Finite Element Analysis of the Human Left Ventricle in Diastole and Systole

Thesis presented for the degree of Doctor of Philosophy

by

M.C.Beecham MSc, BSc (Hons.), GIMA

Department of Mechanical Engineering

Brunel University

February 1997

Abstract

Previously, at Brunel University, two computer programs had been developed to facilitate the analysis of the diastolic material properties of the human left ventricle. These two computer programs consisted of; a finite element program, "XL1", which ran upon a Cray-1S/1000 and a post-processor and pre-processor, "HEART", which ran upon the Multics computer system. The computer program "HEART" produced the finite element model, which was then solved by "XL1", and it also allowed for plotting the results in graphical form. The patient data was supplied by the Royal Brompton Hospital in the form of digitised cine-angiographic X-ray data plus pressure readings.

The first stage was to transfer the two separate computer programs "HEART" and "XL1" to the Sun Workstation system. The two programs were then combined to form a single package which can be used for the automated analysis of the patient data.

An investigation into the effect that the elastic modulus ratio has upon the deformation of the left ventricle during diastole was performed. It was found that the effect is quite small and that using this parameter to match overall shape deformation would be extremely sensitive to the accuracy of the initial data.

The main part of this work was the implementation of active cardiac contraction, by means of a thermal stress analogy, into the finite element program. This allows the systolic part of the cardiac cycle to be analysed. The analysis of the factors that affect cardiac contraction, including the material properties and boundary conditions was performed. This model was also used to investigate the effect that conditions such as ischaemia and the formation of scar tissue have upon the systolic left ventricle.

The use of the thermal stressing analogy for cardiac contraction was demonstrated to mirror global and local deformation when applied to a realistic ventricular geometry.

Acknowledgements

I would firstly like to extend my deepest gratitude to Prof. A.L.Yettram for his assistance and advice during this work and also for supervising it. I would also like to thank Dr. D.Gibson for freely giving his time and advice on medical matters and also for providing inspiration.

Lastly I would like to thank the Mechanical Engineering Department of Brunel University for providing the equipment and the opportunity to study towards a PhD. I would also like to thank the department for financially supporting me with a departmental stipend for my three years of study.

Table of Contents

Chapter 1

1. Introduction

The current work is a continuation of that started by Vinson (1977 [1]) and expanded by Grewal (1988 [2]), who developed the basic finite element model used in the following work. The formation of the model is discussed later in this text and in greater detail by Yettram *et al.* [3] [4] [5].

In 1993 45 per cent of all deaths in England and Wales were as a result of cardiovascular disease (1993 [6]). This figure is also reflected throughout the other developed nations and is seen year on year. The largest causes of heart related death are due to coronary artery disease. There are however also genetic, viral and bacterial causes. The figures for heart disease in the developing countries, insofar as they are known, are probably far lower, although as these countries turn to a more western lifestyle the incidence of heart disease there will almost certainly rise to a rate comparable to that of the United Kingdom. This high figure for mortality is a reduction on that of previous years. This reduction was at first due to improvements in early detection and the long-term treatment of cardiovascular disease, and more recently due to a reduction in the incidence of cardiovascular disease.

With heart disease being such a major health problem, as well as a large financial strain upon healthcare, it is no wonder that over recent years cardiology has featured prominently as a field for medical research. It is with a hope of a better understanding of the function of the heart and the effect diseases have that research has continued, since greater knowledge may enable earlier detection of disease, resulting in medical intervention at a less advanced stage. Research may also pave the way towards more effective treatment.

The heart is extremely complicated in terms of its structure and function. It would be a grave mistake to model it purely from a mechanical point of view without an appreciation of its complex biological nature. It is for this reason that many groups researching the function of the heart bring together those with a medical as well as an engineering background.

For many years researchers have attempted to analyse and quantify the behaviour of the human heart, and in particular the left ventricle. Some have worked with a simplified geometry for the left ventricle, spheres, ellipses of revolution and general ellipsoids using a theoretical approach. Others have done experimental work involving precise measurement of the ventricles of animals, most commonly those of dogs. Only a few have used data gathered from human subjects, since this data is usually hard to come by and is often severely limited in terms of quality and quantity.

Attempts to overcome the inadequacies of experimental data have resulted in a marriage between the above two methods. The limited geometric data is used to generate models using the above simplified geometric shapes. The problem with this is that when attempting to define relationships between various mechanical properties of the myocardium, the muscle which is the wall of the left ventricle, its function and possible effects from disease, one cannot be sure that relationships found are applicable to real patients, since there may be artefacts introduced by assuming a simplistic geometry. Consistently proposed relationships found from spherical or ellipse of revolution models could not be substantiated by Yettram *et al.* [4], using a mor realistic geometry derived from orthogonal bi-plane cineangiograms.

There has been much work done by a number of authors using canine ventricles, Guccione *et al.* [7] went to great lengths to obtain exact geometric information. This involved isolation of the left ventricle from a dog and from the rest of the heart. The authors however expressed concern over whether their results could be extrapolated to an intact heart still within the subject's chest. An extensive investigation of the effect of assuming different ventricle geometric configurations upon myocardial properties and function was carried out by Mirsky (1969 [8]). The effect of assuming spherical, ellipse of revolution and general ellipsoidal geometries was compared. He found that there was a large disparity between the results obtained for each shape. It was claimed that the human left ventricle is most closely allied to an ellipse of revolution and canine ventricles are most closely represented by a general ellipsoid. He therefore concluded that it was not possible to directly extrapolate results from work done with canine subjects to humans.

1.1 Anatomy of the Human Heart

The human heart is a hollow muscular organ approximately conical in shape with large blood vessels protruding from its base (the great vessels). It is situated approximately mid-chest height such that one third is in the right side of the chest and the remaining two thirds are in the left half of the chest. The heart is surrounded by the pericardium which protects the heart. The pericardium consists of two layers, the outer layer is the fibrous pericardium and the inner layer is a thin membrane, the serous pericardium. The serous pericardium is folded back on itself to form a double thickness. The outer layer, the parietal pericardium, is attached to the fibrous pericardium, while the inner layer, the visceral pericardium or epicardium, is attached to the heart proper and also covers the great vessels. Between these two layers of the membrane is pericardial fluid which acts as a lubricant to prevent excessive friction and thus damage to the membrane.

The heart is not a single pump but is in fact two pumps in one. The left half is responsible for circulating blood around the body (systemic circulation) and the right half around the lungs (pulmonary circulation). Each of these pumps is divided into the atrium and the larger ventricle. Between the atria and the ventricles is an 8-shaped atrioventricular (fibrous) ring.

The atria are composed of cardiac muscle fibres (the myocardium) arranged in two layers: the inner layer belonging to the atrium proper and an outer layer common to both atria. The ventricles have thicker walls and the left ventricle wall is the thickest. The left and right atria (LA and RA) are separated by the atrial septum and the left and right ventricles (LV and RV) are separated by the ventricular septum. The interior of the heart is lined with the endocardium which is similar in form and function to the epicardium, it also extends to cover the interior of the great vessels. The interior of the ventricles is far from smooth as muscular structures protrude from the surface and these are known as trabeculæ carnæ.

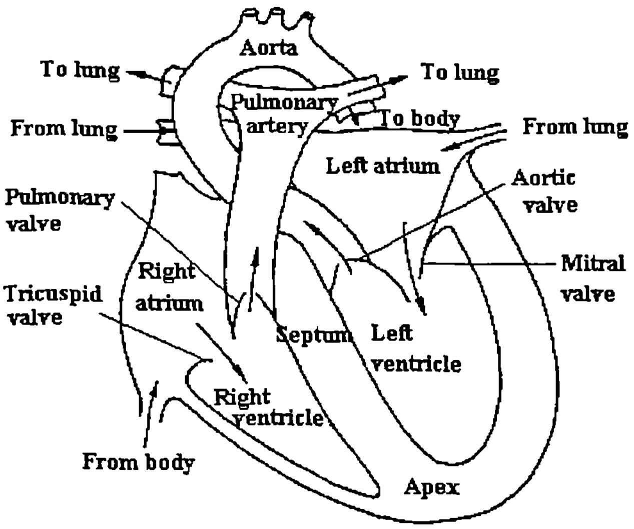


Fig 1.1 This is a diagrammatic representation of the human heart. It shows the relative location of the main structures and the direction of blood flow. (From Campen *et al.* [49])

The fibres of the myocardium are arranged uniformly such that as one moves through the thickness of the myocardium the angle the fibres make with the horizontal changes. From experiments (Streeter 1969 [9]) it has been found that angular change, for the left ventricle, as one moves from the epicardium to the endocardium is approximately -60° to

60° in diastole. This angular change is smooth and is often assumed to be linear and not as had been previously believed arranged in discrete shells.

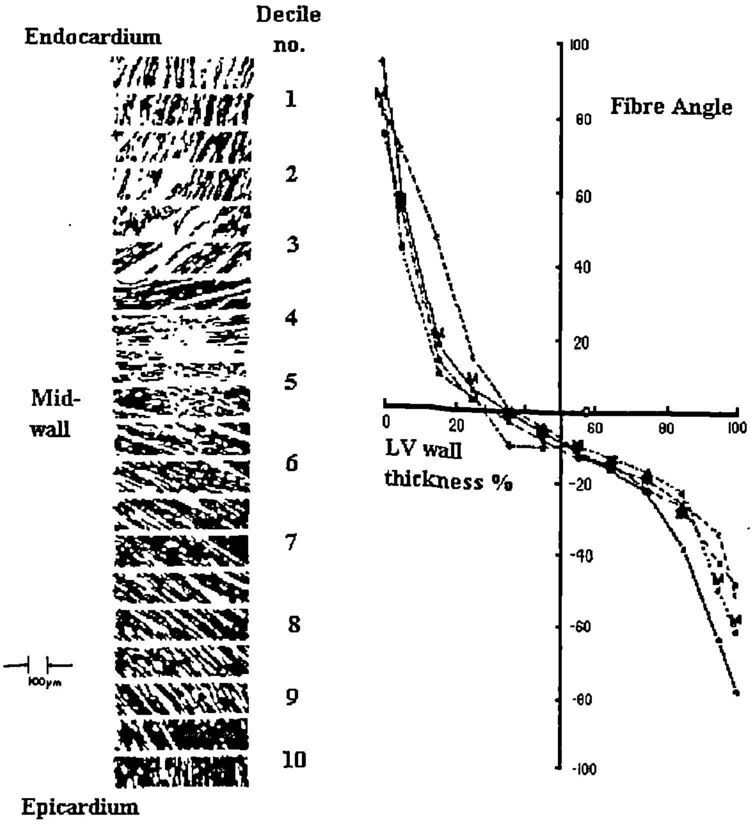


Fig 1.2 This shows how the orientation of the fibres within the myocardium vary through the thickness of the ventricular wall. On the left are slices taken at various depths through the wall. (From Streeter *et al.* [9])

The great vessels are responsible for the transport of blood into and out of the heart. Oxygenated blood enters the left atrium via the pulmonary vein; the pulmonary vein is the only vein to carry oxygenated blood. This oxygenated blood leaves the left ventricle via the aorta. Deoxygenated blood enters the right atrium through the superior vena cava (this blood comes from the upper part of the body) and the infenor vena cava (this blood comes from the lower part of the body).

The heart contains a number of valves which ensure that blood flows through the heart chambers in only one direction. Surrounding each valve is a fibrous ring and this acts as anchors for the valves to be attached to and helps maintain the shape of the valve orifice. The four pulmonary veins that empty into the left atrium do not possess any valves. The left atrioventricular or mitral valve lies between the left atrium and the left ventricle, it is a bicuspid valve consisting of two triangular cusps. The cusps are attached by chordæ tendineæ to papillary muscles which ensure that the valve cannot be turned in-side-out. This valve ensures that blood only passes from the left atrium into the left ventricle. The aortic valve consists of three semilunar cusps and is similar in structure to the pulmonary valve only much larger and stronger. This valve prevents blood entering the left ventricle via the aorta. The superior vena cava does not have a valve at all and the inferior vena cava has only a rudimentary semilunar valve to restrict blood flow from the right atrium. The right atrioventricular (tricuspid) valve consists of three triangular cusps. These cusps are secured by chordaæ tendineæ to the papillary muscles, as in the case of the mitral valve. This valve prevents blood from entering the right atrium from the right ventricle. The pulmonary valve consists of three semilunar cusps and prevents blood flowing from the pulmonary vein into the right ventricle.

Located close to the superior vena cava upon the right atrium is the sinu-atrial node or pacemaker. The pacemaker is responsible for ensuring the timing of cardiac contraction. An electrical signal generated in the pacemaker causes the left and right atria to contract (atrial systole). The electrical impulse is carried through the atrial myocardium along preferred pathways. These pathways are where the muscle fibres are thicker than those in the rest of the atrial myocardium. When the electrical impulse reaches the atrioventricular node, which lies in the septal fibres of the right atrium, the impulse is conducted to the ventricular myocardium. The impulse travels long the bundle of His which is composed of Purkinje fibres. Purkinje fibres have an approximately ten fold greater rate of electrical conduction compared to the rest of the myocardium. The bundle of His divides into two parts one that activates the right ventricle and the other the left ventricle. Both these two halves subdivide many times to provide electrical activation at many points within the myocardium of the left and right ventricles.

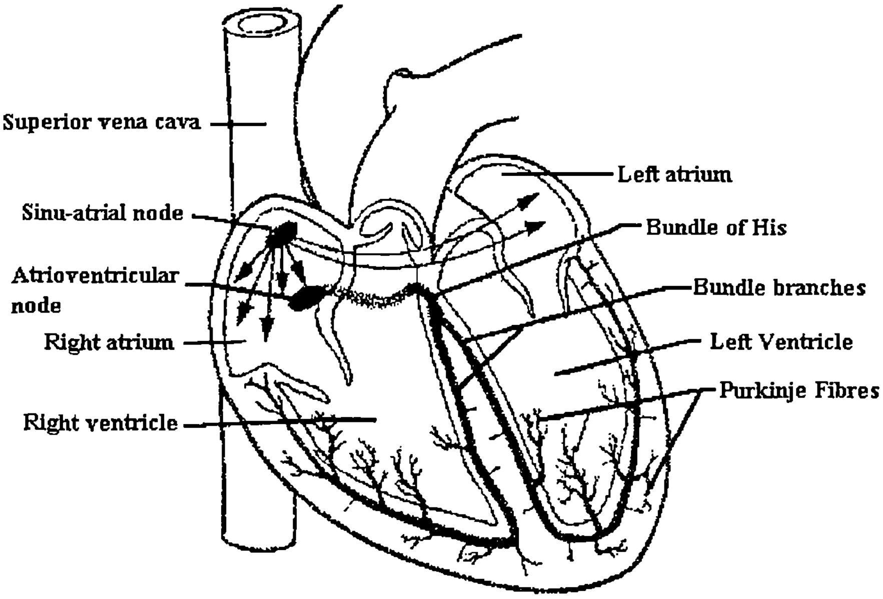


Fig 1.3 This diagram shows the location of the sinu-atrial node, the atrioventricular node and the bundle of His. All of these features are shown in heavy black. Arrows protruding from the sinu-atrial node represent the movement of electrical activation from that node. (From Berne and Levy [11])

To prevent the heart from moving within the chest cavity it is bound to three structures: the parietal pericardium is attached to the root of the lungs and thus in turn to the thorax and is also bound to the diaphragm. Lastly the heart is restrained by the super vena cava which is attached to the structures of the neck including the surrounding fibrous material (Keith 1907 [10]).

The above is only an outline of the major structures of the human heart. For an excellent introduction to the human heart "Principles of Physiology" (1996 [11]) provides a good grounding and when combined with the canonical text "Gray's Anatomy" (1947 [12]) the information obtained should be sufficient for most needs.

1.2 The Cardiac Cycle

The cardiac cycle can be broken down into a number of distinct stages which are easily identified from the pressure/time curve for the left ventricle, the left atrium and the aorta (Fig 1.4). There are apparent rapid changes in pressure and these correspond to valve openings and closures and changes in myocardial activity.

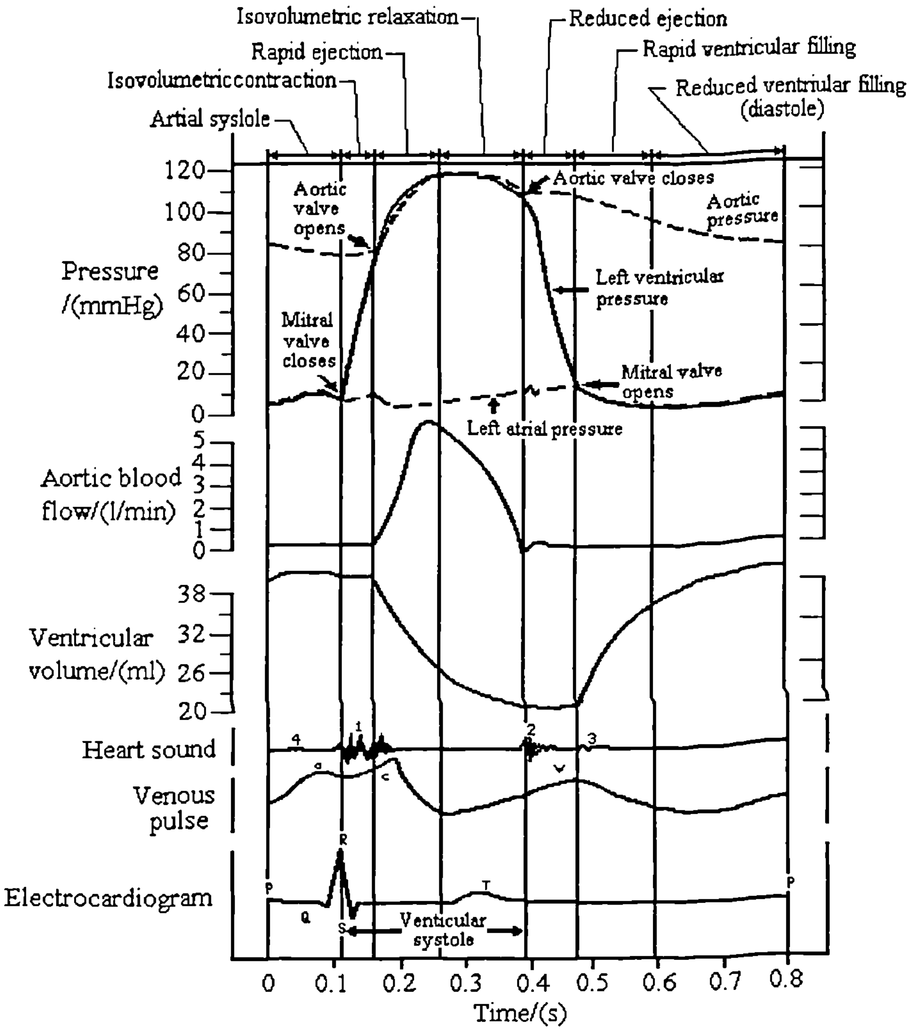


Fig 1.4 The pressure time trace for the left half of the human heart, the atrium, the ventricle and the aorta. The right hand side s similar in shape with reduced pressures. Other detectable features are also shown; the venous pulse, the heart sound and the electrocardiogram. (From Berne and Levy [11])

Atrial Systole: This marks the onset of the systolic (contraction) phase of the cardiac cycle. The electrical activation of the sinu-atrial node results in a peristalsis-like wave of atrial contraction. The blood in the atria can flow into the great veins as well as the ventricles, as there are no valve structures to prevent a back flow of blood. This in practice is minimal as the brevity and the weakness of atrial systole is not normally sufficient to overcome the inertia of the blood entering the atria. Atrial systole contributes about 30% to the stroke volume of the left ventricle. Despite this it is possible with complete atrial systolic failure for sufficient blood flow to be maintained.

Electromechanical Delay: This is the earliest stage of ventricular systole and corresponds to the time when electrical stimulation has begun and before the mitral and tricuspid valves close due to the increased cavity pressure.

Isovolumetric Contraction: This is the time when the valves of the left and right ventricles are closed and the myocardium is contracting. As the name suggests the volume remains constant, as there is nowhere for the blood to flow, and the cavity pressure increases.

Ejection: The ejection phase is marked by the opening of the aortic and pulmonary valves. This part of the cardiac cycle may be further divided into an early, rapid ejection, phase and the longer, reduced ejection, phase. The early phase is characterised by a sharp rise in aortic and ventricular pressures, a sudden decrease in ventricular volume and a large aortic blood flow.

Isovolumetric Relaxation: This part signals the beginning of diastole, the passive part of the cardiac cycle. The pressure in the aorta is now greater than that of the left ventricle. This pressure difference forces the aortic valve shut and results in a notch being formed in the pressure trace for the aorta. The atrioventricular (mitral and tricuspid) valves remain closed during this stage and the cavity volume of the ventricles remains constant as the ventricles relax, hence isovolumetric relaxation.

Rapid Filling Phase: The atrioventricular valves now open due the reduction in pressure within the ventricles, caused by their relaxation, and the increased pressure in the atria, due to an inflow of blood from the great veins. This phase accounts for most of the ventricular filling. During this stage the pressure continues to fall within both the ventricles and the atria despite the increasing volume. This is due to the fact that the ventricles and atria are relaxing and hence continue to exert less and less force upon the blood within them.

Diastasis: This is the truly passive phase. The pressure and volume within all the heart chambers rise under the pressure from the inflow of blood from the great veins. This part is characterised by long slow filling of the chambers.

The above process is repeated many times a minute, approximately 60-80 times (beats) per minute at rest. The heart rate may under loading increase to 180 beats per minute, an increase above this value is possible for only a short period. The resting heart rate also declines with increasing age.

1.3 Structure and Function of the Myocardium

The myocardium consists of muscle fibres held together by collagen fibres (Fig 1.5). The muscle fibres or myocytes make up approximately 70 per cent of the myocardial volume. Only about 1.5 per cent of the myocardium is accounted for by the network of collagen fibres. Although they are only a very small proportion of the total myocardium they can greatly affect deformation of the myocardium (Borg Caulfield 1981 [13] and Horowitz *et al.* 1988 [14]). This is due to the fact that they are several orders of magnitude stiffer than the myocytes. The collagen mesh is sometimes referred to as the "fibrous skeleton" and helps maintain the shape to the heart. It may also store large amounts of elastic energy and may thus contribute to the restoring forces in diastole. The remaining myocardium is composed of various interstitial components including a large fluid component.

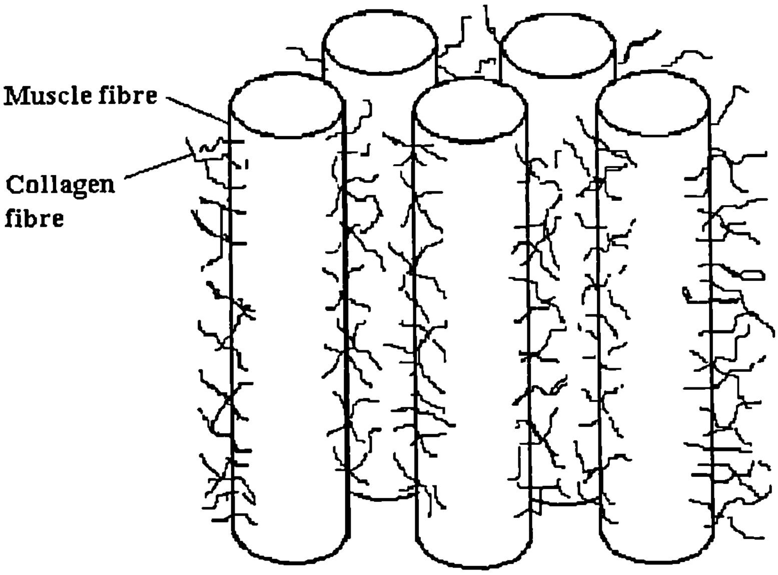


Fig 1.5 The myocardium is composed of muscle fibres bound together by a mesh of collagen fibres. (After Horowitz *et al.* [14])

The sarcomere (Fig 1.6 and Fig 1.7) is the basic contractile element and as such is the only part that actively contributes to the performance of the myocardium. That is not to say that other parts do not play an important role, but their contribution is purely passive on their part. These non-active elements ensure that the force generated by the sarcomeres is efficiently transferred to the blood in the ventricular cavity. From the diagram it can be seen how the basic structure is formed. The contraction of the sarcomere is produced by a series of chemical reactions known as chemomechanical transduction. This results in a cyclic process where the head or crossbridge attaches at an angle of 90° to the actin filament, bends to an angle of 45° and then releases (Fig 1.8). This single process results in approximately a 10nm reduction in the length of the sarcomere, which itself may have a length of 1-4μm. This process is repeated may times with some crossbridges contracting while others hold the actin filament. The single cycle may result in only a tiny overall length reduction, but when the process is repeated many times the length reduction can be substantial.

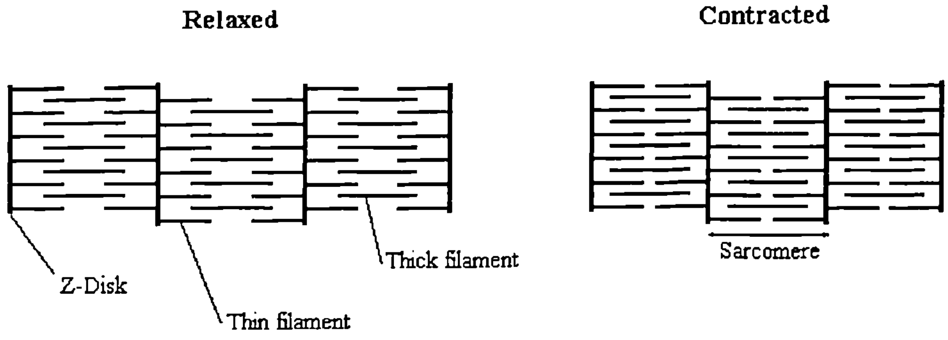


Fig 1.6 This diagram shows how the basic units of a muscle fibre (sarcomeres) fit together. It should be noted that no part of the sarcomere reduces in length during muscle contraction. The reduction in fibre length is due to parts sliding over each other and interleaving. (After Berne and Levy [11])

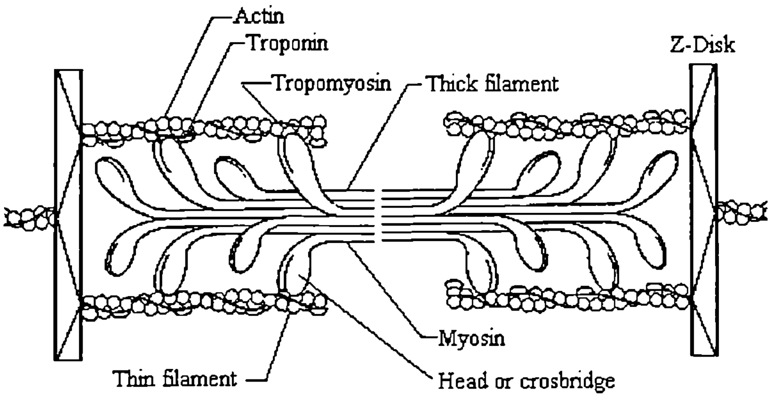


Fig 1.7 This shows the elements that form a single sarcomere. This diagram is much reduced in terms of the number of components. A real sarcomere may be composed of approximately 300-400 myosin molecules. (From Berne and Levy [11])

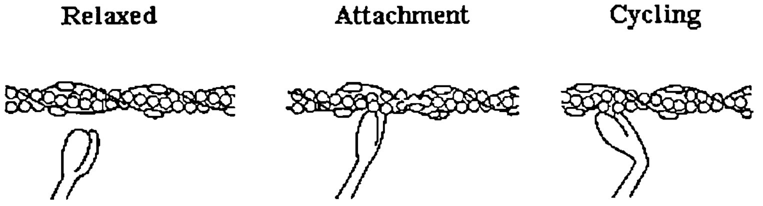


Fig 1.8 This diagram illustrates the process by which the crossbridge elements cause the length of the sarcomere to shorten. (From Berne and Levy [11])

1.4 Diseases of the Heart

The heart is a complicated organ consisting of many parts any of which may become damaged by infection or excessive mechanical wear. A defect of a single part of the heart may affect the performance of the organ as a whole as there is a great interdependency between the individual parts. For instance if the left ventricle fails to fill to the correct volume then it is obvious that this will affect the output of the ventricle during contraction. Thus even though the initial fault was in diastole the effect may be more obvious in systole. It is also the case that when one part of the heart malfunctions it may put an excessive load on the other parts of the heart. It is for this reason that many heart diseases are associated with each other and it is often the secondary condition that causes the clinical condition of "heart failure."

Diseases that affect the heart are characterised by the type of condition and the part of the heart affected. For example carditis is inflammation and can be associated with the pericardium, the endocardium or the myocardium. Stenosis and insufficiency are valvular conditions and may affect any of the four cardiac valves. There are many diseases that can affect the performance of the heart and some of the more common are described below:

Coronary Artery Disease (CAD): This disease is due to a narrowing of the lumen of the coronary arteries. The main cause of this is cholesterol in the blood depositing on the walls of the arteries. This most commonly occurs at either a bifurcation of an artery or at a sharp bend. Over time there are fibrous reactions with the deposit and crystallisation and calcification may occur. The arteries may become further blocked and the blood supply to the myocardium can be severely restricted. This restriction can result in angina pectoris (chest pain), myocardial infarction and ischaemia or sudden death, depending upon the extent of the occlusion.

Myocardial infarction and Ischaemia: Coronary artery disease is the largest cause of this and it results in more deaths in western countries than any other heart condition. Ischaemia is when there is an impairment of myocardial function due to a loss of coronary blood flow. Myocardial infarction is the death of the myocardium, which may be for any reason. Myocardial infarction is not instantaneous following a complete loss of blood flow but takes several minutes. If adequate blood supply is restored within 18 minutes then there may be no lasting damage to the myocardium (Levine and Gaasch 1985 [15]). However after 20 minutes there is a sliding scale of damage leading ultimately to total necrosis of the affected tissue after six hours. This dead tissue is gradually replaced with stiff non-functioning scar tissue over a period of a few weeks.

Hypertrophy (Hypertrophic Cardiomyopathy): With increased load the myocardium naturally increases in mass to compensate for the increased cavity pressure and this keeps the maximum systolic wall stress approximately constant. In Hypertrophy this increase in myocardial mass is excessive and the deeper parts of the myocardium may not get sufficient blood supply. Coupled with this the internal cavity volume may also be reduced and hence further impede cardiac performance, particularly during diastole.

Dilated (Congestive) Cardiomyopathy: Is the commonest form of cardiomyopathy. There is a decrease in the contractile force of the ventricles and a corresponding increase in cavity size. Coupled with this increase in cavity size there are often associated electrical problems such as arrhythmia.

Carditis: Is an inflammatory disease and can affect the endocard um, pericardium or the myocardium. In endocarditis the endocardium becomes damaged and wart-like growths can be formed. This is normally most severe around the valves, since these are subject to the greatest amount of friction. The wart-like growths are also a site for deposition of fibrin that will further interfere with blood flow. In pericarditis the pericardium becomes dry and ridges of fibrin are formed due to the movement of the two surfaces over each other. This condition can result in a rapid weak pulse and low blood pressure. In myocarditis the myocardium itself becomes inflamed and may result in sinus tachycardia and arrhythmia. In the severe cases the above conditions can result in sudden death.

Valvular Disease: This can affect any of the four cardiac valves, mitral, pulmonary, tricuspid and aortic. In stenosis the orifice within which the valve is situated becomes reduced in size. The result is that a greater pressure is now needed to maintain the same rate of blood flow through the narrowed opening. In the more severe cases this pressure increase will have already reached a maximum and sufficient blood flow will not be possible. In regurgitation the valve leaflets may become damaged or the valvular orifice may become enlarged. Whatever the reason the effect on blood flow is the same. The valve is now unable to close effectively and this means that blood will be able to flow backwards, be regurgitated. Both these conditions may lead to compensation by the heart chambers and this in turn may lead to other conditions affecting the heart.

Electrical disorders: These include disorders affecting the rhythm of the heart and conduction in the myocardium. The rhythm disorders can be in the form an accelerated, decelerated or uneven rhythm (tachycardia, bradycardia and arrhythmia). It is also possible for only some parts to be affected and for rapid uncontrollable activation to occur. Conduction problems may, in a minor case, result in an occasional atrial or ventricular beat being missed. It is also possible for conduction problems to result in uncoordinated contraction of the myocardium or part of the myocardium failing to contract.

1.5 Summary

It is clear the heart is an extremely complicated structure both in terms of its geometry and the microstructure of the myocardium. It should also be obvious that it will not be possible to model every aspect of the heart. A truly all encompassing model would include the following: the active and passive elements of the myocardial micro structure, all the structures of the heart, forces exerted upon the heart from outside it (the lungs, diaphragm and the chest wall) and lastly the action of the blood within the heart chambers.

It may be possible to devise a model that incorporated all the above features. There are however two major problems. Firstly, to solve such a sophisticated model would be far beyond today's computing power in terms of both storage and execution time. Secondly, there is a lack of highly accurate data, which such a high precision model would require. Obviously radical simplification must be made. The aim is to reduce the heart to its computationally simplest form and yet still be able to gain useful and reasonably accurate data.

Chapter 2

2. Review of Cardiac Research Pertaining to the Mechanical Behaviour of the Left Ventricle

It has long been known that the heart is an important organ, even before the days of the early anatomists. In the beginning of man's civilisation the heart was considered by many cultures to contain magical powers and was often used to symbolise life itself. It is therefore hardly surprising that there has been a great deal of interest in the structure and function of the heart. Despite many years of research it is perhaps unexpected that a complete understanding of the heart is still not available. This is not due to a lack of quality research but is a testimony to the complexity of the heart and the mathematical and computational intractability of highly complicated models. Simplified models can however be useful to aid understanding of the function of the heart and may even be of use in the clinical diagnosis of disease. It is with this hope that research in this area has continued over the years. The advent of affordable and powerful desktop computers has been a great boost to cardiac research. It has allowed complex finite element models of the heart and in particular the left ventricle to be developed.

2.1 The Development of Cardiac Models

Cardiac research began properly with the early anatomists. The worked to detail the structure of the heart and hypothesised upon the function of the individual parts. In the 15th century Lenardo Da Vinci wrote about the movement of the heart wall which had been observed by the implantation of metal pins through an animal's chest wall. This early interest could only advance understanding of the heart by a limited amount. It is only when high precision instruments became available that the properties of the myocardium, the heart muscle itself, could be directly measured. This however only gives some of the information necessary for an understanding of the heart. Just as important are the developments in mathematical modelling techniques, medical imaging and pressure measuring techniques. The mathematical techniques allow more realistic geometric and material modelling and the data acquisition techniques make it possible to obtain *in-vivo* geometric data and also *in-vivo* pressure information.

2.1.1 Development of Thin Walled Models

The first attempts to model the left ventricle were made in 1892 and exploited Laplace's Law. This provides a relationship for every point between pressure, tension and the orthogonal radii of curvature in a thin shell structure. This method was not used again until sixty years later by Burch (1952 [16]) who used a spherical model to derive the forces, generated by the cavity pressure, within the ventricle wall. An analysis of the importance of ventricular size and shape was also performed by Burton (1957 [17]). A refinement of this type of analysis was produced by Sandler and Dodge (1963 [18]) where an ellipse of revolution was taken as the ventricle geometry. The shape of the ellipse of revolution was based on single plane angiography of human patients. This model was later used by a number of other authors to analyse human and animal patient data. The major drawback with this type of model is in the assumption that the left ventricle can be modelled as a thin walled vessel. It may be justified to use this approach for a vessel whose radius is an order of magnitude greater than its wall thickness, but this is not the case with the left ventricle where the ratio is approximately 4:1 in diastole and up to 2:1 in systole. The membrane model does not allow for stress distribution through the wall neither does it permit shearing of the material both of which may greatly affect ventricular behaviour. Despite this it has been shown that not only do these models give wall stresses to within an order of magnitude but may also be of some clinical use

(Mirsky 1969 [8]) provided local stress values are not required. A more thorough review of thin walled models can be found in Vinson (1977 [1]).

It should be obvious that the next developments should include non-uniform wall thickness, and allow for shearing and bending and uneven stress distribution through the wall. Thick walled models were capable of these features and were developed using thick shell theory, which is an extension of Laplace's Law. Not all of the advantages of thick walled models were utilised from the outset and the new features were incorporated gradually.

2.1.2 Development of Thick Walled Models

In 1968 Wong and Rautaharju [19] developed the equations which allowed the stress distribution calculations for a thick walled ellipsoid of revolution. The model was similar to that previously used by Sandler and Dodge (1963 [18]) except that the stresses were allowed to vary through the thickness of the wall. Analysis of the diastolic phase showed that the stresses were highest at the endocardium and decreased towards the epicardium where they were lowest. This model did not include the possibility of shear within the material which was included by Ghista and Sandler (1969 [20]). They used single plane cineangiography to calculate the size and shape for their ellipsoid of revolution model. They were able to show that the stress distributions in a viscoelastic and an elastic model were similar.

Despite their inclusion in earlier models some still did not include bending and shear. A truncated ellipsoid of revolution model was used by Streeter *et. al.* [21] in 1970 which was composed of nested tethered shells each with a unique fibre direction. The geometry was based on that obtained from canine hearts at end diastole and systole. The stress analysis was performed by considering fibre orientation and wall curvature.

It has been shown that these simple models could provide reasonable estimates to overall stress levels in the myocardium. In 1970 Ghishta and Sandler [22] were able to use these simple models to predict the oxygen consumption of the heart. This method only involved catherisation of the left ventricle and cineangiocardiography. The previously used method had required the measurement of the blood flow into and out of the heart as well as the oxygen concentration in the blood. This new method is far less invasive and hence much less risky to patient health.

Further sophistication was achieved in 1973 by Wong [23] who used a thick walled truncated ellipsoid with non-uniform wall thickness. The diastolic material properties were assumed to be isotropic, viscoelastic and homogenous. The myocardium was modelled using a Hill's model and Huxley's sliding filament theory. This method gives passive and active fibre tension as separate quantities within each fibre and was used to analyse isovolumetric contraction.

In 1980 Janz [24] compared the equatorial stresses of an ellipsoid of revolution model with those of a prolate spheroid model. It was concluded that the prolate spheroid was the most reliable model for stress prediction. The conclusion reached from investigating the systolic stress in a dog was "that the ratio of equatorial wall thickness to cavity semi-minor radius appears to be the dominant geometric factor in determining this stress." The fibre stress and fibre force during ejection were also found to decrease more rapidly than left ventricular cavity pressure. It was noted that the models were very sensitive to errors in measurement of the wall thickness which is also the hardest parameter to measure.

In 1982 Janz [25] produced two equations which could predict the averaged local circumferential wall stresses. The thick walled model was produced as a solid of revolution from a single plane angiogram. The stress results were compared to that for a similar finite element model and the results were found to be relatively consistent. The stress values were found to differ by 5-25% depending on the area of the model being investigated. There were however problems with each of the two formulae. For the first equation there is a need to calculate perpendicular lines to the model's surface. In practice this is prone to large computational errors and may make this formula unusable in general. For the second equation in an area of simulated aneurysm the values differed by more than a factor of two. It was however felt that "judicious use of both" equations "yield meaningful estimates of local averaged circumferential stress."

A cylindrical model was demonstrated by Tözeren in 1983 [26]. The myocardium was modelled as solid fibres in an invicid fluid matrix. The cylinder was thick walled with the fibre angle varying through the wall thickness. It was found that wall thickness and fibre orientation do not affect the diastolic pressure volume relationship but the latter does affect the geometric deformation. In systole the pumping efficiency was shown to increase with increasing wall thickness and fibre contractibility. The rotation of the cylinder was found to be small during diastole but much larger during systole. The model did show some qualitative agreement with experimental data, but the constitutive equations were purely theoretical and based on animal data.

The use of contractile filament stress and strain with circumferential and longitudinal fibre strain was employed by Phillips and Petrofsky (1984 [27]). Introduced into the model were compressive strains which were used to generate the fibre stresses. These stresses and strains were then used to calculate the active systolic elastic modulus for the circumferential and longitudinal axes. The values for these paran eters were determined in an *ad hoc* manner for only four points in the systolic phase. The data used came from thirty-nine human patients with various pathological conditions and was of the form of single plane cineangiography plus pressure readings. It was found that the active systolic moduli decreased during systole in an exponential manner. There was also some grouping of active moduli values for patients with similar pathologic conditions.

A truncated thick wall ellipsoid of revolution model was used in 1985 by Kim *et al.* [28] to derive stresses from observed strains obtained from coronary cineangiography of dogs. The bifurcation points of the coronary arteries were used as local markers and the accuracy of using these was compared with the more invasive method of using implanted lead beads. The new method was found to perform reasonably well, it is however necessary for the coronary arteries to be in good condition otherwise the bifurcation points may not be visible. The two main advantages of this method are that it can be performed with data that would already be gathered, during assessment of the condition of the coronary arteries, and it is far safer than implantation of lead beads.

A more extensive review of thick walled models has been done by Grewal (1988 [2]) who also provides references to other complementary reviews. Most of the thick walled models were used to calculate approximate stress distribution within the vessel walls. The simplistic geometry makes the values obtained much less meaningful than this model type might have provided had the geometric approximations been more suitable. One example of this simplification is that the models define the internal surface of the ventricle as being everywhere concave. In reality parts of the ventricle wall are convex especially at the apex end of the septal region. Due to this thick wall models have contributed relatively little to the understanding of ventricular mechanics. They do give variation of stress though the wall, but the local values are somewhat suspect due to the geometry and thin walled models give reasonably good global values. It is also the case that similar thick walled models have given stress distribution that have not been compatible with each other (Huisman *et al.* 1980 [29]). It has been the conclusion of several reviews that the thin walled models are of far greater clinical use.

To incorporate a more realistic geometry as well as other considerations, including inhomogeneity, anisotropy and non-linear material properties, a more advanced method of modelling must be used. The finite element method is ideal for this sort of problem and was developed in the late 1950's. The only drawback this method has is that it is computationally expensive and that has been the reason for its lack of widespread use as an analytical tool. The rapid increase from the mid 1980's in affordable computing power has negated this earlier bar to its use and has seen this method become the tool of choice for this sort of modelling.

2.1.3 Development of Finite Element Models

This part of the review will concentrate on finite element based research from 1990 with only an overview of earlier work. More comprehensive reviews of the preceding years can be found in Vinson (1977 [1]) and Grewal (1988 [2]).

In 1972 Janz and Grimm [30], who had previously used thick shell theory, began working with the finite element method to analyse the mechanical behaviour of the left ventricle. This first attempt at left ventricular finite element modelling involved ring shaped elements to form a solid of revolution model. The geometric data used was obtained from cross-sections of post-mortem rat hearts. The myocardium was modelled as partly isotropic and as partly transversely isotropic. It was found that the model deformation most closely matched the measured deformation when the inner third of the myocardium was transversely isotropic and the rest was isotropic.

The 20-noded isparametric brick element was developed in 1971 by Zienkiewicz [31] and is ideally suited to the finite element modelling of the left ventricle. This element type was used in 1974 by Hamid and Ghista [32] to build a model able to predict left ventricular chamber stresses as well as the stresses on the aortic valve. This was to enable the design of prosthetic aortic valves and artificial heart chambers. The geometry was produced from a single plane cine-angiographic image at mid-ejection. Their model predicted peak stress levels in the left ventricle three times higher than had been predicted by the previously used thick shell models. Comparisons were made between the effect of normal and partially infarcted left ventricles. It was found that the tissue surrounding the area of infarct took some of the load off the area of infarct.

Plane-strain finite element analysis was employed by Pao *et al.* [33] in 1976 to analyse the mid-plane between the base and the apex during diastole. Data had been obtained from an isolated canine left ventricle using roentgenographic recordings and the myocardium was modelled as an isotropic homogeneous material. It was found from the model that for the anterior and posterior regions the maximum circumferential stresses occurred on the endocardial surface. Contrary to this for the septal and free walls the maximum circumferential stresses were on the epicardial surface. It was also noted that the stress gradients were always steepest near the endocardial border. The need for extension to three dimensions was already commented upon as was the fact that an intact *in-vivo* heart would most likely behave differently to the excised isolate left ventricle.

A method for obtaining far better finite element reconstructions than offered by single or even bi-plane cineangiography was presented by Nikravesh *et al.* (1981 [34]). This involved taking 4 to 5 short axis cross-sections and one long axis cross-section using echocardiography. The method not only gives greater fidelity for reconstructed cross-sections than cine-angiography but also gives wall thickness which is partially lacking from cine-angiography. Although finite element reconstructions were shown there was no analysis performed on the finite element meshes obtained.

In 1986 Horowitz *et al.* [35] introduced a new finite element model able to model the entire cardiac cycle for the left ventricle. The data used was fror a human source and had been obtained by computer tomography. The model was formed from two element types. The basic 20-noded isoparametric brick element formed the basic structure. "Truss" elements were superimposed upon this structure to simulate the anisotropic nature of the myocardium with the varying fibre angle. They also provided the directions for the active contraction forces during simulated systole. The elastic modulus of the myocardium is assumed to obey an exponential stress-strain relationship. The active contraction of the model is the only thing altered to produce the required cavity volume changes. Good agreement was found between the deformations obtained from the model and that of the measured data. The only area of concern is that the model increased in length during systole instead of reducing. A consequence of this was that radial contraction was slightly greater to compensate for this. The increase in ventricle length during systole was attributed to the exclusion of the papillary muscles which contract tightly during systole and may thus cause the ventricle to shorten. It should also be noted that the left ventricle "winds up" during systole and "unwinds" during diastole and this was clearly evident from the model.

