1,iquad)
call quam8(fun, ifun, lder, ilder, jlder, xi, eta, itest)
call matrnu!lder, jlder, geom, igeom, jgeom, jac,
* ijac, jjac, dimen, node!, dimen, itest)
call matinv(jac, ijac, jjac, jacin, jjacin, dimen, det, itest)
call matmul(jacin, ijac, jjacin, ijac, jjac, jjac, ider, gder,
* igder, jgder, dimen, node!, itest)
call scaprd(geom(l,l), igeom, fun, igeom, igeom, prdct, itest)
rad(iquad)=prdct

Formation of strain-displacement matrix b and formation of integrand for element stiffness matrix elk

call b2p2(b,ib,jb,igerd,jgerd,fun,ifun,
* geom,igeom,geome,nodel,itet)
call matmul(d, id, jd, b, ib, jb, db, idb, jdb, numss,
* numss,dofel, itest)
call matran(b, ib, jb, bt, ibt, jbt, numss, dofel, itest)
call matmul(bt, ibt, jbt, db, idb, jdb, btjdb, ibtjdb, jbtjdb,
* dofel, numss, dofel, itest)
c
Formation of integrand for element
mass matrix elm

call shapfn(shp, ishp, jshp, fun, ifun, nodel, dofnod, itest)
call matran(shp, ishp, jshp, tshp, itshp, jtshp, dofnod, dofe1, itest)
call matmul(tshp, itshp, jtshp, shp, ishp, jshp, ntn, intn, jntn, dofe1, dofnod, dofe1, itest)

quot=dabs(det)*wght(iquad)
do 580 i=1,dofel
do 570 j=1,dofel
ntn(i,j) = 1.5d0* ntn(i,j)*quot*dens
if(i.eq.j) then
  dum2(i,j)=dum2(i,j)+ntn(i,j)
else
  dum2(i,j)=0.0d0
end if
btdb(i,j) = btdb(i,j)*quot
duml(ij)=duml(ij)+btdb(ij)
570 continue
580 continue
590 continue

c Assembly of system stiffness matrix
c elk, system lumped mass matrix elm
c
cc1 = alpha + 1.d0/(theta*deltim)
ccc2 = beta + theta*deltim
ccc3 = (1.d0-theta)*deltim
ccc4 = beta - (1.d0-theta)*deltim
c 5 = 1.d0/(theta*deltim)
c 6 = (1.d0-theta)/theta
c 7 = theta*deltim
do 512 i=1,dofel
do 513 j=1,dofel
elk(i,j)=cc1*dum2(i,j)+cc2*dum1(i,j)
elw(i,j)=cc1*dum2(i,j)+cc4*dum1(i,j)
elm(i,j)=dum2(i,j)/theta
513 continue
512 continue

c Assemble right hand side
c
k=1
do 173 i=1,nodel
ki=nfs(iabs(nop(nell,i)),1)
kJ=nfs(iabs(nop(nell,i)),2)
1519  l=1
1520  do 174 j=1,node1
1521    kk=nfs(iabs(nop(nell,j)),1)
1522    kl=nfs(iabs(nop(nell,j)),2)
1523    vec1(ki)=vec1(ki)+elm(k,l)*da0(kk)
1524    1+elm(k,l+1)*da0(kl)
1525 174 continue
1526  k=k+2
1527  continue
1528  do 169 i=1,numint
1529    if(iabs(nop(nell,1)).eq.intnod(i)) then
1530      z1=cords(iabs(nop(nell,1)),2)
1531      z3=cords(iabs(nop(nell,7)),2)
1532      r2=0.5d0*(z3-z1)
1533      ij=nff(iabs(nop(nell,1)),4)
1534      ik=nff(iabs(nop(nell,7)),4)
1535      p1=uinc(istep,ij)-uinc(istep-1,ij)
1536      p3=uinc(istep,ik)-uinc(istep-1,ik)
1537      vec2(kk)=vec2(kk)+0.6666d0*p1*r2
1538      vec2(kl)=vec2(kl)+1.3333d0*(p1+p3)*r2
1539      vec2(km)=vec2(km)+0.6666d0*p3*r2
1540    end if
1541 169 continue
1542  do 175 i=1,totdof
1543    rl(i)=r1(i)+vec1(i)+vec2(i)
1544 175 continue
1545 1414 continue
1546 1414 continue
APPENDIX D