The effect of acute ischaemia was investigated by McPherson *et al.* (1987 [36]) with data obtained from canine subjects using the method of data acquisition and reconstruction proposed by Nikravesh *et al.* (1981 [34]). The myocardium was assumed to be homogeneous, isotropic and linearly elastic. Areas of acute ischaemia were shown to have higher elastic moduli than areas of normal diastolic myocardium.

A theoretical porous-medium finite element model was employed by Huyghe *et al.* [37] in 1991. The myocardium is assumed to consist of an incompressible liquid and solid which is viscoelastic, with the muscle fibre angle varying linearly through the myocardium. Since the myocardium is considered porous intracoronary blood is free to enter or leave the myocardium allowing for volume strain and redistribution of mass. The model is a truncated cofocal ellipsoid with 96 20-noded isoparaı etric brick elements with three layers through the wall. During passive filling the apex is seen to rotate in a clockwise direction relative to the base, for an observer looking from the apex to the base. In the pressure range 0-3kPa the rotation was about 0.1rad., which was considered to reasonably approximate the rotation of a real left ventricle. This was an initial investigation into the use of a porous-medium model of the myocardium.

In the same year Bovendeerd *et al.* [38] proposed another method to model the myocardium using the same geometric data as Huyghe *et al.* [37]. Myocardial muscle was modelled as an incompressible material consisting of muscle cells and connective tissue embedded in fluid. The active stress generated by the sarcomeres is dependent on time, fibre strain and fibre strain rate and is directed along the muscle fibre direction. Muscle fibres were assumed to be twice as stiff in the fibre direction as in the cross-fibre direction. The model considers varying fibre orientation across the left ventricle wall, anisotropy of passive myocardial tissue, dependency of active stress on time and aortic afterload. The global deformation pattern was found to be asymmetric with respect to an ischaemic region. The left ventricle wall is seen to become thin and the fibres stretched during the ejection phase. Muscle fibre shortening and active fibre stress around the ischaemic region was seen to depend strongly upon the orientation of the border zone with respect to the ischaemic region and the angle of the fibres within it. The ischaemic region made up approximately 12% of the ventricle and was designed to simulate occlusion of the left anterior descending artery. With the ischaemic region the left ventricular pressure was found to be about 12% lower, the ejected volume was 20% lower and aortic flow was also reduced compared to a simulation without ischaemia.

Again in the same year Han *et al.* [39] applied finite element analysis and optimisation to a reconstruction of the geometry of a canine left ventricle, in order to assess myocardial material properties during diastole. The model was subjected to an external or pericardial pressure as well as an internal pressure and the elastic modulus changed in order to match the left ventricle volume at the next time-step. The geometry was obtained from eight short axis cross-section echocardiographic images taken at several intervals during diastole. The pressures in the left and right ventricles were obtained with catheter tipped pressure transducers inserted into both left and right ventricles. The average pericardial pressure was calculated from the right ventricular pressure. The computed pericardial pressure was found to play a significant role in the performance of the left ventricle. The one drawback of this work is that results were only presented in respect to the effect of pericardial pressure on left ventricular cavity pressure and myocardial elasticity, and not to its effect upon global deformation.

Again in this year Fann *et al.* [40] placed twenty two radiopaque markers into the myocardium of canine subjects. These were located at the anterior, lateral and posterior subepicardium and subendocardium at the midventricular level. Eight hours later bi-plane videofluoroscopy was used for imaging of the ventricles of the closed-chested canine subjects. It was found that circumferential shortening occurred in all layers and regions, however longitudinal shortening did not occur in the posterior endocardium. Principal shortening was found to be greater in the subendocardium than in the subepicardium. Some concern was expressed as to whether such an invasive method of obtaining data would affect the global deformation of the heart.

Still in 1991 Han *et al.* [41] produced cross-sectional views of a canine left ventricle during diastole using ultrasound. The myocardium was assumed to be a homogeneous and isotropic elastic material. Using the finite element method the elastic modulus of the myocardium was calculated in order to match the volume at successive time-steps. The results suggest that during diastole the elastic modulus and pressure are linearly related, the pressure/volume curve is exponential and this would suggest that the passive myocardium has an exponential stress-strain relationship. Due however to the limited number of points at which this analysis was performed the e idence is far from conclusive. This is particularly true for the pressure/elasticity curve which could easily be interpreted to be quadratic rather than linear. The assumed material properties for the myocardium also give cause for concern when trying to prove relationships between material properties and global function.

Two single plane X-ray angiographic images were taken by Roy *et al.* [42] in 1992 one at end systole and the other at end diastole. The myocardium was then divided into triangular finite element regions and each region then assigned an independent elastic modulus. The boundary at end systole is then deformed under pressure so as to match the boundary at end diastole. The elastic modulus of each of the elements was varied so that the best least squares fit for the boundary could be obtained. Ischaemic regions should show up as areas of significantly greater elastic modulus. The method was used to successfully identify a patient with an ischaemic region. The major drawbacks to this method are, that since it is 2-D an area of ischaemia may be in a position so as not to be adequately represented by this slice and thus not be detected. The wall thickness was also assumed to be constant and is thus far too thick at the apex.

Also in 1992 Bovendeerd *et al.* [43] used the model by Huyghe *et al.* [37] as the basis for their work. The muscle fibre angle distribution through the myocardium is varied in order to make the active muscle fibre stress homogeneous throughout the myocardium layers. The active stress is the stress that is generated in the sarcomeres and is dependent on time. Fibre strain and fibre strain rate are directed along the muscle fibre direction. The fibre angle is taken as the angle between the fibre and the local circumferential direction, this is also known as the helical angle. With an endocardium angle of 60°, mid-wall angle of 0° and an epicardium angle of -60° the maximum active muscle fibre stresses at the equatorial region were 110kPa, 30kPa and 40kPa, in the respective corresponding myocardium layers. However, when the mid-wall angle is changed to 15° these active muscle fibre stresses are all within the range 52kPa-55kPa. The change in muscle fibre strain is also seen to be more homogeneous. The problem with this approach is that no allowance is made for the effect that geometry may play in stress distribution. The left ventricle is never stress free so these are stress changes and not absolute stresses as is implied.

A year later in 1993 McCulloch *et al.* [44] used a 128 element finite element model with linear nodal interpolation to represent the intact left and right canine ventricles. The original geometry had been obtained from anatomical measurement of the canine heart. Myocardial properties were taken as homogeneous and anisotropic with the fibre angle taken to vary through the wall from -51° (epicardium) to 59° (endocardium). The diastolic phase of the cardiac cycle was simulated and the results compared to those obtained from an invasive study upon a living dog. Significant regional variations in stress and strain were observed. The strains were compared to those obtained by direct measurement of a real canine heart. Radial, circumferential and longitudinal strains agreed closely with experimental results. However transverse shear strains were wildly inconsistent. It was concluded that ventricular geometry contributes greatly to the heterogeneity in the mechanical function of the heart.

The same year saw Han *et al.* [45] use echocardiographic imaging to obtain data from canine subjects. The subject had 3-4 plexiglass markers inserted at the mid-papillary muscle level of the myocardium. The animal was given a week to recover from this before imaging occurred. During imaging the animal was mechanically ventilated. Eight short axis views of the left ventricle were taken at 4-5 intervals during diastole using echocardiography. The images were hand digitised to obtain the left ventricle epicardium and endocardium outline. Pressure readings were obtained with a transducer-tipped catheter inserted into the left ventricle via the femoral artery. The model used 64 8-noded isoparametric solid elements. The myocardium was assum d homogeneous and isotropic. An optimisation routine was employed to obtain the best least squares node for node match for the finite element model between successive time intervals. It was alleged from the results obtained that there was a linear relationship between the elastic modulus of the myocardium and the pressure within the ventricle. In this study they also calculated the value of Poisson's ratio between 0.425-0.485, thus the assumption that the myocardium is incompressible would appear reasonable. As with Han *et al.* (1991 [41]) only a few data points were obtained (three or four) so it is impossible to say with any authority that the relation between passive elastic modulus and pressure is linear.

A study of an infarcted region of the left ventricle of a dog, due to coronary occlusion, was made by Pao *et al.* [46] in 1993. Tomographic images of cross-sectional ventricular shapes were obtained for early and end diastole using 14 X-ray sources. Pressure readings were taken using a catheter inserted into the left ventricle via percutaneous arterial and venous routes. The plane-strain finite element method was used to analyse a 2-D short axis ventricular slice. It was found that the muscle in infarcted regions became stiffer than the rest of the diastolic myocardium and this became more evident at higher ventricular pressures. This method is similar to that used by Roy *et al.* (1992 [42]) and is also likely to suffer from the same problems. The short axis cross-section may however give better results as areas of infarction are more likely to be longitudinal in orientation due to the direction of coronary arteries.

The effect upon wall stress of simulated infarction was performed using an experimental and a finite element model devised by Eberhardt *et al.* (1993 [47]). Both models used a spheroidal shape and the experimental model was constructed from 840A-urethane which has an elastic modulus thought to be similar to that of passive myocardium, about 280kPa. It was found that the circumferential stresses in the finite element and experimental models differed by less than 1% at the outer wall and 4% at the inner wall, when both models were subjected to an equivalent internal pressure Simulated infarction was found to only affect surrounding material and had no effect upon the opposing wall. Despite the fact that this model is isotropic and spherical it is still probably reasonable to assume the main result, that ischaemia causes only local distortion in diastole, to be correct.

Also in 1993 Guccione *et al.* [7] used a finite element mesh formed from anatomical measurements of the isolated left and right ventricles of a dog, which was subjected to an internal pressure. The model assumed the myocardium to be homogeneous though not isotropic and incorporated the varying fibre angle through the left ventricular wall. It was found that sarcomeres at the anterior epicardium contribute less to left ventricle pressure than those elsewhere. Differences were noticed in the sarcomere length between the anterior and posterior free walls during ejection. These variations are believed to be due to different geometries in those areas, since the myocardium is homogeneous. This effect of geometry highlights the pitfall of using unrealistic geometry (spheres, ellipses of revolution, etc.). In simplified geometric studies the geometric effect upon local deformation would have been interpreted as a material difference.

In 1994 Bovendeerd *et al.* [48] proposed another enhancement of the model first described by Huyghe *et al.* [41]. A further investigation was carried out into the effect of altering the muscle fibre orientation. This time the muscle fibres are assumed to spiral inwards from the epicardium to the endocardium. Three runs of the model were made; the first run had the angle of spiral always zero and the helical angle -60° at the epicardium, 0° at mid-wall and +60° at the endocardium. The second was the same except the mid-wall angle was 15°. The third was the same as the second except that the angle of spiral varied transmurally and had a maximum at the mid-wall and was zero at the endocardium and epicardium. It was found that a change in the helix angle had only a slight effect on the deformation pattern, though stress and strain are very sensitive to the spatial distribution of muscle fibre orientation. This however has very little effect upon pressure/volume relationships. Transmural crossover of muscle fibres reduced the transmural shear loading of passive tissue. Circumferential-radial shear strains calculated with this model qualitatively disagreed with experimental data. This is explained by the model being very sensitive to transmural crossover angle and the limited data available on this.

In 1994 Campen *et al.* [49] compared how well the two-phase axisymmetric porous medium non-linear finite element model, Huyghe *et al.* [41], and the three-dimensional finite element model, Bovendeerd *et al.* [48], predicted results obtained from canine experiments. With the two-phase model transmural intramyocardial pressure gradients are qualitatively consistent with experimental data. Stiffening of the myocardium due to an increase in intracoronary blood volume was also qualitatively correctly predicted by this model. The three-dimensional Bovendeerd model shows that regional distributions of local ventricular wall stresses are very sensitive to the spatial distribution of muscle fibre orientation. This however has little effect upon ventricular pressure or aortic flow. Important aspects of the mechanics of an ischaemic left ventricle are predicted by this model, such as a change in the pressure/volume loop, fractional decrease in stroke work and spatial redistribution of wall stress and strain. The comparison shows that both models have strengths and weaknesses and the recommendations for improvement are too numerous to list here.

Again in the same year Hashima *et al.* [50] attached twenty five lead markers to the epicardium of the anterior free wall of the left ventricle in an open-chested canine. Bi-plane cine-angiography was used to track the position of the markers before and during induced ischaemia. The strains during the cardiac cycle were calculated using marker triplets. Large strain gradients were observed across the infarcted reg ons.

The work done Huyghe *et al.* [37] and Bovendeerd *et al.* [38] [43] [48] on modelling the myocardium is extremely impressive though the two models each has its own strengths and weaknesses. The linear and exponential relationships between certain myocardial properties suggested by Han *et al.* [41] [45] cannot be given much weight since these curves were only plotted though three or four points. Any data obtained from invasive methods such as those used by Fann *et al.* [40], Hann *et al.* [44], Hashima *et al.* [50] must be treated with caution since it would be naïve to assume that the function of the heart remained unaffected during these procedures, indeed Fann *et al.* [40] expressed these very concerns. Geometry is being seen by researchers as an important factor in the function of the heart and great lengths have been gone by Pirolo *et al.* [51] and Pao *et al.* [46] to obtain a more realistic geometry.

In the last few years heart modelling seems to have polarised into three basic areas. The first of these is myocardial modelling, where sophisticated models of the myocardium are used to describe cardiac muscle, such as those used by Huyghe *et al.* [37] and Bovendeerd *et al.* [48]. These models have the advantage that since they have active components they can be used in systolic analysis. However, neither of the above two models can predict complete left ventricular properties and are based around a simplified geometry. Secondly, authors have tried to improve the geometry used to model the left ventricle, using either dissection or modern imaging techniques. The third group involves invasive methods to obtain strain information directly from the ventricle wall. This is far from ideal as the myocardium is damaged due to the attachment of the markers to it. It is not known whether this has a great effect upon the global function of the myocardium, although local changes in myocardium function seem inevitable. Several authors including Fann *et al.* [40] have expressed concern over the damage caused by insertion of markers into the ventricle wall. Some of the work done is extremely impressive, the only questionable note is that the recent work has exclusively been done with canine hearts, and it is not clear if conclusions drawn from this are relevant to humans.

The finite element method is extremely powerful and can certainly describe many aspects of the left ventricle and the microstructure of the myocardium. The main constraint with this method is not as with thin and thick shell models the method itself but a lack of accurate data. The models used to describe complex myocardial structure are not being made full advantage of due to the use of extremely poor geometric approximations. It is only with the use of accurate geometric information that any advantage from these models can be properly gauged.

2.2 Related Areas of Research

The following sub-sections are not intended to be complete reviews. They are intended as general information to aid in the understanding of the problems associated with cardiac modelling. It is also intended to provide information that will aid in the understanding of advancements in this and other works.

2.2.1 Imaging and Digitisation

The finite element method, as has been seen above, allows the description of complex and unique geometry, such as that of the human heart. The restricting factor has now become the availability of data used to define this geometry. It is thus of great importance that research be concentrated in the area of imaging and digitisation. The following is a short review of some of the advances in less/non-invasive imaging and digitisation of cardiac geometry.

The problems with imaging techniques such as X-ray and ultrasound are that they tend to produce poor quality data. Implantation of markers into the myocardium as used by Hashima *et al.* (1994 [50]) is far too invasive and dangerous to be of general clinical use. The reconstruction method performed by Pirolo *et al.* (1995 [51]) which involved dissection of the post-mortem heart can be of research use only. What is needed is a non-invasive method that gives geometric data at least as good as implanted markers.

In 1988 a new method of marking the myocardium for motion tracking was proposed by Zerhouni *et al.* [52] called magnetic resonance imaging tagging (MRI tagging). This involves using radio frequency saturation to give "hyperintense stripes" on the MRI image. These lasted between 60-450 msec, which is long enough to image the systolic part of the cardiac cycle. This provided the first real hope that a non-invasive method to track local myocardial deformations would be a reality.

An enhancement to this method was reported by Axel and Dougherty [53] a year later. A new method of tagging was described which gave a two dimensional grid as well as sharper stripes. A method known as spatial modulation of magnetisation (SPAMM) was responsible for the improved clarity of the stripes. The 2-D grid effect now allows the tracking of points of the myocardium instead of stripes thus giving an extra dimension to the tracking process.

An investigation into the use of MRI tagging was carried out by Young and Axel [54] in 1992. One finding of this study was that the error associated with tracking tagged points was 0.3mm, which is very close to that of implanted markers. The technique of implanting markers has long been considered the "Gold Standard" for cardiac imaging. This figure for the error was later confirmed by O'Dell *et al.* [55] in 1994.

In 1995 McEachen *et al.* [56] pointed out some drawbacks to the method of MRI tagging. One was limited resolution due to pixel size and the other was the inability to monitor the entire cardiac cycle. Despite these drawbacks the method still provides greater detail than any other completely non-invasive method.

Meanwhile, the use of ultrasonography and computer tomography has been extended by the use of data extraction methods that make better use of the available data or of some *a priori* knowledge of the heart. There are two main drawbacks to the use of the above two methods; firstly, the data tends to be somewhat blurred which necessitates the use of a suitably qualified person to perform the digitisation process. Secondly, there can be blank areas on the image. Both these problems result in digitisation being slow and expensive and prone to observer error.

In 1994 Cho and Kim [57] demonstrated a method of edge detection for 2D echocardiograms using a modified Hough Transform. This method allowed automatic contour detection using a weighted function that could give greater weighting to areas of greater clarity/certainty. Because of this it is able to cope with "dropout" and "speckle noise" which are common problems affecting clarity of echocardiograms.

In the same year Chen *et al.* [58] used a knowledge based approach with an adaptive K-mean clustering algorithm to segment some computer tomography (CT) scan data. The algorithm was found to create a reconstruction with a volume error of 6.7% and a surface area error of 1.9%. The knowledge based part of the algorithm was needed to remove and spurious data produced by the imaging technique.

The next year Chen *et al.* (1995 [59]) suggested the use of *a priori* shape knowledge as a method to further improve the quality of reconstructions from CT scans. This method means the shape of the final reconstruction is considered during the reconstruction process. For this the use of 3D modelling primitives, "superquadratics", were used since these allow deformations that simpler primitives cannot provide. To achieve this the segmentation process considers optimisation in both image and parameter spaces as well as the deformation progression of the reconstruction at different time intervals. This method thus makes use of known characteristics of the left ventricle as well as the changes in spatial configuration with time.

A similar method was proposed by Tu *et al.* [60] also in 1995. This method also made use of spatio-temporal data in order to gain maximum available information from the data. Two sets of 4D canine data were extracted using this method and compared with those obtained manually. The method had problems in extracting part of the ventricle and this resulted in errors of between 5% and 32% between computed and manually obtained reconstructions. When however this problem area was removed the errors were greatly reduced to between 1% and 5%. The method does show some promise though it is not totally automated as it requires the use of some manual editing of the raw data.

If these automated methods for data extraction can perform as they seem to promise, then this could result in the improvement in data acquisition that the finite element models require. Failing that MRI data may improve and if combined with spatio-temporal smoothing may reach the accuracy necessary for finite element models.

2.2.2 Analysis and Modelling of the Myocardium

Another area of continued interest is the determination of the material properties of the myocardium through the mechanical testing of excised strips of myocardium. These strips are subjected to a variety of loads and the stresses and associated strains recorded. From this data the stress/strain relationship can then be derived. Originally these tests were uni-axial but more recently they have been biaxial. Biaxial tests are now the norm as it was discovered that uni-axial tests significantly underestimated the elastic stiffness of the myocardium.

The normal testing procedure is thoroughly documented by Lundin (1944 [61]) who performed an extensive array of tests upon frog myocardium. The tests were performed both on passive and contracting muscle. One property discovered was that the stiffness of contracting muscle was approximately eight fold that of passive muscle. It was suggested that this was due to the increased tension in the muscle and was not due to another intrinsic property of the myocardium. This publication provides a great deal of data as well as a good introduction to this field.

In 1983 Demer and Yin [62] performed extensive biaxial tests on human myocardial material. They used material from a number of different sites within the left ventricular wall. It was discovered that the ratio of long fibre to cross fibre elastic modulus can vary by as much as 2:1 to 1:2, traditionally most researchers have taken a ratio of 2:1 in their models. It might seem plausible in the light of this to assume isotropic material properties in models, this however was not advised as statistical analysis of the test results suggested that the myocardium was indeed anisotropic.

Four years later in 1987 Yin *et al.* [63] performed further biaxial loading tests on canine myocardium. In these tests thin sheets of myocardium were loaded *such that the ratio of* strains in the fibre and cross-fibre directions remained constant. The use of a new exponential strain energy function using non-integral powers of the strains was proposed to model the myocardium. The new function was found to be more self consistent than the more commonly used exponential strain energy function with quadratic powers of the strains. The tests also gave a large spread of values of the ratio of long and cross-fibre elastic modulus. Values of the elastic modulus ratio were found to vary from 6.5:1 to 1:4. It should be noted that there is a great deal of non-repeatability in calculating coefficients for strain energy functions from new data. The reason for this is thought to be due to either "numerical instability" of the algorithms used, or due to the "strain-history" of the myocardium.

These tests are not merely limited to testing myocardial material but also involve fitting functional descriptions to the properties observed. In 1987 Pinto [64], from work with pig papillary muscle, suggested an equation that would give the force generated by a contracting muscle fibre. The form of the curve produced is similar to the X^2 statistical distribution. It has a steep rise up to the time when maximum force is achieved and is then followed by an exponential decay.

Attempts have also been made to devise a functional descriptor of the passive properties of myocardium. In 1990 Humphrey *et al.* [65] proposed a "pseudostrain-energy function" to describe the passive myocardial properties of canine left ventricles. Slabs of myocardium were subjected to biaxial tests and results used to find the parameters of the “pseudostrain-energy function” which was of a polynomial form. The function was found to be useful for prediction purposes as well as being descriptive and was felt, due to its biaxial nature and the larger range of loading conditions for which it was applicable, that it was far more useful to the intact heart studies than previously used functions. Great faith was placed in this function, although caution was advised in its use, as well as further research recommended. The main area of concern was the damage caused to the myocardium, by excising the slabs, and the effect this has upon the properties of the material. It should also be pointed out that the function is only applicable when passive mechanical forces are far greater than the active contraction forces.

In 1994 Novak *et al.* [66] studied the human myocardium at the equatorial region of the left ventricle. It was discovered that the myocardium at the epicardium and endocardium was significantly stiffer than at the mid-wall level. This has a significant impact upon those authors who study stress distribution as even small differences in material properties through the wall are likely to greatly affect the stress distribution.

There are two main problems that arise from this type of work over and above that of preserving functionality of the myocardium during the testing procedure. That is that most of these tests are carried out on non-human tissue and as a result may not be directly applicable to humans. The requirements of the myocardium are quite likely to vary wildly between animal species since different animals will require their heart to function within different parameters. The second problem is that it may not be directly applicable to finite element models of the heart. The reason for this is due to the nature of the myocardium which is known to be quite heterogeneous as regards its material properties. It would be quite unreasonable to expect that just because the individual fibres follow an exponential stress strain relationship that the myocardium as a whole would behave this way also.

What is really needed is a formula for the bulk properties of the myocardium. The repeated attempts to prove that an exponential stress strain relationship exists for the bulk myocardial properties have been less than convincing.

The real benefit that comes from this type of work is an understanding of how different material properties may relate to each other. That is to say how along-fibre, cross-fibre and shear elastic moduli are interrelated. Once a fuller understanding of how these different material properties relate to each other is achieved a great deal of the arbitrariness in the choice of these for modelling purposes can be removed.

2.3 Summary

It would appear that there are two major drawbacks with much of the current research. There is an excessive over reliance upon simplified geometry to represent the left ventricle. This is despite the importance of shape and change in shape being recognised as major factors in the performance of the left ventricle (Mirsky 1969 [8]).

The other cause for concern is in the rise of more complex models for myocardial material properties. These have largely been produced by consideration of the results of tests on excised myocardial tissue. Due to the heterogeneity of the myocardium it does not necessarily follow that the behaviour of the whole intact myocardium is mirrored in these localised tests. Models performed using exponential stress/strain relationships for the myocardium are imposing somewhat artificial constraints upon the myocardial performance.

Chapter 3

3. Previous Work Relevant to the Current Project

This Chapter provides a brief summary of the work, previously carried out by Vinson [1] and Grewal [2], upon which this work is based. The work in question is that pertaining to the three-dimensional reconstruction and the method used to calculate the material properties as well as some general information about the original data. In this chapter the material properties previously found by Grewal (1988 [2]) will be presented.

3.1 Data Acquisition

The data used in the modelling process was obtained by X-ray cine-cardiography by the Royal Brompton Hospital, London. Two orthogonal views 60° left anterior oblique (LAO) and 30° right anterior oblique (RAO) were taken at a rate of 50/second. The two frames are slightly out of phase by 10ms due to the necessary staggering of the X-ray exposures. The inaccuracy introduced by this is thought to be negligible compared to the assumptions made in definition and interpretation of the images. The images were taken at mid-inspiration with cranio-caudal tilt, this was found to give the most vivid images with the easiest reconstruction.

In order to obtain the X-rays a contrast medium must be introduced into the ventricle. This is achieved by the use of a catheter inserted into the left ventricle, via the pulmonary vein. The tip of the catheter houses a micromanometer which is used to measure the pressure within the ventricle, atmospheric pressure is taken as the zero. It should be noted that the contrast medium only allows the endocardium to be visible, the epicardium in most cases is only partially visible. The X-ray frames were hand digitised using a Summagraphics digitiser connected to Brompton Hospital's Prime 300 computer system. This process involves the projection onto a screen of each individual X-ray frame which must then be hand digitised using a pen follower. To make the digitisation process easier the screen has a grid with 0.1mm divisions. By repeating the digitisation process several times it was found that the reconstructed volume was consistent.

There is an important point that should be made about the above digitisation process. The process was only shown to give a reproducible volume when the reconstruction was used. It was not however shown that any other property of the reconstruction was adequately catered for in the digitisation process. This will be of importance later on when consideration of overall shape will be made.

It must be also be noted that the digitisation process is extremely difficult and liable to inaccuracy for two main of reasons. Firstly, the contrast medium is not evenly distributed within the ventricular cavity. This leads to areas with little or no contrast medium, which results in X-rays that have areas where the endocardium is not visible. Secondly, the endocardium is not smooth as it is covered by the trabeculæ carnæ. This means that instead of a distinctive change from cavity to myocardium there is in fact a blurred area of uncertainty. It is for these reasons that digitisation requires the use of a trained cardiologist.

3.2 Extraction of the Finite Element Mesh

A method for reconstructing the geometry of the left ventricle from bi-plane cine-angiography was proposed by Yettram *et al.* [5]. A selected right anterior oblique (RAO) and left anterior oblique (LAO) endocardium image of the ventricle are chosen. They are aligned at the apical point, which is defined as the lowest point on the on each image. Definition in the RAO image was always found to be more complete than the LAO, especially at the aorta, thus it is often necessary to extend the LAO image so that it matches the highest point on the better defined RAO image. This is done by projecting vertically the end points of the LAO image.

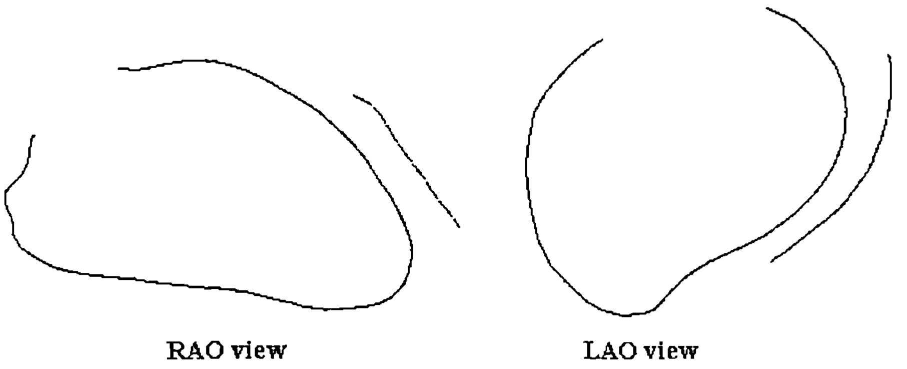


Fig 3.1 This shows the typical data produced by the digitisation process from which the left ventricular reconstruction is made. Typically a patient data file may consist of between 30 - 70 such images plus the cavity pressure.

The RAO image is now sliced at fifty levels at 90° to the mid-aortic apical line. This gives one hundred points which define the RAO image. These points are now projected onto the LAO view and the two corresponding points, for each RAO point, are found. This gives two hundred points which form an envelope within which the ventricular geometry must lie. Each set of four points, corresponding to the same level on the RAO view, form a quadrilateral.

The epicardium is treated in a similar way except that since this information is not complete the non-digitised epicardium must first be constructed from the limited information available from the X-ray. This is achieved by projecting parallel to the endocardium the first and last points of the digitised epicardium. Thus the first point is projected backwards and the last point is projected forwards. It is known that the apical point of the myocardium is only 2mm thick. A quadratic curve is fitted around the apex of epicardium from 5mm above the apex of the endocardium to 2mm below this point.

The four profiles defining the two three-dimensional envelopes for the endocardium and epicardium are now rotated so that the mid-aortic apical line is vertical. The structure can now be sliced horizontally, at fifty levels for volume calculation or twelve levels for finite element mesh generation so that both envelopes are each defined by a set of horizontally stacked quadrilateral discs. Every point of the endocardium and epicardium must lie within their respective envelopes. How to construct the actual ventricle geometry within these envelopes is to some extent arbitrary. The method used is to construct a circular arc in each corner of all of the envelope discs so that the arc touches both sides of the quadrilateral which make the corner. This arc can be subdivided to provide the points needed to generate the finite element mesh or to calculate the volume of the ventricle.

It would seem wise to discuss the merits and drawbacks of the above method of reconstruction. With good data the above method has been shown to give realistic left ventricular geometry (Vinson 1977 [1]). The problem occurs when the data is not quite so suitable. The best possible reconstruction would be possible when the RAO and LAO views are both aligned along the apex to mid-base line. This results in two images which can truly be said to be longitudinal. In practice although the views are orthogonal they will not lie along the apex to mid-base line. As the angle between the apex to mid-base line for the two X-ray frames increases the LAO image becomes less of a longitudinal image and more of an end on image. The result is not only a poor reconstruction but also results in some finite elements being very much smaller than the others. The result of having these smaller elements is that the mesh is not evenly distributed throughout the wall volume of the model and thus full use of the elements is not made.

This problem of poor element formation could possibly be overcome by implementing a more specialised meshing scheme. The advantages of this however would probably be lost due to the poor reconstruction potential of these frames of data.

3.3 The Finite Element Model

The finite element method is widely used in many areas of engineering and in recent years has become the main method for modelling the left ventricle. It is a numerical method employing the use of so called shape functions to describe the solution space. The solution space in this case is the wall of the ventricle. These shape functions provide a functional description of the geometry of the left ventricle. The solution of problems containing complex geometry is made possible by reducing it to a number of simpler components, finite elements. A good description of the finite element method as used is given by Vinson [1] and for a more advanced guide Zienkiewicz [31] is an excellent text. The finite element model is generated from the defined geometry such that it has 72 elements with a total of 401 nodes. There are 60 isoparametric bricks with 20 nodes and twelve wedge shaped elements with 15 nodes, which form the apex. The elements are arranged to form two layers through the wall, with the fibre angle varying linearly through each element. Pao *et al.* [67] suggested a three layer model although Grewal [2] found that a three layer model did not provide significantly greater accuracy to warrant such a increase in computation time. The model assumes the myocardium to be homogeneous, except for the apex which is particularly stiff, though not isotropic. It is assumed that the muscle fibre angle varies linearly between -60° on the outer surface to 60° on the inner surface.

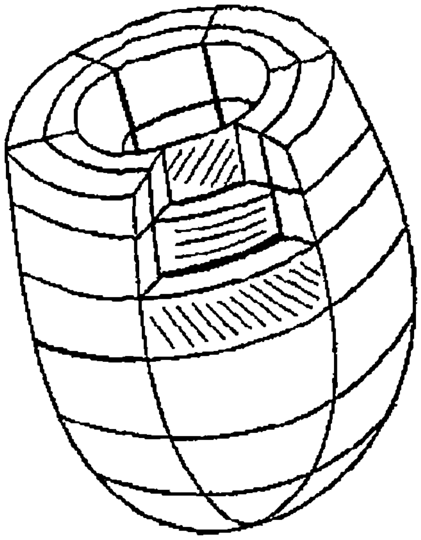


Fig 3.2 This shows an idealised left ventricular F.E. mesh and clearly shows, by means of a cut away section, how the fibre angle varies through the wall of the model. (From Grewal [2])

3.4 Volume Matching and the Elastic Modulus

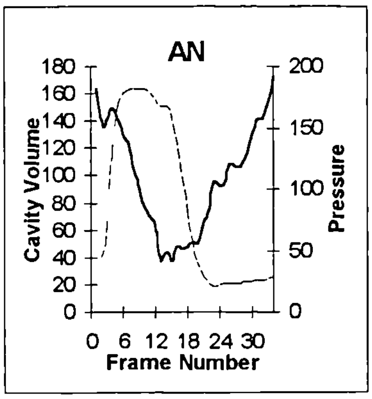
The finite element model described above is used to find the elastic modulus of the myocardium during the diastolic phase. The diastolic phase is when the left ventricle relaxes and the pressure and the volume within it are both rising. It is assumed that in this part of the cardiac cycle the myocardium is completely passive. Thus it is only the pressure increasing within the left ventricle that causes the left ventricle to deform. This is not entirely true although it is widely believed that the internal pressure is the major driving force, especially in the latter part of diastole. The process employed to find the material properties is to guess the solution and this guess is then iteratively refined until the solution is found to the required tolerance. In this case the v lume calculated at one time-step is matched to the volume calculated at the next time-step. This is achieved by subjecting the finite element mesh, representing the left ventricle at one time-step (frame), to the internal pressure increase that occurs between that interval and the next. A guess is made for the value of the elastic modulus and finite element analysis is performed upon the system. The finite element mesh is distorted by the increase in internal pressure and will now have a new volume. If this new volume is larger than that at the next time-step then the ventricle needs to be stiffer and the elastic modulus is increased. On the other hand if the new volume is less than that required then the ventricle is too stiff and the elastic modulus is reduced. This iterative process is then repeated until the distorted mesh volume matches that at the next time-step. When this process has been applied to all the diastolic data the progression of the elastic modulus can then be analysed.

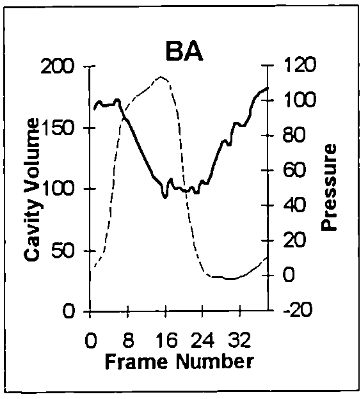
A process known as "smoothed volume matching" was developed by Grewal [2] and this is now described. The late diastolic phase of the cardiac cycle is chosen. This is when both the pressure and volume of the left ventricle are increasing. The volume pressure plot for these frames is then smoothed through using cubic splines. The weighting of the interpolating splines is two orders of magnitude greater for the first volume. This ensures that the smoothed spline curve passes through the first point. This is important as this volume and associated mesh are the starting point for the volume matching process.

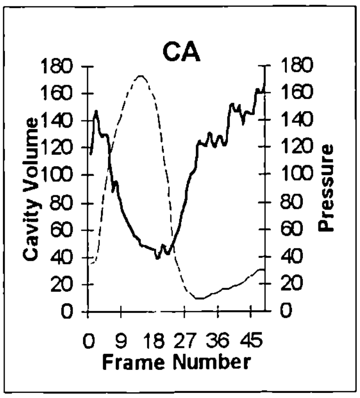
The finite element mesh is constructed for this first frame of late diastole. The finite element part of the program is then used to find the displacement of the mesh nodes that would occur when subjected to the pressure increase between this and the next frame. For this run the minor modulus is assigned an arbitrary value. The displacements are then scaled in such a way that when added to the nodal coordinates the new mesh volume is the smoothed volume of the next frame. The elastic modulus which was chosen is then divided by the same scaling factor as the displacements where mu tiplied by and the finite element part is initiated again and the matching process is repeated. At this stage, since the displacements are small and the system is approximately linear, the modulus that gives a volume match has been found. However some frames have larger displacements and may require more applications of the finite element part to obtain a match. At this stage the displacements are added to the mesh nodes and this becomes our new finite element mesh. The above process is repeated until the moduli for all the diastolic frames have been calculated.

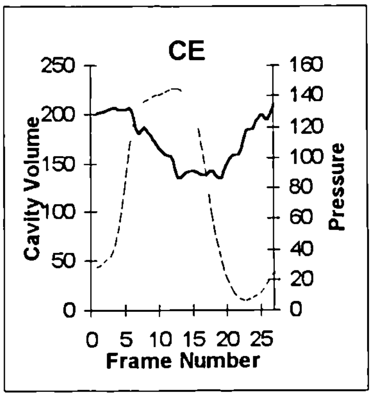
3.5 The Patient Data

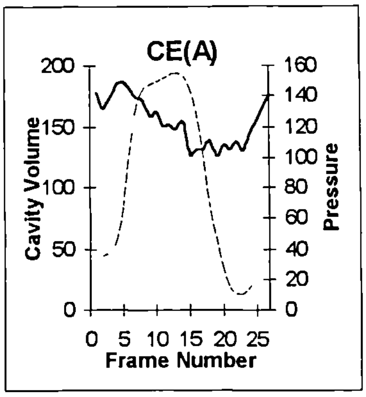
There were twenty one sets of patient data available for analysis. Sadly, it had only been possible to use the finite element method on twelve sets of data. A suitable finite element mesh could not be formed for the remaining nine sets of patient data. Below are the volume/pressure plots for the twenty one patients.

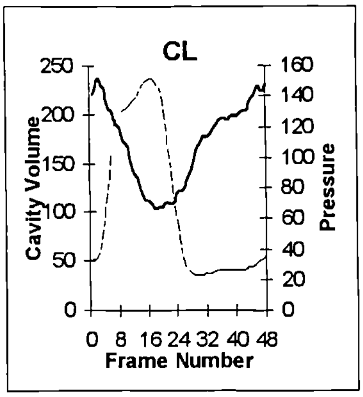


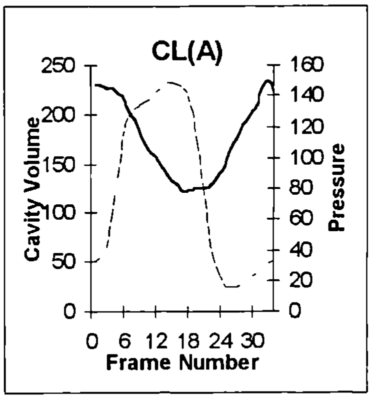


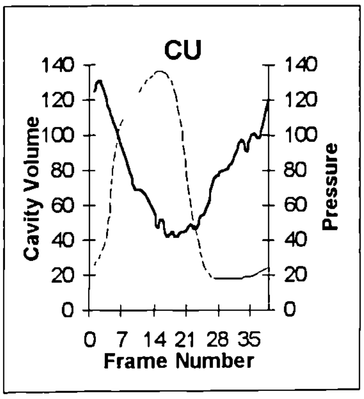


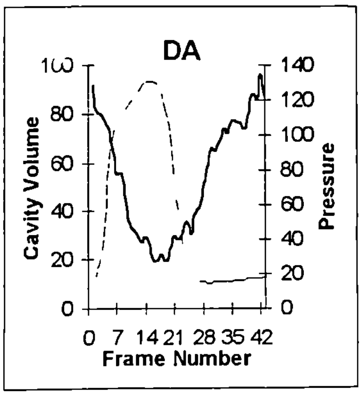












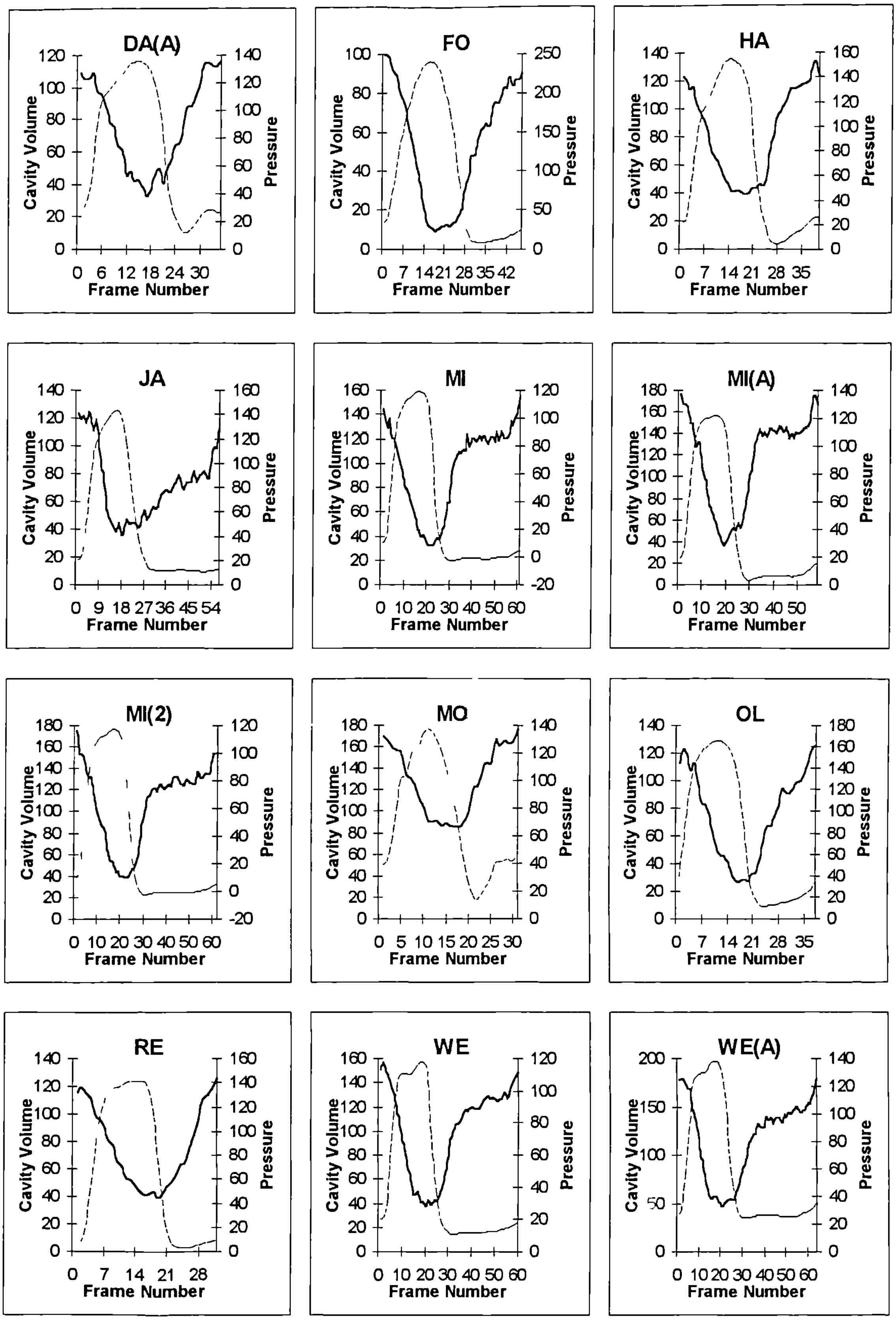
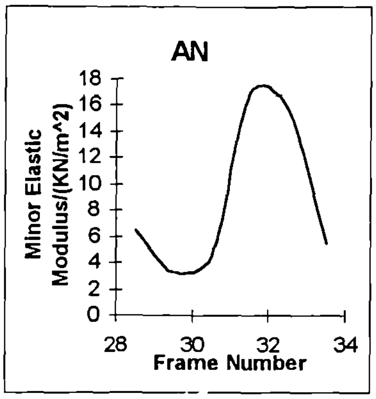
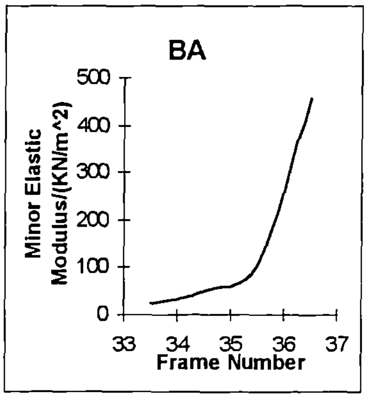
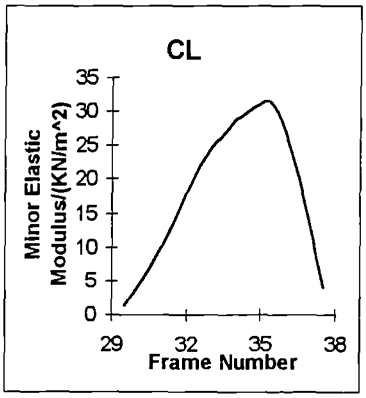


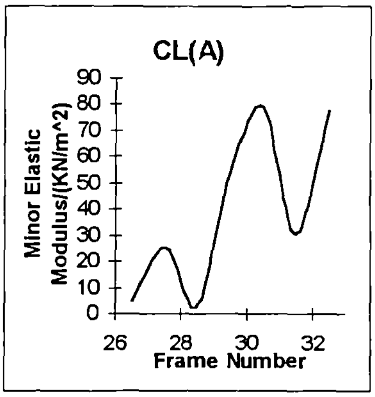
Fig 3.3 These graphs show the cavity volume (cm^3) and cavity pressure (mmHg) against frame number for each of the twenty one sets of patient data. The volume is given by the solid line and the pressure by the broken line.

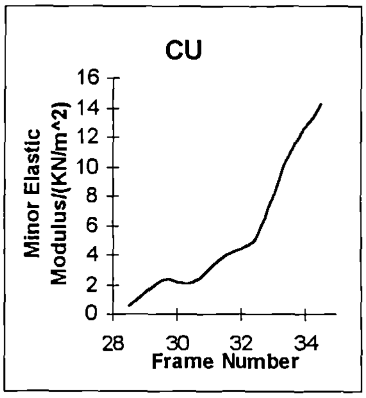
The period of diastole can clearly be identified towards the right hand side of the above graphs. The reason for the use of smoothing through the data is clearly evident from the volume data which tends to be far from smooth. The area of diastole for each patient was identified from each graph and the process of "smoothed volume matching" was applied to this region. The plots of the minor elastic modulus against frame number are given below.

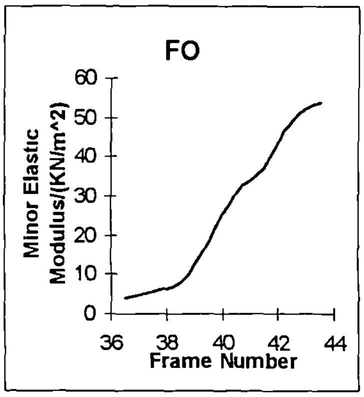


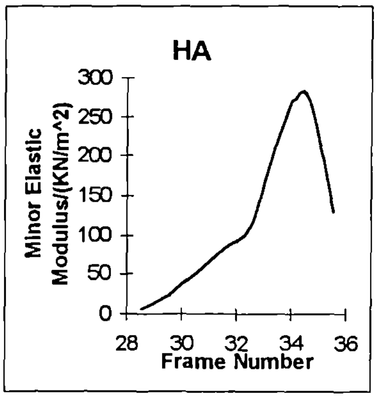


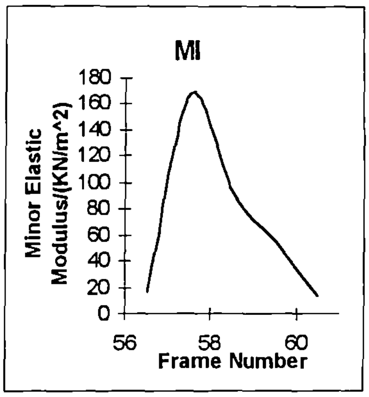


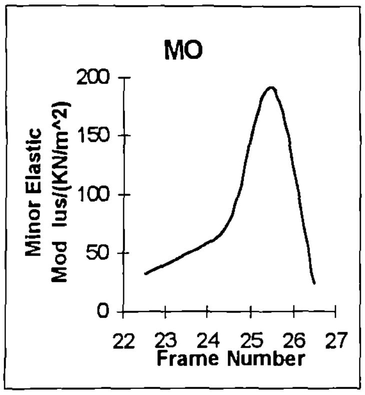












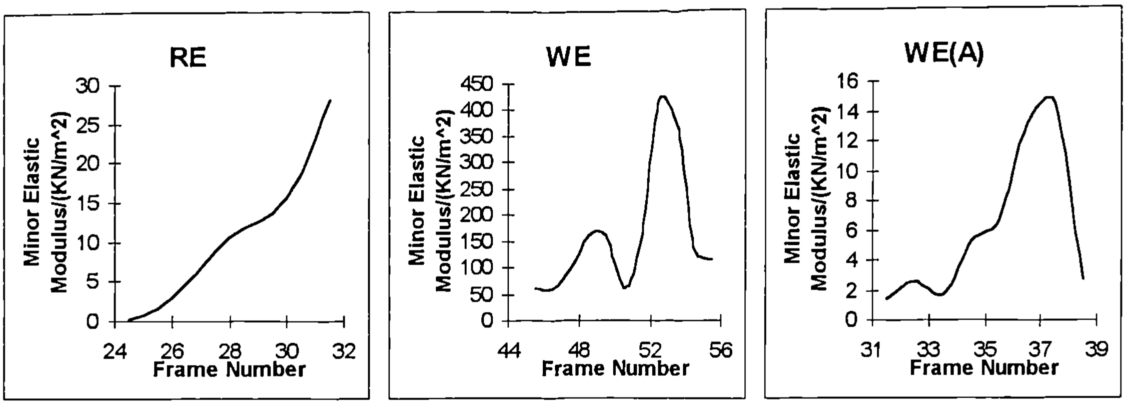


Fig 3.4 These graphs show the calculated minor elastic moduli during diastole for twelve patients. These are the values obtained by Grewal [2].

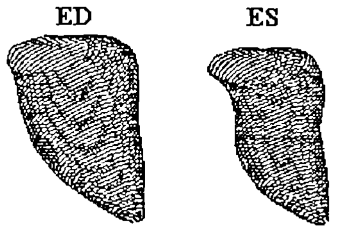
Since the method used to gain the patient data is invasive there is some risk to the health of the patient when this data is obtained. This means that the procedure can only be performed when there is a clear medical reason. As a result all the patient data available is for patients who suffered from a variety of medical conditions. A table of the patients and their medical conditions is given below.

Patient	Medical Condition
AN	Mild mitral stenosis with regurgitation
BA	Coronary arterary disease with LV dysfunction
CA	Coronary arterary disease
CE	Coronary arterary disease
CE(A)	Coronary arterary disease with induced angina
CL	Coronary arterary disease
CL(A)	Coronary arterary disease with induced angina
CU	Coronary arterary disease
DA	Coronary arterary disease
DA(A)	Coronary arterary disease with induced angina
FO	Aortic valve dysfunction with LV hypertrophy
HA	Coronary arterary disease
JA	Coronary arterary disease with LV aneuurysm
MI	Coronary arterary disease
MI(A)	Coronary arterary disease with induced angina
MI(2)	Coronary arterary disease
MO	Congestive cardiomyopathy
OL	Coronary arterary disease
RE	Coronary arterary disease
WE	Coronary arterary disease
WE(A)	Coronary arterary disease with induced angina

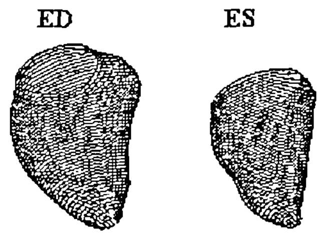
Table 3.1 This table is a complete list of patients, for whom data was available, and the medical conditions from which they suffered.

The diagrams below show the endocardial surface of the three-dimensional reconstruction. The end-diastolic (ED) and end-systolic (ES) frames are shown. These represent the time when the left ventricular cavity volume is at its largest and smallest. The diagrams are all to the same scale and clearly show the variation that occurs in size and shape between patients and during the cardiac cycle.

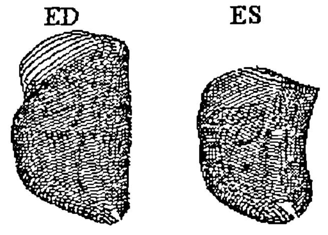
AN



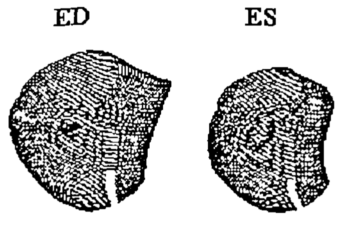
BA



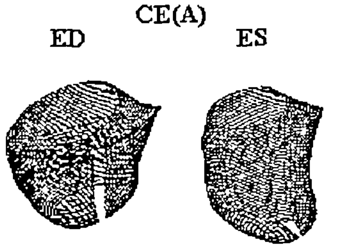
CA



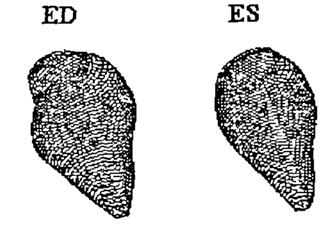
CE



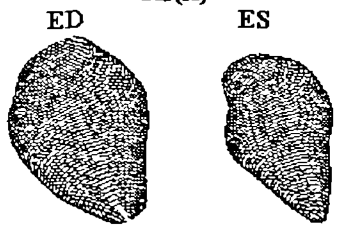
CE(A)



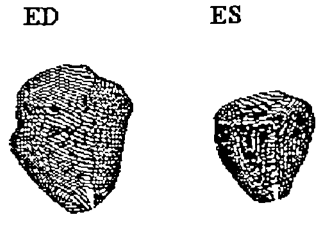
CL



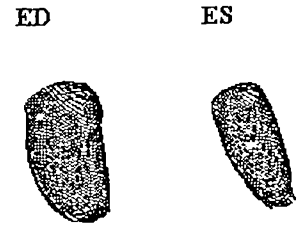
CL(A)



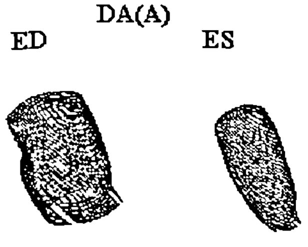
CU



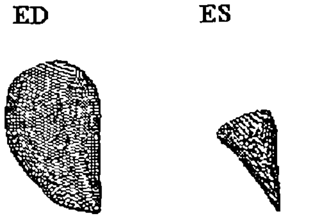
DA



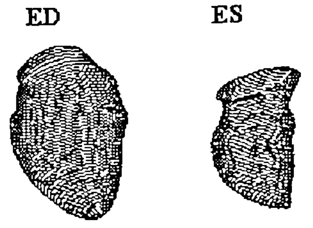
DA(A)



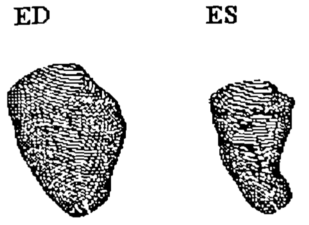
FO



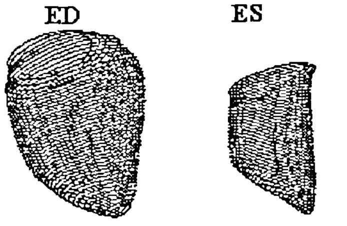
HA



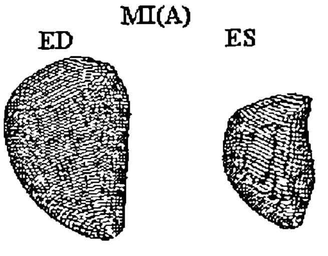
JA



MI



MI(A)



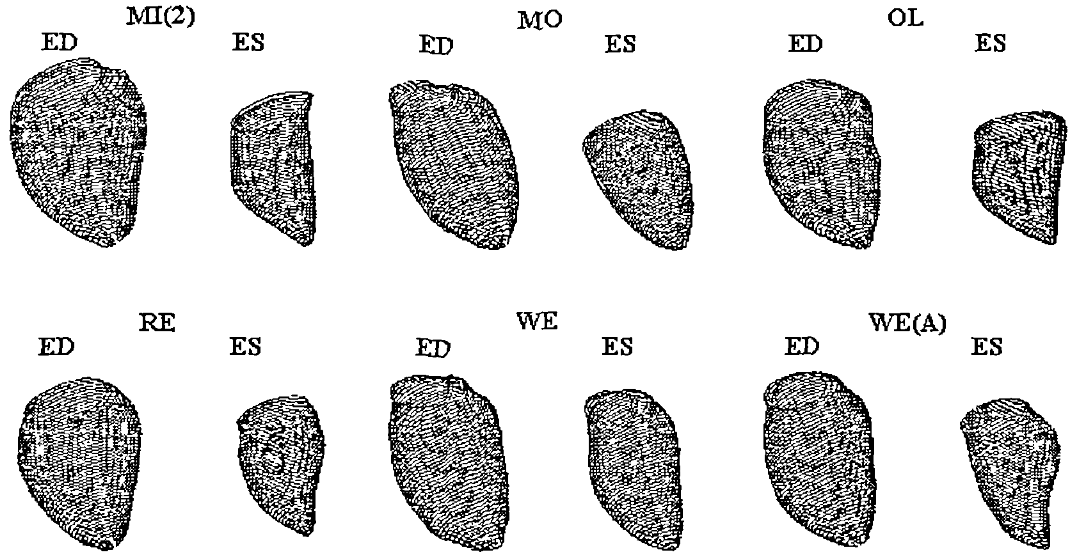


Fig 3.5 These diagrams show the end-diastolic (ED) and end-systolic (ES) endocardial surface for each of the patients.

Chapter 4

4. Programming Developments

The first stage of this work was to resurrect the computer programs which had previously been developed. There were two programs, the finite element program which had been developed to run upon the Cray-1S/1000 at the University of London and the pre-processor and post-processor which ran upon Brunel University's Honeywell Multics. The idea was to combine the two programs into one which could then be run upon a Sun Workstation; both programs were written in Fortran 77. Once they had been combined the process of volume matching, described in detail in Chapter 3, could be automated to speed up the analysis.

4.1 The Finite Element Part

The finite element program was fairly standard and only required a few minor changes. The main feature that needed to be altered was to align double precision variables within the common blocks. This had not been a problem on the Cray since the Cray uses software double precision whereas the Sun uses hardware double precision. Double precision is achieved by allowing two consecutive single precision words to be used to define one double precision word. The two single precision words used for this are important as not using the preallocated words can cause serious computational inefficiency. This is because before computation each incorrectly aligned variable must be aligned properly. This was very easily remedied by reordering the common blocks so that the double precision variables appear first.

There were also several arrays that had been incorrectly defined, for example two integer arrays had been defined as double precision and one two dimensional array was inconsistent in one of its dimensions. Some alterations were also made to the way output from the program is treated. The output was originally to standard output, (this is a file on the Cray but to the screen on the Sun system) this was changed and output is now to two files. One file contains the general information and the stress and strain values while the other contains the computed displacements of the nodes of the finite element mesh. The treatment of characters was also changed, originally they were treated as packed integer arrays. Instead of storing a character string as a character variable, two characters of the string are stored in each position of an integer array. However this can cause problems when it comes to writing them to a file. They are written as the numerical values which represent the characters instead of their character values, so they are now treated as character variables throughout the program.

4.2 The Post-processor and Pre-processor

The first step in modifying the pre-processor and post-processor was to arrange for it solely to use the graphical package SimplePlot. Previously the packages Gino and SimplePlot had both been used to provide graphical output. However Gino was no longer available on the computer network. The first thing that was necessary was to discover which of the routines belonged to the Gino package. All the "calls" to routines were then characterised as one of the following: internal subroutine, NAG library routine, SimplePlot routine or Gino routine. Once all the internal, NAG and SimplePlot routines were identified the ones remaining must be Gino routines.

Now that the routines to be replaced have been identified it is important to discover their purpose so as to choose a suitable replacement. This had to be done by examining their arguments and the context in which they were used, since a manual was not available for the package Gino. SimplePlot however, since it was still a package supported upon the network, had a full complement of user and reference manuals ([68] [69]). Below is a list of Gino routines, in bold type, their function and the SimplePlot routines that were used to replace them:

UNITS(UMILS) Sets the measurement units to UMILS for plotting purposes. It is removed and not replaced.

DEVPAP(XPAP,YPAP,ITYPE) Sets the paper size in the current units to width XPAP and length YPAP and output to device ITYPE. The paper size is specified by routine PAGE(XCMS,YCMS) and the output device is specified by routine DEVNO(ITYPE).

LINTO2(X,Y) Draws a straight line from the current position to point (X,Y). It is replaced by BRKNT(X,Y,LTYPE), the value of LTYPE here is unimportant although it must be in the range -6 to 6.

MOVETO2(X,Y) This moves the cursor to the point (X,Y). It is replaced by the two routines BREAK and BRKNT(X,Y,LTYPE), the value of LTYPE specifies the type of line and as before must be in the range -6 to 6.

LINBY2(DX,DY) Draws a straight line from the current position to a point (DX,DY) relative to the starting point. It is replaced by the routine BRKNT(X+DX,Y+DY,LTYPE) where the point (X,Y) is the current position and LTYPE is as above.

DASHED(MODE,REPEAT,DASH,DOT) This routine sets the type of broken line to be used for drawing with. In SimplePlot this is set at the time of drawing using the parameter LTYPE.

DEVEND This routine terminates plotting in Gino. It is replaced with the SimplePlot equivalent, ENDPLOT.

CHAHAR(NSIZE,NHV) This routine sets the character size, NSIZE, and the orientation, NHV=0 gives horizontal and NHV=1 gives vertical. It is replaced with the

SimplePlot routine TEXTMG, although the orientation must remain horizontal. The Y-axis however is automatically rotated vertically.

CHASIZ(WIDTH,HEIGHT) This routine sets the width and height of the subsequent text. As above it is replaced with TEXTMG.

CHAANG(ANGLE) This routine sets the angle to the horizontal at which all subsequent text will be written. There is no SimplePlot equivalent.

CHAARR(ICHA,I,N) This routine writes a character string which has been stored as a packed integer array to the specified output device. The integer array has length I and density N. It is replaced with the SimplePlot routine CP7PT(X,Y,15,CHA) where the position on the plotting device is the point (X,Y) and the text is stored as a character array in CHA.

SCALE2(SX,SY) This routine is used to scale X and Y-axes by amounts SX and SY respectively. It is not directly replaced with any routine as setting up and scaling of the output area is easily accomplished with the SimplePlot routine SCALES.

APDS4, T4014, VH7221 These three routines are used to initialise the output device to be used for plotting purposes. The first two are for different terminal types while the third is for a pen plotter. In SimplePlot the output device can be specified by the routine DEVNO(DEV) where DEV is an integer representing a logical output device.

ENCODE(IARRAY,N) 'CHA' This is a function which is used to convert the character string "CHA" to a packed integer array IARRAY of length N. It is not needed under SimplePlot since SimplePlot is able to directly output chara ter strings as character strings and does not require conversion to a packed integer array.

As well as the changes to SimplePlot routines from Gino it was also necessary to change one of the NAG library routines. One of the previously used routines was removed from the NAG library of routines when the version changed from Mark 15 to Mark 16. The function of routine "M01AJF" was split between three new routines "M01DAF", "M01ZAF" and "M01CAF" ([70]). The routine "M01AJF" had previously been used for the purpose of ranking a vector such that it was ascending. The role of this part of the routine is now performed by "M01DAF" and this routine has now been used to replace the one previously used.

After the replacement of the above routines the program would now compile and run to a limited extent. At this point a number of problems came to light. As with the finite element program the common blocks required alignment and this was achieved by rearranging the lists of variables. The other more serious problems can be split into two basic types, programming and logical. Some of these may have been previously corrected since there were several copies of this program and it was not known which was the most recent version.

4.2.1 Programming Alterations

On the whole these were hardest to track down, since the effects they had on the function of the program were not necessarily directly related to the routines in which the errors occurred. For example a variable may take an inappropriate value but it would only be when that variable was required that any problem could be noticed. It was often the case that it was only the cumulative affects of an earlier error that caused a program failure. Most of these occurred as a direct result of the way in which Multics Fortran and SunPro Fortran handle certain situations. Thus situations that would be handled as the programmer intended under Multics Fortran caused catastrophic failure in SunPro Fortran. However it was also known that the program had problems processing some of the frames of geometric data for several of the patients, but since none of these occurred during diastole no attempt had previously been made to correct them. These problems now require attention since we will be interested in examining systole.

Many Fortran compilers add extensions to the ANSI (American National Standards Institute) standard, this is with the aim of making programming easier and programs more robust. Relying on these extensions however is often a false economy. This is especially so if it later becomes necessary to change to another version of Fortran. It is thus wise to avoid compiler specific features as they will change from compiler to compiler and sometimes between different versions of the same compiler. The problems introduced as a result of the use of non-standard features have now been remedied and are discussed below.

Arrays Too Small for Required Data: In some versions of Fortran, if this problem is encountered, the program may write extraneous data to memory after the last element in the array so that it can be assessed later. In SunPro Fortran when there is an attempt to write data to the n+1th position of an array of length n the data may be written to position 1. In some cases it was also seen to overwrite other variables corrupting the data stored in them. This was remedied by increasing the size of all the arrays used to store patient data from 101 elements to 151 elements. This type of problem is relatively easy to find as SunPro Fortran has a compiler option, "-c", which aborts execution of the program when there is an attempt to access an array element out of the defined array bounds. Checks were also added so that if, in future, this value is too small a warning message is given to the user and execution of the program will be suspended.

Inconsistent Declaration of Variables in Parameter Lists: Some versions of Fortran ensure a complete mapping of each variable in a call list o to each element in a subroutine's parameter list. SunPro Fortran does not do this, thus if a variable is incorrectly defined in one parameter list then the information after and including the mistake will be corrupted. There were several instances where there was a mismatch in variable declaration where variables and arrays changed from real to double precision or integer to real. To correct this problem was a considerably time consuming task, as every call to every subroutine must be checked to ensure that variables in a parameter list are declared as the same type in the subroutine and the routine from where it is called. If a variable type mismatch occurs then the most suitable type of variable must be chosen and all conflicting definitions changed to that.

Calculation Within a Parameter List: Under SunPro Fortran performing simple calculations within a call list leads to variables being incorrectly overwritten when control reverts back to the routine from which the call was made. The effect of this is that attempting to use this variable results in a large amount of spurious text being produced. This text is in fact the output from the "error variables". Multics and indeed most other major versions of Fortran do not exhibit this problem. These calculations must be carried out prior to the subroutine call, the result stored as a temporary variable and that then placed in the call parameter list in place of the calculation.

Unassigned Variables Assumed to Take a Value of Zero: In some versions of Fortran when entering a subroutine any unassigned variables are assigned a value of zero. Other versions of Fortran do not even initialise unassigned variables to zero upon program execution. For this reason the value of unassigned variables should never be relied upon as there is no guarantee what value they may have. In SunPro Fortran these variables retain the value they had when last exiting the routine and are only assigned a value of zero when the routine is entered for the first time. Thus all variables that should have value zero upon entering a subroutine are explicitly assigned the value zero.

One specific instance when the above caused a problem was in the formation of the finite element mesh. The mesh is produced from a vertically stacked series of reconstructed cross-sectional slices. Each slice itself being formed from four circular arcs. When the routine which constructs these is called for the first time the mesh produced was correct. When however another frame was reconstructed first the mesh would be different. The problem was that the array index that selected which cross-sectional slice was being produced was not set to zero upon entering the routine. The result of this was that the first or top cross-sectional reconstruction was actually that of the first reconstruction done during that run of the program.

4.2.2 Logical Alterations

Though there are quite a few changes within this category they only affected a small minority of the frames within diastole. The effect upon the shape and volume of the reconstruction was also often quite minor. Thus it is not envisaged that previous results were greatly affected qualitatively or quantitatively.

When data was read from the patient's data file the possibility of an unequal number of right anterior oblique (RAO) and left anterior oblique (LAO) frames was not catered for. This would mean that about half of the patient data files could not be directly analysed without some hand editing of the file. To add this facility only required the addition of a few lines of code and was quickly corrected.

There existed a subroutine that aligned the RAO and LAO frames. It had been assumed that the LAO frame always had its apex at a height zero. Another assumption was that the apex was also at the point furthest from the midpoint between the first and last point on the frame image. Neither of these two is universally true, the second however is always true for the LAO image. When the apex on the RAO image is not the furthest point from the midpoint between the first and last point on the frame image the finite element mesh cannot be formed. The solution is to translate vertically the LAO image so that its apex is at a height of zero and to translate the RAO image so that the lowest point of that is also at a height of zero.

When attempting to construct the ventricle envelope it may not be possible to obtain the two distinct values needed to form one of the sides of the envelope box. Thus two of the points forming the corners of the box are the same. The method used to reconstruct the ventricle however requires a quadrilateral and cannot cope with a triangle. The solution to this problem is, after alignment to the RAO and LAO images, to move any RAO points that have height zero to a height of 0.1mm, such a small difference does not affect the overall volume, but it is enough to ensure that a quadrilateral can always be formed. This problem is most acute when the RAO has a number of points at zero height and is made worse when some of these points are not close to the apex.

The subroutine "sketch" had a number of problems associated with it although few of these had a major effect upon the diastolic frames. This routine reconstructs the epicardium outline from the usually very limited information gained from X-ray cineangiograms. However for one patient this information was complete but the routine did not allow for the first point on the epicardium to be higher than the first point on the endocardium. The result was that the entire digitised epicardium would be lost as the reconstruction process resulted in no epicardial points being found. This meant that no volume or geometric data could not be produced for these frames. Fortunately none of these were in the diastolic phase. The solution was to add in a check within the reconstruction algorithm so that if the epicardium was already complete the reconstruction would not be performed.

The routine also could not cope effectively with concave sections of endocardium wall, and also if the image became horizontal at the end points. Two routines had to be written to correct these flaws in the epicardium reconstruction. Routine "TIDYUP" ensures that the first and last points on the pericardium are o tside the first and last points on the endocardium, since if the aorta needs to be drawn on, and the first or last point of the pericardium are inside the first or last point of the endocardium, this area would have a negative thickness. This is achieved by searching along the pericardium, from the first point, until a point where the horizontal distance between this point and the first point of the endocardium is greater than the thickness of the myocardium at this point. The x value (horizontal position) of the point previous to this is given the value of the x position of the first point of the endocardium plus the thickness of the myocardium at this point. The position of this newly formed point in the y (vertical) plane is calculated using linear interpolation between the two points either side of the new point being formed. The arrays holding this coordinate data are then reordered so that the point that has just been formed is the first point held in the arrays. The process for the end of the epicardium is the same except that the points are traversed in the opposite direction.

When the endocardium had a concave part a loop could then be formed when the pericardium was reconstructed. A routine "NOLOOP" was written which removes these loops and also removes rogue points which were introduced when digitising occurred. The rogue points are removed first. These are easily identified visually as a sudden spike in the outline. The test used in the program for a rogue point is as follows. If the distance between point n and point n+1 is more than ten times greater than the distance between point n and point n+2 then point n+1 is removed. The removal of loops is somewhat more complex. To aid this, the subroutine "INSECT" was constructed. This routine, when supplied with two sets of two points, returns the value "TRUE" if the two line sections formed from joining the two points in each set intersect. The algorithm used constructs the line equations for each of the two lines then, providing the lines are not parallel, the point of intersection is found. This point is then checked to see if it lies in the domain of the two line sections and if it does then the val e "TRUE" is returned otherwise the value "FALSE" is returned. The routine "NOLOOP" calls routine "INSECT" for the first two points and then for each subsequent pair of points after that. The process is repeated in turn for every pair of starting points with the exception of the last pair, which will already have been checked with every other pair of points. Should routine "INSECT" return with the value "TRUE" then all the points in between the two pairs are removed including the second point of the first pair and the first point of the second pair. The points are removed by reordering the arrays in which they are stored, so that the unwanted values are overwritten by the subsequent data. Thus the length of the array is reduced by the number of points being removed. This routine is also used on the endocardium data to remove any rogue points found there.

There was also a subroutine "CHECKPT" which was supposed to remove duplicate points introduced during the digitising process. The method used was to perturbate the duplicate point. This however could cause problems since the perturbation was always in the same direction. The result of this is that the outline can double-back upon itself and this can cause the algorithm used to extract the finite element mesh to fail. These duplicate points are now removed completely by reordering the arrays so that each duplicate point is overwritten by the next non-duplicate value. The overall length of the array is thus reduced by one for each duplicate point being removed.

4.3 Combining the Computer Programs

The rationale in combining the two programs was that the process of obtaining the elastic moduli for the myocardium could be automated. Combining the two programs was by no means trivial as there were in excess of ten thousand lines of code, only some of which were documented and very little of which was commented. Previously the post-processor and pre-processor program had been used to generate the data file for the finite element program and then process the results for volume matching. The scaling factor for the displacements which gave a volume match was then found by trial and error. This is a very time consuming process and it requires the user to be present throughout the operation of the program. Automating the process means that once all the parameters that the program requires are specified by the user, the program can be left to run without any further intervention. Thus the operator is now free to pursue other work and is not restricted by having to continually provide input. The program normally takes about eight hours to process one set of patient data, thus it is usually left to run overnight.

To automate the volume matching process a number of changes needed to be made. New versions of existing routines for the generating of the finite element mesh needed to be formed. The original versions were interactive, requiring the user to select options at each stage. Several new subroutines also had to be written to automate the volume matching process. A subroutine "PART4" was written to control the automated process, it allows the selection of all the options necessary for volume matching. Routine "NEWFIL" updates the data file, "XL1-data", for the finite element part with either new values for the material properties or new ventricular geometric data depending upon which is required. The process of finding the scaling factor for the displacements which gives a volume match is carried out by subroutines "VOLMATCH" and "GOLDSEC". Routine "VOLMATCH" checks to see if the new calculated volume (finite element mesh plus displacements) is within the required tolerance. If it is then the mechanical properties are written to a file, otherwise routine "GOLDSEC" is invoked. This routine uses a linear step method to find a region in which the required scaling factor exists. The golden section method (Fibonacci search) is then applied to the region to find the scaling factor that gives the matched volume to the required tolerance, typically < 0.1cm^3. The scaling factor found is then passed to routine "NEWFIL" to scale he elastic moduli and the finite element data file is then updated. The finite element and volume matching parts are repeated until a volume match is achieved (finite element mesh plus displacements gives the required volume). The newly formed mesh coordinates for the next frame are then substituted into the finite element data file. The newly formed coordinates being the original mesh data plus the displacements. This process continues until all the diastolic frames have been analysed.

A number of cosmetic changes were also made to aid the use of the program. Automatic naming of input and output files was introduced so that only the patient data file to be used need be specified and all other necessary files will be either generated, opened or appended. Better error trapping to improve the stability of the program was developed, e.g., if an incorrect or inconsistent input is given, the program will inform the user of the mistake instead of crashing or producing spurious output. A more efficient user interface was created as originally options were presented as a list of questions to be answered one after another. This has been changed to a menu system where any option can be chosen without having to go through several layers of unrelated options. Thus the program was made more structured, improving efficiency, readability and ease of future development. The program originally contained a large number of "GOTO" statements, and many of these have now been replaced with if blocks and do loops.

Excessively long routines were split into several smaller more manageable ones. Format statements were moved away from the executable statements to the end of the routines. This makes the executable statements more compact, tidier and easier to follow when examining the code.

4.4 Speed Improvements

The program took on average about twenty minutes, with low system load, to perform one application of the finite element process. When faster machines became available such as the Ultra Sparc™ a significant speed increase was not achieved. On average the speed increase was only in the order of 10% despite these machines being many times faster than the previously use IPX Workstations™. The reason for this was that the tasks were IO (input output) bound, that is although the newer machines were computationally faster than the older ones they were still subject to the same speed restrictions when accessing filestore which was used to store the global stiffness matrix. A normally slow process was further exacerbated as access to the global stiffness matrix was made very often in small amounts as the matrix was accessed element by element.

It was obvious that the analysis was being seriously slowed by this continual reading and writing of data to disk. There are two possible solutions to this problem; one is to lump all the reads and all the writes together. This enables better use of the system's data caches and reduces the time lost due to lag in requesting file data and receiving that data. The other is to bypass the system read/writes for the global stiffness matrix altogether. This was the method that was employed. A new subroutine was set up called "STOREMAT" this sets up a large array in which the global stiffness matrix is stored. All the previous reads and writes to the stiffness matrix are replaced with calls to this subroutine.

Original		**New**
READ(1,PBMK,REC=I)	Became	CALL STOREMAT(PBMK,0,I)
WRITE(1,PBMK,REC=I)	Became	CALL STOREMAT(PBMK,1,I)

This resulted in a substantial increase in speed. On the Ultra Spa ™ the execution time for one run of the F.E. program fell from approximately 20 minutes to a little under 3 minutes. This represents a substantial saving in time and will allow for significantly faster processing of data in the future. Even if the machine in question does not have enough conventional memory to store the entire stiffness matrix the operating system can use virtual memory which will still be significantly more efficient than the previous storage and retrieval method.

Chapter 5

5. A New Volume Matching Method

In Chapter 3 the method of "smoothed volume matching" was described. An alternative method is now proposed, "target volume matching", which is the same as the previous process with two exceptions. Firstly, instead of matching the smoothed volume we match a target volume. The target volume is given by the equation below,

$$T_{i+1} = \frac{V_i(S_{i+1} - S_i)}{S_i}$$

where T_{i+1} is the target volume for the next frame, V_i is the volume for the present frame and S_n is the smoothed spline volume for the F.E. mesh at frame n. The smoothed spline volumes are the same as those found for "smoothed volume matching". The finite element mesh is constructed for the first frame of late diastole. The finite element part of the program is then used to find the displacement of the mesh nodes, for an arbitrary value of the elastic modulus. The displacements are then scaled so that when added to the nodal coordinates the new mesh volume is the required target volume. The minor elastic modulus which had been chosen is then divided by the scaling factor and the finite element part is initiated again and the matching process is repeated. The modulus that gives a volume match has usually been found at this stage. The results of this are now written to file for later use. This is so far identical to "smoothed volume matching" except for the use of the target volume instead of the smoothed volume. Even this however takes the same value as the smoothed volume for the first volume match. The part that is different now follows. A new finite element mesh is now reconstructed from the RAO and LAO images for the next frame and the above process is repeated. This process continues until moduli have been calculated for all the frames in diastole. The advantages of this method are that errors will not be propagated from one frame to the next and also the mesh will become less distorted which is another source of inaccuracy. The disadvantages are that the continuity gained by using the same set of coordinates is lost and there is likely to be some loss of smoothness to the results. The other main disadvantage is that the cumulative effect upon the mesh of the simulation is lost. Thus it is more difficult to quantify the areas in which the model may perform poorly in predicting deformation.

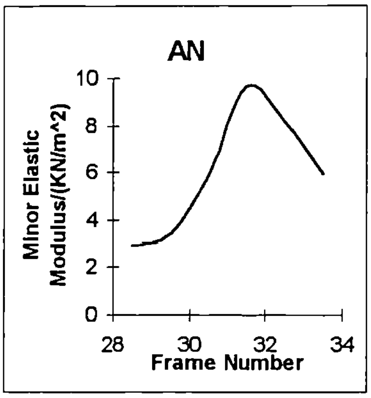
5.1 Results and Discussion

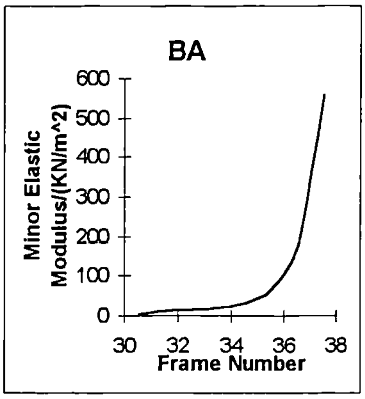
The minor modulus of elasticity has been calculated using target volumes and also using smoothed volumes. On the whole the moduli obtained from target volume matching tend to be slightly lower, the system is more flexible, than for the smoothed volume matching. This probably explains why features of these curves are more extenuated than when using the smoothed volume matching, since in that process those features are damped out by the increased stiffness of the system. However the overall shapes of both sets of graphs are fairly similar.

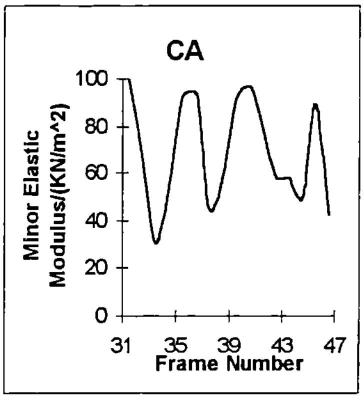
There is no apparent divergence between the values obtained by the two methods towards the end of diastole. This suggests that any distortion of the ventricular model that occurs with repeated application of matching the smoothed volumes does not greatly affect the value of the elastic modulus obtained.

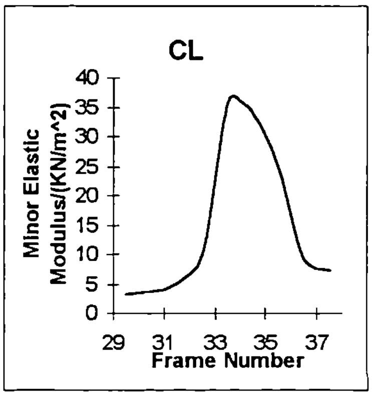
On the whole the results given in Fig 5.2 are similar to those obtained previously by Grewal [2] and presented in Chapter 3, Fig 3.4. However there are a number of differences which can be accounted for by the points raised in programming developments.

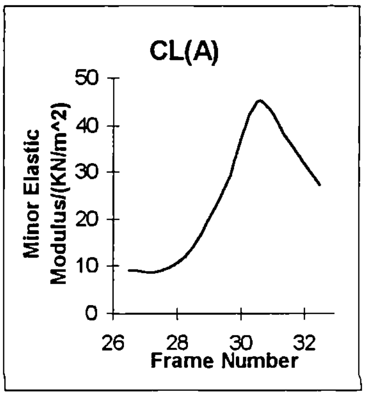
Included in graphs Fig 5.1 and Fig 5.2 are a number of graphs from previously not analysable sets of patient data. With the improvements described in Chapter 4 these sets of patient data can now been analysed. The results obtained from these previously unused sets of data agree strongly with the main conclusion of Grewal [2]. That is that there is no evidence that the left ventricular bulk properties follow an exponential stress/strain relationship.

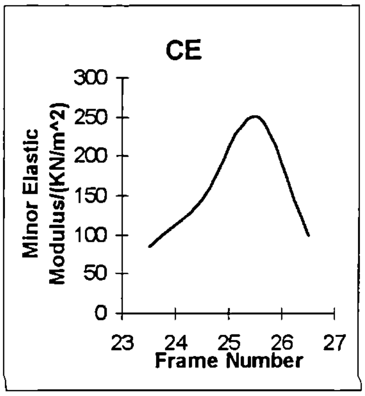


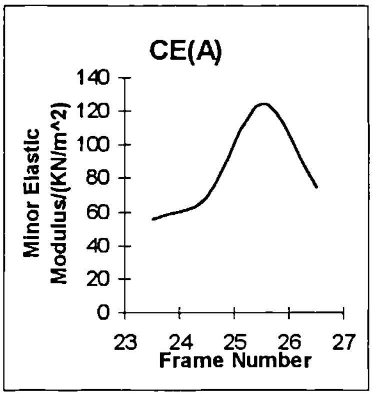


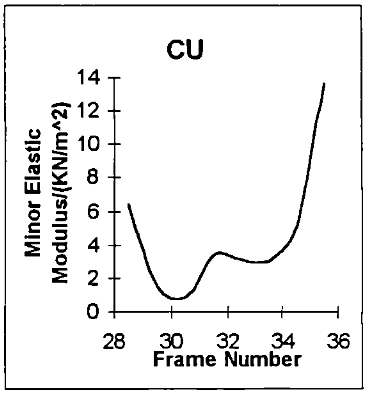


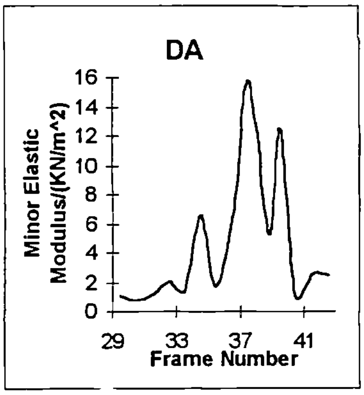












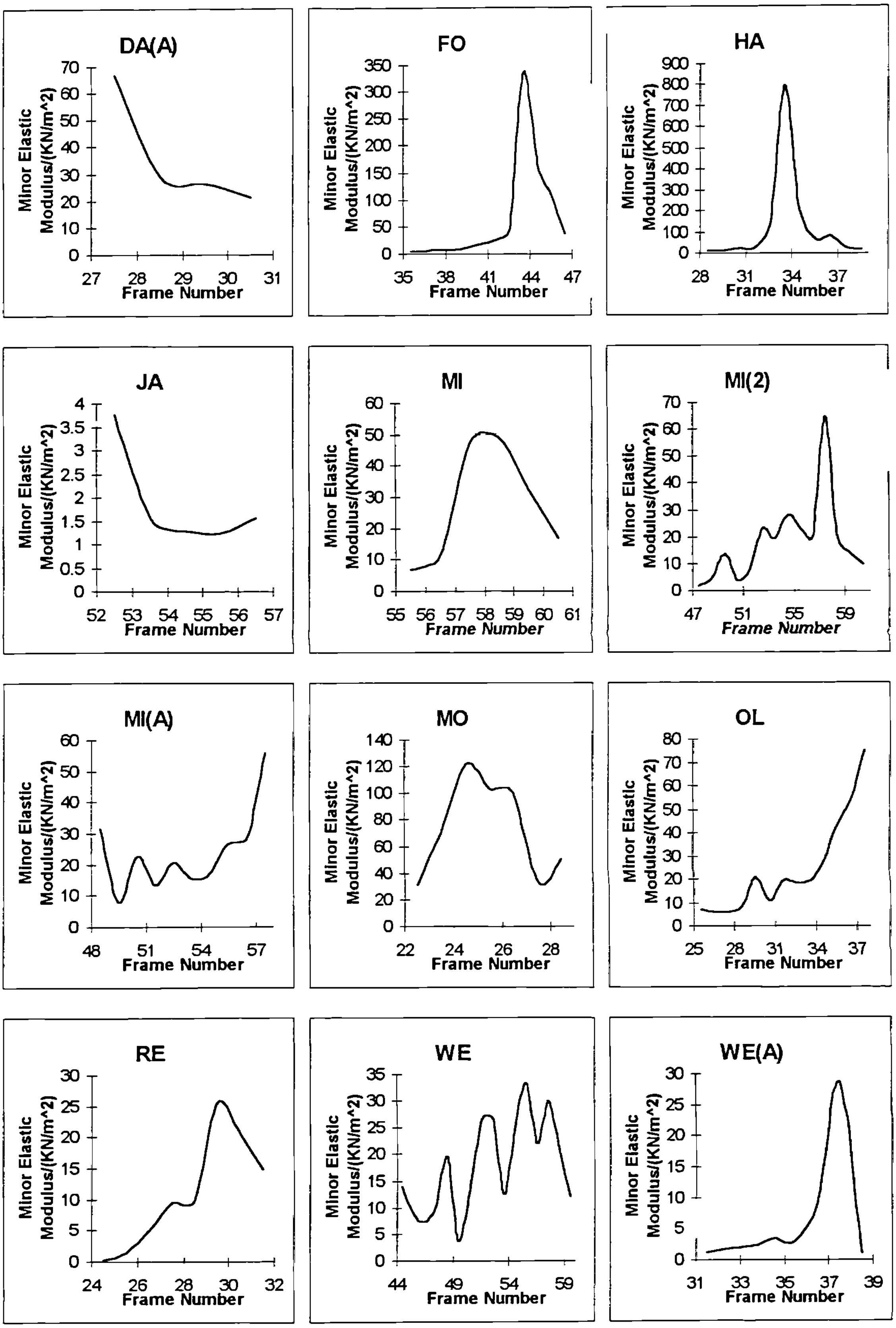
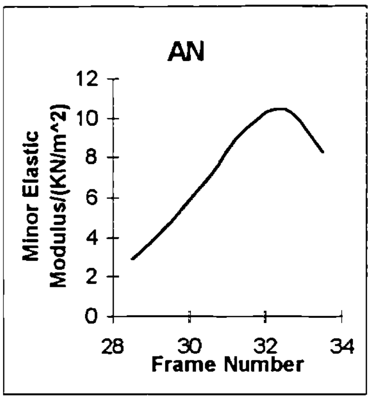
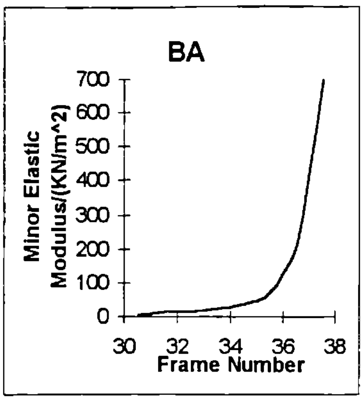
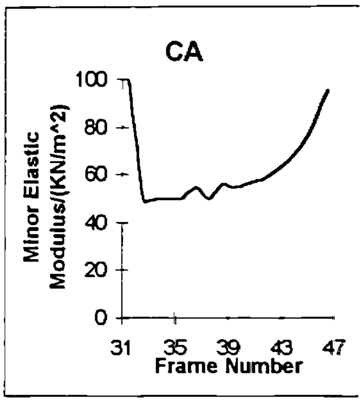
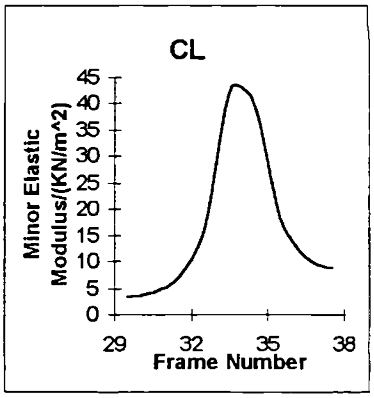


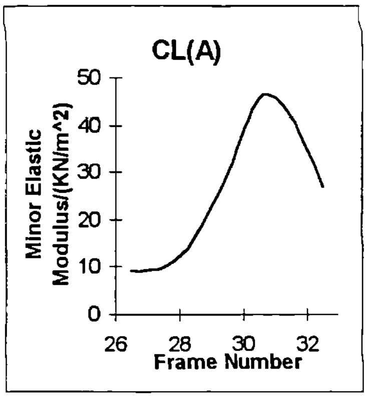
Fig 5.1 These graphs show the calculated minor elastic moduli during diastole for all the patients. The volumes used for matching were the target volumes.