Mesh Generation Program Listings

D.1 Program Description
The two programs MESHL and MESHX generate finite element mesh data for programs PFECTL and PFECTLX respectively. The programs generate the element topology and node coordinates from input data consisting of the number of rows and columns of elements in MESHL and the number of radial and circumferential elements in MESHX. The programs generate the lists of boundary nodes, the boundary variables to be restrained and the value of the variable, as appropriate, for the fluid Dirichlet and Neumann boundary conditions, together with the wall displacement constraints. The boundary conditions are generated along lines of the mesh. All fluid and wall material properties are read in, to be repeated in the input files for the finite element programs.

D.2 List of Variables

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>amu</td>
<td>fluid dynamic viscosity</td>
</tr>
<tr>
<td>coord</td>
<td>node coordinates</td>
</tr>
<tr>
<td>deltim</td>
<td>time increment</td>
</tr>
<tr>
<td>dens</td>
<td>wall density</td>
</tr>
<tr>
<td>eltop</td>
<td>element topology</td>
</tr>
<tr>
<td>emod</td>
<td>wall modulus of elasticity</td>
</tr>
<tr>
<td>fdof</td>
<td>total number of fluid dof</td>
</tr>
<tr>
<td>fels</td>
<td>total number of fluid elements</td>
</tr>
<tr>
<td>intnod, outnod</td>
<td>nodes on wall surfaces</td>
</tr>
<tr>
<td>ncod</td>
<td>boundary condition code</td>
</tr>
<tr>
<td>ncodf</td>
<td>boundary condition code-fluid</td>
</tr>
<tr>
<td>ncods</td>
<td>boundary condition code-solid</td>
</tr>
<tr>
<td>ndof</td>
<td>total number of dof</td>
</tr>
<tr>
<td>nel</td>
<td>element number</td>
</tr>
<tr>
<td>nits</td>
<td>number of (Picard) iterations</td>
</tr>
<tr>
<td>nff</td>
<td>dof number - fluid</td>
</tr>
<tr>
<td>nfs</td>
<td>dof number - solid</td>
</tr>
<tr>
<td>nodedod</td>
<td>corner node identifier</td>
</tr>
<tr>
<td>ns, nsteps</td>
<td>number of pressure steps</td>
</tr>
<tr>
<td>numint</td>
<td>number of interface nodes</td>
</tr>
<tr>
<td>numnod</td>
<td>node number</td>
</tr>
<tr>
<td>pinc</td>
<td>step pressure</td>
</tr>
<tr>
<td>poisrt</td>
<td>wall Poisson's ratio</td>
</tr>
<tr>
<td>r0</td>
<td>tube radius of curvature</td>
</tr>
<tr>
<td>pocord</td>
<td>node polar coordinates</td>
</tr>
<tr>
<td>rho</td>
<td>fluid density</td>
</tr>
<tr>
<td>sdof</td>
<td>number of dof - solid</td>
</tr>
<tr>
<td>sk</td>
<td>solution for current iteration</td>
</tr>
<tr>
<td>totels</td>
<td>number of elements</td>
</tr>
<tr>
<td>tottime</td>
<td>pressure cycle time</td>
</tr>
<tr>
<td>totnod</td>
<td>number of nodes</td>
</tr>
</tbody>
</table>
D 3 Pre-processor Program Listings