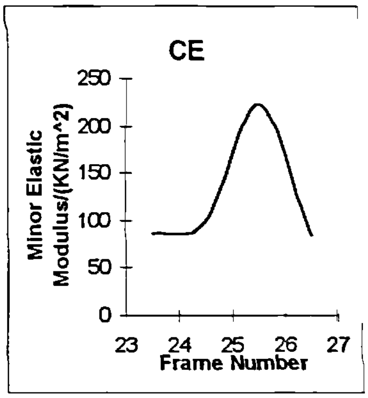

AN
Minor Elastic Modulus/(KN/m^2)
12
10
8
6
4
2
0
28
30
32
34
Frame Number

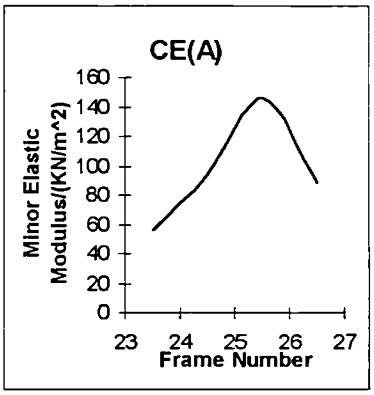

BA
Minor Elastic Modulus/(KN/m^2)
700
600
500
400
300
200
100
0
30
32
34
36
38
Frame Number

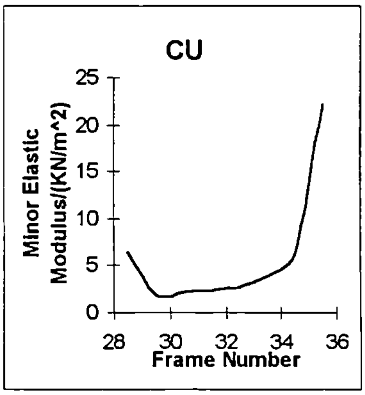

CA
Minor Elastic Modulus/(KN/m^2)
100
80
60
40
20
0
31
35
39
43
47
Frame Number

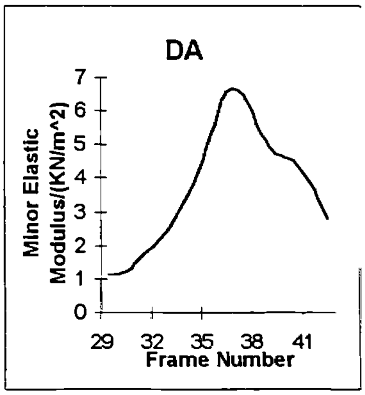

CL
Minor Elastic Modulus/(KN/m^2)
45
40
35
30
25
20
15
10
5
0
29
32
35
38
Frame Number

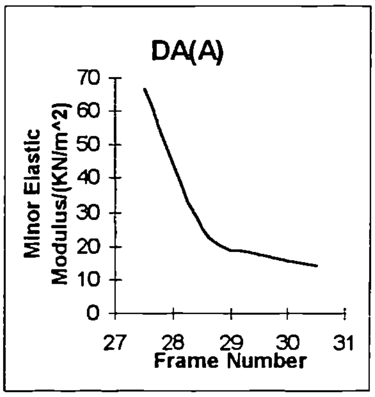

CL(A)
Minor Elastic Modulus/(KN/m^2)
50
40
30
20
10
0
26
28
30
32
Frame Number

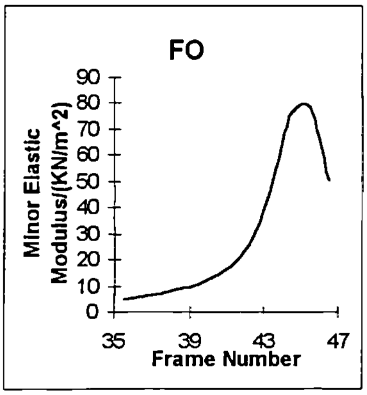

CE
Minor Elastic Modulus/(KN/m^2)
250
200
150
100
50
0
23
24
25
26
27
Frame Number

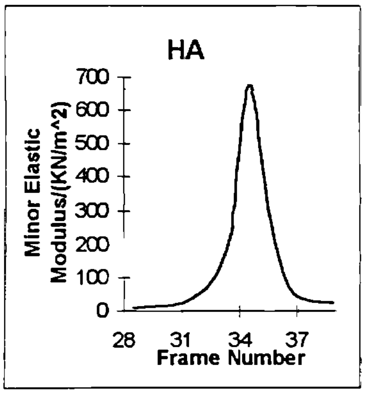

CE(A)
Minor Elastic Modulus/(KN/m^2)
160
140
120
100
80
60
40
20
0
23
24
25
26
27
Frame Number


CU
Minor Elastic Modulus/(KN/m^2)
25
20
15
10
5
0
28
30
32
34
36
Frame Number


DA
Minor Elastic Modulus/(KN/m^2)
7
6
5
4
3
2
1
0
29
32
35
38
41
Frame Number


DA(A)
Minor Elastic Modulus/(KN/m^2)
70
60
50
40
30
20
10
0
27
28
29
30
31
Frame Number


FO
Minor Elastic Modulus/(KN/m^2)
90
80
70
60
50
40
30
20
10
0
35
39
43
47
Frame Number


HA
Minor Elastic Modulus/(KN/m^2)
700
600
500
400
300
200
100
0
28
31
34
37
Frame Number

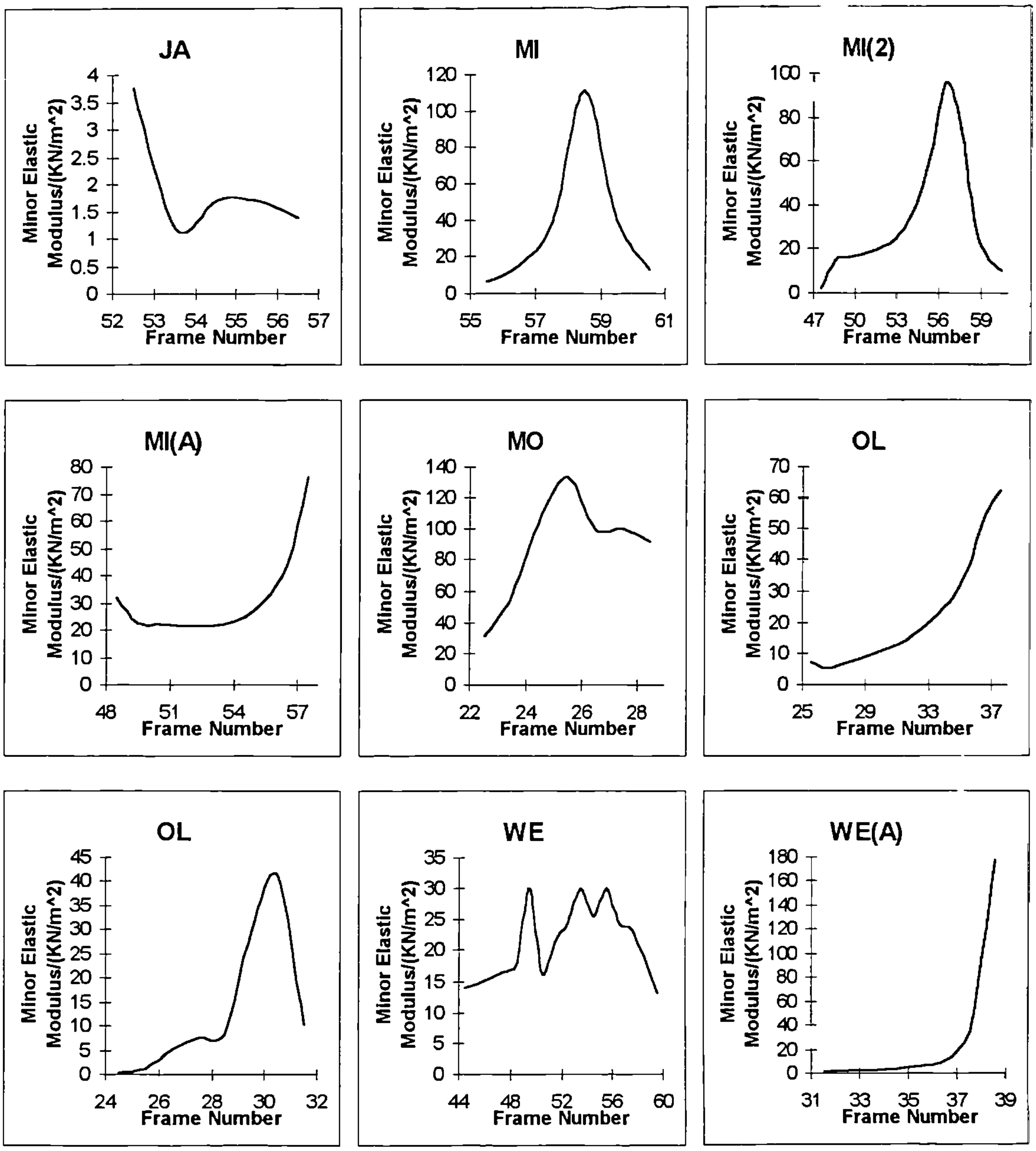


Fig 5.2 These graphs show the calculated minor elastic moduli during diastole for all the patients. The volumes used for matching were the smoothed volumes.

Chapter 6

6. Matching Both Length and Volume To Derive the Elastic Modulus Ratio

The aim of this Chapter was to gain an insight into the value of the modulus ratio, and to discover whether there was any evidence that this value varied during the diastolic phase To achieve this it was necessary to bring in another independent variable into the system. Up to the present time only the volume has been matched. To allow the two elastic moduli to vary independently, another property must also be matched, the length of the ventricle was the property chosen for this.

Very little work has been done to determine a suitable value of the elastic modulus ratio to be used for modelling purposes. Many authors have assumed a value of 2 for the major to minor modulus ratio in their finite element analysis, although there appears to be no firm evidence that this value is better than any other. A value of 2 has also been used in the work described in Chapter 3 and Chapter 5. This value had been derived from work done by Vinson [1] with thin shell ellipsoids and has no real connection to the later finite element analysis, other than that the finite element analysis grew out of the work with thin shell ellipsoids.

Work has been done by Yamada [71] on measuring the material properties of various biological materials, including that of the myocardium of the left ventricle. In his study small pieces of myocardium were taken from the mid-walls of the left ventricles of 61 patients. When these were tested for various material properties he concluded: "The ultimate elongation, in the direction transverse to the course of the fibres, is about 1.3 times that in the parallel direction." This would suggest that the major and minor modulus ratio was 1.3. However since no information on the magnitude of the stresses and strains was explicitly given, or even whether the stresses in the two directions were the same, we must therefore be cautious in assuming this ratio to be valid for this finite element analysis, although it would seem to be the obvious conclusion. The other cause for concern is that this muscle tissue had been removed from the myocardium and hence results obtained may not give an adequate indication of *in-vivo* material properties.

Other authors have also measured the elastic modulus ratio of excised myocardial tissue. The work of Yin *et al.* [62] showed that the value of this ratio could not only vary significantly from patient to patient but could also vary from site to site within the same heart. It was also suggested that the ratio was also history dependent. Thus the ratio could not only vary with position within the left ventricle but could also vary with time. This work therefore does not give any indication of what a reasonable choice for the bulk elastic modulus ratio would be.

The effect upon the length of a ventricular model of altering the modulus ratio has been investigated by Neckyfarow *et al.* [72]. Using the averaged frame data obtained from several cardiac cycles a thick shell solid model was produced from this composite frame. To reconstruct the geometry from the single cine-angiography frame a circular cross-section was assumed. The model was applied using three values for the minor to major modulus ratio 1.0, 0.8 and 0.5. It was stated that: "with the ratio E_2/E_1 less than 0.8 the predicted deformation agrees favourably with observed ventricle elongation." It is important to note that the ratio referred to here is the reciprocal of that referred to elsewhere in this text, as E_1 refers to the along-the-fibre and F_2 to the cross-the-fibre direction. Thus the major to minor modulus ratio that was prescribed is one that is greater than 1.3. The method used here produces *in-vivo* results and is thus more closely allied to this work than that done by Yamada [71]. However there must be some concern over the method used to average the multiple readings and align them to be at exactly the same point in the cardiac cycle. There must also be concern that these results were obtained using only single plane cine-angiography which does not give as accurate volume information as bi-plane cine-angiography. Another concern is that the volume obtained from the single plane reconstruction will be far more dependent upon the length measured from the X-ray frame than is the true volume. Lastly this analysis was performed for only one subject and it would require results from further subjects for the result and the procedure to prove its validity.

Two independent sources using quite different methods have therefore both arrived at a value of approximately 1.3, although the second did seem to imply that a value slightly higher than this would give a better match with the observed ventricular behaviour The two methods complement each other well and the value arrived at gains greater validity because of this. However both approaches have their shortcomings when considering the *in-vivo* case.

6.1 Resolving Myocardial Stiffness into Circumferential and Longitudinal Directions

A small block of material, analogous to the myocardium in its structure was investigated mathematically. This was to gain an understanding of the likely consequences of altering the modulus ratio and discover how fibre angle affected the strains within the myocardium as well as the global deformation of the myocardium. In particular this is an attempt to investigate what effect the elastic modulus ratio has upon the stiffness of a small element.

A small cuboid that is composed of elastic fibres was considered. Just as in the myocardium these fibres are straight and always in a plane parallel to the two opposing faces. The angle of the fibres to the horizontal are 60^O at one face and -60^O at the other face with the angle varying linearly between these two values within the cuboid (see Fig 6.1).

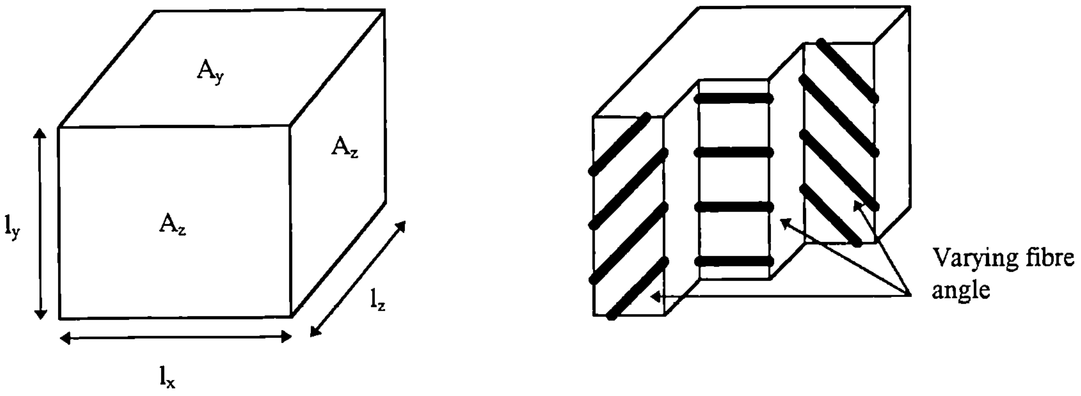


Fig 6.1 This diagram shows a cuboid with a fibre angle varying linearly from 60° to -60° through the depth of the block. Where, A_x, A_y, A_z are the areas of the three faces and l_x, l_y, l_z are the lengths of the three sides.

The stiffness of a piece of material in one of the x, y, z, Cartesian coordinate directions is given below,

$$k_i = \frac{A_i \sigma_i}{\varepsilon_i l_i} \qquad \text{Eqn. 6.1}$$

Thus to find the stiffness in the i-direction the strain in the i-direction that occurs in response to the stress in the i-direction must be found.

The stiffness of a slice at any depth within the block in either of the x, y, Cartesian coordinate directions will vary, provided the modulus of elasticity is not the same in the along the fibre direction as in the across the fibre direction. A "thin" slice at an arbitrary depth within the block was considered, the term "thin" meaning that there is no change in fibre orientation within the slice.

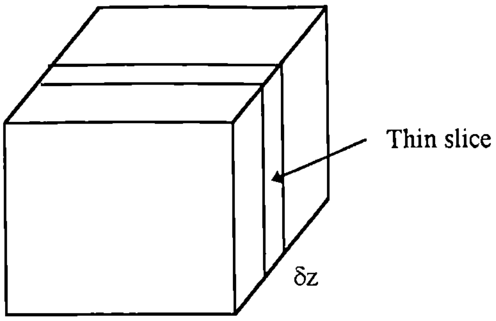


Fig 6.2 This is the same block as Fig 6.1. We now consider a thin slice of this block at some arbitrary depth of thickness δz.

This block will have the fibres running at some angle θ to the horizontal, where θ is in the range -π/3, π/3. This angle is linearly dependent upon the depth the slice is within the block. We shall first consider the case of θ = 0 and then generalise it to an arbitrary angle by applying a transformation matrix. The slice or lamina below has the fibres running parallel to the 2-direction and perpendicular to the 1-direction.

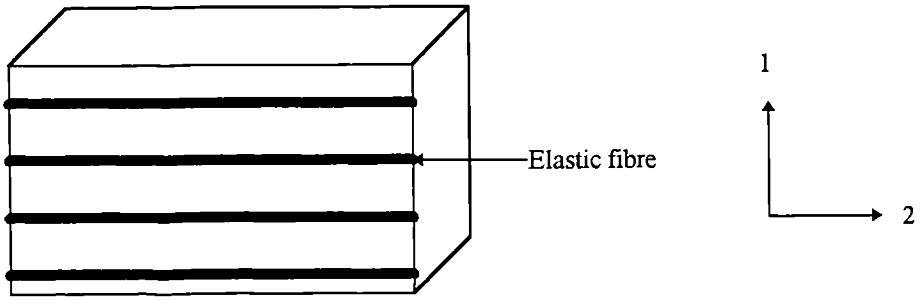


Fig 6.3 This shows a thin slice of material with the elastic fibres running parallel to one of the sides. The directions 1 and 2 are the perpendicular and longitudinal directions relative to the elastic fibres.

The material properties for a lamina such as this are given by the tensor equation.

$$\begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \gamma_{12} \end{bmatrix} = \begin{bmatrix} 1/E_1 & -v_{21}/E_2 & 0 \\ -v_{12}/E_1 & 1/E_2 & 0 \\ 0 & 0 & 1/G_{12} \end{bmatrix} \begin{bmatrix} \sigma_1 \\ \sigma_2 \\ \tau_{12} \end{bmatrix} \qquad \text{Eqn. 6.2}$$

This equation relates the normal and shear strains in the 1 and 2-directions to the normal stresses, the shear stresses and the material properties in the 1 and 2-directions.

If E_2 is assumed to be related to E_1 by some ratio r then to preserve the symmetry of the material properties matrix ν_{21} must equal r ν_{12}. The above matrix equation then simplifies to,

$$\begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \gamma_{12} \end{bmatrix} = \begin{bmatrix} 1/E & -\nu/E & 0 \\ -\nu/E & 1/(Er) & 0 \\ 0 & 0 & 1/G \end{bmatrix} \begin{bmatrix} \sigma_1 \\ \sigma_2 \\ \tau_{12} \end{bmatrix} \qquad \text{Eqn. 6.3}$$

The subscripts on the material properties can now be dropped, E_1 is replaced by E, E_2 by Er, ν_{12} by ν and ν_{21} by rν, as there is now no need to distinguish between material constants in the different directions.

To convert the strains from the local 1, 2-coordinate system to the global x, y-coordinate system requires a transformation matrix. The matrix below can be deduced by resolving forces between the two coordinate systems (Holmes and Just [73]).

$$\begin{bmatrix} \varepsilon_x \\ \varepsilon_y \\ \gamma_{xy} \end{bmatrix} = \begin{bmatrix} \cos^2\theta & \sin^2\theta & -\sin\theta\cos\theta \\ \sin^2\theta & \cos^2\theta & \sin\theta\cos\theta \\ 2\sin\theta\cos\theta & -2\sin\theta\cos\theta & \cos^2\theta - \sin^2\theta \end{bmatrix} \begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \tau_{12} \end{bmatrix} \qquad \text{Eqn. 6.4}$$

If the right hand side of tensor equation 6.3 is multiplied out and the transformation from the local to the global coordinates is applied then the global strains can be expressed in terms of the local stresses. The equations for the global normal strains are given below. The global shear strain is not given, as shear deformation of the block is not being considered in this analysis.

$$\left.\begin{aligned} \varepsilon_x &= \frac{1}{\mathbf{E}}\{(\sigma_1 - \nu\sigma_2)\cos^2\theta + (\sigma_2/\mathbf{r} - \nu\sigma_1)\sin^2\theta\} - \frac{1}{\mathbf{G}}\tau_{12}\sin\theta\cos\theta \\ \varepsilon_y &= \frac{1}{\mathbf{E}}\{(\sigma_1 - \nu\sigma_2)\sin^2\theta + (\sigma_2/\mathbf{r} - \nu\sigma_1)\cos^2\theta\} + \frac{1}{\mathbf{G}}\tau_{12}\sin\theta\cos\theta \end{aligned}\right\} \text{Eqn. 6.5}$$

The next stage is to replace the local stresses in equations 6.5 with expressions for them in terms of the global stresses. As with the strains there is a transformation matrix to convert from local to global coordinates and vice versa. This matrix is similar to the one required for the strains and is obtained in a similar manner.

$$\begin{bmatrix}\sigma_1\\ \sigma_2\\ \gamma_{12}\end{bmatrix}=\begin{bmatrix}\cos^2\theta & \sin^2\theta & 2\sin\theta\cos\theta\\ \sin^2\theta & \cos^2\theta & -2\sin\theta\cos\theta\\ -\sin\theta\cos\theta & \sin\theta\cos\theta & \cos^2\theta-\sin^2\theta\end{bmatrix}\begin{bmatrix}\sigma_x\\ \sigma_y\\ \gamma_{xy}\end{bmatrix}\qquad \text{Eqn. 6.6}$$

Equations 6.5 now become,

$$\left.\begin{aligned}\varepsilon_x &= \frac{1}{\mathbf{E}}\{(\cos^2\theta-\nu\sin^2\theta)(\sigma_x\cos^2\theta+\sigma_y\sin^2\theta+2\tau_{xy}\sin\theta\cos\theta)\\ &\quad+({\sin^2\theta}/_{\mathbf{r}}-\nu\cos^2\theta)(\sigma_x\sin^2\theta+\sigma_y\cos^2\theta-2\tau_{xy}\sin\theta\cos\theta)\}\\ &\quad-\frac{1}{\mathbf{G}}\sin\theta\cos\theta(-\sigma_x\sin\theta\cos\theta+\sigma_y\sin\theta\cos\theta+\tau_{xy}(\cos^2\theta-\sin^2\theta))\\ \varepsilon_y &= \frac{1}{\mathbf{E}}\{(\sin^2\theta-\nu\cos^2\theta)(\sigma_x\cos^2\theta+\sigma_y\sin^2\theta+2\tau_{xy}\sin\theta\cos\theta)\\ &\quad+({\cos^2\theta}/_{\mathbf{r}}-\nu\sin^2\theta)(\sigma_x\sin^2\theta+\sigma_y\cos^2\theta-2\tau_{xy}\sin\theta\cos\theta)\}\\ &\quad+\frac{1}{\mathbf{G}}\sin\theta\cos\theta(-\sigma_x\sin\theta\cos\theta+\sigma_y\sin\theta\cos\theta+\tau_{xy}(\cos^2\theta-\sin^2\theta))\end{aligned}\right\}\qquad \text{Eqn. 6.7}$$

These two equations are unnecessarily long and can easily be simplified. If when considering the strain in the x-direction we assume there is no stress in the y-direction and vice versa. This will eliminate half the stress terms, almost halving the length of the above two equations. We are also at liberty to distribute the strain over the element in such a way that there will be no shearing in the x, y-plane. Thus the above two equations become,

$$\left.\begin{aligned}\varepsilon_x &= {}^{\sigma_x}/_{E}\left\{\cos^4\theta+\left({}^{E}/_{G}-2\nu\right)\sin^2\theta\cos^2\theta+\sin^4\theta/r\right\}\\ \varepsilon_y &= {}^{\sigma_y}/_{E}\left\{\sin^4\theta+\left({}^{E}/_{G}-2\nu\right)\sin^2\theta\cos^2\theta+\cos^4\theta/r\right\}\end{aligned}\right\}\qquad \text{Eqn. 6.8}$$

The shear modulus G is independent of the elastic modulus and the modulus ratio for all but the isotropic case, r=1, when G is given by the equation,

$$G=\frac{E}{2(1+\nu)}$$

Since we are only interested at this stage with the effect of the modulus ratio and not that of the modulus of rigidity it would seem sensible to keep its value constant while varying the modulus ratio. Replacing the isotropic value of G into equation 6.8 yields,

$$\left.\begin{aligned}\varepsilon_x &= {}^{\sigma_x}\!/_{E}\left\{\cos^4\theta + 2\sin^2\theta\cos^2\theta + \sin^4\theta/r\right\}\\ \varepsilon_y &= {}^{\sigma_y}\!/_{E}\left\{\sin^4\theta + 2\sin^2\theta\cos^2\theta + \cos^4\theta/r\right\}\end{aligned}\right\} \qquad \text{Eqn 6 9}$$

Substituting these values for the strains into the equations for the stiffness in the Cartesian coordinate directions gives.

$$\left.\begin{aligned}k_x &= \left\{\frac{EA_x}{l_x}\right\}\left\{\cos^4\theta + 2\sin^2\theta\cos^2\theta + \sin^4\theta/r\right\}\\ k_y &= \left\{\frac{EA_y}{l_y}\right\}\left\{\sin^4\theta + 2\sin^2\theta\cos^2\theta + \cos^4\theta/r\right\}\end{aligned}\right\} \qquad \text{Eqn 6 10}$$

The equations for k_x and k_y are made up from two parts. The first is the stiffness for the element, assuming it is isotropic, and the second is a non-dimensional term (the "non-dimensional stiffness factor") which modifies this for the non-isotropic case It is this factor that produces the change in behaviour with varying fibre angle and modulus ratio. Hence it is only this part that is of interest to us. Below are two plots of this factor one for that of k_x and one for k_y. Plots are made for three values of the modulus ratio 1,1.3 and 2. for θ in the range -π/3 to π/3.

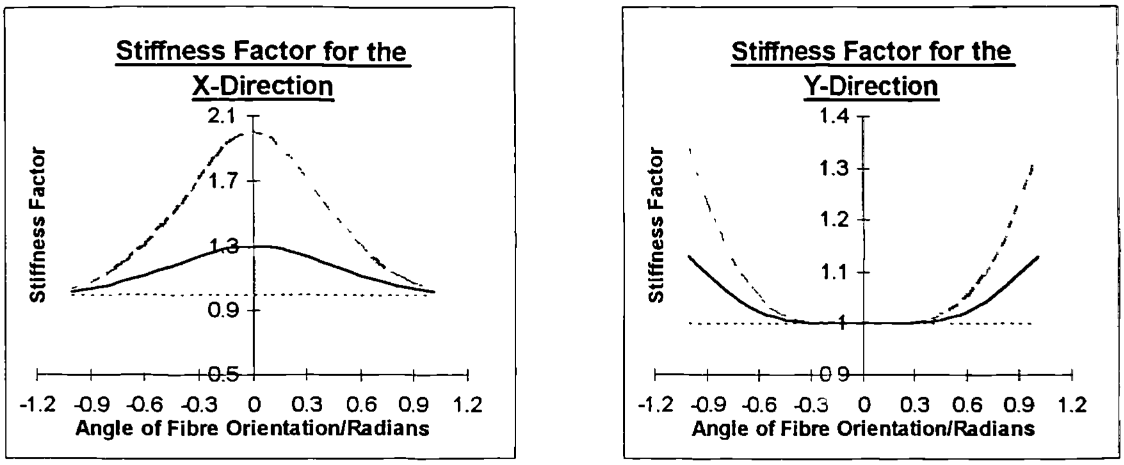


Fig 6.4 These two graphs show how the elastic modulus ratio affects the stiffness of an element as the fibre angle changes. The dotted line is for a modulus ratio of 1, the solid for a value of 1.3 and the broken line for a value of 2.

As should obviously be the case for the isotropic situation the factor has a constant value of 1. This is also obviously going to be so from inspection of the above equations. As can be seen the area under both graphs increases as the modulus ratio increases. This is to be expected as altering the modulus ratio changes the overall stiffness of the element. There is however a much greater change for k_x. This vast difference is because the maximum value for k_y is at multiples of $\pi/2$ but this value is well outside the range of angles that are exhibited within the myocardium. Intuitively a material will be stiffest in a particular direction if the fibres are parallel to this direction and will be less stiff as the angle of the fibres to this direction increases up to $\pi/2$ radians which will be a minimum. This is true only if r is greater than 1. From inspection of the above graphs this is obviously going to be the case.

To obtain the stiffness for the entire block in the x and y-directions the stiffness of the lamina must be integrated through the thickness of the block. Thus the stiffness of the block in the x and y-directions, K_x and K_y, is given by,

$$\left.\begin{aligned} K_x &= \left\{\frac{El_y}{l_x}\right\}\int_0^{l_z}\frac{1}{\left\{\cos^4\theta+2\sin^2\theta\cos^2\theta+\sin^4\theta/r\right\}}dz \\ K_y &= \left\{\frac{El_x}{l_y}\right\}\int_0^{l_z}\frac{1}{\left\{\sin^4\theta+2\sin^2\theta\cos^2\theta+\cos^4\theta/r\right\}}dz \end{aligned}\right\} \qquad \text{Eqn. 6.11}$$

In order to perform the integration we need, in the above pair of equations, to replace dz with the equivalent in $d\theta$ since θ is linearly dependant upon depth into the block.

$$z=\frac{l_z}{2}\left(1+{}^{3\theta}\!/_{\pi}\right)$$

Therefore,

$$dz=\frac{3l_z}{2\pi}d\theta$$

Upon substitution for dz in the above equation pair,

$$\left.\begin{aligned} K_x &= \left\{\frac{3EI_y l_z}{2\pi l_x}\right\}\int_0^{l_z}\frac{1}{\left\{\cos^4\theta+2\sin^2\theta\cos^2\theta+\sin^4\theta\, r\right\}}dz \\ K_y &= \left\{\frac{EI_x l_z}{2\pi l_y}\right\}\int_0^{l_z}\frac{1}{\left\{\sin^4\theta+2\sin^2\theta\cos^2\theta+\cos^4\theta\ r\right\}}dz \end{aligned}\right\} \qquad \text{Eqn. 6.12}$$

An attempt was made to integrate the above two equations analytically using the symbolic manipulation package Mathematica. This however did not prove possible and so it was integrated numerically, using the default options of Mathematica. Below are two graphs of the definite integral between -π/3 and π/3 of the non-dimensional stiffness factor of K_x and K_y for r in the range 1 to 10.

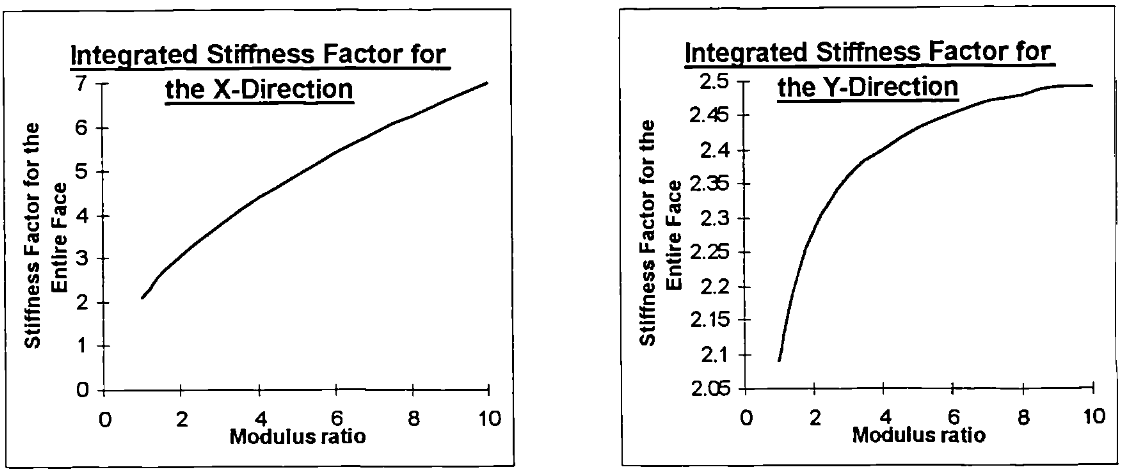


Fig 6.5 These two graphs show how the stiffness factor in the x and y-directions varies with the elastic modulus ratio.

As was to be expected the modulus ratio has a far greater affect upon the stiffness in the x-direction than it does on the stiffness in the y-direction. When increasing the modulus ratio from 1 to 10 the stiffness factor in the x-direction increases by 250% but in the y-direction the increase in stiffness is only 19%. Thus the effect that the modulus ratio has in the y-direction is negligible compared to that in the x-direction.

This is not yet the complete story as we still have the question of how much effect the modulus of rigidity has upon the relative stiffness in the two orthogonal directions. Below are two graphs showing how the stiffness varies with the changing fibre angle, for three different values of the modulus of rigidity. The value prescribed in the isotropic case is shown by the solid line. The dotted line is for a value that is half this and the dashed line is for a value that is twice the isotropic value. Note that Poisson's ratio is assumed to be 0.5 and a modulus ratio of 2 is also assumed.

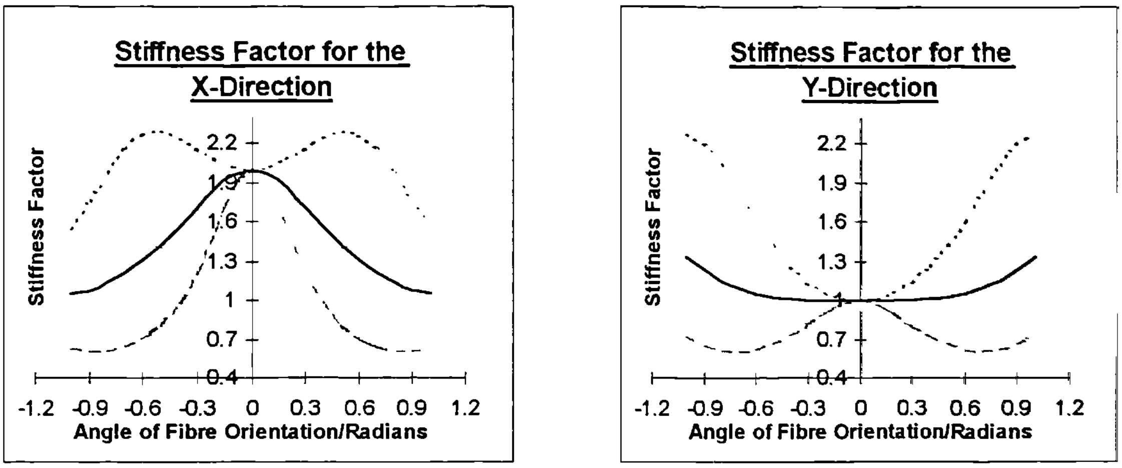
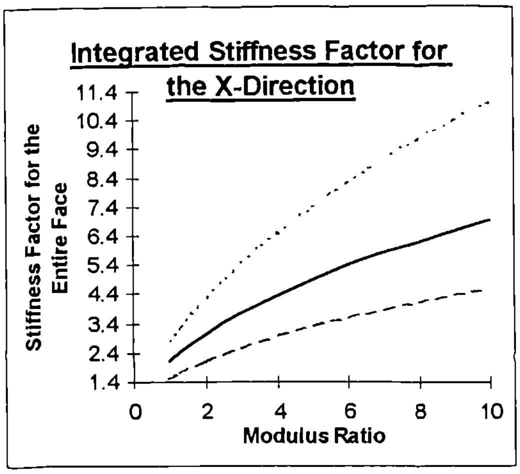


Fig 6.6 These two graphs show the effect that the shear modulus has upon the stiffness of the element as the fibre angle changes. The dotted line is for a shear modulus double the normally used value, the solid line is for the normally used value and the broken line is for a value half this.

As might be expected the effect that changing the modulus of rigidity has on the stiffness in the x and y-directions appears to be fairly consistent, and at this stage one might be inclined to believe that the modulus of rigidity may not be an important factor in the relative normal strains in the x and y-directions.

Using the above three different values for the modulus of rigidity the stiffness factor for the entire x and y faces of the block are plotted below for the modulus ratio in the range 1 - 10. It should be noted that although the plots start at a valu of 1 for the modulus ratio, the modulus of rigidity is in reality fixed for the isotropic case. This value has been included for comparison only.



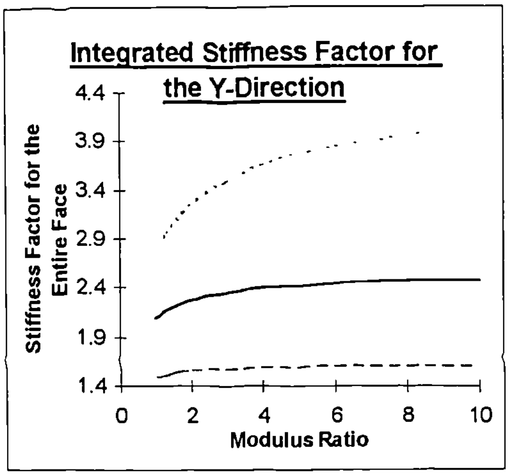


Fig 6.7 These two graphs show how much effect the modulus of rigidity has upon the stiffness in the x and y-direction compared to the effect of the modulus ratio.

The increase in the value of the stiffness factor in the x-direction when the modulus ratio increases from 1 to 10 varies between 209% and 298% for the three values of the modulus of rigidity. In the y-direction for the same values the increase is in the range 8% and 45%. Thus the modulus of rigidity has a greater relative effect upon the stiffness in the y-direction than that in the x-direction. However this is unlikely to be of any importance as the overall effect of the modulus ratio in the x-direction is still going to be extremely dominant when compared to that in the y-direction.

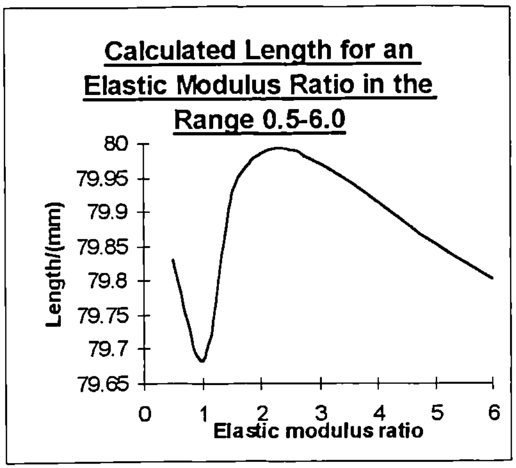
The other thing to note is that when the value for the modulus of rigidity is changed it has a greater effect upon the stiffness in both the x and y-directions than the same change in the modulus ratio. This is most obvious in the y-direction where the effect the modulus of rigidity has is far greater than that of the modulus ratio. In the x-direction the effect of both parameters is of the same order although the effect of the modulus of rigidity is still slightly greater.

At first glance this analysis may appear far removed from the original left ventricle problem, since the ventricle has a complex geometry and is not composed of vertical rectangular elements. However most of the ventricle's surface is approximately vertical.

At the apex, where this is patently untrue the myocardium is exceptionally stiff and any strains here are likely to be negligible compared to those within the rest of the ventricle. The above results suggest that there will be little or no direct effect upon the ventricle length when changing the modulus ratio and that the greatest effect on ventricular expansion in altering the modulus ratio will be in the circumferential direction. The above results also suggest that the modulus of rigidity can have a significant effect upon normal strains and that a poor choice for a value for this parameter may greatly affect results when attempting to match ventricular volumes during diastole.

6.2 Initial Investigation into the Effect of Altering the Modulus Ratio

An initial investigation into the effect which altering the modulus ratio has upon the volume and length of our ventricular model was undertaken. The volume matching process was applied to the first frame of diastole for one patient using values for the modulus ratio in the range 0.5 - 6. Below are four graphs; Fig 6.8 shows the calculated length against the modulus ratio for constant minor modulus, Fig 6.9 shows the calculated volume against the modulus ratio for constant minor modulus, Fig 6.10 shows the length of the ventricle after the volume has been matched to that of the second frame in diastole and lastly Fig 6.11 is the same as Fig 6.10 except that the range of the modulus ratio is much greater. The graphs were produced using ata from frames 33 and 34 of patient "BA". The initial value chosen for the modulus is 30 KN/m^2 and the pressure increase between the two frames is 1.39 mm Hg.



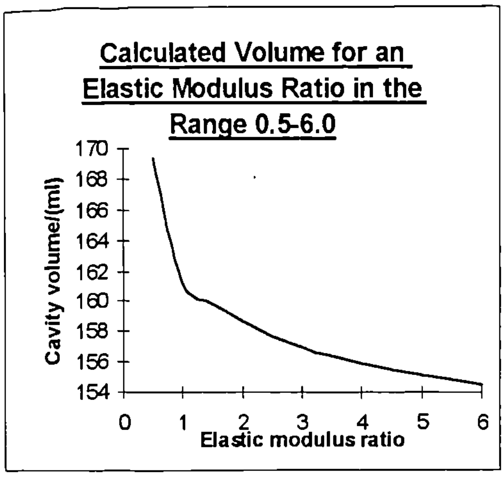
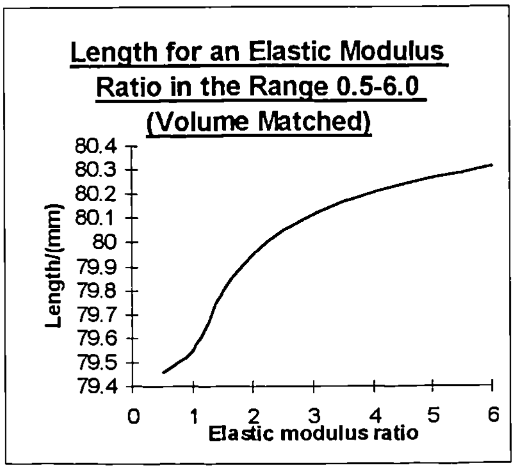


Fig 6.8 and Fig 6.9 The first graph shows how the length of the model is affected by changing the elastic modulus ratio and the second how the volume is affected.



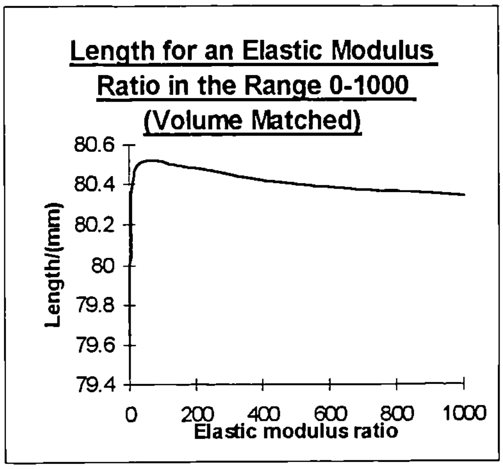


Fig 6.10 and Fig 6.11 Both the graphs show how the length of the cavity volume matched model varies with the elastic modulus ratio.

From the first graph, Fig 6.8, it can be seen that the length of the ventricle does not vary drastically over the range 0.5 - 6.0 for the modulus ratio. The graph appears to have an unexpected shape. The sudden change in the way in which the length is affected when the modulus ratio is below 1 is likely to be due to the change in the calculation of Poisson's ratio (see Section 6.6). The length of the undisplaced ventricle is 79.2mm and the maximum and minimum length increases observed are 0.8mm and 0.5mm respectively, the difference being only 0.3mm. This means that within the range investigated for the modulus ratio the largest difference in the overall length that could be made by varying the modulus ratio was less than 0.4 % of the initial length. This agrees strongly with the findings of section 6.1 which suggested that altering the modulus ratio would have little direct effect upon the calculated length of the ventricle.

The second graph, Fig 6.9, shows the volume of the ventricle calculated with a minor elastic modulus of 30 KN/m^2. The volume decreases in an exponential manner as the modulus ratio increases. The volume decrease, in going from a modulus ratio of 0.5 to one of 6.0, is 14.8 cm^3. This is an 8.7 % decrease in volume over this range. Thus the effect upon the volume of altering the modulus ratio is in excess of an order of magnitude greater than the effect it has upon the ventricle length, again supporting the analysis of Section 6.1.

The third graph, Fig 6.10, shows the length of the ventricle after the minor modulus has been adjusted to ensure that the volume matches that at the next frame. As the modulus ratio increases, the curve appears to approach an asymptote at just under 80.5mm. Therefore the length of the ventricle cannot be matched with a length greater than this by consideration of the value of the modulus ratio alone. It is also worth noting that if a length match is required near to the maximum achievable value, then the modulus ratio would be extremely sensitive to the accuracy of the measured length of the ventricle.

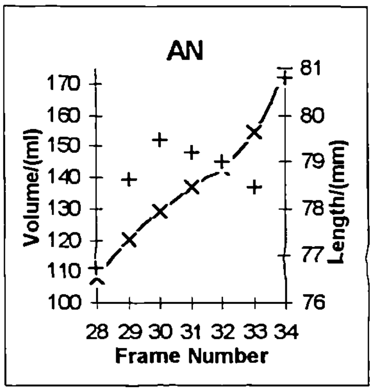
The fourth graph, Fig 6.11, shows how the length of the volume-matched ventricle varies as the modulus ratio becomes large. At first glance this may appear to be strange and unexpected behaviour. The length increases to a maximum after which it reduces slightly and then levels out to a plateau. This is entirely consistent and predictable with the findings of Section 6.1. When the modulus ratio is small and then increased slightly the greatest effect is in the circumferential direction and when the volume is matched the length of the ventricle is increased. On the other hand if the modulus ratio is large then the ventricle is almost rigid in the circumferential direction, and changes to the modulus ratio will have little effect upon the stiffness in this direction. The longitudinal direction however may still be far from rigid and can be affected by changes in the modulus ratio.

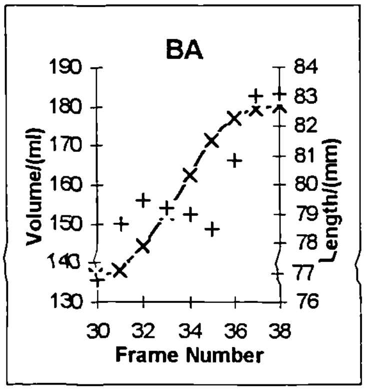
Although these changes are small they are now dominant since the effect in the circumferential direction is now negligible.