D.3.1 Program MESHX

- Mesh generation for program PFECTX
- Cross section through tube

```fortran
program meshx
 implicit double precision(a-h,o-z)
 integer eltop,totels,totnod
 1,hband,dif
 dimension cord(1500,2),eltop(1500,10)
 1,node(100,100)
 2,md(350)
 3,neu(200),
 4xpts(1500),ypts(1500),ang(1500),r(1500)
 5,innod(300),outnod(300),pocord(1500,2)
 dimension pinc(25)
 nout=8
 open(file='meshx.dat',unit=nout,form='formatted')
 write(*,7000)
 open(file='diag',unit=11,form='formatted')
 7000 format(/,'Mesh data: rin? rout? nelr? nelt? nelw? r0?'/)
 read(*,*) rin,rout,nelr,nelt,nelw,r0
 tote1s=nelt*(nelw+nelr)
 write(*,*) rin,rout,nelr,nelt,nelw,r0
dirnen=2
 write(*,7010)
 read(*,*) amu
 write(*,7002)
 rho
 write(*,7012)
 read(*,*) incs
 write(*,7013)
 read(*,*) time
 write(*,7014)
 read(*,*) nsteps
 ii=2*nelr+1
 jj=2*nelt+1
 kk=2*nelw+1
 write(*,*) ii,jj
 do 90 j=1,jj,2
 md(j)=1
 md(j+1)=2
 dang=3.14159d0/(jj-3)
 ang(1)=0.0d0
 ang(2)=dang/2.0d0
```
ang(3)=dang
ang(jj)=3.14159d0
ang(jj-1)=ang(jj)-dang/2.0d0
ang(jj-2)=ang(jj-1)-dang/2.0d0
do 500 j=4,jj-3
ang(j)=ang(j-1)+dang
500 continue
rcen=0.01d0
rbar=1.0d0
dr=rin/ii
dw=(rout-rin)/nelw
r(1)=rcen
k=3
do 601 i=3,ii,2
r(i)=r(i-2)+k*dr
k=k+1
if(i.gt.(ii/2)) then
  k=k-1
end if
if(i.gt.(3*ii/4)) then
  k=k-2
endif
rsum=r(i)
601 continue
do 600 i=1,ii,2
r(i)=r(i)*rin/rsum
600 continue
do 602 i=ii+2,ii+kk,2
r(i)=r(i-2)+dw
602 continue
ii=ii+kk-1
do 700 i=2,ii,2
r(i)=(r(i-1)+r(i+1))/2.0d0
700 continue
k=1
do 900 j=1,jj
do 900 i=md(j),ii-1+md(j),md(j)
node(j,i)=k
n=i+1-md(j)
pocord(k,1)=r(n)
pocord(k,2)=ang(j)
write(11,*) k,pocord(k,1),pocord(k,2)
k=k+1
900 continue
totnod=k-1
do 1000 j=1,jj
if(md(j).eq.1) then
  kl=2*(nelr+nelw)+1
  km=2*nelr+1
else if(mdG.eq.2) then
  kl=2*(nelr+nelw+1)
  km=2*(nelr+1)
end if
outnodG)=nodeG,kl)
intnodG)=nodeG,km)
1000 continue
totnod=k-1
idim=2
write(nout,7050) totnod,idim
do 300 k=1,totnod
write(nout,7030) k, pocord(k,1),pocord(k,2)
300 continue
nele=1
do 200 i=1,ii,2
do 200 j=1,ij-1,2
  eltop(nele,3)=nodeG+2,i)
  eltop(nele,4)=nodeG+2,i+ 1)
  eltop(nele,5)=nodeG+2,i+2)
  eltop(nele,6)=nodeG+1,i+3)
  eltop(nele,7)=nodeG,i+2)
  eltop(nele,8)=nodeG,i+ 1)
  eltop(nele,9)=nodeG,i)
  eltop(nele,10)=nodeG+1,i+ 1)
if(eltop(nele,10).eq.0) then
  eltop(nele,2)=6
else
  eltop(nele,2)=8
end if
nele=nele+ 1
200 continue
fels=totels-nelw*nelt
write(nout,7050) totels,fels
do 1010 nele=1,totels
write(nout,7050) nele,(eltop(nele,i),i=1,10)
1010 continue
nodmax=0
hband=0
do 2000 i=1,totels
do 2001 j=3,10
  if(eltop(i,j),lt,nodmin) then
    nodmin=eltop(i,j)
  end if
  if(eltop(i,j),gt,nodmax) then
    nodmax=eltop(i,j)
  end if
dif=(nodmax-nodmin)*2+1
if(dif.gt.hband) then
  hband=dif
end if
continue