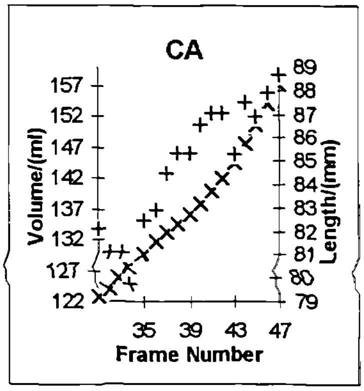
6.3 Attempts at Matching the Ventricle Length by Varying the Elastic Modulus Ratio

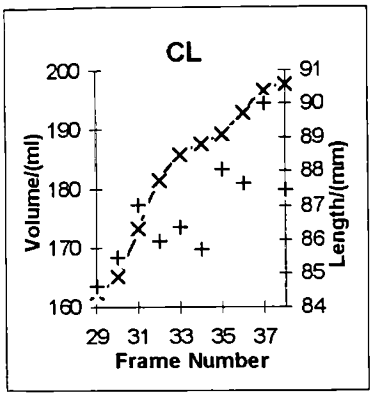
The main aim of this investigation has been to try to match the computer generated model's length during diastole with that measured from the X-rays taken of the ventricle during the same time period. This unfortunately has proved less than completely successful. Despite this some interesting results have come out of this piece of work.

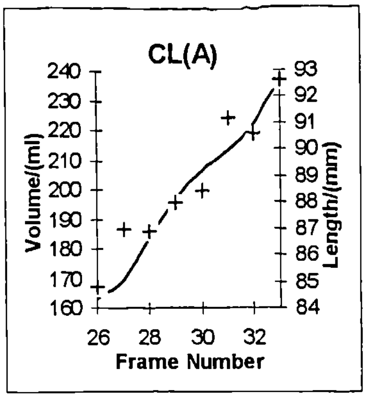
The ventricle length during the diastolic phase was plotted for each of the twenty one sets of patient data used in this study. There were two main things to notice. Firstly, the length was not monotonically increasing and secondly, the increase in length of the ventricle during this stage was very small compared to the change in volume. Typically during the diastolic period the volume would increase by approximately 20% while the length would only increase by about 5%.

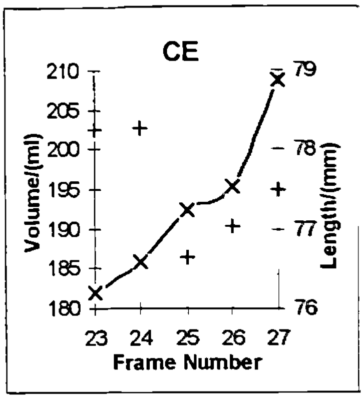


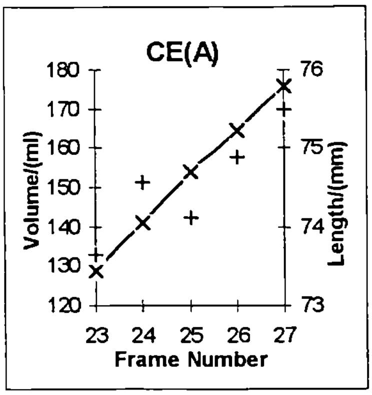


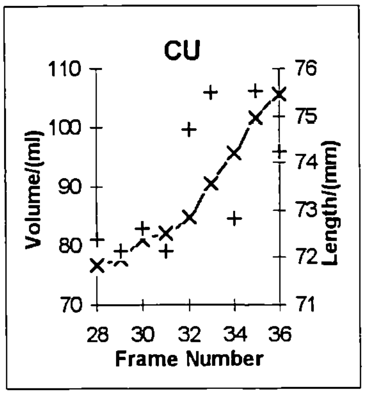


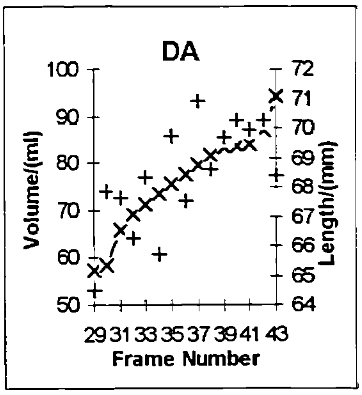

CL
Volume/(ml)
Length/(mm)
200
190
180
170
160
91
90
89
88
87
86
85
84
29 31 33 35 37
Frame Number

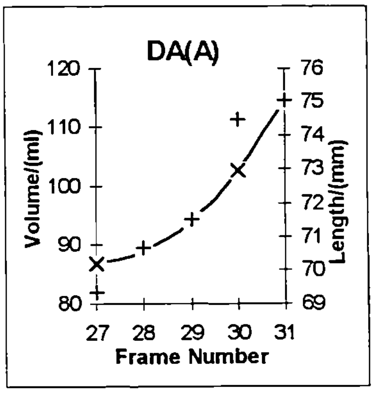

CL(A)
Volume/(ml)
Length/(mm)
240
230
220
210
200
190
180
170
160
93
92
91
90
89
88
87
86
85
84
26 28 30 32
Frame Number

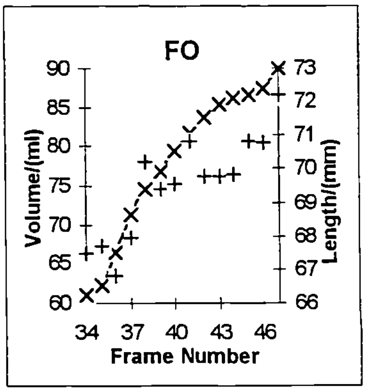

CE
Volume/(ml)
Length/(mm)
210
205
200
195
190
185
180
79
78
77
76
23 24 25 26 27
Frame Number

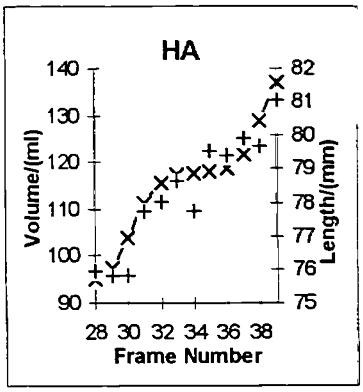

CE(A)
Volume/(ml)
Length/(mm)
180
170
160
150
140
130
120
76
75
74
73
23 24 25 26 27
Frame Number

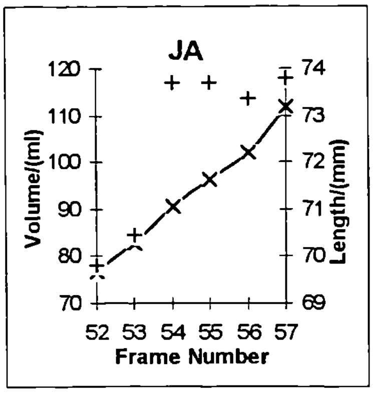

CU
Volume/(ml)
Length/(mm)
110
100
90
80
70
76
75
74
73
72
71
28 30 32 34 36
Frame Number

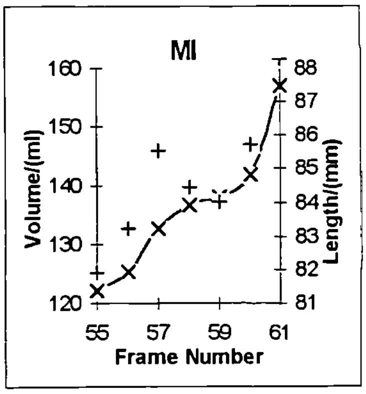

DA
Volume/(ml)
Length/(mm)
100
90
80
70
60
50
72
71
70
69
68
67
66
65
64
29 31 33 35 37 39 41 43
Frame Number

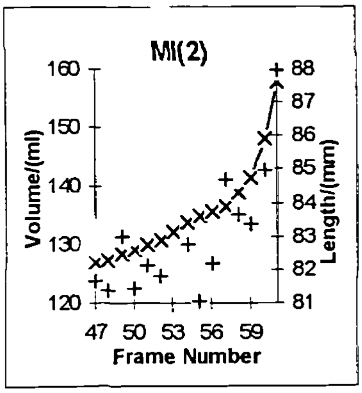

DA(A)
Volume/(ml)
Length/(mm)
120
110
100
90
80
76
75
74
73
72
71
70
69
27 28 29 30 31
Frame Number


FO
Volume/(ml)
Length/(mm)
90
85
80
75
70
65
60
73
72
71
70
69
68
67
66
34 37 40 43 46
Frame Number


HA
Volume/(ml)
Length/(mm)
140
130
120
110
100
90
82
81
80
79
78
77
76
75
28 30 32 34 36 38
Frame Number


JA
Volume/(ml)
Length/(mm)
120
110
100
90
80
70
74
73
72
71
70
69
52 53 54 55 56 57
Frame Number


MI
Volume/(ml)
Length/(mm)
160
150
140
130
120
88
87
86
85
84
83
82
81
55 57 59 61
Frame Number


MI(2)
Volume/(ml)
Length/(mm)
160
150
140
130
120
88
87
86
85
84
83
82
81
47 50 53 56 59
Frame Number

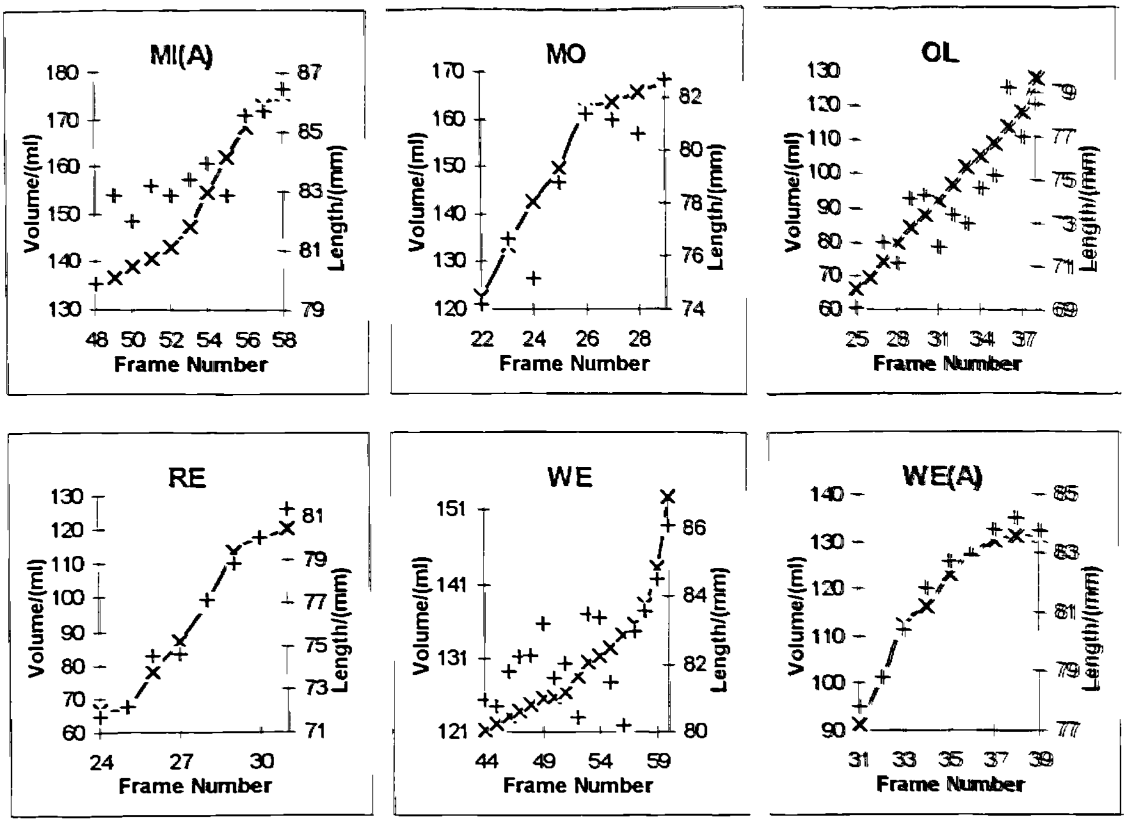


Fig 6 12 This set of graphs shows how the smoothed cavity volumes and the measured ventricle lengths change over the diastolic phase. Where "x" denotes the smoothed volumes, "+" denotes the measured length

The length increase in going from one frame to another is typically 0 8mm and the order of the error in digitising may be in the region of 0.5 mm, possibly much larger The digitisation error occurs in two main areas. Firstly, the X-ray is digitised on a mesh with 0.1 mm divisions so there is a 0.1 mm error from this. Secondly, and possibly a much larger source error occurs when attempting to distinguish the endocardium outline from the myocardium and the blood stained with the contrast medium. For the sake of argument the error from this is assumed to be 0.15 mm. This value not meant to be an exact value but just a value to enable some idea of the effect of possible errors in the length calculations. This level of uncertainty obviously causes serious problems The only solution is to smooth through the pressure/length data. This smoothing was accomplished by linear regression with the added constraint that the line must pass through the first point.

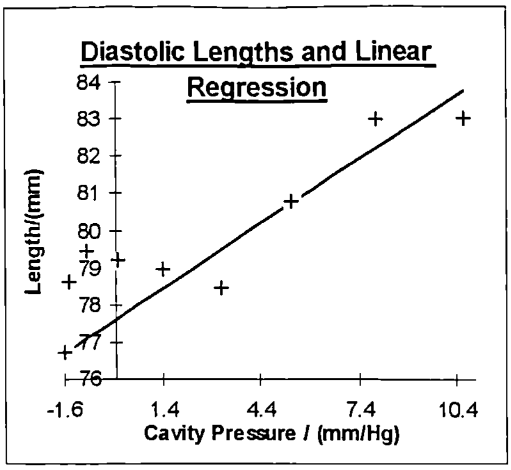
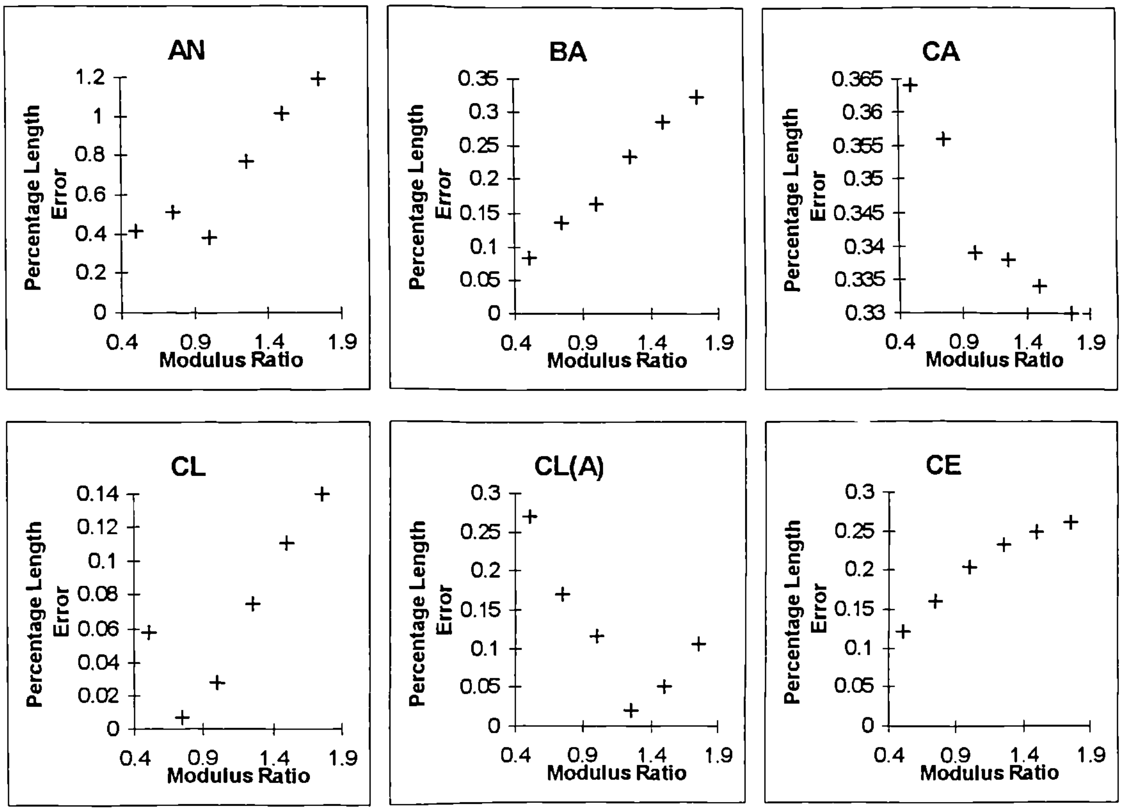
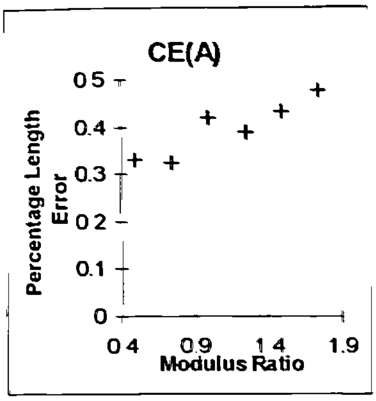


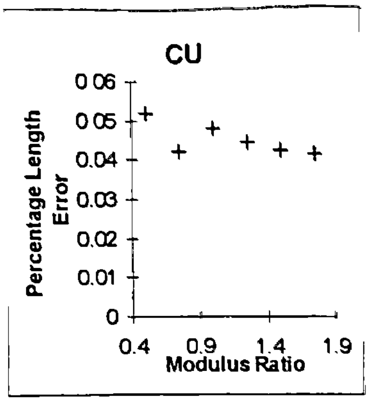
Fig 6.13 This graph shows how the pressure/length data for patient "BA" is smoothed through using linear regression.

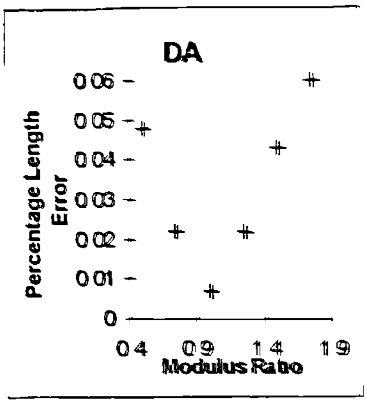
Linear regression was used since, as the length increase was small and the possible digitisation errors were relatively large, it would have been difficult to justify any other form of smoothing for the data. It was also the easiest way to ensure that the smoothed lengths for the matching process were always monotonically increasing.

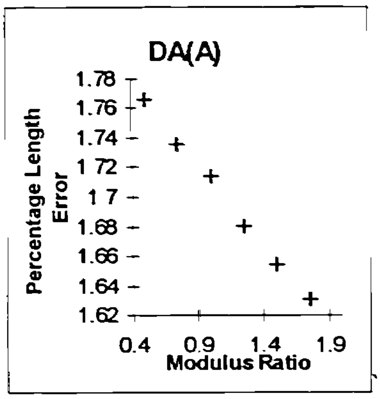
The volume matching process was then applied to the first frame of each patient with various modulus ratios in the range 0.25 - 2.0.

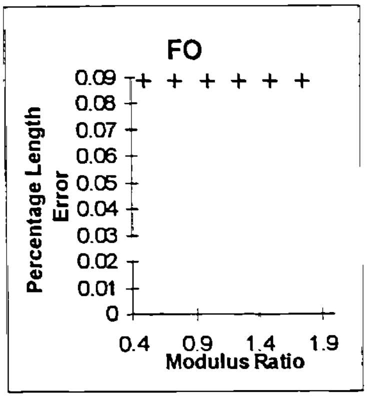


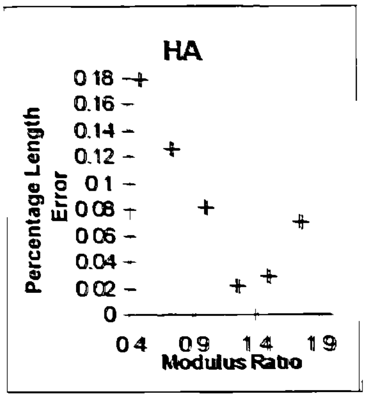

CE(A)
Percentage Length Error
0.5
0.4
0.3
0.2
0.1
0
0.4
0.9
1.4
1.9
Modulus Ratio

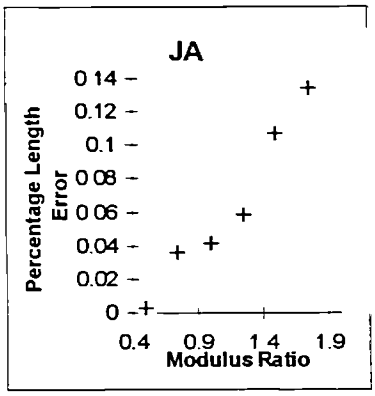

CU
Percentage Length Error
0.06
0.05
0.04
0.03
0.02
0.01
0
0.4
0.9
1.4
1.9
Modulus Ratio

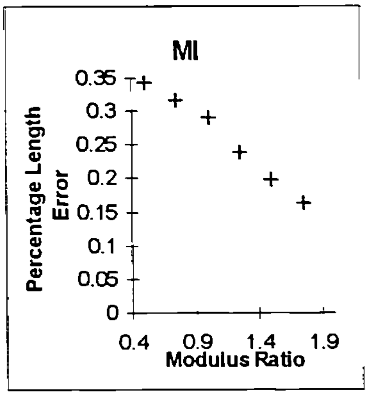

DA
Percentage Length Error
0.06
0.05
0.04
0.03
0.02
0.01
0
0.4
0.9
1.4
1.9
Modulus Ratio

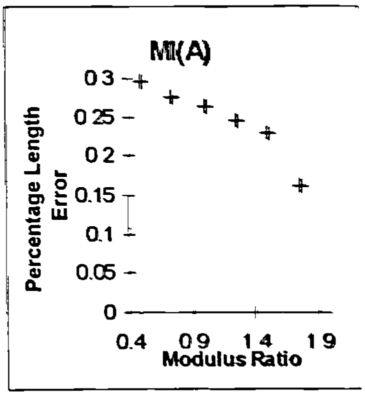

DA(A)
Percentage Length Error
1.78
1.76
1.74
1.72
1.7
1.68
1.66
1.64
1.62
0.4
0.9
1.4
1.9
Modulus Ratio

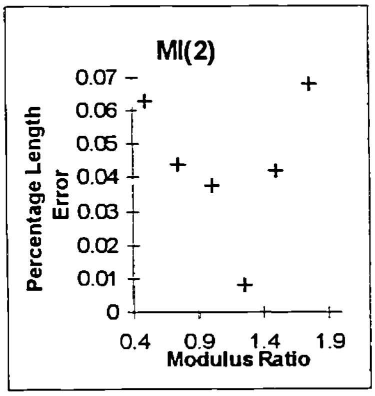

FO
Percentage Length Error
0.09
0.08
0.07
0.06
0.05
0.04
0.03
0.02
0.01
0
0.4
0.9
1.4
1.9
Modulus Ratio

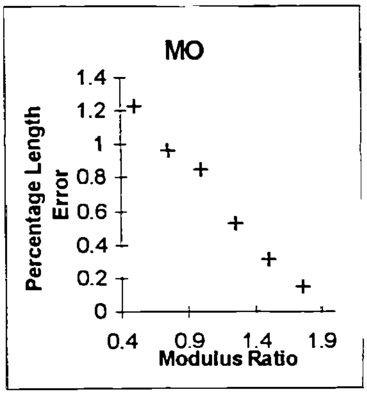

HA
Percentage Length Error
0.18
0.16
0.14
0.12
0.1
0.08
0.06
0.04
0.02
0
0.4
0.9
1.4
1.9
Modulus Ratio

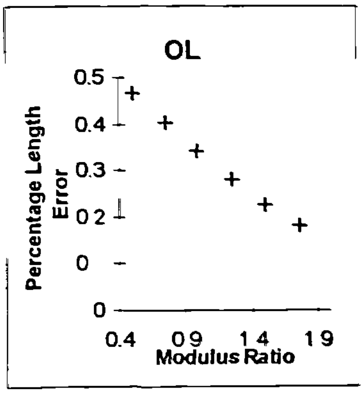

JA
Percentage Length Error
0.14
0.12
0.1
0.08
0.06
0.04
0.02
0
0.4
0.9
1.4
1.9
Modulus Ratio


MI
Percentage Length Error
0.35
0.3
0.25
0.2
0.15
0.1
0.05
0
0.4
0.9
1.4
1.9
Modulus Ratio


MI(A)
Percentage Length Error
0.3
0.25
0.2
0.15
0.1
0.05
0
0.4
0.9
1.4
1.9
Modulus Ratio


MI(2)
Percentage Length Error
0.07
0.06
0.05
0.04
0.03
0.02
0.01
0
0.4
0.9
1.4
1.9
Modulus Ratio


MO
Percentage Length Error
1.4
1.2
1
0.8
0.6
0.4
0.2
0
0.4
0.9
1.4
1.9
Modulus Ratio


OL
Percentage Length Error
0.5
0.4
0.3
0.2
0
0
0.4
0.9
1.4
1.9
Modulus Ratio

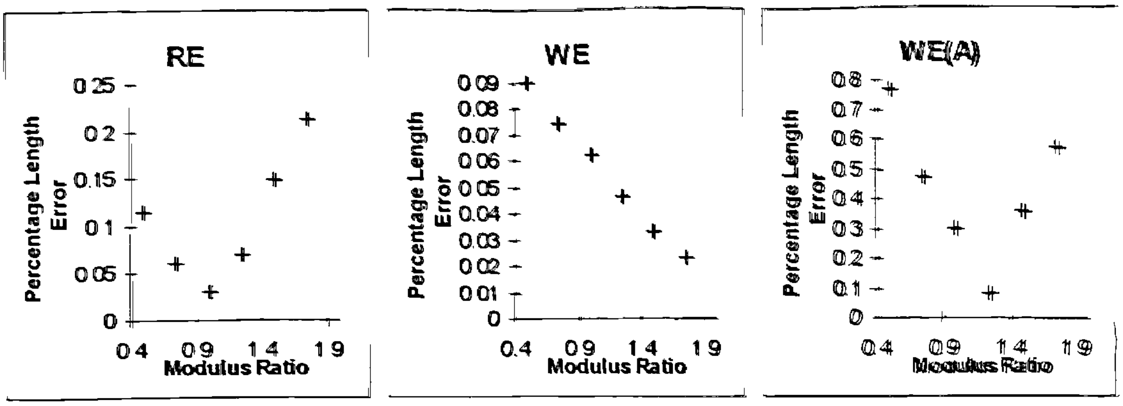


Fig 6 14 These graphs show the percentage absolute error between the calculated length of the model and the target value obtained by linear interpolation.

The results from this appeared to suggest that a modulus ratio of between 0 75 - 1 25 would allow a length and volume match for most of the patients The length was matched to the target length obtained from linear regression of the pressure/length data When an attempt was made to try to match subsequent frames, using the length and volume matched ventricle geometry, this became impossible, as the length increases necessary to achieve a match could not be attained. The early low values for the modulus ratio were artefacts generated as a result of smoothing through the pressure length data

6.4 Start and End Frame matching

It is obvious that the errors that occur in the process of digitisation make it impossible to obtain meaningful values of the modulus ratio between each frame of diastole An attempt was made to match the volume and length of the first frame of diastole with that of the last frame. In doing this the lack of accuracy in digitising become much less significant since the length increase would become greater than the assumed error in the lengths due to digitising. Table 6.1 shows the values of the major and minor elastic moduli plus the modulus ratio for the diastolic start and end frames.

Patient	Minor Modulus	Major Modulus	Modulus Ratio
AN	6.90	7.30	1.06
BA	29.54	99.77	3.38
CA		Not Possible	
CL	11.32	22.63	2.00
CL(A)	4.21	92.70	22.02
CE	899.76	9.00	0.01
CE(A)	205.75	88.23	0.43
CU	7.37	10.46	0.27
DA	3.26	4.62	1.42
DA(A)	10.18	81.43	8.00
FO	12.75	38.47	3.02
HA	17.95	97.23	5.42
JA		Not Possible	
MI		Not Possible	
MI(A)		Not Possible	
MI(2)		Not Possible	
MO		Not Possible	
OL	6.09	51.26	8.42
RE		Not Possible	
WE		Not Possible	
WE(A)	2.91	7.28	2.50

Table 6.1 Effective elastic moduli for the complete diastolic period with ventricular length matched.

As can be seen from the table there is a large spread in the results, the highest value for the modulus ratio being 22.02 and the lowest 0.01. It was not possible to obtain results from patient data "CA", "JA", "MI", "MI(A)", "MI(2)", "RE", and "WE" since the lengths to be matched were not in the range of values that could be achieved by altering the modulus ratio. The mean value from the above data is 4.5. This is somewhat higher than the traditionally assumed value of 2, and the standard deviation is 5.7.

It is important to remember that results obtained are liable to be skewed towards higher values for the modulus ratio, since errors that require the ventricle to be increased in length more than necessary will cause the modulus ratio to increase far more than the reverse would cause the modulus ratio to be reduced. Thus a value of 2 or even 1.3 may not be a totally unreasonable estimate for the value of the modulus ratio. Unfortunately without far greater accuracy in data acquisition it is not possible to gain meaningful values for the modulus ratio.

6.5 Effect of Digitisation Errors Upon the Modulus Ratio

The questions arising from the previous section are whether:

- the values obtained are consistent with each other
- there is evidence that values vary between patients
- there is evidence for some other factor affecting the length of the ventricle during diastole.

If the digitising process is assumed to be accurate to within 0.25mm one digitised frame could be as much as 0.5mm longer or shorter than the original X-ray frame, since the highest point and the lowest point may both be as much as 0.25 mm higher or lower than those on the original X-rays. The start and end frame of diastole may thus have a combined error of as much as ±1 mm. The length matching process was repeated for all the patients with the length to be matched increased and also reduced by this "maximum" error value.

Patient	Minor Modulus	Major Modulus	Modulus Ratio
AN	26.04	3.00	0.12
BA	38.94	77.88	2.00
CA		Not Possible	
CL	14.07	17.45	1.24
CL(A)	8.86	66.63	7.52
CE		Not Possible	
CE(A)		Not Possible	
CU	16.82	4.61	0.27
DA	3.71	3.71	1.00
DA(A)	19.00	55.50	2.92
FO	16.49	29.16	1.77
HA	26.12	77.29	2.96
JA		Not Possible	
MI		Not Possible	
MI(A)		Not Possible	
MI(2)		Not Possible	
MO	24.93	299.16	12.00
OL	7.96	43.17	5.42
RE		Not Possible	
WE	8.35	66.79	8.00
WE(A)	3.35	6.13	1.83

Table 6.2 Modulus ratio obtained from matching to the target length -1.0mm.

Patient	Minor Modulus	Major Modulus	Modulus Ratio
AN	6.78	7.95	1.17
BA	17.93	131.29	7.32
CA		Not Possible	
CL	6.98	32.76	4.69
CL(A)		Not Possible	
CE	201.42	125.29	0.62
CE(A)	71.72	228.65	3.19
CU	4.72	17.35	3.68
DA	2.68	5.90	2.20
DA(A)		Not Possible	
FO	7.45	59.59	8.00
HA	10.96	119.90	10.96
JA		Not Possible	
MI		Not Possible	
MI(A)		Not Possible	
MI(2)		Not Possible	
MO		Not Possible	
OL	4.05	64.75	15.99
RE		Not Possible	
WE		Not Possible	
WE(A)	2.28	9.13	4.00

Table 6.3 Modulus ratio obtained from matching the target length +1.0mm

There are several points raised by the above results:

- Firstly, even allowing ±1.0mm error between the target length and the actual length, it is still not possible for all the patients to obtain a value for the modulus ratio by matching the ventricular length. The implication of this is that either the digitisation errors are far greater than those assumed or there is some other factor present that does, in certain patients, greatly affect the change in ventricular length during diastole.
- Secondly, a value for the modulus ratio cannot be chosen which would be within the range of values calculated, for all the patients when the error is assumed to be ±1.0mm
- Thirdly, when the length to be matched is increased this tends to affect the modulus ratio obtained to a greater extent than reducing the length to be matched. The mean value for the modulus ratio when the length to be matched is increased by 1mm more

than doubles from 3.0 to 6.5. When the length to be matched is reduced by 1mm the mean modulus ratio becomes 1.8. Note that to calculate these mean values only those patients for which a value for the modulus ratio could be obtained in all three cases are used. This is a further indication, that when obtaining a mean value for the modulus ratio, there is a tendency for the value to be an overestimate, as errors that result in an increase in the length to be matched have a greater effect than those that result in a decrease.

6.6 Programming Enhancements

It was necessary to make several changes and extensions to the computer program above and beyond the obvious ones needed for the inclusion of the length matching process.

The data required for the finite element calculations are all stored in the file "XL1-data". The major and minor elastic moduli are stored here rather than the minor modulus and the modulus ratio. In order to represent more accurately the modulus ratio, the accuracy to which these values are stored was increased from five to seven significant figures. The format that the pressure values are stored in was also changed to allow for larger pressure increases between matched frames. This was required when matching start to end diastolic frames.

Since the material properties matrix must be symmetric it is necessary for two of the Poisson's ratios to be equal to the other divided by the modulus ratio. Either, one can have its value fixed and the other two set to that value divided by the modulus ratio or, two are set to some value and the other is given this value multiplied by the modulus ratio. Originally in the program the former of these was used. There are however constraints upon what values Poisson's ratio can take. For instance it cannot be greater than one in absolute magnitude. Another constraint is that they must also satisfy the below inequalities,

$$\nu_{12}\nu_{23}\nu_{31} < \frac{1}{2}(1-\nu_{12}\nu_{21}-\nu_{23}\nu_{32}-\nu_{31}\nu_{13}) < \frac{1}{2}$$

In order for these inequalities always to be true, it is necessary, when the modulus ratio is below 1, for the cross-fibre Poisson's ratio (ν_{31}) to be set and the two along-fibre Poisson's ratios (ν_{12} and ν_{23}) to take this value multiplied by the modulus ratio Thus the computer program has been altered to calculate the Poisson's ratio in the appropriate manner depending upon the value of the modulus ratio. It may seem strange that the modulus ratio should be below 1, since this would be extremely unusual for a fibred material, where the along-fibre elastic modulus is usually greater than the cross-fibre modulus. However in this analysis a value below 1 is sometimes needed when attempting to match the ventricular length. From experimentation it has also been shown that some specimens of myocardium appear to possess a modulus ratio below 1 (Demer and Yin [62]).

Two new subroutines were necessary to perform the length matching process A subroutine "SMOOLEN" returns an array containing the target lengths for each frame. If start to end diastolic frame matching is being performed then only the length at end diastole is returned in the array. The length matching process is performed by the subroutine "LENMATCH", this routine returns the value for the next estimate of the modulus ratio and also tests for convergence. The method used t obtain the next estimate of the modulus ratio was a linear step method followed by quadratic interpolation. This was found to be significantly faster than a linear step method followed by a Fibonacci search. The quadratic interpolation is of most use when matching lengths frame to frame throughout diastole, as this method is extremely effective at minimising the difference between calculated and target length. Though this was not possible with the data used at present, it has been kept as an option in the hope that more accurate data may be obtained in the future allowing this process to be used.

6.7 Conclusions

The main conclusion of this Chapter is that, in order to ascertain whether the modulus ratio is constant or varying through the diastolic cycle and what value or range of values it takes, far greater accuracy in the digitisation is needed. The most important points for the length and volume matching process are the base and apical points. An increase in an order of magnitude for the accuracy of these points should be sufficient to enable length and volume matching to a reasonable tolerance. This is not a simple task since most of the error here is likely to be due to the fact that the ventricular outline is difficult to distinguish from the X-ray data. However with the advances in powerful imaging techniques such as Magnetic Resonance Imaging, this may no longer be the problem that it was a few years ago. Magnetic Resonance Imaging has become a more frequently used tool in recent years as it has several advantages over more traditional methods such as ultrasound and X-rays.

There is also the possibility that the modulus ratio may not be the only or even the most significant factor contributing to the ventricle length. It may also be the case that pericardial pressure may significantly affect the way in which the ventricle lengthens during diastole. In a paper written by Han *et al.* [39] it was suggested that pericardial pressure played a significant role in ventricular performance during diastole and to completely ignore this would be detrimental to any model. The pericardium and parietal pericardium may also restrict ventricular expansion in a non-uniform way.

From equations 6.12 it would also appear that the shear modulus may be more important to global ventricular deformation than the value of the modulus ratio.

To summarise:

- Without extremely accurate length information, a tolerance of below 0.1mm should be sufficient. It is not possible to obtain accurate values for the elastic modulus ratio by matching both the length and volume of the left ventricle.
- Also the likely significance of pericardial pressure on the length change of the ventricle could not easily be qualitatively or quantitatively assessed without more accurate data.
- It may also be the case that the modulus ratio is not a dominant parameter in the global deformation of the left ventricle.

Chapter 7

7. Inclusion of Muscle Fibre Contraction

In this Chapter we are now looking to expand on our model. In the past, from a structural point of view, the myocardium had almost always been treated as passive. This is widely accepted to be a reasonable model for determining the diastolic properties of the myocardium, as during diastole the ventricle can be assumed to be completely passive, i.e. it deforms under an increase in internal pressure. This assumption is not applicable when one wishes to investigate the left ventricle in systole. Here the myocardium generates internal stresses by the contraction of the muscle fibres. The net result of this is that the cavity volume decreases even though the cavity pressure increases. This is a situation which cannot possibly be accounted for without the introduction of some active contraction mechanism into the model.

To simulate left ventricular systole the finite elements must be able to simulate the contraction of the myocardium. That is, the finite elements must shorten along the direction of the muscle fibres and must widen in the cross fibre direction. The method to be used here to simulate self activation of cardiac muscle is that proposed by Rosen (1968 [74]). This involves an analogy with that of thermal stressing, where internal strains in a structure are generated in response to a change in temperature. In order to simulate the contraction in the fibre direction and the corresponding thickening of the fibre it is necessary for the pseudo coefficients of thermal expansion to have opposite sign in the fibre direction and the across fibre direction. An implementation of this pseudo thermal method had previously been made by Kwak *et al.* (1977 [75]) and it was anticipated that, with some extension, their program known as "GTQSA" could be used in the systolic analysis of the available patient data.

7.1 Revival of the FE Program "GTQSA"

The program developed by Kwak *et al.* [75] had several features which made its use in this work desirable. Firstly, it included the needed thermal stressing type activation code. Secondly, it had built in stability analysis and thirdly, it had significantly greater documentation than did the presently used finite element code. It did however have several drawbacks; the program was only available in the form of a listing in the appendix of the report. This had previously been typed up from this into an electronic form, although as is to be expected it contained many typographical errors. Secondly, it did not include the varying fibre angle through an element or even the ability to specify fibre direction in local coordinates. The fibre direction could only be described in terms of the global coordinate system. Thirdly, the program gave only the ability to specify elements of hexahedral topology, whereas specification of the apex of the ventricle requires wedge shaped elements. Despite these drawbacks this program was considered to be the best choice, as future development would be more easily accomplished due to the availability of documentation for the program.

The problem of typographical errors would be a major task to eliminate. The situation was made worse due to the poor quality reproduction of the program listing within the report. The difficult task of tracking down the typographical errors was begun by compiling the program using the "-u" and "-C" options of the SunPro Fortran compiler. The "-u" option turns off implicit typing of variables. Implicit typing of variables is a process whereby the type of variable, either integer or real, is decided by the first letter of the variable's name. The use of implicit typing is a false economy as it makes it easy for typographical errors in variable names to go unnoticed. The net result of using this option is that error messages for non-declared variables are reported at compilation. This makes it relatively easy to track down typographical errors in variable names, that is of course provided the misprint is not itself a valid variable name. Implicit typing of variables had been used by the author in the program so it was first necessary to explicitly declare all variables.

The use of the "-C" option gives compilation errors or run time errors for arrays when the subscripts of an array are not within their defined limits. It should be noted that subscripts attached to an undefined array automatically give a compilation error.

Once all the compilation errors had been removed it was necessary to devise a simple test element that could be used to find the more obscure typographical errors. The most simple element that the program could accept was of an 8-noded isoparametric brick type. After a number of further corrections, mainly to the calculation of the shape functions, the program was able to calculate the nodal displacements for this element subjected to thermal stress and mechanical loading. The program allowed the specification of an even number of nodes per element within the range 8 to 20. Elements with 12 and 14 nodes were also tested and found to give the expected values. These elements were tested next as there were results within the report obtained using elements of these types.

Contained within the report were some tabular results from the application of the program to a two layer eight element orthotropic cylindrical model. The data file for this was contained within the report, although it was not fully readable. With the aid of a description of the model and a description for the using the program a complete data file was produced. The program appeared to run correctly and the results were of the correct order of magnitude. They did not however correspond to the values given within the report. The situation was made more uncertain as the description of the cylindrical model and the material properties used did not correspond to those in the data file given in the report's appendix. Despite painstaking subroutine by subroutine checking and the correction of several further errors the situation was not rectified. It was obvious that a different approach would be required.

The report contained a test model used to show that the finite element program worked correctly. This took the form of a cylinder identical to the above with the exception that it was isotropic and was only subjected to an internal load. This model could be easily solved using the previously used F.E. program "XL1". The advantage of this is that the values of the variable at each stage of calculation could be compared between "GTQSA" and "XL1" and the point at which these values diverged would signify where the bug lay. Upon running this isotropic model it was immediately evident that there was a problem, the results from "XL1" for the radial expansion agreed with those of the report however the length reduction did not. The report gave the length reduction as 0.1mm whereas "XL1" calculated the length reduction as 0.3mm. It was necessary to seek independent validation as to which value was correct. Consulting Roark and Young [76] the analytic solution was found to be 0.3mm.

In the light of these revelations it seemed unwise to continue further with a program that was still not fully working and there were now some doubts over its accuracy. Despite the fact that over four months had been spent working on this program not all the time had been wasted as a greater understanding of the implementation of thermal stressing/muscle contraction had been gained.

7.2 Adaptation of F.E. Program XL1

With the failure of the finite element program, "GTQSA", to perform as expected it left adaptation of the originally used finite element program, "XL1", as the next course of action. From the work with "GTQSA" it had been realised that inclusion of thermal stressing was relatively straight forward. It required the nodal forces that resulted from the simulated muscle contraction being calculated and added to the passive mechanical forces due to internal pressure loading. The solution of the finite element problem then continues as before with the forces, derived from the simulation of muscle contraction, being treated as though they were ordinary passive nodal forces.

The only alteration to the finite element program that was necessary was to create subroutines that would calculate the forces generated by the muscle contraction The calculation of the force generated by muscle contraction is described below with a more detailed description available in Chapter 8.

7.2.1 The Calculations Required For Muscle Contraction

The method used to simulate muscle contraction is computationally identical to anisotropic thermal stressing. The reasoning behind this approach and the calculations that are necessary for this are explored below.

In thermal stressing a piece of material is subjected to a thermal load, either a temperature increase or a temperature decrease. The result of this is that the material will generate internal stresses due to differential deformation. The stress/strain equation for a one dimensional fibre is given below.

$$\sigma = E(\varepsilon - \alpha T) \qquad \text{Eqn. 7.1}$$

The working for the case of a two dimensional lamina is performed by Rosen [74]. Here however we shall content ourselves with the one dimensional case at this stage. In muscle contraction the stresses are generated within the muscle fibres by the shortening of the sarcomeres. We shall assume that there exists a function, F(t), that gives the strain produced, with time, by a piece of contracting muscle. The stresses generated within the myocardium in general will be partly due to this contraction and partly due to pressure from the blood within the ventricular cavity. An equation representing this process is given below.

$$\sigma = E(\varepsilon - F(t)) \qquad \text{Eqn. 7.2}$$

In practice the function F(t) which gives the strain rate of a muscle fibre depends on many physiological and historical factors relating to the fibre. But in this case we are only concerned with propagation of strain with time and not the other contributing factors.

Both equations are relatively similar and even though F(t) is an unknown function it can be replaced by the numerical approximation C_p^i. This is a numerical approximation of F(t) at time t=i.

$$\sigma = E\left(\varepsilon - C_p^i\right) \qquad \text{Eqn. 7.3}$$

The equations for thermal stressing and muscle contraction are now quite similar and the analogy easily understood. To preserve the analogy with thermal stressing each point in the mesh is assigned a "temperature" of minus one and the value of C_p^i is varied to achieve the required material contraction. To preserve the wall volume of the model it is necessary that the fibres thicken as they contract. This is achieved by applying a cross-fibre expansion stress which is related to the along fibre contraction by some ratio, r, which must also be determined.

$$\sigma_c = E_c\left(\varepsilon_c - rC_p^i\right) \qquad \text{Eqn. 7.4}$$

This results in equations for contraction with two unknowns and thus we have a two-variable problem. In the three-dimensional case there will be two cross-fibre stress equations. For the sake of simplicity the amount of active strain generated is assumed to be the same in all cross-fibre directions. It is also likely to be the case that the fibres expand relatively uniformly about their cross-section. It should be noted at this stage that the shear stress component resulting from the cavity pressure loading is not directly affected by the fibre contraction. This is as the fibre is assumed to only generate longitudinal and cross-sectional stresses. The finite elements will however experience a shear component as a result of the fibre contraction due to the course of the fibre direction changing through the depth of the element.

To calculate the nodal forces generated by the muscle contraction required the creating of a number of new subroutines, the function these perform will be discussed here An in-depth description of the function of the new subroutines can be found in Chapter 8

The basic code for muscle contraction is based on that of "GTQSA". Each finite element node has an associated "nodal temperature" or "contraction potential". This allows for variation in the force of contraction from element to element as well as within an element. It is not presently envisaged that the use of this variation will be required, however once implemented it provides for future flexibility. Using the element's shape functions the contraction potential is calculated at each integration point. The value of this is constant throughout the element provided the contraction potential is the same at each node.