do 1201 i=1,ii
continue

do 100 n=1,totnod
cord(n,1)=cord(n,1)*b/cord(totnod,1)
xpts(n)=cord(n,1)
ypts(n)=cord(n,2)
continue

xmin=xpts(1)
ymin=ypts(1)
xmax=xpts(totnod)
ymax=ypts(totnod)
q2=ymax/xmax
imesh=1
do 301 nele=1,totels
continue

write(nout,7040) r0
write(nout,7040) amu
write(nout,7040) rho
emod=10.0d+23
poisrat=3.00d-01
write(nout,8004) emod,poisrat
dens=10000.0
write(nout,7040) dens
time=10000.0
write(nout,7050) incs
write(nout,8001) time
write(nout,8002) nsteps
do 8009 i=1,nsteps
continue

write(nout,7050) jj
numint=jj
do 1111 kk=1,jj
write(nout,7050) kk, intnod(kk),outnod(kk)
continue

c Dirichlet Boundary Conditions

.. D.5
184  dirich=jj*4+ii*2-1
185    write(nout,7050) dirich
186    do 800 j=1,jj
187    do 800  k=1,3
188      b=0.0d0
189      write(nout,7075) intnod(j),k,b
190    800     continue
191    do 801 i=1,ii
192      k=3
193      b=0.0d0
194      write(nout,7075) node(1,i),k,b
195      write(nout,7075) node(jj,i),k,b
196    801    continue
197    do 802 i=1,totnod
198      if(pocord(i,1).eq.r(1)) then
199        if(pocord(i,2).ne.0.0d0) then
200          if(pocord(i,2).ne.3.14159d0) then
201            n=2
202          end if
203        end if
204      end if
205    802    continue
206      i=intnod(numint)
207      k=4
208      write(nout,7075) i,k,b
209    c Wall Boundary Conditions
210    c
211      wallbc=6
212      write(nout,7050) wallbc
213      ib2=2
214      bv=0.0d0
215      do 3000 i=intnod(1),outnod(1)
216        write(nout,6035) i,ib2,bv
217    3000    continue
218      do 3001 j=intnod(numint),outnod(numint)
219        write(nout,6035) j,ib2,bv
220    3001   continue
221    c Neumann Boundary Conditions
222    c
223      neuman=1
224      ii=2*nelr+1
225      jj=2*nelt+1
226      do 730 j=1,jj
do 729 i=1,ii
if(pcord(node(j,i),1).le.rin.and.node(j,i).ne.0) then
  neu(neuman)=node(j,i)
  neuman=neuman+1
end if
continue
continue
neuman=neuman-1
stop

6000 format(2f10.3,2i5)
6010 format(f10.5)
6020 format(i5)
6030 format(3i5,2f10.5)
6035 format(2i5,f10.5)
7002 format(' Wall modulus?'
7003 format(' Wall Poisson s ratio?')
7004 format(' Wall density?'
7013 format("Total time?'
7014 format(' Number of steps?')
7015 format(' number of boundaries?')
7010 format(' viscosity?')
7011 format(' density?')
7012 format(' number of increments?')
7020 format(" boundary node number? row or column? u,v or p?,value?")
7030 format(i5,2f10.5)
7040 format(f10.2)
7050 format(200i5)
7070 format(" Number of Boundary Nodes?")
7075 format(2i5,f10.5)
7080 format(" Type? Node number? Row or Column? Value?")
7090 format(" Number of Boundaries?")
7100 format(i5,8d10.4)
7110 format(/'Pressure for step',i5/)
D.3 Program Listings

D.3.2 MESHL

c Mesh generation for program PFECTL
c Longitudinal section through tube
c
1        program mesh
2        implicit double precision(a-h,o-z)
3        integer b1,b2,b3,btop,bnum,bvar,totels,totnod
4        1,btype
5        2,hband,dif
6        dimension cord(990,2),eltop(990,10),b1(350),b2(350),b3(350),
7        1 bnum(350),node(200,200)
8        2,md(350),btype(350)
9        3,bvar(350),bval(350),idir(350),linenode(350)
10       4,press(350),vnorm(350)
11       5,dx(990),shift(990)
12       6,xpts(990),ypts(990)
13       dimension area(300),arnod(1000),darnod(1000)
14       dimension pinc(20)
15       5,nout=8
16       open(file='mesh.dat',unit=nout,form='formatted')
17       write(*,7000)
18       7000    format(/,' Model mesh data: ht? lth? nella? nelb? rad? ',/)
19       read(*,*) a,b,nela,nelb,r0
20       write(*,7001)
21       7001    format(/,' Wall mesh data: thk? nelt? '/)
22       read(*,*) thk,nelt
23       totels=nela*nelt
24       nodref=a*b-totels
25       wels=nelt*nelt
26       fels=totels-wels
27       imesh=1
28       if(imesh.eq.0) then
29        end if
30       write(*,7010)
31       read(*,*) amu
32       write(*,7011)
33       read(*,*) rho
34       write(*,7002)
35       read(*,*) emod
36       write(*,7003)
37       read(*,*) poisrat
38       write(*,7004)
39       read(*,*) dens
40       write(*,7012)
41       read(*,*) incs
42       write(*,7013)
read(*,*) time
write(*,7014)
read(*,*) nsteps
node=8
idim=2
ndof=3
delx=b/(2*nelb)
dely=a/(2*nela)
i=2*nela+1
jj=2*nelb+1
do 90 j=1,jj,2
md(j)=1
md(j+1)=2
90 continue
ii=2*nela+l
jj=2*nelb+1
do 90 j=1,jj,2
md(j)=1
md(j+1)=2
90 continue
sdelx=0.0d0
sdely=0.0d0
k=1
do 250 j=1,jj
do 260 i=1,ii,md(j)
cord(k,1)=sdelx
cord(k,2)=sdely
sdely=sdely+md(j)*dely
node(i,j)=k
k=k+l
260 continue
sdelx=sdelx+delx
sdely=0.0d0
250 continue
totnod=k-1
if(imesh.eq.0) go to 1111
write(nout,7050) totnod,idim
write(nout,7050) ii,jj
nel=1
do 200 i=1,ii-1,2
do 201 j=1,jj-1,2
eltop(nele,3)=node(i,j)
eltop(nele,4)=node(i+1,j)
eltop(nele,5)=node(i+2,j)
eltop(nele,6)=node(i+2,j+1)
eltop(nele,7)=node(i+2,j+2)
eltop(nele,8)=node(i+1,j+2)
eltop(nele,9)=node(i,j+2)
eltop(nele,10)=node(i,j+1)
nele=nele+1
201 continue
200 continue
1111 nodmin=nodref
nodmax=0
hband=0
do 2000 i=1,totels
doo 2001 j=3,10
if(eltop(i,j).lt.nodmin) then
  nodmin=eltop(i,j)
end if
if(eltop(i,j).gt.nodmax) then
  nodmax=eltop(i,j)
end if
2000 continue