The material also has three other properties defined for it; the contraction coefficient in the fibre direction and the two orthogonal cross fibre expansion coefficients which as stated above are taken to be the same as each other. The cross-fibre contraction coefficient is related to the along-fibre contraction coefficient by the contraction coefficient ratio. Presently this ratio is set at 1:-0.5, this means that the cross-fibre contraction coefficient will be half the size and of opposite sign to the along-fibre one This ratio of 1:-0.5 was suggested by Kwak *et al.* (1977 [75]). When these contraction coefficients are multiplied by the contraction potential at an element's integration point a strain tensor is formed. This tensor defines the strain generated at that point in the material due to muscle contraction. Since we require the forces generated at this point and not the active strains, the strains are converted into stresses by multiplication with the compliance matrix for the material. The contraction at our integration point is now defined in terms of the stresses necessary to generate the required contraction The integration point however does not correspond to an elemental node, and thus the forces required to produce this integration point value are calculated at each element node by using the element's "strain shape function" matrix.

This is how the contraction forces are calculated for an element. There is however more than this to simulating muscle contraction. Another important part is the orientation of these forces in 3-D space since the contracting material is simulating muscle fibres of non-uniform direction.

7.2.2 Fibre Direction Calculation and Correction

Here we shall discuss how the strains calculated for the contracting material are orientated in three dimensional Cartesian space. The calculation of the fibre orientation is a two fold process. Firstly, the orientation of the fibre relative to the idealised element form, (brick or wedge) is calculated and the strains are then transformed using a transformation matrix. This transformation matrix is formed by considering the directional cosines associated with the specific fibre directions. The strains are now orientated in the local element coordinates. Secondly, since the element is unlikely to be an idealised shape a transformation into Cartesian coordinates is also necessary. The strains in local coordinates are then multiplied by another transformation matrix to convert them into Cartesian coordinates. This transformation has an identical form to the previous one. This time however the matrix is produced using the inverse Jacobian of the finite element at each integration point. Both of these transformation matrices are applied to the compliance matrix so that the strains can be easily converted into stresses.

During the testing and validation of the wedge shaped elements it became apparent that an oversimplification had previously been made in the calculation of the fibre direction within the material. This was more obvious when considering the wedge shaped elements as historically they had always been taken as isotropic. When the transformation was applied to the isotropic material property matrix it should remain unchanged, this however was not the case. The problem and its solution are discussed below.

The problem had occurred due to the assumption that the inverse Jacobian would give an orthogonal set of axes representing the fibre direction at some point within the element. This is not the case, the inverse Jacobian gives the local elemental coordinates. These are not in general going to be orthogonal in Cartesian space in which the material properties are always defined. In this case the material properties and the fibres are not to be mapped from the local curvilinear coordinate system to the Cartesian system. The only transformation needed is that necessary to orientate the material property matrix within three dimensional Cartesian space.

To illustrate the problem consider the two dimensional element below,

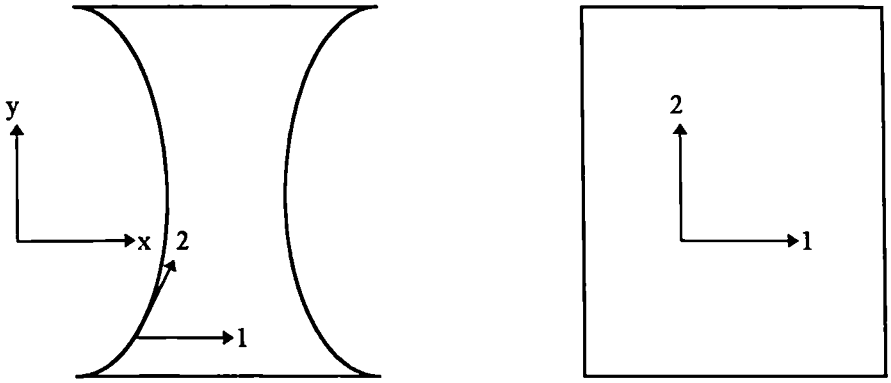


Fig 7.1 This depicts an element in 2-D Cartesian x,y-space (left) that has two parallel sides and two concave sides, along with the same element in local curvilinear 1,2-coordinates (right).

The fibres run in the elemental 2-direction and have a curved trajectory with respect to Cartesian space but are parallel to the curvilinear 2-direction. The inverse Jacobian would give the elemental 1-direction as being in the global x-direction. It should be obvious that the elemental 1 and 2 axes are not in general orthogonal in the global x and y coordinate system. The coordinate system for the fibre will not in general be the same as the local curvilinear elemental coordinate system. For the fibre the y-direction should be taken as the trajectory of the fibre and the fibre x-direction should be some direction orthogonal to this.

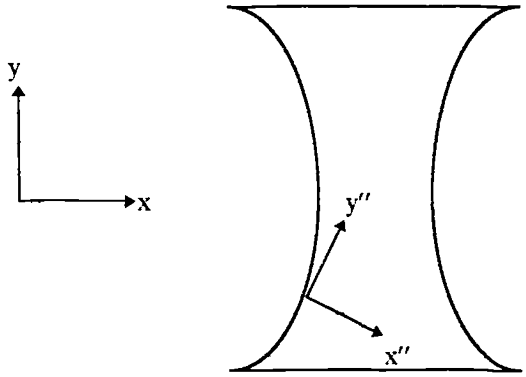


Fig 7.2 This shows the element in Cartesian space with the required fibre material directions superimposed upon one of the fibres within the element.

The solution to this problem in three dimensions is to set one direction given by the inverse Jacobian as one of the fibre axes and then make the other two orthogonal to it as well as mutually orthogonal. Previously the inverse Jacobian had been used as a set of vectors, $\underline{x}$, $\underline{y}$ and $\underline{z}$ which spanned Cartesian space. This they do, however they are not in general going to be mutually orthogonal. The first stage is to choose our fixed direction, this will be the y-direction as the convention in this study has been that the fibres run parallel to the y-axis. The vector $\underline{y}$ is now made into a unit vector by dividing each term in it by its magnitude. Next we must make the $\underline{x}$ and $\underline{z}$-vectors orthogonal to the vector $\underline{y}$. This is done by removing the y-component from the $\underline{x}$ and $\underline{z}$ vectors.

$$\begin{aligned} \underline{x}' &= \underline{x} - \underline{y}\left(\cos\left(\frac{\underline{x}.\underline{y}}{\sqrt{\underline{x}.\underline{x}}}\right)\right) \\ \underline{z}' &= \underline{z} - \underline{y}\left(\cos\left(\frac{\underline{z}.\underline{y}}{\sqrt{\underline{z}.\underline{z}}}\right)\right) \end{aligned} \qquad \text{Eqn. 7.5}$$

We now have two vectors $\underline{x}'$ and $\underline{z}'$ which are now orthogonal to the fixed y-direction. The only thing now left is to make the $\underline{x}'$ and $\underline{z}'$-vectors mutually orthogonal. They are firstly made into unit vectors and the angle between them is calculated. It is necessary for this angle to be 90°. The angular difference between the two vectors is then halved and each vector is then rotated by this amount plus $\pi/4$ in opposite directions. The two vectors are now orthogonal to each other as well as to the $\underline{y}$ vector.

$$\begin{aligned}\underline{\mathbf{x}}'' &= \underline{\mathbf{x}}' - \underline{\mathbf{z}}'\left(\cos\left(\frac{1}{2}\frac{\underline{\mathbf{x}}'.\underline{\mathbf{z}}'}{\sqrt{\underline{\mathbf{z}}'.\underline{\mathbf{z}}'}}+\frac{\pi}{4}\right)\right) \\ \underline{\mathbf{z}}'' &= \underline{\mathbf{z}}' - \underline{\mathbf{x}}'\left(\cos\left(\frac{1}{2}\frac{\mathbf{x}'.\underline{\mathbf{z}}'}{\sqrt{\underline{\mathbf{z}}'.\underline{\mathbf{z}}'}}+\frac{\pi}{4}\right)\right)\end{aligned} \qquad \text{Eqn. 7.6}$$

This process will always yield an orthogonal set of fibre axes provided the initial set of vectors given by the Jacobian span Cartesian space. The only case when they will not do so is if the element is incorrectly formed in which case a finite element solution would be impossible.

7.2.3 Testing and Validation

It is always wise to perform comprehensive testing of any program or program modification. Presented here is a summary of some of those test procedures which were carried out.

The simplest test is that of a cube with its vertices corresponding to those of the finite element fundamental. This is particularly simple since the Jacobian will be the unit matrix. The fibre direction for this element was taken to be in the elemental y-direction. The element was found to deform as expected for a variety of contraction coefficients. As a further check the finite element analysis was carried out for the element in various orientations to the Cartesian axes. The element was also split into two small elements to check for the correct handling of multiple elements.

The next stage was to expand upon this simple element by having the fibre direction at some angle to the elemental y-direction. Since the results from this test are not so intuitively obvious the test needed to be compared against another package. For this purpose all the following tests were compared against the results from the finite element package PAFEC™. The results for various angles were identical between the two programs.

It must also be shown that the program can deal with non-cubic elements. To this end an element which was a quarter of an annulus was created. There unfortunately is a drawback since PAFEC™ cannot handle local curvilinear coordinates. For the PAFEC™ comparison model ten elements were used to simulate the changing fibre direction as one moves around the arc of the quarter annulus. As is to be expected the results were not exactly the same, however the overall deformation was reasonably consistent between the two test models.

One further test also needed to be performed and that was for the varying fibre angle through the thickness of the block. Again PAFEC™ does not include facilities to perform this so a model of this element type was constructed in PAFEC™. This involved splitting a cube into six slices and having each slice with a slightly different fibre angle. The results from this were not exactly the same as from "XL1" but were reasonably close considering the two models were not identical in their structure.

Some sample results are included in Appendix I along with sample data files for both programs.

7.2.4 Preliminary Results Using a Cylindrical Model

A cylindrical model has been used here as a simple demonstratio that the basic requirements of the left ventricle during systole can be modelled with the process described above. A model of the left ventricle must be able to produce a contracting cavity volume despite an increasing internal pressure. The cavity volume and pressure changes are taken from patient "BA", and the values of the contraction coefficients are found such that the volume changes in the cylindrical model are similar to those for the "BA" data.

	Frame Number Starting at the Onset of Systole								
	0	1	2	3	4	5	6	7	8
Cumulative Pressure/(mmHg)	0.00	6.1	9.9	12.4	14.2	16.0	18.5	21.3	22.9
Target Volume/(cm^3)	162	156	149	139	130	122	115	107	104
Calculated Volume/(cm^3)	N/A	156	149	140	130	122	115	107	103
Contraction Coefficient	0.00	0.11	0.20	0.29	0.38	0.46	0.54	0.64	0.69

Table 7.1 Table of pressures, volumes and contraction coefficients required to simulate the systolic part of the cardiac cycle for patient "BA" (frames 7-15) using a cylindrical model.

The model is an eight element two layers through the wall type. The angle of the muscle fibres is taken to be -60° at the external surface, 0° at the mid-wall level and 60° at the inner surface.

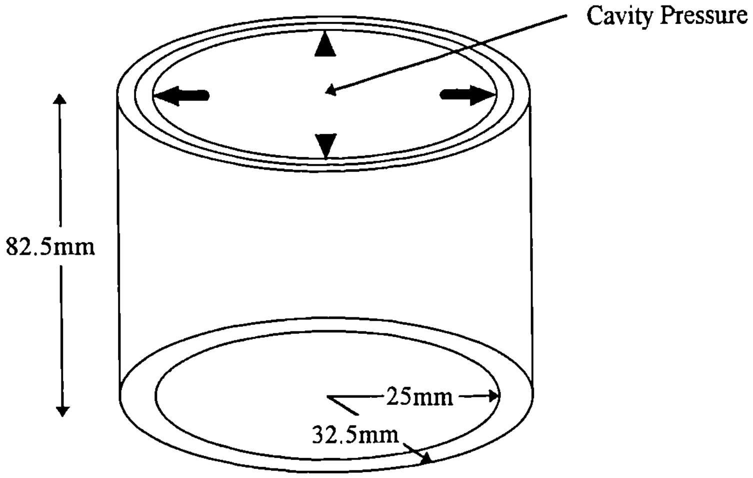


Fig 7.3 The cylindrical model used as an initial demonstration of the potential of the thermal stress analogy in modelling the left ventricle during systole.

	Radial Contraction/mm			Height Change/mm			Twisting/degrees	
Frame	Inner	Middle	Outer	Inner	Middle	Outer	Middle	Outer
1	0.59	0.51	0.414	0.742	0.716	0.669	0.106	0.230
2	1.23	1.05	0.85	1.45	1.40	1.32	0.199	0.434
3	2.04	1.74	1.43	2.27	2.21	2.08	0.300	0.659
4	2.92	2.47	2.02	3.14	3.06	2.90	0.409	0.911
5	3.63	3.06	2.48	3.91	3.81	3.62	0.511	1.15
6	4.20	3.50	2.80	4.62	4.50	4.27	0.619	1.404
7	4.89	4.03	3.18	5.53	5.39	5.12	0.762	1.759
8	5.19	4.25	3.31	5.97	5.832	5.53	0.837	1.942

Table 7.2 This is a table of the results obtained using the cylindrical geometry and the data of table 7.1.

The changes in shape of the model agree favourably with observations of ventricular deformation. The most important feature is that cavity volume is reduced during the application of the modelling process. It is also noticed that the wall thickens and that radial motion of the outer surface is less than that of the inner surface. Both these morphological changes agree with known ventricular systolic properties. It has also been observed that the ventricle during the cardiac cycle undergoes a significant amount of rotation. In effect the ventricle "winds up" during systole and "unwinds" during diastole. This "winding up" process is clearly evident from Table 7.2. The values for the rotation are however smaller than one might expect. The reason for this may be due to the material properties and the geometry used and even to the way in which the model is restrained.

The only real cause for concern is that the model tends to increase in length when a reduction would be expected. This however may be due to either the poor approximation of the ventricle by a cylinder or a poor choice for the material properties. The true reason for this and factors contributing to global deformation will be investigated in Chapter 9.

7.3 Summary

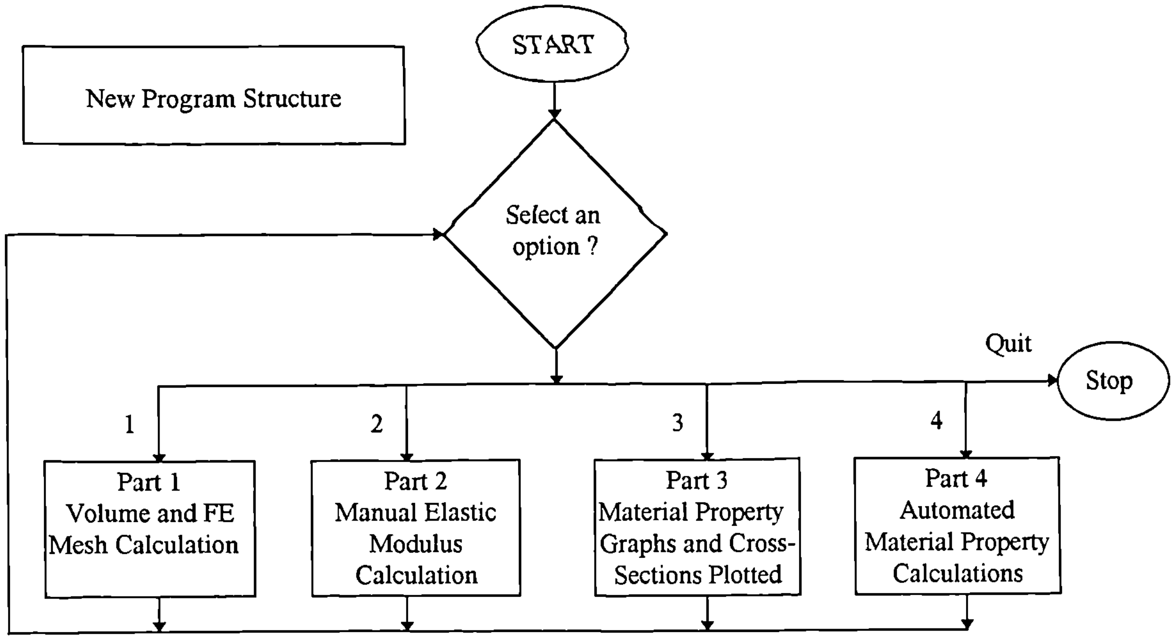
The process of simulating muscle contraction by a thermal stress analogy certainly does produce an effect similar to what would be expected during myocardial contraction. The cylindrical model used shows many of the important features of v ntricular systole. Firstly, the cavity volume reduces and can overcome an internal pressure. Secondly, the inner, middle and outer surfaces all show rotation relative to each other. The only concern is that the cylinder does not reduce in length as the left ventricle is generally known to do during ventricular systole. The reason for this will become apparent during analysis of the patient data in Chapter 9.

Chapter 8

8. An Overview of the Computer Program

This Chapter will be concerned with explaining the structure and function of the new automated program routines. Here the extensions that were necessary for the expanded functionality of the program will be discussed as well as the restructuring and the new menu system of the program. The program originally consisted of three parts and little reference to these will be made here. The full details of these can be found in Grewal [2]. Some alterations have been made in these parts of the program, however they are fairly superficial and do not alter the overall function of these parts. The alterations made were only to improve the clarity of the code and enable these parts to be more stable with respect to input errors. As such the flow diagrams that can be found in Grewal [2] Chapter 7 are still valid.

This Chapter will concentrate on the newly formed part of the program, "PART4", which controls the process of finding material properties. These properties can either be the passive diastolic properties or the active systolic properties.


New Program Structure
START
Select an option ?
Quit
Stop
1
2
3
4
Part 1 Volume and FE Mesh Calculation
Part 2 Manual Elastic Modulus Calculation
Part 3 Material Property Graphs and Cross-Sections Plotted
Part 4 Automated Material Property Calculations

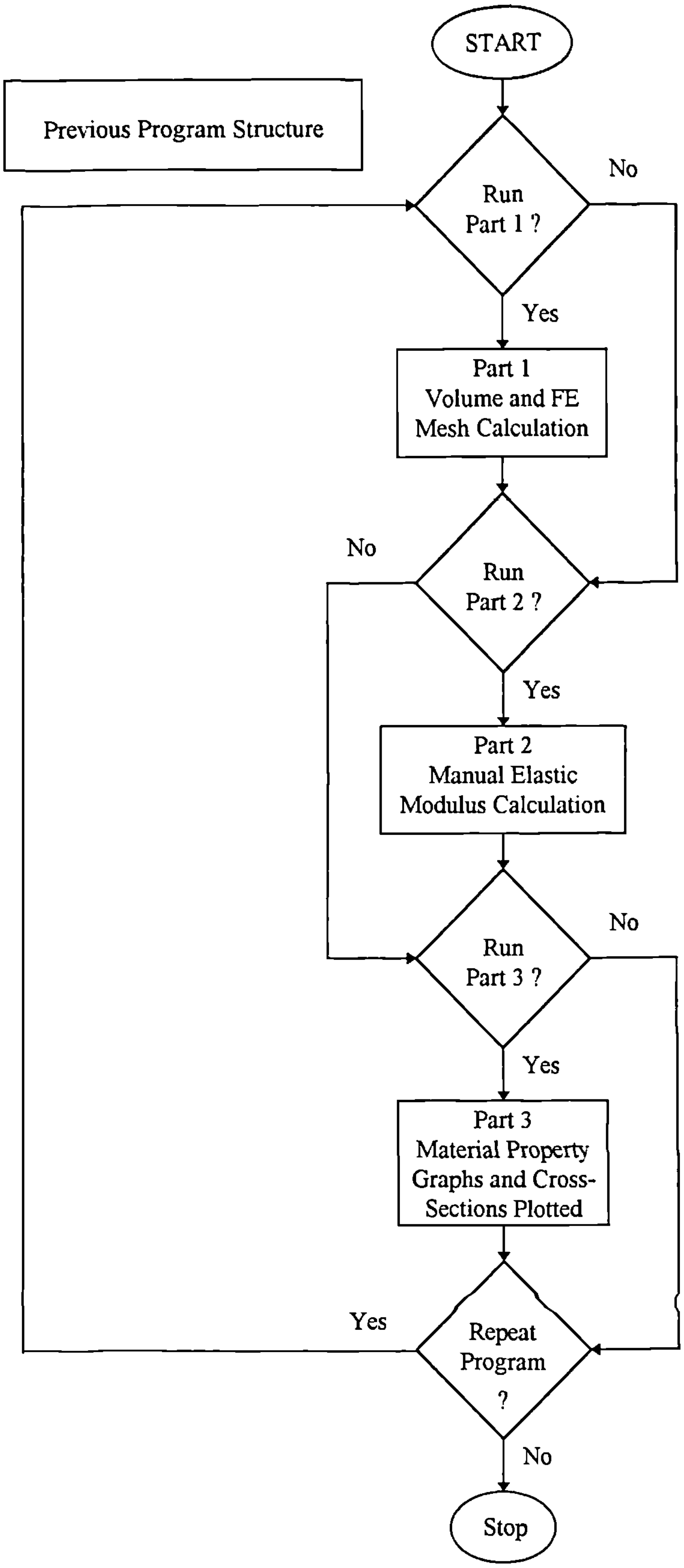

START
Previous Program Structure
Run Part 1 ?
No
Yes
Part 1 Volume and FE Mesh Calculation
No
Run Part 2 ?
Yes
Part 2 Manual Elastic Modulus Calculation
No
Run Part 3 ?
Yes
Part 3 Material Property Graphs and Cross-Sections Plotted
Yes
Repeat Program ?
No
Stop

Fig 8.1 These two flow diagrams illustrate how the parts of the program were originally accessed compared to how they are now accessed.

As can been seen each part of the program can now be directly accessed from the main menu. Whereas before the individual parts could only be accessed sequentially one after another.

8.1 The Original Function of the Program Heart

The program "HEART" was the post-processor and pre-processor for the finite element program "XL1". As stated above it was originally composed of three parts. The first part, "PART1", was used to obtain the volume and pressure information from the X-ray data. The pressure and volume graphs could also be plotted from within this part of the program. As well as this the data file, "XL1-data", which the finite element program requires is also generated within this part of the program. The mesh generated for the finite element analysis can also be viewed here to ensure that it is well formed and may also be saved to a separate coordinate file.

The second part, "PART2", deals with the matching process of finding the elastic modulus of the model required for the observed volume and pressure change. This is done interactively and can produce an updated data file with a new value for the elastic modulus.

The third part, "PART3", is responsible for most of the plotting routines. It allows for the plotting of the material property graphs and the deformed and undeformed short axis cross-sectional views. This part has been slightly enhanced by the inclusion of multiple graphs on a single page. A further set of plotting options has also b en included. This new option allows a three dimensional plot to be made of the deformed and undeformed model data. The endocardium and epicardium can be selectively plotted and up to three sets of data may be overlaid. The endocardium is plotted with solid lines and the epicardium with broken lines. Each set of coordinate data is also plotted with a different colour to make each set easily distinguishable from one and other.

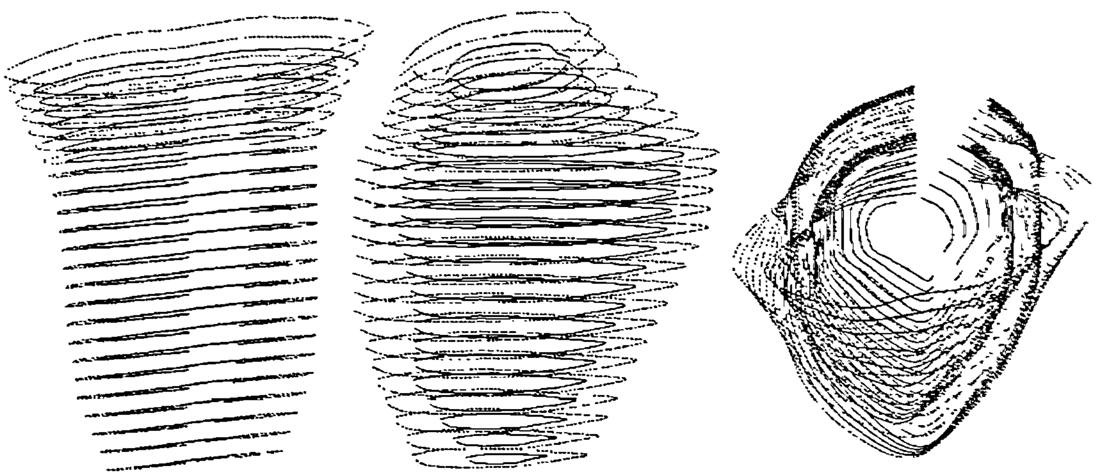


Fig 8.2 This shows the shape of a typical reconstruction. The endocardium is shown by the solid lines and the epicardium in broken lines. The same frame is shown at three different orientations.

The structure of this part of the program has changed slightly from that previously described by Grewal [2]. The changes are shown below and further information on the unchanged parts can be gained from Grewal [2] Chapter 7.

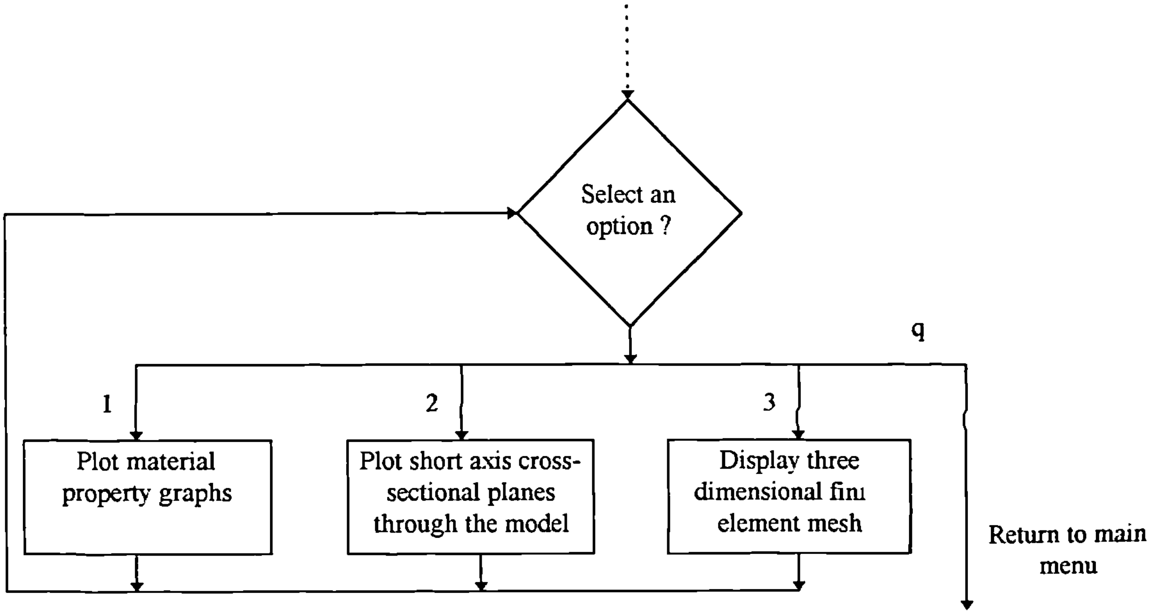

Select an option ?
q
1
2
3
Plot material property graphs
Plot short axis cross-sectional planes through the model
Display three dimensional fim element mesh
Return to main menu

Fig 8.3 This flow diagram shows the structure of the plotting routines. The plotting options are largely unchanged from the previous version of the program as described in Grewal [2] Chapter 7.

The use of the data file generating part of "PART1" and the whole of "PART2" are no longer needed due to the inclusion of "PART4". This part of the program has now taken over the formation and updating of the necessary data files. The only function of the previous two parts that is still required is the formation of the volume data file. This file contains the original cavity volumes as well as the smoothed and target volumes.

8.2 An Overview of the Structure and Function of "PART4"

Here we shall look at the overall structure of the automation part of the program. This shall be done by consideration of a flow diagram for this part of the program.

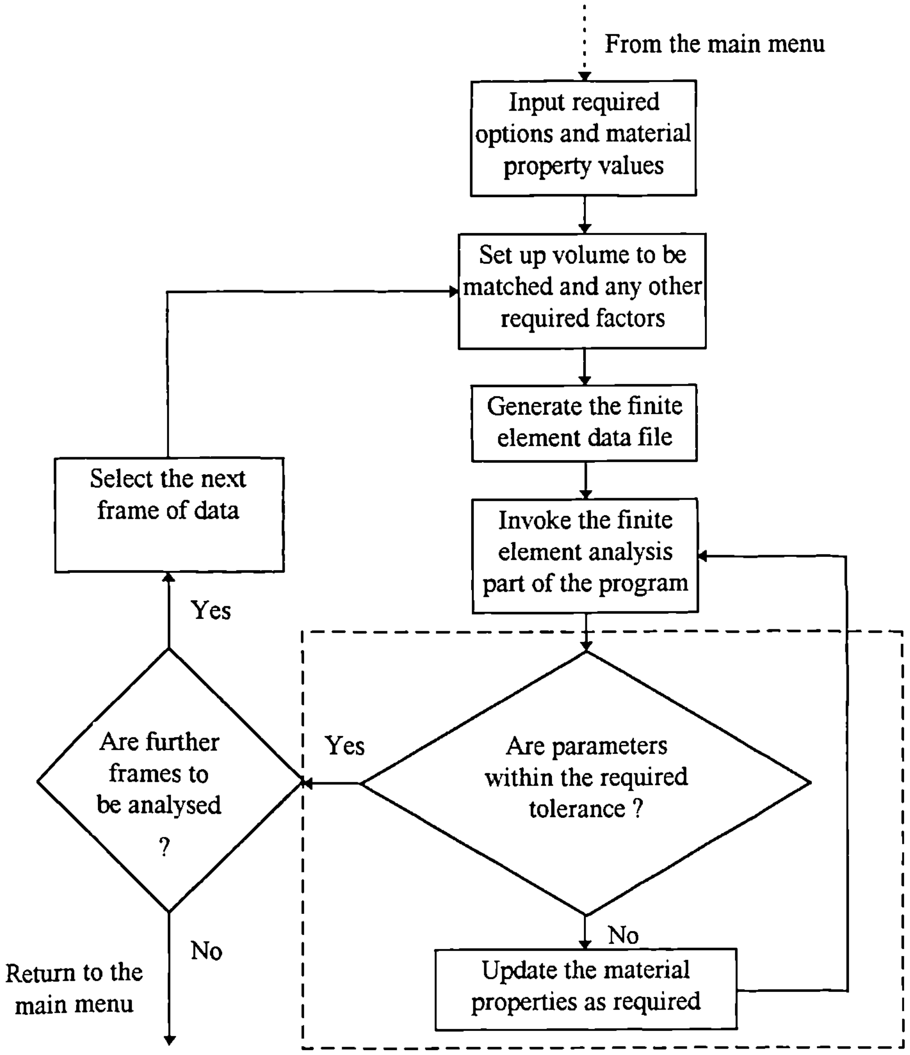

From the main menu
Input required options and material property values
Set up volume to be matched and any other required factors
Generate the finite element data file
Invoke the finite element analysis part of the program
Are parameters within the required tolerance ?
No
Update the material properties as required
Yes
Are further frames to be analysed ?
Yes
Select the next frame of data
No
Return to the main menu

Fig 8.4 This is a flow diagram which demonstrates how the material properties are found for the model using an automated process.

This diagram is applicable to all the various options that exist for the material property calculations within the program. The only part that requires any alteration from one option to another is that part which is enclosed in the large dashed box. The material properties which are altered in order to match the required parameters of the model will depend upon the type of model being used, either active or passive.

Further discussion of the contents of this boxed area are described below in separate subsections for the diastolic model and the systolic model.

8.3 Automation For Diastolic Work

A description of the automation process for the diastolic investigation was given in Chapter 3. Here we will expand upon that with a flow diagram.

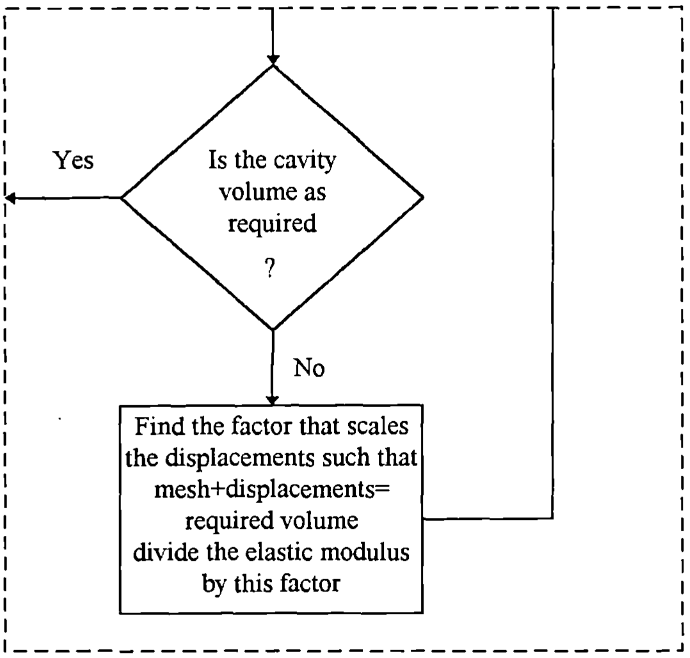

Yes
Is the cavity volume as required ?
No
Find the factor that scales the displacements such that mesh+displacements= required volume divide the elastic modulus by this factor

Fig. 8.5 This flow diagram shows how the material properties are found for the model which is produced from the X-ray and pressure data. (This flow diagram comes from the flow diagram for "PART4" of the program, Fig 8.4)

There are two options that allow for diastolic volume matching. The first is option "1", this is used to select "target" volume matching. The process involved in the matching of "target" volumes is described fully in Chapter 5. The second option is "2", this allows for smoothed volume matching, which is described fully in Chapter 3.

When the process is attempting to match the length of the model by varying the modulus ratio the problem is slightly more complex. From the work with matching only the cavity volume it is known that, if the finite element method is applied and then the displacements obtained are scaled so that the cavity volume is as desired, then the required elastic modulus is that obtained from reciprocal scaling of the elastic modulus previously used. Thus the elastic modulus is effectively found with one iteration. In order to match the length as well the modulus ratio must be altered too. From an initial investigation it was found that by increasing the modulus ratio the model would be made to lengthen when the cavity volume is again matched. Thus if we require a the model to be found that has a length greater than has been found with the presently used value then the modulus ratio is increased and vice versa. When values of the modulus ratio have been found which give a model that is too short and a model that is too long then quadratic interpolation is used to produce the next estimate. One such estimate is usually sufficient to provide a solution that is accurate to a tenth of one per cent of the actual solution. This is the tolerance normally used throughout this work. The program has limits placed on the region that it will search for a solution which is in the range 0.01 to 50., should a modulus ratio outside these limits be indicated the execution of the program will cease. The reason that these limits have been set are that when the modulus ratio reaches either of these limits then further refinement of the modulus ratio has negligible effect upon the length of the model.

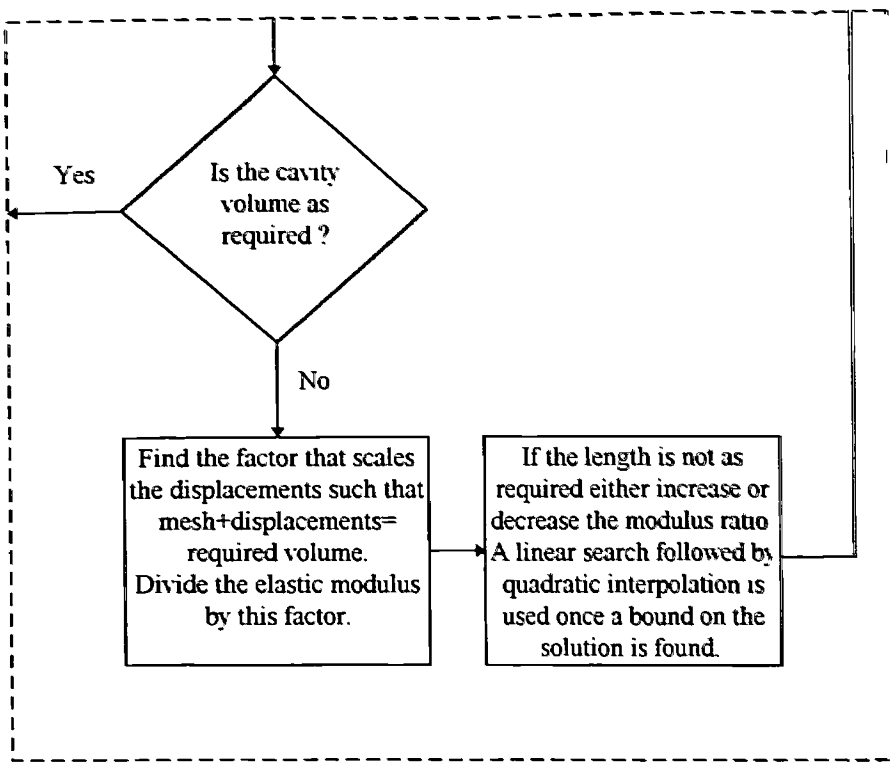

Yes
Is the cavity volume as required ?
No
Find the factor that scales the displacements such that mesh+displacements= required volume. Divide the elastic modulus by this factor.
If the length is not as required either increase or decrease the modulus ratio A linear search followed by quadratic interpolation is used once a bound on the solution is found.

Fig. 8.6 This flow diagram shows how the cavity volume and the mid-base to apex length are matched. (This flow diagram comes from the flow diagram for "PART4" of the program, Fig 8.4)

There is a choice of two volume and length matching options available within the program. The first is option "3", which is like option "2" of the program in that it matches smoothed volumes, the difference here is that an attempt to also made to match a ventricle target length. The target length was produced by linear interpolation of the measured length values. The use of this type of process was less than successful though may be of future use should more suitable data become available. The other length matching option is "4", this allows the volume and length to be matched to a frame that is not successive. It has been used to obtain a cavity volume and length match between the first and last frames of diastole.

8.4 Calculation of the Forces Generated by Muscle Contraction

As was previously stated in Chapter 7 the method used to simulate muscle contraction is similar to that of thermal stressing. Ultimately within the program the active contraction part generates the nodal forces that result from the material contraction. It is important to note that this is not contraction in the normal sense since no volume strain occurs. It is more a reshaping process whereby the material contracts in one direction while expanding in another and thus it preserves its overall volume. The simplest way to provide a measure of the material contraction is to supply the strain values for the fibre and cross-fibre directions. This has the advantage that it is independent of the chosen material properties and is thus far simpler to interpret and compare between patients.

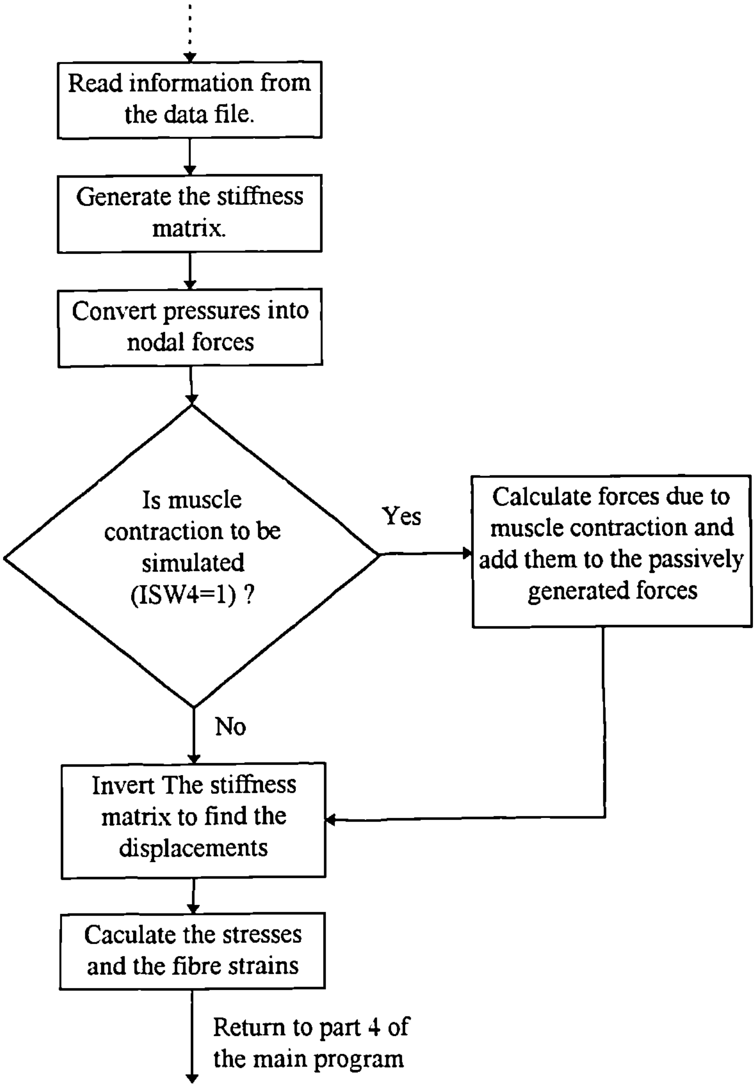

Read information from the data file.
Generate the stiffness matrix.
Convert pressures into nodal forces
Is muscle contraction to be simulated (ISW4=1) ?
Yes
Calculate forces due to muscle contraction and add them to the passively generated forces
No
Invert The stiffness matrix to find the displacements
Caculate the stresses and the fibre strains
Return to part 4 of the main program

Fig 8.7 This flow diagram shows how the simulated muscle contraction is incorporated into the original finite element program XL1.

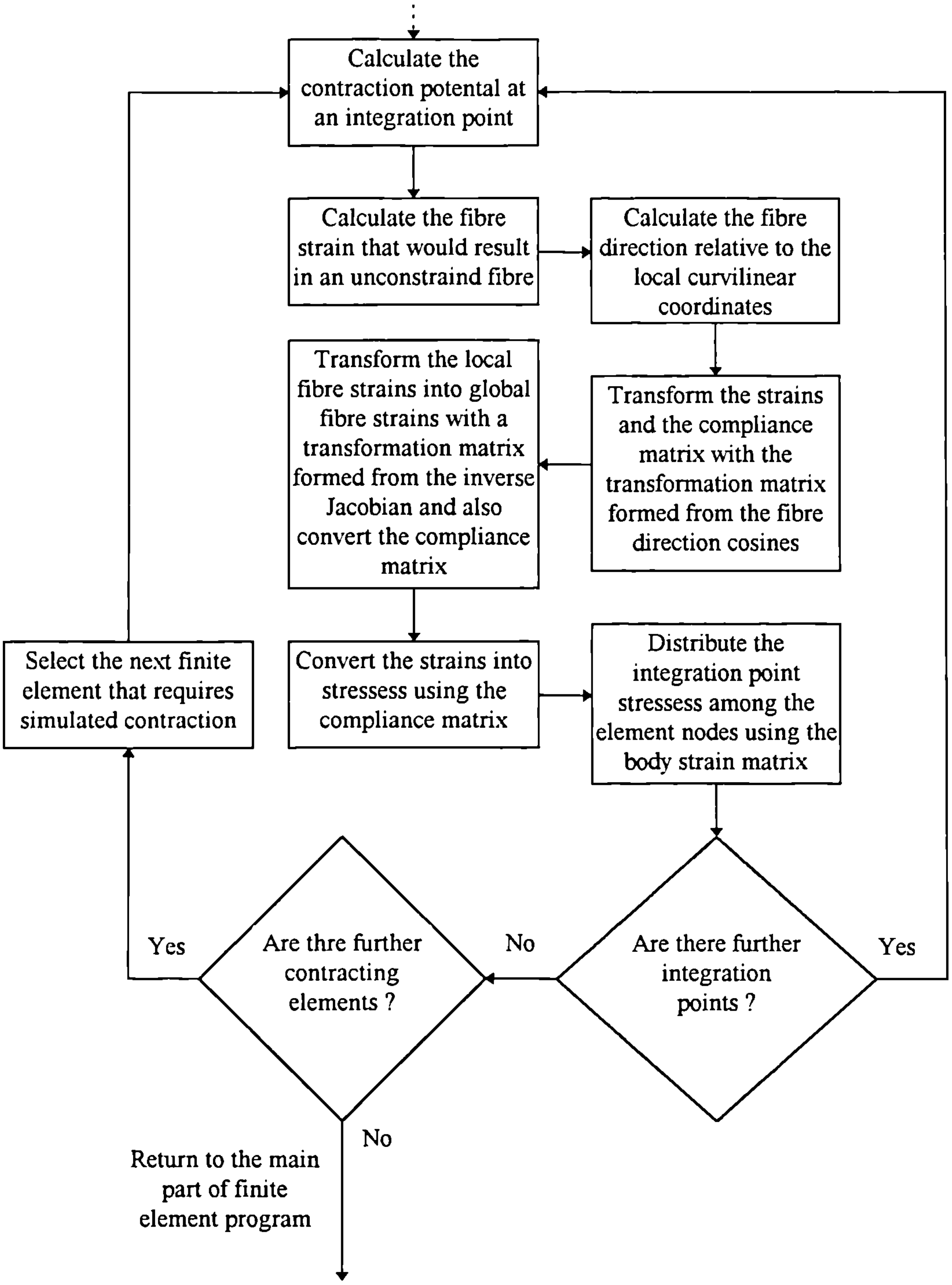

Calculate the contraction potental at an integration point
Calculate the fibre strain that would result in an unconstraind fibre
Calculate the fibre direction relative to the local curvilinear coordinates
Transform the local fibre strains into global fibre strains with a transformation matrix formed from the inverse Jacobian and also convert the compliance matrix
Transform the strains and the compliance matrix with the transformation matrix formed from the fibre direction cosines
Select the next finite element that requires simulated contraction
Convert the strains into stressess using the compliance matrix
Distribute the integration point stressess among the element nodes using the body strain matrix
Yes
Are thre further contracting elements ?
No
Are there further integration points ?
Yes
No
Return to the main part of finite element program

Fig 8.8 This flow diagram shows how the nodal forces attributed to the simulated muscle contraction are calculated.