dif=(nodmax-nodmin)*2+1
if(dif.gt.hband) then
  hband=dif
end if
2000 continue

do 1201 i=1,ii
1201 continue

do
  n=1,totnod
  cord(n,1)=cord(n,1)*b/cord(totnod,1)
  write(nout,7030) n, cord(n,1),cord(n,2)
  xpts(n)=cord(n,1)
  ypts(n)=cord(n,2)
100 continue
 xmin=xpts(1)
ymin=ypts(1)
 xmax=xpts(totnod)
 ymax=ypts(totnod)
 q2=ymax/xmax
if(imesh.eq.0) go to 8095
write(nout,7050) totels,fels
301 continue
nele=1,totels
write(nout,7050) nele,nodel,(eltop(nele,i),i=3,10)
301 continue
write(nout,7040) r0
write(nout,7040) amu
write(nout,7040) rho
write(nout,8003) emod,poisrat
write(nout,7040) dens
write(nout,7050) nele,nodel,(eltop(nele,i),i=3,10)
301 continue
write(nout,7040) r0
write(nout,7040) amu
write(nout,7040) rho
write(nout,8003) emod,poisrat
write(nout,7040) dens
write(nout,7050) incs
write(nout,8001) time
write(nout,8002) nsteps
8009 continue
c
c
c Dirichlet Boundary Conditions
write(*,9000)
c
cwrite(*,7090)
read(*,*) nbnds
nconds=0
nk=0
nl=0
do 400 nb=1,nbnds
write(*,7020)
read(*,*) linenode(nb),idir(nb),bvar(nb),bval(nb)
do 401 i=1,ii
do 402 j=1,jj
if(linenode(nb).eq.node(i,j)) then
if(idir(nb).eq.1) then
ll=i
icount=bvar(nb)/4+1
do 403 mrn=1,jj,icount
if(node(ll,mrn).eq.O) then
go to 403
end if
write(nout,7075) node(ll,mrn),bvar(nb),bval(nb)
nl=nl+1
continue
else if(idir(nb).eq.2) then
icount=bvar(nb)/3+1
mm=j
do 404 ll=1,ii,icount
if(node(ll,mm).eq.0) then
go to 404
end if
write(nout,7075) node(ll,mm),bvar(nb),bval(nb)
nk=nk+1
continue
end if
402 continue
401 continue
400 continue
nconds=nk+nl
write(*,7050) nconds
c
cNeumann Boundary Conditions
write(*,9001)
c
cc 
write(*,7070)
read(*,*) nb
do 500 n=1,nb
write(*,7080)
read(*,*) btype(n),linenode(n),idir(n),press(n)
write(nout,8002) nb
write(nout,7040) press(n)
do 501 i=1,ii
do 502 j=1,jj
if(linenode(n).eq.node(ij)) then
if(idir(n).eq.l) then
l=i
write(nout,7050)
btype(n),jj,(node(l,mm),mrn= l,jj)
else if(idir(n).eq.2) then
mm=j
write(nout,7050)
btype(n),ii,(node(l,mm),l= l,ii)
504 continue
end if
end if
502 continue
501 continue
500 continue
8095 if(imesh.eq.0) then
stop
end if
6000 format(2f10.3,2i5)
6010 format(f10.5)
6020 format(i5)
6030 format(3i5,2f10.5)
6035 format(2i5,f10.5)
7002 format(' Wall modulus?')
7003 format(' Wall Poisson s ratio? ')
7004 format(' Wall density? ')
7013 format('Total time? ')
7014 format(' Number of steps?')
7015 format(' number of boundaries?')
7010 format(' viscosity?')
7011 format(' density?')
7012 format(' number of increments?')
7020 format(' boundary node number? row or column? u,v or p?,value?')
7030 format(i5,2f10.5)
7040 format(f10.5)
7050 format(200i5)
7070 format(' Number of Boundary Nodes?')
7075 format(2i5,f10.5)
7080 format(' Type? Node number? Row or Column? Value?')
7090 format(' Number of Boundaries?')
7100 format(i5,8d10.4)
7110 format('Pressure for step',i5)
8001 format(f10.5)
8002 format(i5)
8003 format(2f10.5)
8888 format(i5,f10.5)
9000 format('Dirichlet Boundary Conditions',//)
9001 format('Neumann Boundary Conditions',//)
end
Program Listings

D.3.3 MESHXY

c Program to generate an x-y mesh on a semi-circular frame

dimension dx(200),dy(200),x(200),y(200)
dimension cart(2,200),cord(2,100)
dimension s1(200),s2(200),s3(200),s4(200)
open(file='meshxy.dat',unit=6,form='formatted')
read(*,*) nels,rad,rad2
ncols=nels+1
nrows=nels+1
x1=0.0d0
y1=0.0d0
x2=0.0d0
y2=rad
x3=rad*0.7071d0
y3=rad*0.7071d0
x4=rad
y4=0.0d0
df=2.0d0/ncols