The process is relatively simple to incorporate into a non-active finite element program. It requires only the inclusion of the forces generated by the simulated muscle contraction. Thus once this has been done the solution of the finite element tensor equation is identical to the passive case. A flow diagram for the whole finite element program can be found in Vinson [1] Chapter 7.

8.5 Automation For Systolic Work

There are six options for systolic volume matching and their purpose is described here. The flow diagram shows how the finite element program is used to obtain a solution for the contraction coefficient that allows the model to mirror the observed behaviour of the left ventricle.

It was discovered early on that the ratio of along-fibre to cross-fibre contraction coefficient ratio was not only important to preserve wall volume but also depends upon the size of the contraction coefficient. Thus the value of the contraction coefficient ratio must be calculated at the same time as the contraction coefficient itself, if wall volume is to be preserved.

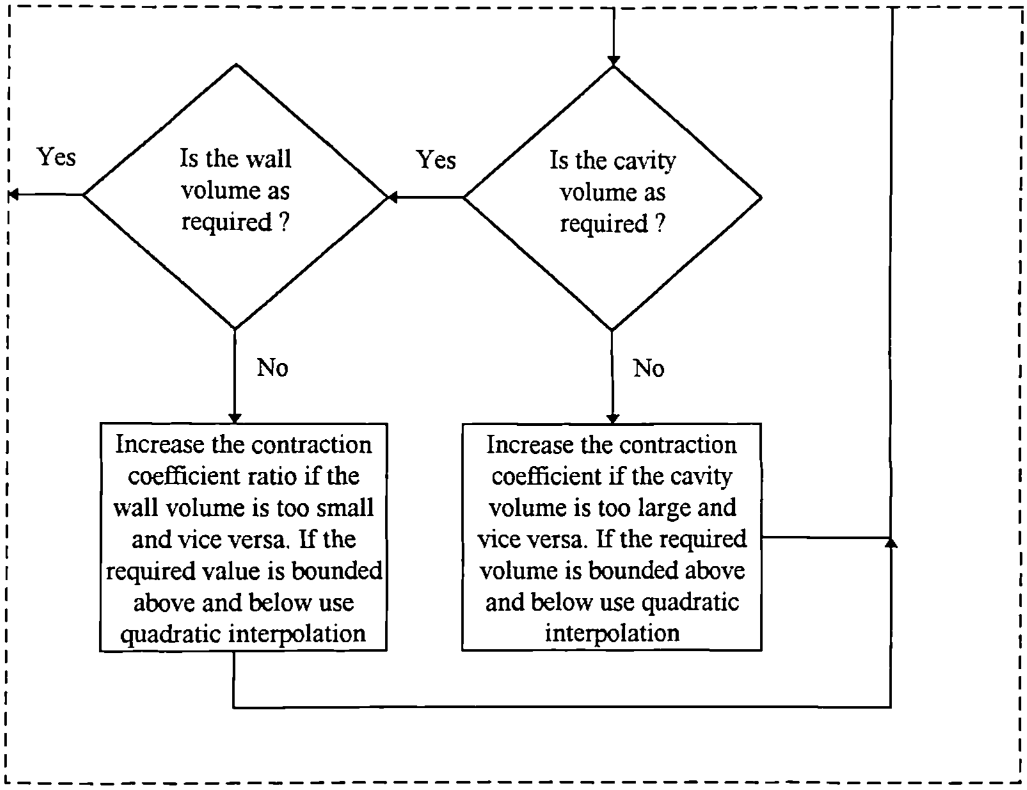

Yes
Is the wall volume as required ?
Yes
Is the cavity volume as required ?
No
No
Increase the contraction coefficient ratio if the wall volume is too small and vice versa. If the required value is bounded above and below use quadratic interpolation
Increase the contraction coefficient if the cavity volume is too large and vice versa. If the required volume is bounded above and below use quadratic interpolation

Fig 8.9 This flow diagram gives an overview of how the wall volume and contraction coefficient ratio are simultaneously calculated. (This flow diagram comes from the flow diagram for "PART4" of the program, Fig 8.4)

As can be seen from the diagram this part of the program has been structured into two distinct sets of subroutines. These can easily be combined together to provide any combination of functionality. The option to match the model's length by varying the shear modulus has been retained. The fact that varying this parameter has a much reduced effect when the fibre angle is high means this option is not of great use. The reason that it has been retained is that is clearly shows how extensions can be easily made to the programs functionality. It is also relatively straightforward to adapt this option to provide a new function.

8.6 Matching Routines

The purpose of the matching routines is to produce a new estimate for the material properties based upon the results obtained from previous estimates. The nature of these routines is important only in that they can produce convergence to the required solution. It is also greatly desirable that this process should seek also to minimise the number of iterations that is necessary for suitable convergence to occur.

In order to obtain convergence in the volume matching process during the simulation of diastole the fact that the model is linearly elastic is made use of, that is that the displacements of the model's nodes are linearly dependent upon the material properties. This is made use of here by scaling the displacements from an initial estimate to obtain a volume match and then scaling the material properties by the reciprocal of this value.

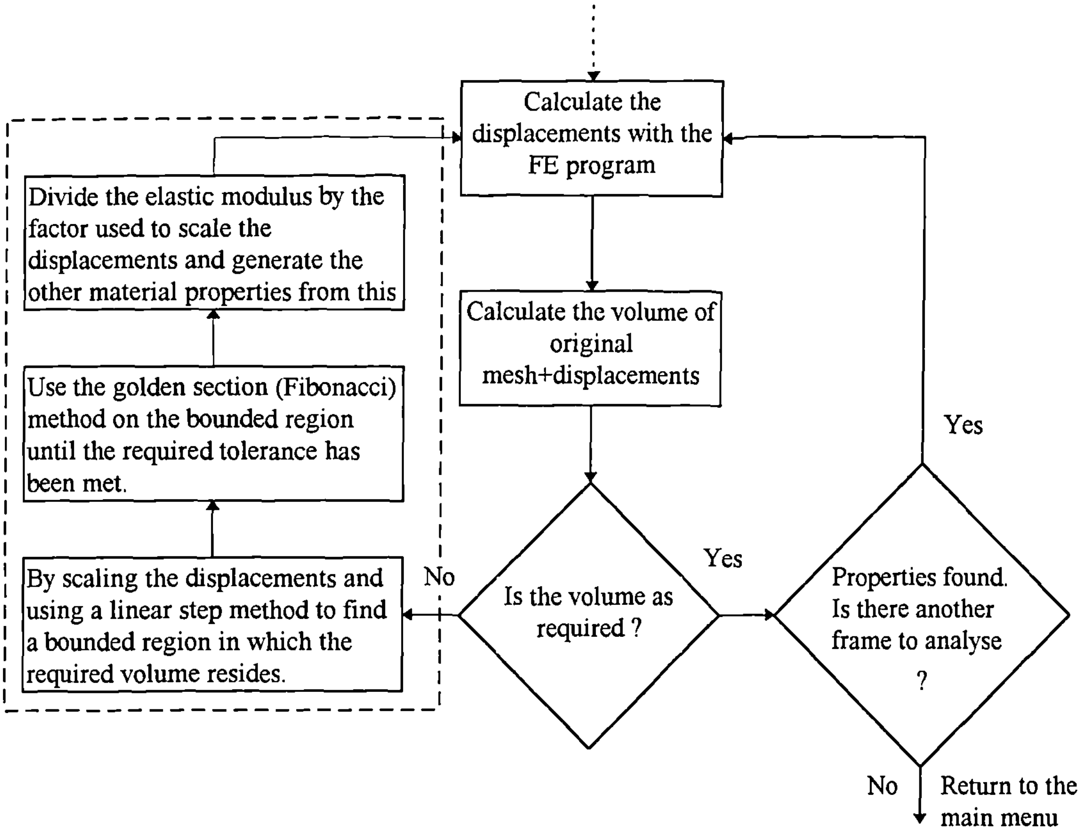
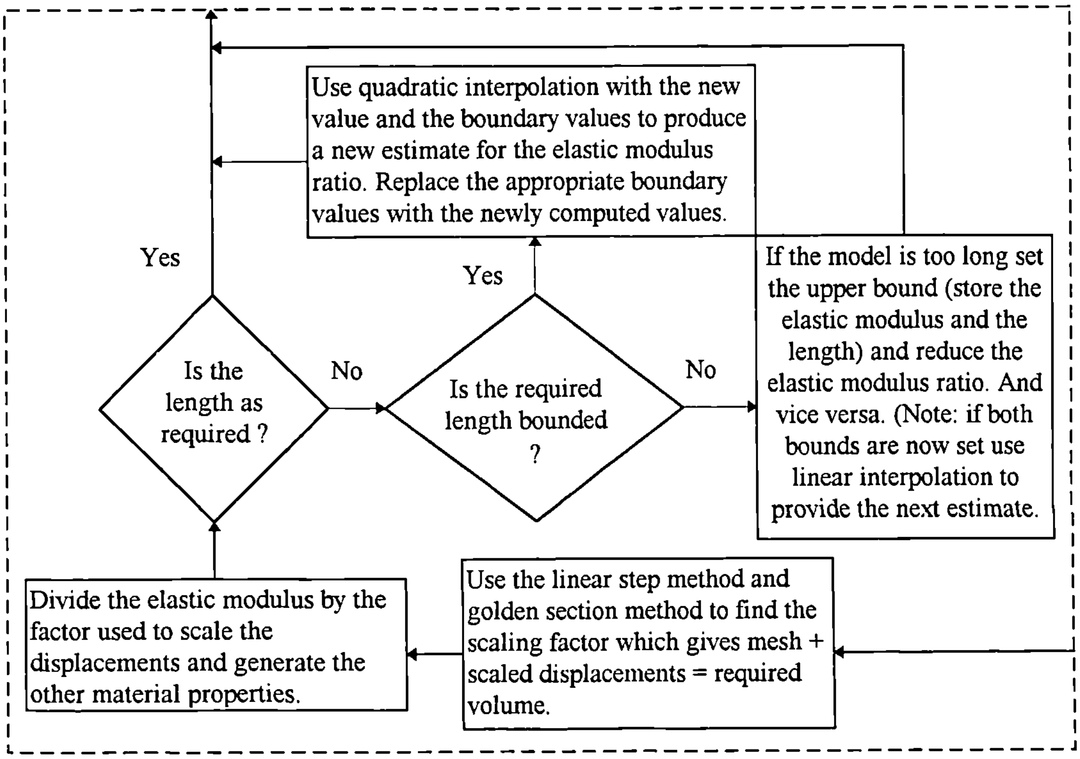

Calculate the displacements with the FE program
Calculate the volume of original mesh+displacements
Is the volume as required ?
Yes
No
Properties found. Is there another frame to analyse ?
Yes
No
Return to the main menu
By scaling the displacements and using a linear step method to find a bounded region in which the required volume resides.
Use the golden section (Fibonacci) method on the bounded region until the required tolerance has been met.
Divide the elastic modulus by the factor used to scale the displacements and generate the other material properties from this

Fig 8.10 This shows the process by which the material properties are found during diastole. This flow diagram is valid for the description of options "1" and "2".

The golden section/Fibonacci method is a search method which incorporates trisection of a bounded region in order to minimise a function. In this case the function is the absolute error between the required volume and that obtained from the original F.E. mesh plus the scaled displacements. The method is well known and most books on optimisation should contain a good description of the method.

The area surrounded by the dashed box is the only part which requires alteration for each of the matching options. For this reason only the part within the aashed box will be reproduced in the following flow diagrams.

When matching the active material properties or the ventricle length by varying the elastic modulus ratio the model is no longer linearly dependent upon the parameters which are varied. A more sophisticated method must be used to obtain solution values for the required parameters. The fact that the system is continuous is made use of as is *a priori* knowledge about the response of the system to parameter changes. The process in each case involves finding a bounded region in which the desired solution exists and then using quadratic interpolation to refine this boundary. In the case of matching only cavity volumes during systole only one application of quadratic interpolation is usually necessary to achieve a cavity volume match to within 1.0% of the required cavity volume. In fact it is often the case that the difference between the calculated and required volume is often an order of magnitude less than this.



Use quadratic interpolation with the new value and the boundary values to produce a new estimate for the elastic modulus ratio. Replace the appropriate boundary values with the newly computed values.
Yes
Yes
Is the length as required ?
No
Is the required length bounded ?
No
If the model is too long set the upper bound (store the elastic modulus and the length) and reduce the elastic modulus ratio. And vice versa. (Note: if both bounds are now set use linear interpolation to provide the next estimate.
Divide the elastic modulus by the factor used to scale the displacements and generate the other material properties.
Use the linear step method and golden section method to find the scaling factor which gives mesh + scaled displacements = required volume.

Fig 8.11 This shows the process by which the material properties and the elastic modulus ratio are found during diastole, such that the cavity volume and length are matched. This flow diagram is valid for the description of options "3" and "4". (This flow diagram comes from the flow diagram given in Fig 8.10)

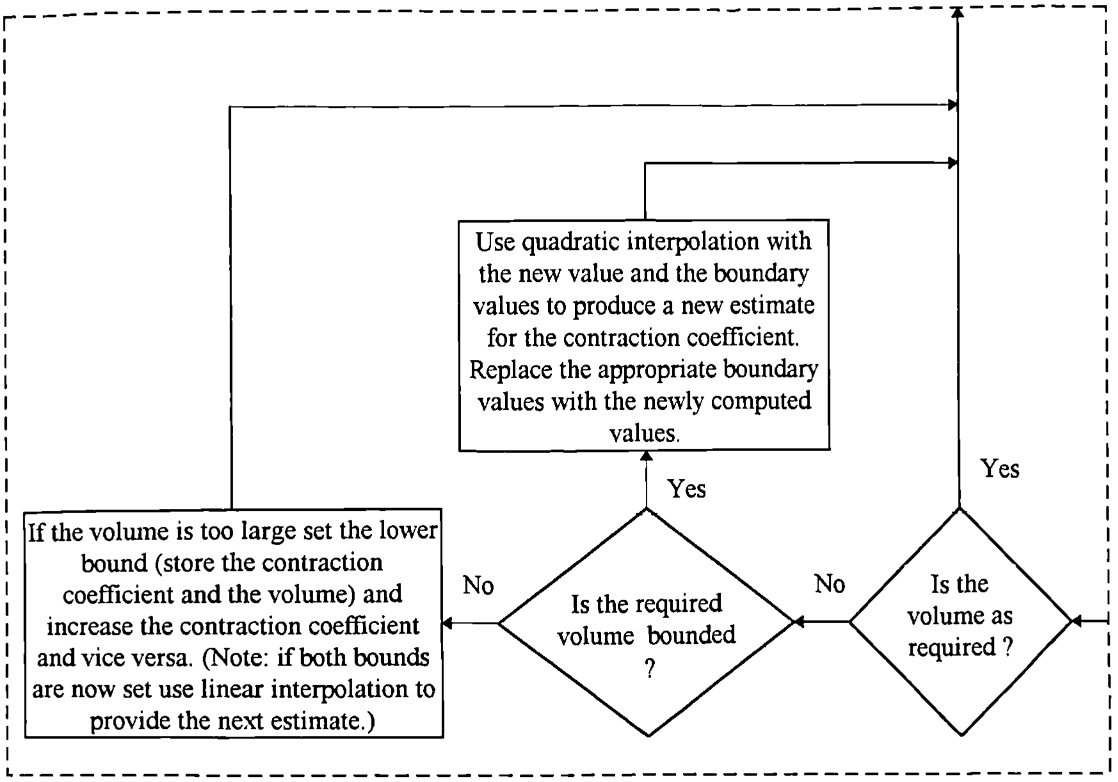

Use quadratic interpolation with the new value and the boundary values to produce a new estimate for the contraction coefficient. Replace the appropriate boundary values with the newly computed values.
Yes
Yes
If the volume is too large set the lower bound (store the contraction coefficient and the volume) and increase the contraction coefficient and vice versa. (Note: if both bounds are now set use linear interpolation to provide the next estimate.)
No
Is the required volume bounded ?
No
Is the volume as required ?

Fig 8.12 This flow diagram shows how the systolic properties are derived for matching options 5 to 7. (This flow diagram comes from the flow diagram given in Fig 8.10)

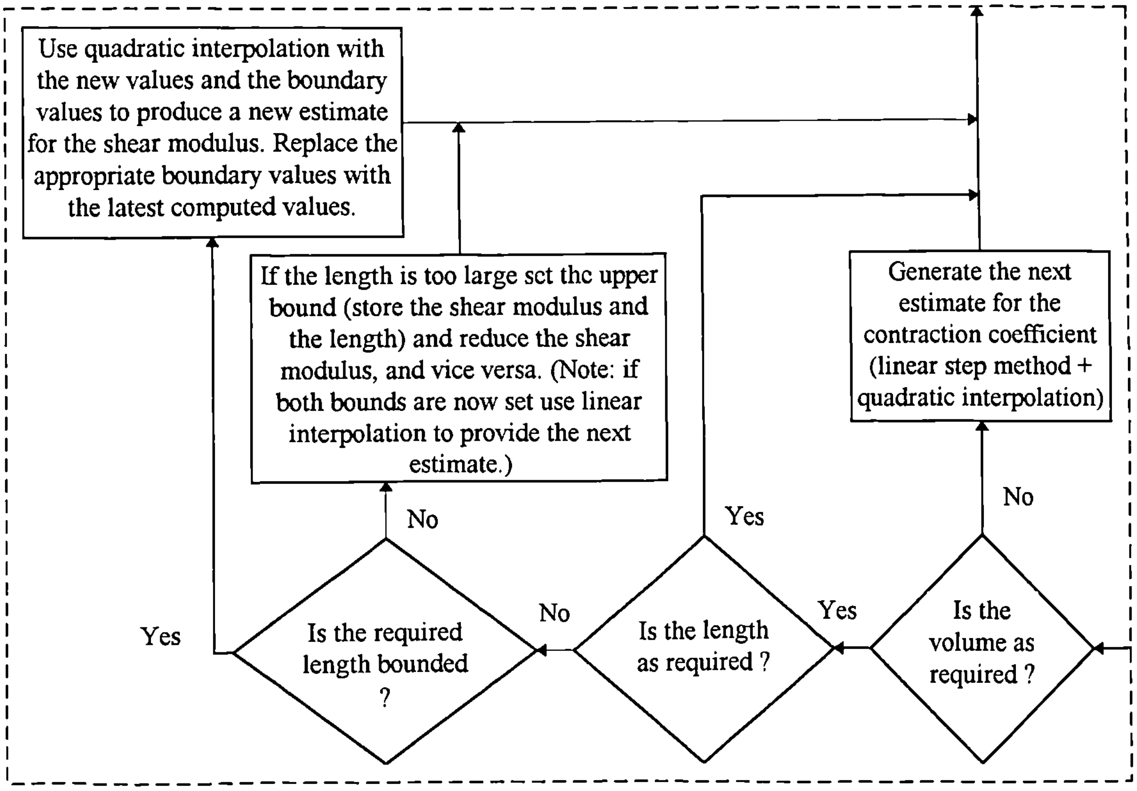

Use quadratic interpolation with the new values and the boundary values to produce a new estimate for the shear modulus. Replace the appropriate boundary values with the latest computed values.
If the length is too large sct thc upper bound (store the shear modulus and the length) and reduce the shear modulus, and vice versa. (Note: if both bounds are now set use linear interpolation to provide the next estimate.)
Generate the next estimate for the contraction coefficient (linear step method + quadratic interpolation)
No
Yes
No
Yes
Is the required length bounded ?
No
Is the length as required ?
Yes
Is the volume as required ?

Fig 8.13 This flow diagram shows how the systolic properties are derived for option 8 which also attempts to match the model's length. (This flow diagram comes from the flow diagram given in Fig 8.10)

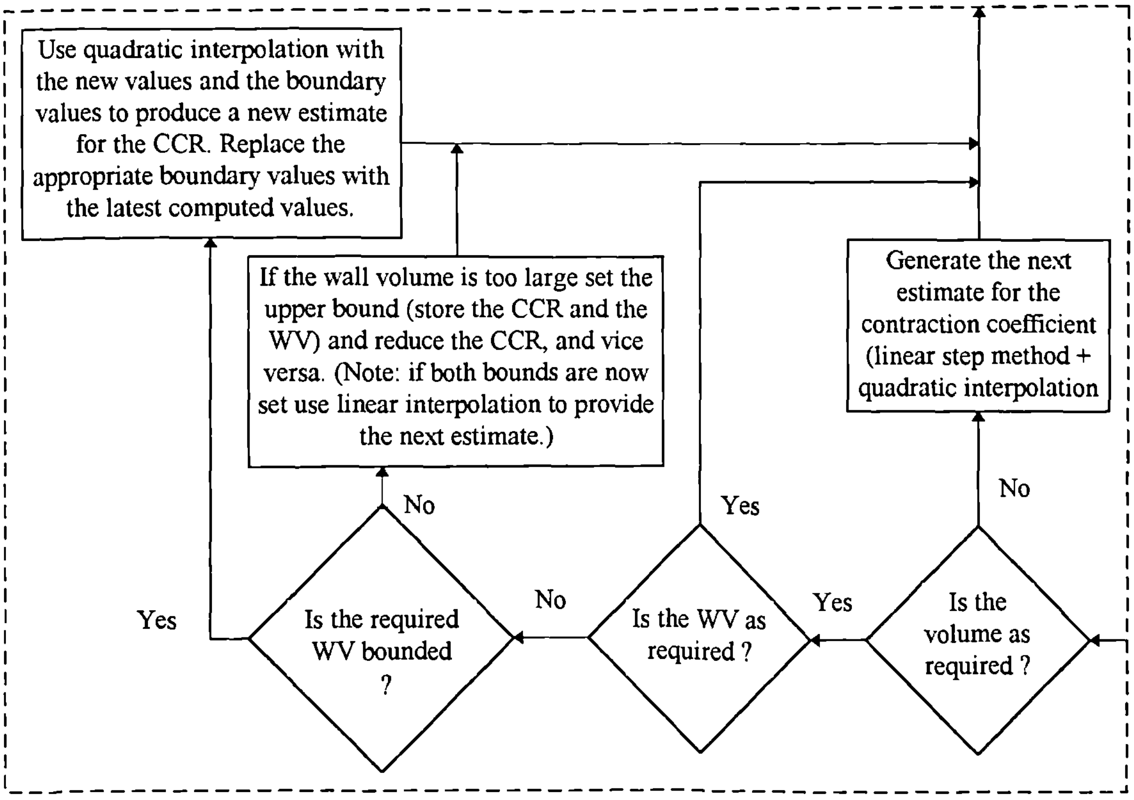

Use quadratic interpolation with the new values and the boundary values to produce a new estimate for the CCR. Replace the appropriate boundary values with the latest computed values.
If the wall volume is too large set the upper bound (store the CCR and the WV) and reduce the CCR, and vice versa. (Note: if both bounds are now set use linear interpolation to provide the next estimate.)
Generate the next estimate for the contraction coefficient (linear step method + quadratic interpolation
No
Yes
No
Yes
Is the required WV bounded ?
No
Is the WV as required ?
Yes
Is the volume as required ?

Fig 8.14 This flow diagram shows how the systolic properties are derived for matching options 9 and 10. In these two cases the wall volume (WV) is kept constant by varying the contraction coefficient ratio (CCR). (This flow diagram comes from the flow diagram given in Fig 8.10)

The idea behind the matching process when more than one thing is to be matched is to reduce the problem to a quasi-one-dimensional problem. That is solving only one part at a time. This means that simple methods can be used to search for a solution. The result is that solution methods do not require the use of arbitrary objective functions and known relationships between cause and effect of the parameter changes can easily be made use of. The result is that these tailored methods can give much more rapid convergence that general optimisation techniques or similar convergence with a lower verhead.

8.7 Summary

The program as it stands provides for the automatic calculation of the active and passive material properties. It provides a number of useful and possibly not so useful options which can be chosen for the calculation of the material properties. It is also relatively easy to extend the program with further procedures to enable other aspects of the model to be more accurately simulated.

Another useful feature of the program's structure is that the finite element part can be easily separated from the unified program so that it may easily be run upon super computers should the need for very high performance be required. It is also equally possible to separate the whole of "PART4" of the program including the finite element program to allow the automated process to be run upon other computer systems which do not support the packages Nag and SimplePlot which are used in other parts of the program.

For further detail the program listing can be found in Appendix III and a sample data file and a sample of the program output can be found in Appendix II.

Chapter 9

9. Factors Influencing General Systolic Performance

In this Chapter an investigation will be carried out to find some of the important mechanical factors which are responsible for global deformation of the left ventricle. These factors include boundary conditions, material properties and fibre orientation. Originally it was intended that the analysis would be performed over several frames of data. This would mean that the small changes in the geometry of the model would be more easily quantified and might also be less susceptible to inaccuracies in the original data. It was however not possible to perform the analysis successively over a large number of frames, i.e. over a large time period as the mesh became badly deformed and was thus likely to be a source of great inaccuracy. This would then compromise the validity of any results obtained. In some cases this deformation was severe enough to cause a total loss of integrity in some of the finite elements.

The reason that the finite element mesh became unusable was that the deformation of myocardium has a very large shear component. This results in the finite elements becoming badly distorted. This shearing effect has been seen in the ventricles of real human patients by using the technique of Magnetic Resonance Myocardial Tagging (Buchalter *et al.* 1990 [77] and Buchalter 1992 [78]).

It was decided that the analysis would have to be performed over only one time period or frame interval, as has previously been described in Chapter 5 for the method of "target volume matching". For this process the last pair of the frames in systole were used as they underwent the largest relative volume change. The result of this would be that global geometric changes would thus be the greatest of any part of the systolic phase and thus the effects of parameter changes would be more easily quantified.

It is important that the most significant factors influencing performance of the left ventricle are found as this will be of benefit in the development of ventricular finite element models. This will lead to models that are better able to quantify ventricular performance and will hence be of greater research and clinical use.

9.1 The Analysis Procedure and Programming Enhancements

To enable us to evaluate the effect of changes to the model we must have another descriptor for the model other than the cavity volume. What is required is some measure of "shape". Previously there had been a shape index defined as (Grewal 1988 [2]),

$$\text{Shape Index} = \frac{6\pi^{\frac{1}{2}}\text{Volume}}{\text{Surface Area}^{\frac{3}{2}}} \qquad \text{Eqn. 9.1}$$

which had been an expansion into three dimensions of the two dimensional shape index (Gibson and Brown 1975 [80]),

$$\text{Shape Index} = \frac{4\pi\text{Cross} - \text{Sectional Area}}{\text{Perimeter}^2} \qquad \text{Eqn. 9.2}$$

The first equation relates the surface area of the three dimensional ventricle to its volume in order to obtain an index, which is unity for a sphere. The second equation relates the area of a cross-section to its circumference and provides an index which is unity for a circle.

There had previously been an oversimplification in the calculation of the shape function. The surface area was not obtained accurately and had been approximated as the curved surface of a cylinder. This cylinder was defined as the length of the ventricle (mid-base to apex length) and the average radius of the base. The result of this was that the change in shape index during finite element analysis was as a result of length change only (the endocardial base is held fully as the boundary condition so the average radius is constant). This would then be equivalent to using length as the index of shape. Length has previously proved not to be a very good parameter to use due to the nature of digitisation errors.

It was necessary to create several new subroutines to enable the calculation of the endocardial surface area of the model directly from the finite elements. This involves the calculation of the area Jacobian, for that surface of a finite element, and numerical integration of it over that surface. This is done for the "brick" as well as the "wedge" shaped elements.

Another thing that was changed was the size of the apex region of the model. In the real left ventricle the stiff apex is a small "button". The model however was divided equally by height for all elements including the apex ones. This means that the size of the "wedge" elements forming the apex was similar to that of the "brick" elements. The meshing routine was altered so that the apex was fixed as the last 4mm of the model. Typically the apex elements could previously have accounted for as much as 15mm of the model's height. This was the smallest the apex could be made without compromising the integrity of the "wedge" shaped elements. To make the "stiff apex" smaller and still preserve the integrity of the mesh would require a significant rethink of the meshing scheme used.

The investigation of systole is a new area in this work and it is important that global shape change can be easily visualised. It is also important that global shape change can be related to numerical shape change indicators such as length or "shape index". To this end a new set of visualisation routines were created to enable a three dimensional view of the deformed finite element mesh to be produced. Previously only two dimensional short axis cross-sectional views had been available for the deformed data. These are severely limited as they provide no information about longitudinal changes in the model.

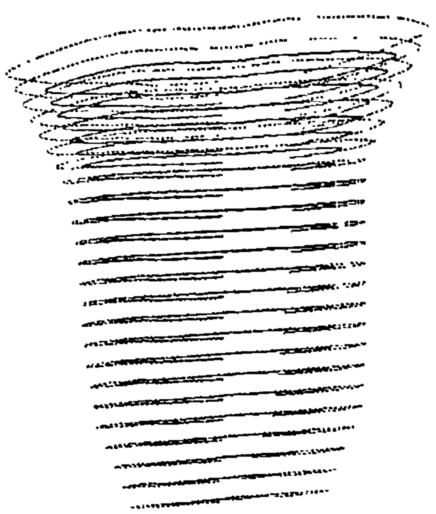


Fig 9.1 This shows the three dimensional reconstruction of the endocardial and epicardial surfaces of a typical finite element model. The endocardium is shown as the solid lines and the epicardium as the dashed lines. The apex region has not been included since it is very small and its inclusion would detract from the clarity of the diagram.

The cavity shape can be displayed at any orientation within three dimensional space. The epicardium and endocardium can be selectively viewed and up to three sets of data can be overlaid. Each of these sets of data will then be displayed in a different colour. To enable an easy comparison between the different sets of data each set is aligned vertically along their mid-base to apex line and the apex is centred at the origin. This is done before plotting and rotation of the data by the user defined amounts.

9.2 The Analysis Procedure

Originally the idea was to use the starting frame in systole and then match the volume change with the frame at the end of systole. This, it was hoped would reduce the significance of the digitisation errors in the data. It would also have the benefit that any divergence between calculated changes in geometry and those taken from the raw data would be more easily quantified. The analysis performed in this way resulted in contraction coefficients far greater than unity for some patients. This is of grave concern as the finite element program is based upon small deflection theory and the associated strains resulting from such large values of the contraction coefficient are certain not to be small (Note: a contraction coefficient is in effect a strain potential). For some sets of patient data it was not even possible to obtain results as the elements would lose their integrity and some would even be reported as having negative volume. Despite these failings the analysis did seem to function reasonably well for some patients. It was decided however that it would be more prudent to proceed with the analysis on a frame by frame basis similar to the "target volume matching" process employed in the investigation of diastole.

9.3 Wall Volume and the Contraction Coefficient Ratio

The following tests described in this section were all originally performed using an along-fibre to cross-fibre contraction coefficient ratio of 1:-0.5, a value originally suggested by Chen *et al.* (1980 [80]). This paper, although not published until 1980, was of work carried out prior to 1977. It was then later suggested that a value in the range of 1:-0.5 to 1:-0.3 (Chen *et al.* 1978 [81]) may be more appropriate. This ratio is important since the values are in effect potential strain values and will affect volume strain changes. The myocardium is normally considered to be of constant volume, however there has been some evidence that there may be as much as 10% volume strain during the cardiac cycle (Horowitz *et al.* [35] and Yin *et al.* [82]). It is therefore the case that consideration as to what is happening to the wall volume of the model is important.

It was necessary to create a number of new subroutines which were able to calculate the wall volume of the model from the individual finite elements. This process is similar to the calculation of the surface area except that instead of using the area Jacobian the volume Jacobian is used and integration is over the volume and not just one surface.

The introduction of wall volume calculations resulted in the discovery that the contraction coefficient ratio of 1:-0.5 resulted in the wall volume increasing. The amount of this increase was largely dependent upon the value of the along-fibre contraction coefficient. The larger this value the greater the volume increase was. For most tests the increase was only a few percent but when the material properties were varied this increase could become significant.

As stated above the following tests were originally carried out using a contraction coefficient ratio of 1:-0.5 and this resulted in the wall volume increasing. This affected greatly the initial results obtained is several ways. Firstly, it resulted in an elevation in the size of contraction coefficient required to match the volume, which in itself will result in yet a further increase in wall volume. In some cases this was so severe that the wall volume would increase at a rate faster than the cavity volume would reduce. The effect of this is that this reduction caused by the increase in contraction coefficient is smaller than the increase caused by the wall volume increasing. Thus it would be impossible to match the target volume.

Secondly, even when the volume matching process was possible the results could be self-contradictory. That is altering a parameter may cause the system to behave in a particular way except when the parameter reached a certain value when the effect would be markedly different. The other part of this problem is that because of the change in wall volume the effect that the parameters being tested had were either exaggerated or masked by the effect of changing wall volume.

9.4 Initial Results Using Standard Values

In order to assess the relative importance of each of the model's parameters it is vital that a reference set of results be obtained. The reference set of parameters was based on those values obtained from the analysis of diastole. The elastic modulus is taken as a representative value of the values calculated in the diastolic investigation. The shear moduli were obtained from the assumed relationship between elastic and shear moduli used in Chapters 3 and 5 and Poisson's ratio was kept the same from the previous analysis. This gives a starting point against which variations in all the relevant factors can be compared.

In this section we use data from four patients taken at the end of systole. The reason for this is that the relative volume change is much larger at this point which means that the changes in deformation due to altered conditions will be more pronounced. This is important since geometric changes are likely to be small and the even smaller variations in these changes would otherwise be dwarfed by approximations in the calculation and may lead to the incorrect interpretation of results.

Patient	Frame Used	Minor Elastic Modulus /(KN/m^2)	Pressure Increase /(mmHg)	Volumes /(cm^3)		Original Shape Index	Original Length /(mm)
				Original	Target		
BA	14	50	1.62	107.19	104.08	0.852	75.71
CL	14	20	3.68	122.72	114.04	0.896	82.83
MI	15	60	2.01	56.21	44.62	0.660	77.58
WE(A)	15	15	2.36	58.40	54.99	0.709	73.14

Table 9.1 This table shows the original shape index and length obtained from the patient data. It also shows the value of elastic modulus used for each patient as well as the pressure and volume changes between the original and the target frames. The other material properties can be d rived from the information given in Chapter 3.

Patient	Contraction Coefficient	Contraction Coefficient Ratio	Deformed Shape Index	Deformed Length /(mm)	Percentage Length Reduction
BA	0.121	-0.455	0.850	74.80	1.21
CL	0.252	-0.440	0.841	86.81	-4 81
MI	0.295	-0.442	0.571	76.59	1.27
WE(A)	0.158	-0.432	0.666	73.40	-0.35

Table 9.2 Here the changes that occur when systole is simulated using our reference data, while keeping the wall volume constant, are shown.

One of the most striking features in the length changes is that two of the patients exhibit a length increase whereas the other two have a length reduction. From examination of the original X-ray frames it can be seen that the overall trend in length is to decrease during systole.

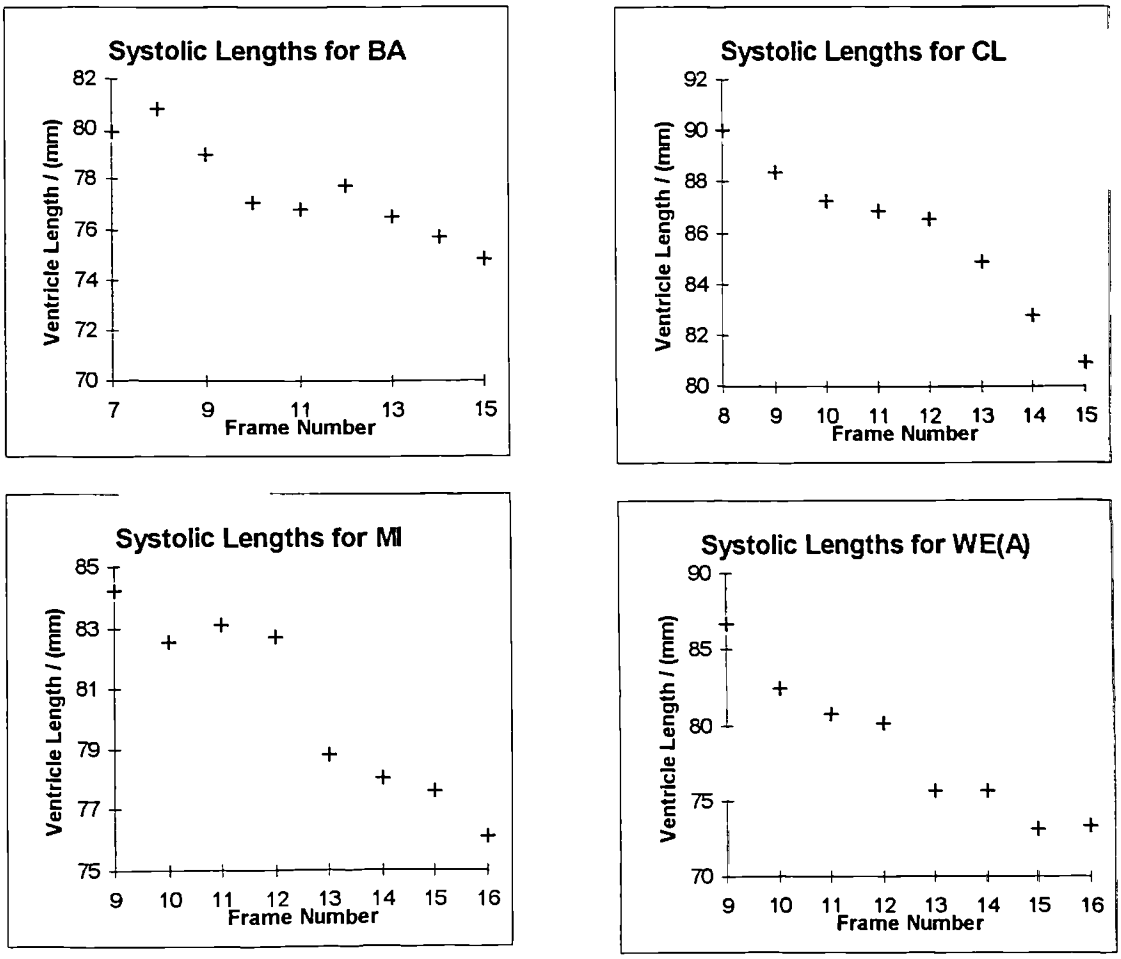


Fig. 9.2 These four graphs show the measured ventricle lengths against frame number for systole.

A model that was able to simulate systole was proposed by Horowitz *et al.* (1986 [35]) it was based on a single set of patient data and also predicted a length increase during systole. This was assumed to be due to the exclusion of the papillary muscles from the model, which contract strongly during systole. It however would be unwise to agree with this view. From patients who have undergone mitral valve replacement, and hence have had their papillary muscles cut, it has indeed been observed that the length reduction is affected. Despite this the ventricle is still seen to reduce in length during systole; albeit in an abnormal manner. It is also the case that information about the force generated by these muscles is not easily quantified.

It should also be noted that the shape index reduces in every case. The reason for this is that the surface area is increasing relative to the volume. This may mean that the ventricle is either becoming more elongated relative to its volume or the surface area is increasing relative to the volume for another reason. The use of the shape index makes it difficult to visualise ventricular geometric changes whereas length change is much easier to visualise. The disadvantage is that length measurements taken from the data are much less accurate than the shape index. This is because the accuracy of the length is proportional to the digitisation errors whereas the shape index is dependent upon the surface area and is thus proportional to the errors squared. This will make the surface area less sensitive to the data accuracy than the length.

Another noticeable feature is that the contraction coefficient ratios are all fairly similar in magnitude. The values are also well within the range -0.3 to -0.5 as p edicted by Chen *et al.* [80].

9.5 Material Properties

The importance of the various material properties upon global deformation in systole is not well known. Here we examine the effect of each material property in turn upon the global deformation of the left ventricle. In this way the relative importance of each upon systolic cardiac mechanics can be quantified. This will enable a better choice for material properties in the analysis of systole and may enable a more targeted approach to experimental work involving direct measurement of myocardial properties.

9.5.1 Elastic Modulus

This is arguably the most important property of the myocardium during diastole since it is almost totally responsible for governing the change in volume in response to a pressure increase. Its importance during systole however is unknown and so the effect of different values is investigated here. It has been suggested that the value of the elastic modulus is almost a factor of 10 greater during systole than during diastole (Lundin 1944 [61]). To circumvent this uncertainty some authors in their systolic models have used assumed stress/strain relationships to get over the ambiguity of the elastic modulus during systole (Horowitz *et al.* 1986 [35]). This is far from ideal as these exponential stress/strain models have been shown not to predict diastolic properties (Grewal 1988 [2]), for which they were intended, and their use in systole must be even more suspect.

Since the volume of the ventricle is decreasing during systole it would seem reasonable to assume that the major factor governing ventricular deformat on was sarcomere contraction. If this is the case then the effect of the internal pressure loading and consequently that of the elastic modulus will be small and possibly even negligible. It was the view of Gibson (1996 [83]) that the cavity pressure of the ventricle was not an important factor in systolic deformation of the left ventricle.

With this in mind the model is tested with an elastic modulus tenfold higher and fivefold lower than the base value and the overall deformation compared to the reference test (Table 9.2).

Patient	Contraction Coefficient	Contraction Coefficient Ratio	Deformed Shape Index	Deformed Length /(mm)	Percentage Length Reduction
BA	0.319	-0.416	0.857	75.77	-0.07
CL	0.778	-0.340	0.769	94.69	-14.32
MI	0.322	-0.435	0.571	76.87	0.92
WE(A)	0.430	-0.338	0.622	78.91	-7.90

Table 9.3 This table shows the deformation information of the model for an elastic modulus a fifth that of the reference data.

Patient	Contraction Coefficient	Contraction Coefficient Ratio	Deformed Shape Index	Deformed Length /(mm)	Percentage Length Reduction
BA	0.067	-0.467	0.847	74.90	1.07
CL	0.139	-0.465	0.860	84.62	-2.16
MI	0.289	-0.443	0.571	76.52	1.37
WE(A)	0.076	-0.458	0.677	72.75	0.54

Table 9.4 This table shows the deformation information of the left ventricle for an elastic modulus tenfold that of the reference data.

Looking at the first table it can be seen that taking a lower value for the elastic modulus has had a great effect upon the deformation of the model. The effect on the deformation of the data "MI" and data "BA" is much less than that experienced by the other two sets of data. This is almost certainly due to the fact that these two have an elastic modulus at least twice that of the other two while also having the two smallest pressure increases.

The second table shows a markedly smaller change from the values obtained in Table 9.2. The "BA" and "MI" data again show the smallest change and this time it is very much smaller than the changes shown in the previous table. The data "CL" and "WE(A)" still however have their deformation significantly affected by an increase in elastic modulus. This effect is much smaller than that experienced by the decrease in elastic modulus.

It can therefore be concluded that only if the elastic modulus is itself small are small changes likely to have a significant effect upon global deformation.

It should be noted that there is some evidence that the shape index is not performing as it had been intended to do. The data sets "CL", "MI" and "WE(A)" all behave as expected. That is their shape indices reduce as the ventricles become relatively more elongated, less spherical. Data set "BA" however exhibits the reverse of what should be expected. The shape index increases as the model becomes relatively more spherical (comparing the values of Table 9.3 and Table 9.4). This means that although the model is becoming more spherical due to shortening along its major axis its surface area is increasing. It must therefore be the case that the model's surface area is increasing by becoming more uneven. The shape index therefore may be influenced by the smoothness of the surface rather than just by the global shape.

9.5.2 Ratio of With-Fibre and Cross-Fibre Young's Modulus

The value of the elastic modulus ratio has been shown from experimentation (Yin *et al.* 1987 [63]) to vary between 6.5:1 to 1:4 (along the fibre to across the fibre). The value of this ratio has been shown to vary from one part of the myocardium to another as well as from patient to patient. It has also been hypothesised that the ratio is history dependent and thus is likely to change with time. The net result is that there is a great deal of uncertainty as to a reasonable value to choose as a representative value for the bulk elastic modulus ratio. It is likely that not only would a bes choice be patient specific but may also vary with time. The importance of this ratio in systole is thus of some interest due to the lack of a specific value.

In altering the modulus ratio the stiffness of the model will obviously be greatly affected. The stiffness of the model has already been shown to have a significant effect upon the deformation of the ventricle. Thus to eliminate the effect of changes in the stiffness of the model the value of the elastic modulus will also need to be changed to keep the stiffness of the model constant between tests involving the modulus ratio.

The method used to find the value of the elastic modulus that keeps the wall stiffness constant will be briefly described. First, the program is run for standard values without any simulated contraction. The resultant cavity volume is then noted. This value is then used to match against using the diastolic options of the program for each of the required elastic modulus ratios. The resultant values for the elastic modulus obtained are then given in the table of results below.

	M.R.=0.5			M.R.=1.0			M.R.=10.0		
Patient	E.M.	C.C	% .LR.	E.M.	C.C.	%L.R.	E.M.	C.C.	%L.R.
BA	107.89	0.115	1.94	65.20	0.121	1.64	18.45	0.144	1.06
CL	39.96	0.216	-1.53	26.87	0.248	-3.03	10.00	0.247	-7.80
MI	117.94	0.347	4.13	73.33	0.326	2.77	24.04	0.261	-1.17
WE(A)	31.93	0.147	1.63	19.99	0.160	0.69	5.83	0.176	-2.43

Table 9.5 This table shows the effect that the elastic modulus ratio (M.R.) has upon the contraction coefficient (C.C.) and the percentage length reduction (% L.R.). The elastic modulus (E.M.) used is also given.