90 do i=1,nrows
  y(i)=(i-ncols/2-1)*df
91 continue

90 continue

nnodes=ncols*nrows
node=1
do 120 j=1,ncols
do 121 i=1,nrows
cord(1,node)=x(j)
cord(2,node)=y(i)
s1(node)=0.25d0*(1-cord(1,node))*(1-cord(2,node))
s2(node)=0.25d0*(1-cord(1,node))*(1+cord(2,node))
s3(node)=0.25d0*(1+cord(1,node))*(1+cord(2,node))
s4(node)=0.25d0*(1+cord(1,node))*(1-cord(2,node))
nodes=node+1
120 continue

nnodes=ncols*nrows
do 130 node=1,nnodes
cart(1,node)=s1(node)*x1+s2(node)*x2+s3(node)*x3+s4(node)*x4
cart(2,node)=s1(node)*y1+s2(node)*y2+s3(node)*y3+s4(node)*y4
polar(1,node)=sqrt(cart(1,node)**2+cart(2,node)**2)
polar(2,node)=atan(cart(2,node)/cart(1,node))
polar(1,node+nnodes)=polar(1,node)
polar(2,node+nnodes)=polar(2,node)+1.5708d0
if(cart(2,node).eq.0.0d0) then
  polar(2,node)=0.0d0
end if
write(6,* ) node,polar(1,node),polar(2,node)
do 135 node=nnodes,2*nnodes
write(6,* ) node,polar(1,node),polar(2,node)
node=2*nnodes+1
do 140 ii=1,nels*4+1
  node=node+1
  polar(2,node)=(ii-1)*3.14159/(4*nels)
polar(1,node)=rad2
write(6,* ) node, polar(1,node),polar(2,node)
continue
end
APPENDIX E
Numerical Integration for Characteristic Equation Terms in Program PFECTL

Let a typical term of Eqtns 5.8-5.10 or 5.4, \( \int_{-\infty}^{\infty} f(b, \theta) \, d\theta \) be expressed in terms of coefficients \( a_i \) and \( b_i^p \), so that

\[
\int_{-\infty}^{\infty} f(b_i, \theta) \, d\theta = a_0 + a_1 b_i + a_2 b_i^2 + a_3 b_i^3 + a_4 b_i^4
\]  \( \text{(E.1)} \)

\( a_0 \) is found immediately from the substitution of \( b = 0 \) in Eqtn E.1

Substituting values for \( b_i, i = 1, 4, \) e.g., 0.05, 0.10, 0.15, 0.20, a typical range, leads to the equations

\[
\begin{bmatrix}
   b_1 & b_1^2 & b_1^3 & b_1^4 \\
   b_2 & b_2^2 & b_2^3 & b_2^4 \\
   b_3 & b_3^2 & b_3^3 & b_3^4 \\
   b_4 & b_4^2 & b_4^3 & b_4^4 \\
\end{bmatrix}
\begin{bmatrix}
   a_1 \\
   a_2 \\
   a_3 \\
   a_4 \\
\end{bmatrix} =
\begin{bmatrix}
   \int_{-\infty}^{\infty} f(b_1, \theta) \, d\theta \\
   \int_{-\infty}^{\infty} f(b_2, \theta) \, d\theta \\
   \int_{-\infty}^{\infty} f(b_3, \theta) \, d\theta \\
   \int_{-\infty}^{\infty} f(b_4, \theta) \, d\theta \\
\end{bmatrix}
\]  \( \text{(E.2)} \)

or

\[ [B] \{A\} = \{C\} \]

Evaluating the terms in \( \{C\} \) numerically, the values for \( a_j \) are found from

\[
[B]^{-1} [B] \{A\} = \{A\} = [B]^{-1} \{C\}
\]  \( \text{(E.3)} \)