The results in the above table are quite interesting in that the elastic modulus ratio has a significant effect upon the deformation of the model. This is despite the active part of the model being identical to the previous tests and the overall stiffness being kept the same as the reference data. The elastic modulus ratio must obviously affect the way in which the stresses are distributed through the wall which in turn results in more or less stress being distributed longitudinally rather than radially. The change in length appears to be far greater than that experienced during the diastolic work in Chapter 6. This is most likely to be due to the greater stresses which occur during systole compared to diastole. The conclusions as to what effect the elastic modulus ratio has upon the model during simulated systole are as follows:

Firstly, it seems that the lower the elastic modulus ratio the greater the length reduction (or the smaller the length increase). With the exception of "BA" all the models show a very significant length change variation between the extreme values taken for the elastic modulus ratio The length change variation is less than 1% for "BA", where as for "CL" the change is greater than 6%.

Secondly, the elastic modulus ratio has a significant effect upon the contraction coefficient. All of the patients with the exception of "MI" experience a reduction in contraction coefficient when the modulus ratio is reduced. It can be concluded from this that a low elastic modulus ratio can in some cases be beneficial to ventricular performance, while in others a high ratio may be beneficial. This may help to explain why there is such a large disparity in the values obtained experimentally from excised cardiac tissue. Whether a high or a low elastic modulus ratio is more desirable for greater ventricular efficiency is likely to be dependent upon one or more of the following shape, cavity pressure and the other material properties. Of these possible contributing factors it is likely that shape is the most significant. This is because shape is the only thing which differs greatly between "MI" and "BA", which exhibit markedly different responses with respect to the elastic modulus ratio.

9.5.3 Modulus of Rigidity

The muscle fibres in the myocardium are held together by a mesh of microfibrils. It has been hypothesised that these are fairly loose throughout the cardiac cycle, with the exception of end diastole when they are thought to become taut (Horowitz *et al.* 1988 [14] and Borg and Caulfield 1981 [13]). This loose coupling of the fibres, if it were the case, is likely to have the effect of reducing the effective shear modulus of the bulk material in the along-the-fibre direction. It is also likely that the effective shear modulus of the material will be reduced in the across-the-fibre direction also. The net result of these reductions in the shear modulus is likely to allow the individual muscle fibres to contract more freely as they would no longer be prevented from sliding over each other.

In this case the shear modulus is in effect dictating how readily the muscle fibres can slide over each other. This should manifest itself as the model requiring a significantly smaller contraction coefficient for a given volume reduction. It is however not clear whether there will be any effect upon the global deformation of the model. Another thing that must be borne in mind is that, as with altering the modulus ratio, altering the shear modulus will have a major effect upon the wall stiffness. The elastic modulus is therefore adjusted in the same way as was necessary in the elastic modulus ratio tests. Below are two tables showing the effect upon the length and upon the contraction coefficient of altering the shear modulus.

	Shear modulus $G^*/1000$			Shear Modulus $10\ G^*$		
Patient	E.M.	C.C	% L.R.	E.M.	C.C.	% L.R.
BA	150.00	0.118	1.99	30.43	0.170	1.15
CL	45.17	0.234	-3.10	13.26	0.255	-8.11
MI	255.31	0.293	5.37	28.73	0.270	-2.96
WE(A)	38.70	0.121	0.74	8.34	0.163	-1.41

Table 9.6 This table shows the effect that altering the longitudinal shear modulus has upon the contraction coefficient (C.C.) and the percentage length reduction (% L.R.). The elastic modulus (E.M.) used is also given. Here G^* is the normally used value for the shear modulus and is given by G^*=E.M./1.47 in the longitudinal fibre direction.

There is certainly evidence that the shear modulus can affect the way in which the model performs under systolic conditions.

Firstly, all the patients show there is a tendency for the model to reduce in length more (or lengthen less) when the shear modulus is smaller. There is also a tendency for the model to lengthen more (or shorten less) when the shear modulus is higher. The data set "BA" shows quite a small variation in length change between the high and low shear modulus values. All the other data sets show a large variation in the length changes between the extreme values of the shear modulus.

Secondly, with the exception of patient "MI" all the contraction coefficients are smaller when the shear modulus is smaller and higher when the shear modulus is higher. It would appear that for some patients at least, a low shear modulus may be beneficial by reducing the magnitude of the contraction coefficient required. This in turn translates into less force being required to generate the same cavity volume reduction.

	Shear modulus $G^*/10000$			Shear Modulus $10\ G^*$		
Patient	E.M.	C.C	% L.R.	E.M.	C.C.	% L.R.
BA	632.42	0.131	2.58	25.82	0.168	0.57
CL	131.60	0.155	0.59	10.93	0.335	-13.57
MI	2962.7	0.119	2.20	21.79	0.257	-0.17
WE(A)	152.85	0.102	1.16	6.88	0.172	-1.17

Table 9.7 This table shows the effect that altering both the longitudinal and cross-fibre shear moduli has upon the contraction coefficient (C.C.) and the percentage length reduction (% L.R.) Here G^* is the normally used value for the shear modulus and is given by; G^*=E.M./1.47 in the longitudinal fibre direction and G^*=E.M./2.94 in the radial fibre direction.

On the whole the results in this table are consistent with those of the previous table and do not require reiteration. There are however some anomalies that do require comment. The data sets "BA" and "CL" experience a greater effect upon their length change than when just the longitudinal shear moduli are changed. This would seem reasonable as greater capacity for the fibres to "slide over each other" is introduced. The data sets "CL" and "WE(A)" both show a much greater reduction or increase in their contraction coefficients than when only the longitudinal shear moduli are altered.

The inconsistency in the effect of the shear moduli upon the deformation of the models would appear to be most likely due to the different shape derived from each set of patient data. It is however likely that the contraction of the left ventricle in at least some cases may be more easily facilitated by having low shear moduli.

9.5.4 Poisson's Ratio

It is for completeness only that the effect of Poisson's ratio is included here. It is not expected that it will have a significant effect upon ventricular deformation. In diastolic investigations Poisson's ratio dictates how compressible the model is in terms of its wall volume. In systole the function of Poisson's ratio is likely to be swamped by the effect of the contraction coefficient ratio which is used to regulate the volume strain of the model. Since the myocardium is normally considered to be virtually incompressible, mainly due to its high water content, Poisson's ratio is normally taken as being close to 0.5. Thus when choosing some suitable values to compare the effect of Poisson's ratio we need not depart far from a value of 0.5. For the above reason a lower value of 0.40 and a higher value of 0.49 were used, the results of which are tabulated below.

Patient	Contraction Coefficient	Contraction Coefficient Ratio	Deformed Shape Index	Deformed Length /(mm)	Percentage Length Reduction
BA	0.129	-0.450	0.852	74.74	1.29
CL	0.255	-0.427	0.847	86.35	-4.25
MI	0.292	-0.443	0.569	76.39	1.54
WE(A)	0.162	-0.426	0.669	73.32	-0.25

Table 9.8 This table shows the effect that taking Poisson's ratio as 0.4 instead of the normally used value of 0.47.

Patient	Contraction Coefficient	Contraction Coefficient Ratio	Deformed Shape Index	Deformed Length /(mm)	Percentage Length Reduction
BA	0.126	-0.459	0.852	74.77	1.25
CL	0.262	-0.438	0.844	86.60	-4.55
MI	0.298	-0.446	0.569	76.41	1.51
WE(A)	0.164	-0.435	0.666	73.41	-0.37

Table 9.9 This table shows the effect that taking Poisson's ratio as 0.49 instead of the normally used value of 0.47.

As was to be expected the effect of Poisson's Ratio is much smaller than the other parameters studied so far and there is little deviation from the reference values (Table 9.2). The effect of Poisson's ratio is completely swamped by the other much more significant parameters. Thus it would seem reasonable to exclude this parameter from any consideration of the factors which may affect global deformation during systole.

9.6 Boundary/Fixation Conditions

The treatment of the boundary conditions is a cause for concern in any finite element model but is more so in biological situations. The reason for this is that in general, and in this case in particular, there are no solid boundary conditions. The heart is held non-rigidly in several places and this is what prevents excessive rigid body movement The finite element method demands that the model be prevented from rotational or translational rigid body movement and so artificial restraints are imposed upon the model in order to prevent this. The model has previously only been restrained at the base where all the endocardial points are prevented from movement in any direction. Part of the reason for this set of boundary conditions is to simulate the valve ring which is extremely stiff. In late diastole when volume changes are much smaller than in systole there is probably no significant effect from this assumption. In systole however the volume changes are far greater and there is some contraction as well as shape deformation of the valve orifice.

In an attempt to assess the effect that previously used boundary conditions had upon ventricular contraction another set of boundary conditions was tried. The apex in our model is extremely stiff and as such experiences negligible strain. The restraints for the model were applied at the top edge of the apex since, as this is already unable to deform, it will not be affected by these restraints. This allowed the base of the ventricle to contract freely and the effect of the base constraints to be easily assessed.

Patient	Contraction Coefficient	Contraction Coefficient Ratio	Deformed Shape Index	Deformed Length /(mm)	Percentage Length Reduction
BA	0.155	-0.463	0.872	73.69	2.67
CL	0.216	-0.436	0.861	86.02	-3.85
MI	0.348	-0.399	0.606	76.50	1.71
WE(A)	0.201	-0.471	0.706	71.66	2.02

Table 9.10 This shows the changes in length and shape index that result from removing the restraints at the base and placing them at the non-deformable apex.

On comparing Table 9.10 with the reference results in Table 9.2 there are several observations to be made. Firstly, the contraction coefficients for three of the patients are greater than for the reference results. This is unexpected as it would be expected that rigidly holding the base would increase stiffness. The cause of this might be that the cavity pressure has greater effect at the base than does the active contraction. It may be that this is an artefact of the pericardial reconstruction algorithm which tends to make the myocardium slightly thinner at the base. This would result in smaller contraction forces and a larger effect due to the cavity pressure. Another possibility is that the model with an unrestrained base deforms in a less efficient manor, for some reason, compared to when it is restrained. There is also some change in the shape index which increases for all of the patients. It is also noted that some patients show a greater change in shape index compared to the reference data while others experience a smaller change. This would suggest that the ventricles are becoming more spherical relative to the reference results. The change in length is also interesting, all the patients experience a relative length reduction compared to the reference data. This change is quite pronounced for all of the data sets with the exception of "MI" for which the change is only 0.64%.

It is important that we know exactly what is happening at the base. Below the original and deformed endocardial surfaces of the finite element meshes are overlaid for each patient. It can be seen that the base areas show large displacements, these displacements are far greater than are required. It is obvious that some form of restraint is required at the base. It is also clear that completely restrained is far better than completely free. It may be worth considering, at some future time, having semi-rigid restraints to simulate the mitral valve ring.

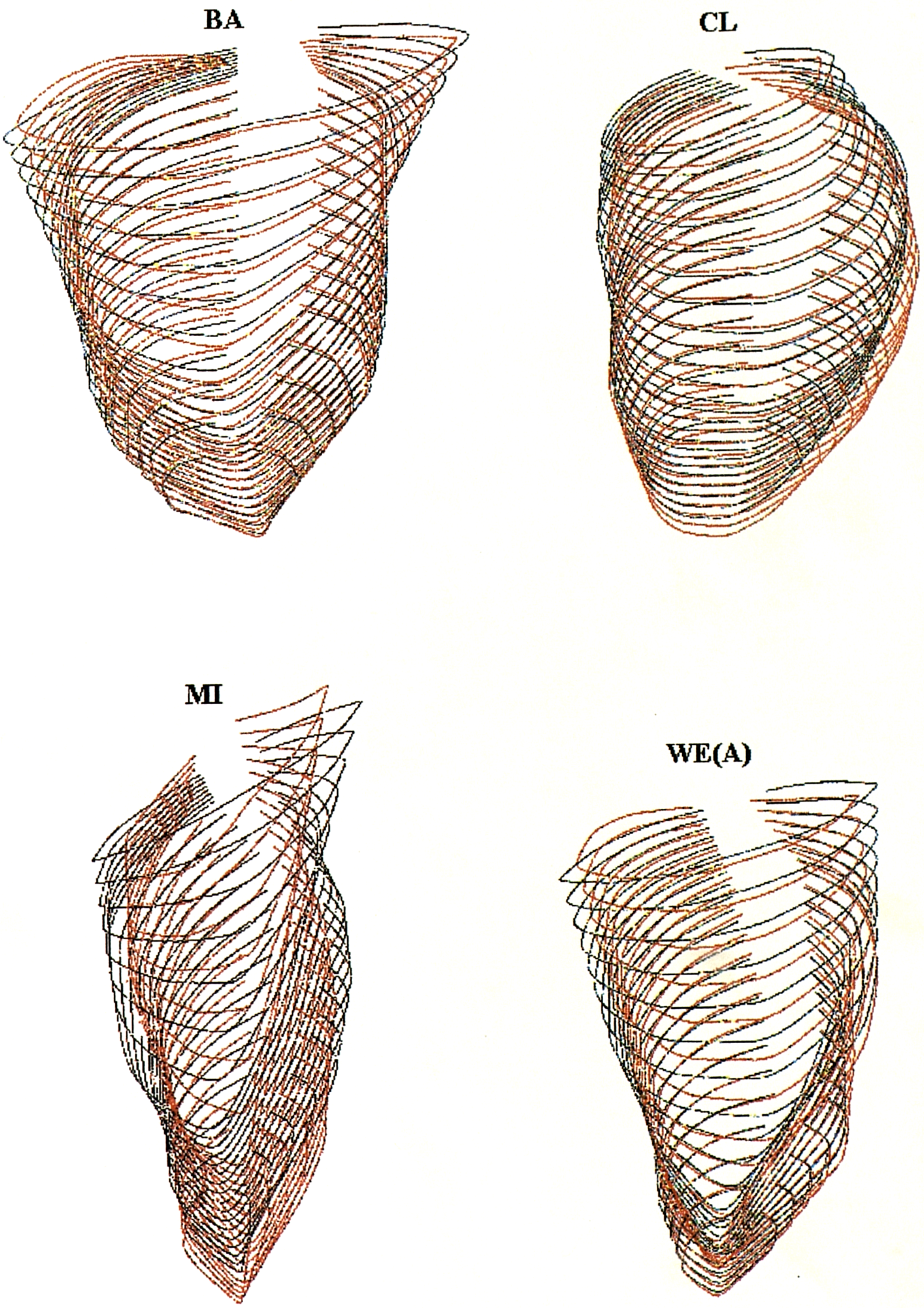


Fig 9.3 These diagrams show the endocardial surface of the deformed (red) and undeformed (black) F.E. meshes. They illustrate how the base region deforms when there are no restraints at the base.

The conclusion from this test is that the way in which the ventricle is held can have a large effect upon global deformation.

9.7 Endocardial and Epicardial Fibre Angle

From the work of Streeter *et al.* (1969 [9]) it has been shown that in systole the fibre angle at the epicardium and endocardium can be as much as -90° and 90° respectively. It should be obvious that the steeper the fibre angle is the more contraction is directed in the longitudinal direction. Thus the model should show a greater tendency to shorten. The model was run with epicardial and endocardial fibre angles of -70° to 70°, -80° to 80° and -90° to 90°. The deformation results are then compared between each other and the standard -60° to 60° angles used in the reference data (Table 9.2).

	60° to -60°		70° to -70°		80° to -80°		90° to -90°	
Patient	C.C	%L.R.	C.C.	% L.R.	C.C.	% L.R.	C.C.	% L.R.
BA	0.121	1.21	0.156	1.63	0.200	2.09	0.290	2.91
CL	0.252	-4.81	0.279	-3.08	0.325	-1.57	0.440	-1.38
MI	0.295	1.27	0.339	1.80	0.412	2.68	0.546	3.93
WE(A)	0.158	-0.35	0.185	0.40	0.226	1.22	0.293	2.19

Table 9.11 This shows the effect upon the length change that the epicardial and endocardial fibre angle has. (C.C. is the contraction coefficient and % L.R. is the percentage length reduction.)

From the results in the table it is clear that the orientation of the muscle fibres has a great influence upon global deformation. It is not a surprise that increasing the relative directional component of the fibre in the longitudinal direction increases the shortening of the model (or reduces the length increase). It is however of some surprise that even with the fibre angle change at the maximum possible there is still a length increase with the data set "CL". This is despite the length reduction being cut from -4.81% in the reference test to -1.38%.

9.8 The Effect of the Wall Volume Reducing During Systole

We will now try to assess the importance of wall volume strain in the deformation of the left ventricle during systole. Volume strains of approximately 10% have been seen to occur during the cardiac cycle (Horowitz *et al.* [35] and Yin *et al.* [82]). This wall volume strain is due to the movement of the intracoronary blood. When the myocardium is subjected to a load some of the blood within it will be forced out and the total volume of the myocardium will be reduced. It would seem reasonable to assume that the majority of this occurs during isovolumetric contraction since this is the phase when the loading on the myocardium increases the most. If this is the case then there will be little intracoronary blood forced out during the ejection phase, which is currently being investigated.

Taking into consideration the above comments it would seem reasonable that between the last two frames of systole the reduction in wall volume is likely to be far less than 1%. We will now investigate the effect of a -1% wall volume strain over the last two frames of systole. The results of this analysis are presented in the table below.

Patient	Contraction Coefficient	Contraction Coefficient Ratio	Deformed Shape Index	Deformed Length /(mm)	Percentage Length Reduction
BA	0.107	-0.418	0.852	74.71	1.33
CL	0.237	-0.421	0.845	86.34	-4.24
MI	0.288	-0.428	0.574	76.35	1.59
WE(A)	0.150	-0.402	0.670	73.10	0.05

Table 9.12 This table shows the effect of a -1% wall volume strain upon the deformation of our model.

Clearly, even taking a possible unrealistically high value for the wall volume strain, the effect upon the model is extremely small. It would thus seem reasonable to conclude that these small reductions in wall volume between successive frames are unlikely to greatly alter the deformation of the left ventricle. If however the 10% wall volume reduction was to occur over a relatively short part of systole then it may play an important role in that part of the cardiac cycle.

9.9 Assessment of the Relative Effect of the Papillary Muscles

It has been stated that the observed length increase of a previous model (Horowitz *et al.* [35]) was due to the papillary muscles being neglected. So far by altering the various material parameters it has been possible to alter the length changing properties of all the sets of data. It is impossible to gauge exactly how much force may be generated by the papillary muscles and whether this can have a significant effect on length changes of a model. It is however possible to place some bounds on how great this force can be. For example the force on the valve leaflets due to the action of the papillary muscles must be less than that resulting from the pressure differential across the left ventricle and the left atrium. As an extremely coarse approximation to this value we shall reverse the pressure on the apex elements. The results are given in the table below.

Patient	Contraction Coefficient	Contraction Coefficient Ratio	Deformed Shape Index	Deformed Length /(mm)	Percentage Length Reduction
BA	0.095	-0.443	0.849	74.61	1.46
CL	0.213	-0.286	0.854	89.55	-8.11
MI	0.292	-0.440	0.572	76.47	1.43
WE(A)	0.140	-0.433	0.681	72.61	0.71

Table 9.13 This table shows the effect of trying to simulate the papillary muscles by reversing the pressure on the apex elements.

For three of the data sets there is a reduction in length relative to the reference data (Table 9.2). This reduction although significant is smaller than that produced by other methods. That is not to say that the papillary muscles do not play an important role only that their effect may be easily accounted for by altering the other properties of the model. All the models show a reduction in the required contraction coefficient. This reduction is significant for all the data sets except "MI". This is strange since "MI" experiences a larger length change than this negligible contraction coefficient change would otherwise suggest. The model produced from the data "CL" behaves contrary to what would be expected. The reason for this is that "CL" produces a model which has a "bend" in it. The result of this is that reversing the pressure on the apex does not produce a longitudinal force. If one wished to perform a test for the effect of the papillary muscles on "CL" the direction of the force would require more care in defining its orientation.

9.10 The Effect of the Cavity Pressure

To some extent altering the elastic modulus has the effect of increasing or reducing the significance of the cavity pressure loading on the model. As such the effect of removing the cavity pressure should follow logically from the results of section 9.5.1. The table below shows the effect of removing the cavity pressure.

Patient	Contraction Coefficient	Contraction Coefficient Ratio	Deformed Shape Index	Deformed Length /(mm)	Percentage Length Reduction
BA	0.062	-0.469	0.846	74.92	1.04
CL	0.126	-0.468	0.862	84.36	-1.85
MI	0.288	-0.444	0.571	76.53	1.36
WE(A)	0.067	-0.462	0.678	72.71	0.58

Table 9.14 This table shows the affect on the deformation of model on removing the cavity pressure.

From the table there are several points that can be made:

Firstly, the cavity pressure causes the models using the "CL", "MI" and "WE(A)" data to lengthen during simulated systole relative to the reference results in Table 9.2. Patient data "BA" gives the reverse of this.

Secondly, "CL" and "WE(A)" which have the two lowest elastic moduli are affected most by the cavity pressure. The other two with the greater wall stiffnesses have their length change barely affected.

Thirdly, the proportion of the contraction coefficient required to overcome the cavity pressure is approximately half for "BA", "CL" and "WE(A)", for "MI" on the other hand only a small fraction is required.

It is certainly the case that the cavity pressure does in some cases have a significant effect upon the deformation of the model and that the amount of this effect unsurprisingly is dependent upon the wall stiffness and cavity pressure of the model.

9.11 Summary

It is perhaps not unexpected that there are many factors which can have a significant effect upon the way in which our model deforms. It is however unexpected that some data sets should respond more strongly than others to these different factors. The single factor which appears to show the greatest effect and consistency of effect upon the deformation of the model is the fibre angle. The patient data "CL" however responds more strongly to a change in the shear modulus used. This is something that will have to be borne in mind in the later work and maybe combined with a high fibre angle. Patient "MI" also experiences significant length changes when the other parameters are varied. The length reductions in this case are likely to be too great for this patient, so whichever parameters cause a greater reduction in length is largely irrelevant.

The contraction coefficients obtained in this Chapter without exception appear excessively high. Since they are effectively strain potentials one might expect the contraction coefficients to be of a similar order of magnitude to the fibre strains which in turn should be lower than the relative volume reductions. Part of the reason for this is likely to be due to the fact that large elements are being used to simulate the myocardium. In reality the myocardium is formed from individual fibres which are able to slide "freely" over each other. This is not possible to incorporate into a model like this without greatly increasing the number of elements. There was some evidence that this was the problem when looking at the effect of the shear modulus. Even though reducing the shear modulus should enable some “pseudo-sliding” effect it is likely to not be remotely as great as that available in the real ventricle.

Another interesting point is that perhaps the shape index is not a good way measure global deformation, as it can vary very little when there is a large change in length and vice versa. Since the shape index depends upon surface area and volume it may be that it is more sensitive to surface smoothness than the overall shape.

Chapter 10

10. Simulating the Left Ventricle Over End Systole

In this Chapter the information gained in the previous Chapter will be used to simulate systole for several frames of data. The analysis procedure will be discussed as well as attempts made to simulate the observed long dimensional changes. This will allow for some understanding to be gained as to how the strength of myocardial contraction varies during systole.

10.1 The Analysis procedure

The process originally to be used for analysis of systole was similar to that of "smoothed volume matching" which is described in Chapter 3. The main difference is that the volumes here are decreasing and not increasing as is the case in diastole. The first attempt at systolic modelling involved starting at the onset of ejection. This however resulted in the finite element mesh becoming badly deformed and element integrity being severely compromised long before end systole was reached. For some patients it was not even possible to simulate systole over the first time step. It was decided that analysis would have to be restricted to a smaller region and end systole was chosen.

The graph of pressure verses time for patient "BA" shows a steep rise in pressure for the first part of systole and then an abrupt change to a less rapid rise in pressure, Fig 10.1. The area chosen for investigation was this second part. There are two main advantages in choosing the latter part of systole for investigation. Firstly, the ventricle is smaller and any interaction with the pericardium is likely to be much less significant. Secondly, the early stage of ventricular electrical activation is not considered so the effect of partial and progressive activation will be less significant and it is thus more reasonable not to model this. These two things are normally ignored when investigating left ventricular systole but at least here it can be done with some justification.

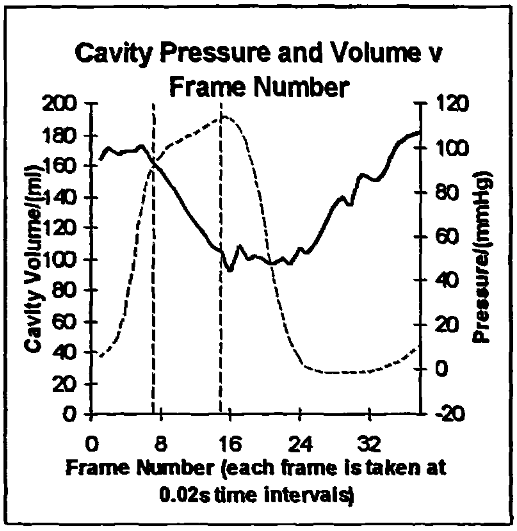


Fig 10.1 This graph shows the cavity pressure (broken line) and volume (solid line) plots for "BA". The area of interest is late systole (ejection) where pressure is rising and volume is falling. This is the area between the two vertical dashed lines (frame 7 to 15).

Since the same mesh could not be used to simulate the whole of systole, a method similar to that of "target volume matching" was employed. Unlike the work on the analysis of diastole there is no need to generate "target volumes" or to smooth through the volume data. This was only necessary in diastole as the volumes were required to always be increasing and the measured values were not always increasing. In systole the volume is always decreasing, as required for the analysis. The volumes also form a smooth curve to which further smoothing would add very little.

It should be mentioned that it was not possible to analyse all the patients which had been previously analysed for their diastolic properties. It was discovered that the method used to create the FE mesh was not always able to produce one which was suitable for analysis. The FE program would return node connection error messages for some of the elements. This means that the elements which are defined by the FE mesh in some instances are not valid elements. The nodes defining each element must describe an element of hexahedral topology. If this is not the case then a node connection error message is given. This was found to result from a number of different reasons Firstly, in systole the X-ray outlines are not only smaller but also tend to have much more convoluted outlines than in diastole. This can result in poorly formed elements Secondly, the base of the ventricle in several of the patients was seen to noticeably tilt during systole. This resulted in some of the elements which form the base being badly formed. Thirdly, as has previously been discussed, the LAO image is not a longitudinal image but in fact has a significant short axis component. In some patients the short axis component may be dominant. If this is the case then the reconstruction algorithm produces a shape which can be quite cylindrical. This can also result in some elements being poorly shaped due to the way in which the mesh is formed.

The first two problems may be solved by employing a far more sophisticated reconstruction algorithm. Unfortunately the third problem cannot be addressed by any means other than by obtaining more suitable data. It is true that by employing a more sophisticated meshing scheme the elements could be made more robust. This does not however help with the fact that the shape will be a very poor representation of the original left ventricle.

10.2 Matching Cavity Volume While Keeping Myocardial Volume Constant

It would be preferable to be able to keep the same finite element mesh for the entire cardiac cycle. This would mean that the effect of the simulation process could be easily assessed by comparison of the deformed model at the end of the simulation of systole to that produced from data for end systole. This was unfortunately not possible due to the cumulative distortion of the finite element mesh which resulted in the integrity of some of the finite elements becoming severely compromised.

The process of "target volume matching" introduced in Chapter 5 would have to be employed. Since the finite element mesh is renewed at each time step, which is over 0.02 seconds, there is now no propagation of the distortions in the finite elements. This does, in most situations, resolve the problem of excessive mesh distortion. On the down side it means that there is also no cumulative change in the model's geometry, with which to compare to the reconstructed end systolic data.

What is required is a method to sum all the geometric changes that occur between one frame to the next frame throughout the cavity volume matching process. A simple and straightforward method is to be preferred. The method chosen is to sum all the percentage length changes which occur when matching one reconstruction to the cavity volume of the next. This overall measure can then be compared to the percentage length change that is observed from the reconstructed X-ray data.

The table below shows how these summed length changes compare to the measured length changes.

Patient	Start	End	E.M.	M % LR	C % LR
BA	7	15	50	6.25	-6.03
CE	8	13	150	2.26	8.39
CL	8	15	20	10.10	-43.52
CL(A)	7	14	30	5.52	-35.24
MI	9	16	60	9.66	6.11
WE(A)	9	16	15	13.29*	-12.04

Table 10.1 This table shows the summed percentage length changes (C % LR) which occur when simulating left ventricular sy ole. These are presented with the length changes measured from the reconstructed finite element mesh (M % LR). Other properties of interest are also presented in this table. The assumed value for the elastic modulus (E.M.), in KN/m^2, is taken to be a representative value for the diastolic period. (* frame 10 of WE(A) was unusable so the M % LR was reduced accordingly from 15.51)

As can be seen there is a large spread in the measured left ventricular length changes. The lowest change being only 2.26% and the largest being 13.29%. This is interesting since it clearly shows that ventricles from different patients deform differently. Some of this difference is likely to be due to the pathologic conditions suffered by each patient. Part of this variability may also be attributed to natural variability in the normal function of human left ventricles of different shapes.

There is also a large spread in the values for the percentage length changes that occur with the model. It is perhaps of some concern that there is such a vast difference in the observed and calculated length changes. In all but two cases the model predicts substantial length increases when length reductions are indicated from the patient data. One patient who exhibits a length reduction exhibits one that is far greater than that required. It is obvious that something must be done to ensure that the model more closely simulates the observed ventricular behaviour.

10.3 The Effect of Fibre Direction Upon Global Deformation

From the previous Chapter it was discovered that the parameter which had the most significant effect upon the length change of the model during simulated systole was that of fibre angle. Here we shall use different fibre angles to alter the way in which the length of the model changes during the simulation of systole.

To show the effect of fibre angle upon the deformation of the model, the model will be used to simulate systole using various fibre angles. The results of these are tabulated below.

		Fibre angle/(degrees)					
Patient	M %L.R	65	70	75	80	85	90
BA	6.25	-3.92	-1.74	0.44	2.69	5.30	9.61
CL	2.26	-36.74	-29.46	-22.03	-14.88	-8.31	-2.84
CL(A)	10.10	-30.00	-24.40	-18.78	-13.26	-7.91	-3.06
CE	5.52	8.85	9.27	9.80	10.19	10.46	10.66
MI	9.66	6.57	7.28	8.24	9.50	11.00	12.87
WE(A)	13.29	-9.30	-6.36	-3.01	1.99	4.23	8.03

Table 10.2 This table shows the effect of fibre angle upon the percentage length “reduction” of the model. The length reduction measured from the start and end systolic frames (M %L.R.) is also given for comparison.

The table shows that the cumulative affect of fibre angle over the end of systole can be quite significant. Patient data "CL" and "CL(A)" show a massive change in their length deformation. This however is not great enough to make the length contract as is observed from the X-ray data. An attempt was made to use the shear modulus to further increase the length reduction of patient "WE(A)" and cause a length reduction for patient "CL" and "CL(A)". Unfortunately the effect of this parameter had been vastly reduced by increasing the fibre angle. It was thus not possible to use the shear modulus and fibre angle in conjunction with each other when attempting to alter the deformation of the model. Patient "CE" decreases in length far more than is required. In order to obtain a more accurate length change the fibre angle was reduced. This however had little effect and therefore these other values are not presented here.

The result of this is that one of four possibilities must be occurring. One, the method used is completely unsuited to the simulation of cardiac contraction. Two, the method used to simulate the fibre contraction results in a diminished ventricular length contraction component. Three, there is some other external factor contributing to the length contraction of the ventricle. Four, the muscle fibres at different depths within the wall may contract with significantly different strengths. The result of this could be that some of the fibres in a more longitudinal direction may be contracting with greater force than those in a more circumferential direction. This would result in a disproportionate amount of longitudinal contraction than would presently be predicted by the model as it is currently employed.

Taking the above points in order. One, it would seem unlikely that the method used to simulate cardiac contraction is completely without merit. The reasons for this are that from the work of Chapter 9 it can be seen that at least in some cases this method can produce results that are quite realistic in terms of global deformation.

Two, it is certainly possible that there may be some problem with the direction and force calculation on the nodes of the finite element mesh. This could lead to forces in the longitudinal direction being reduced in magnitude. One possible reason for this may be the use of the varying fibre angle through the finite element. During systole the elemental forces are much larger than in diastole and the coarseness of the approximation achieved with this sort of element may give rise to an unrealistic nodal force distribution.

The possibility of some external length contracting force may exist but as to what it could be there is little choice. Since the ventricle is becoming smaller the interaction of it with the pericardium is likely to be small and diminishing rapidly as the ventricle contracts. At the early stages of contraction it may be that the ventricle was constrained in the circumferential direction and thus initial contraction would be totally longitudinal. This however does not seem possible since there is no sign of high longitudinal contraction in early systole. It is possible that the papillary muscles contracting to prevent inversion of the mitral valve may provide a significant longitudinal contractile force. It has been observed, Gibson [83], that patients who have had mitral valve replacement and hence have had their papillary muscles cut do show some increase in their ventricular length, *though this still reduces during systole.*

It is certainly true that muscle contraction is not going to be constant throughout the myocardium, either from region to region or through the depth of the ventricular wall. It is however difficult to quantify these differences, although it is known however that strains at the inner surface are approximately half those at the out r surface, Gibson [83]. This unfortunately may be due to geometry rather than any indication of strength of contraction. From analysis of excised myocardium, Novak *et al.* (1994 [84]) , it has been shown that fibres at different depths have different stiffnesses. It would thus be quite likely that the contraction force generated at different depths within the myocardium may also be similarly non-uniform.

The progression of the contraction coefficient during systole appears to show no trend which is consistent between the patients. This may be due to such a trend not existing or that the data/model combination is not able to detect this trend. It is also the case that looking at the contraction coefficient from frame to frame is likely to be extremely sensitive to data accuracy. It may be better to look at the cumulative values which will be less affected by this. This is due to the fact that the overall discrepancy will be the mean of all the individual discrepancies. This averaged difference, if the differences are unbiased, will tend to zero as the sample size increases.

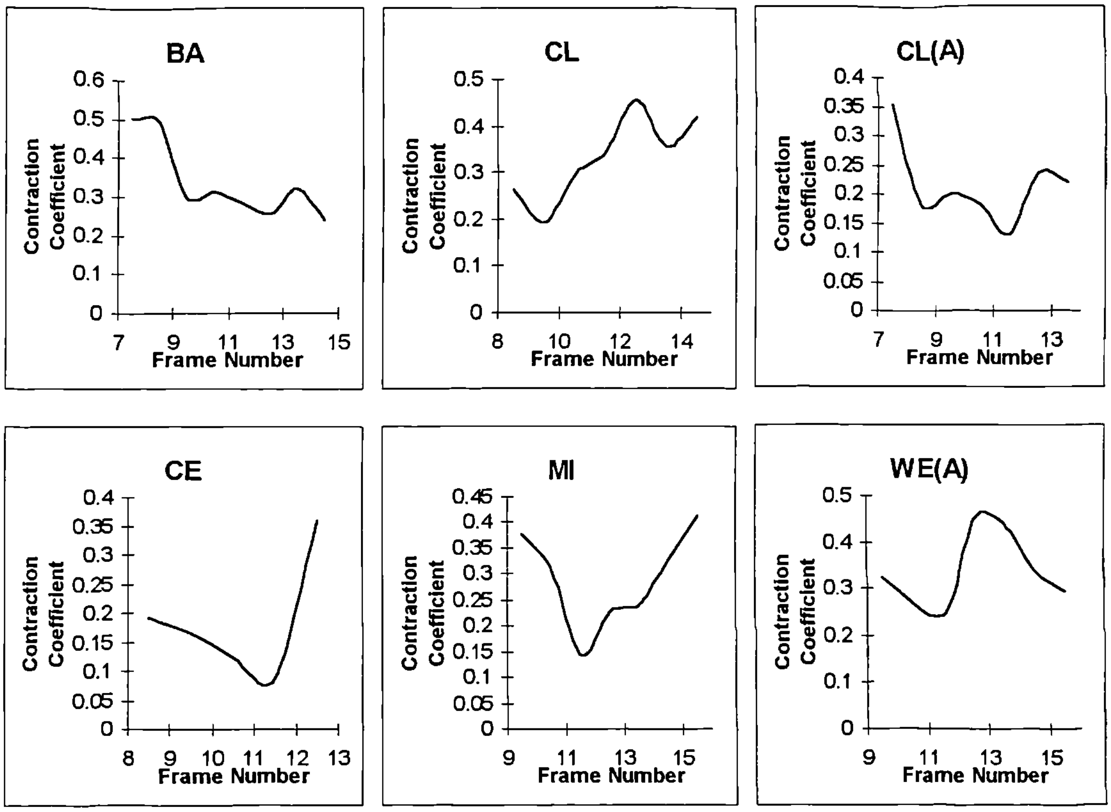


Fig. 10.2 These graphs show the calculated contraction coefficient at each frame for each of the six patients.

The graphs below show the total or cumulative contraction coefficient through systole. These cumulative values appear to show far more consistency between each other. The trend in these graphs seems to be fairly linear. There is however a large variation in the total contraction coefficient between each patient. Much of this difference is due to the fibre angle used for the patients. By increasing the angle the contraction coefficient required for the same volume reduction also increases. This is likely to be due solely to the coarseness of the varying fibre angle approximation and the restriction upon fibre movement within the elements. These are also possibly the reasons for the total contraction coefficients being so large. These large values go to illustrate why it would not be possible to use the same FE mesh for the entire systolic part of the cardiac cycle.

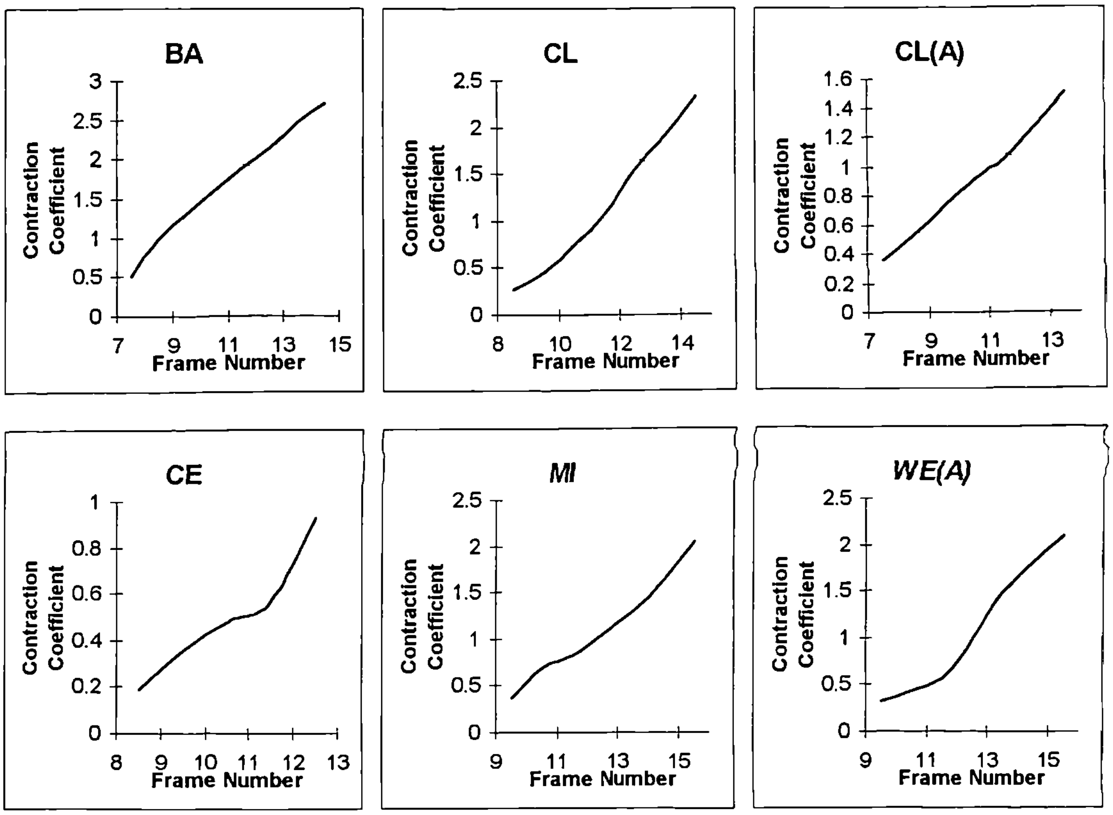


Fig. 10.3 These graphs show the cumulative or total calculated contraction coefficient at each frame for each of the six patients.

10.4 Length Variation During Systole

As well as looking at the overall measure of length change during the simulation of systole it is also worth looking at the length change from frame to frame. From the above table we shall choose the fibre distribution angle which gives the deformation that most closely mirrors the observed ventricular behaviour. The reason there may be some interest in observing the progression of change is that with the same patient data there is sometimes a length increase and sometimes there is a length decrease. This is often seen to happen with consecutive frames and it may be worth comparing these changes with those measured from the patient data. It must be noted that there are likely to be large discrepancies between the calculated values and the measured values. This is because the measured data is dependent upon the accuracy of the digitisation process. We are therefore only looking to see if there is any similarity in the overall trends in length changes between the measured and the calculated lengths.

The graphs below were obtained by using the fibre angles which were identified above as giving the most realistic length change.

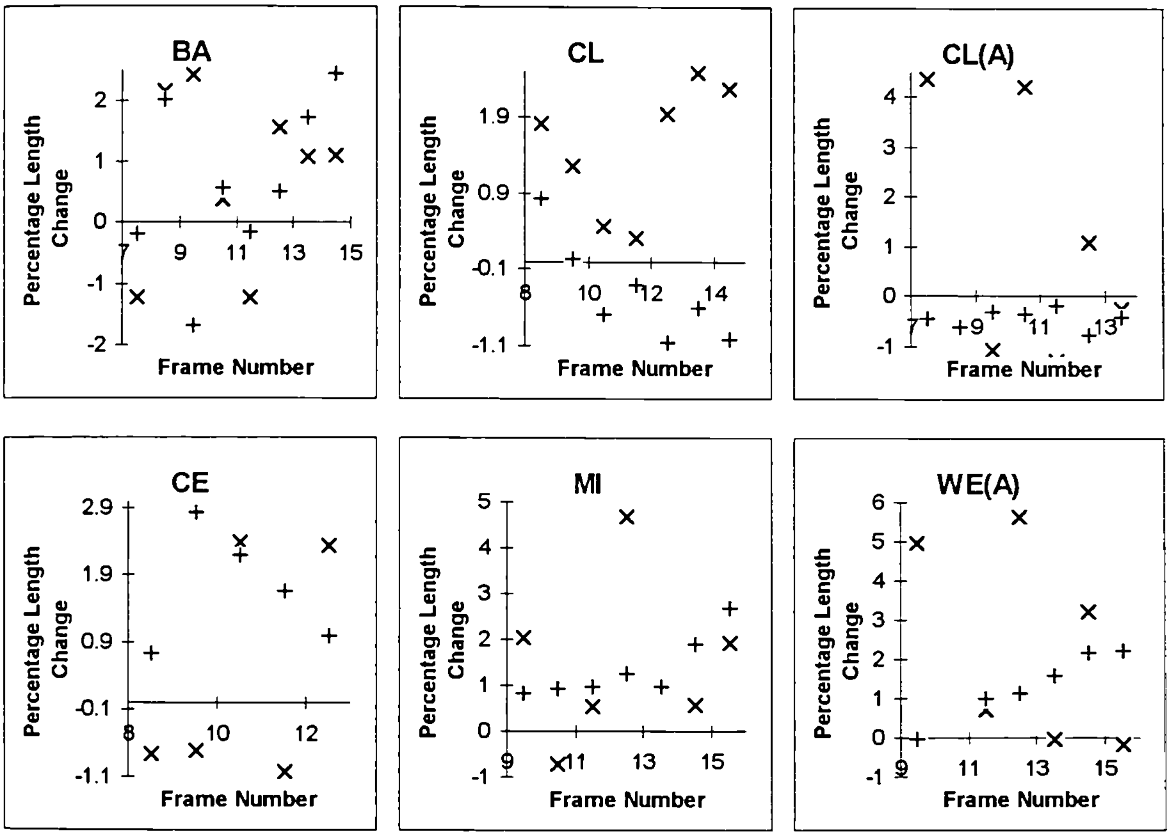


Fig. 10.4 These graphs show how the change in length at each frame compares between measured and calculated. The measured length change is denoted by an “x” and the calculated length change by a “+”.

There appears to a great deal of randomness in the length changes which occur between successive X-ray frames. This may in part be due to the accuracy of the length information obtained from the X-ray data. It may also be the case that the length of the left ventricle does not reduce continually during systole. One thing which may support this is the fact that the model predicts length increases and then reductions on successive frames.

Despite the fact that there is a high degree of scatter in the above graphs there does appear to be some level of correlation between the values obtained for patients "BA", "MI" and "WE(A)". The other patients would not be expected to show any level of agreement since the calculated total length changes were not close to those required. It is also of interest that the spread in the measured values in all cases is far greater than that in the calculated. This would lend credence to the hypothesis that the ventricle lengths are poorly reproduced by the digitisation process.

It does appear that when the total length change is adequately modelled then the changes between individual frames will also be comparable between the observed and the calculated.

10.5 The Effect of Pressure and Volume Changes

In this section we shall look at how the pressure increase and the volume changes contribute to the formation of the contraction coefficient. Again we will use for comparison purposes the fibre angles that were identified above as giving the closest fit to the observed ventricular deformation.

It is worth mentioning that the results in this section are largely invariant to changes in the fibre angle used in the model. This is of course not the case with length change or even the magnitude of the contraction coefficient.

The first stage is to see if there is any relationship between pressure and the contraction coefficient and volume change and the contraction coefficient. At this stage it would seem reasonable to expect that there may be some dependence of the contraction coefficient upon the cavity pressure, but at the same time it would be anticipated that it will be more strongly dependent upon the cavity volume change.

Below are plots of the contraction coefficient against cavity pressure.

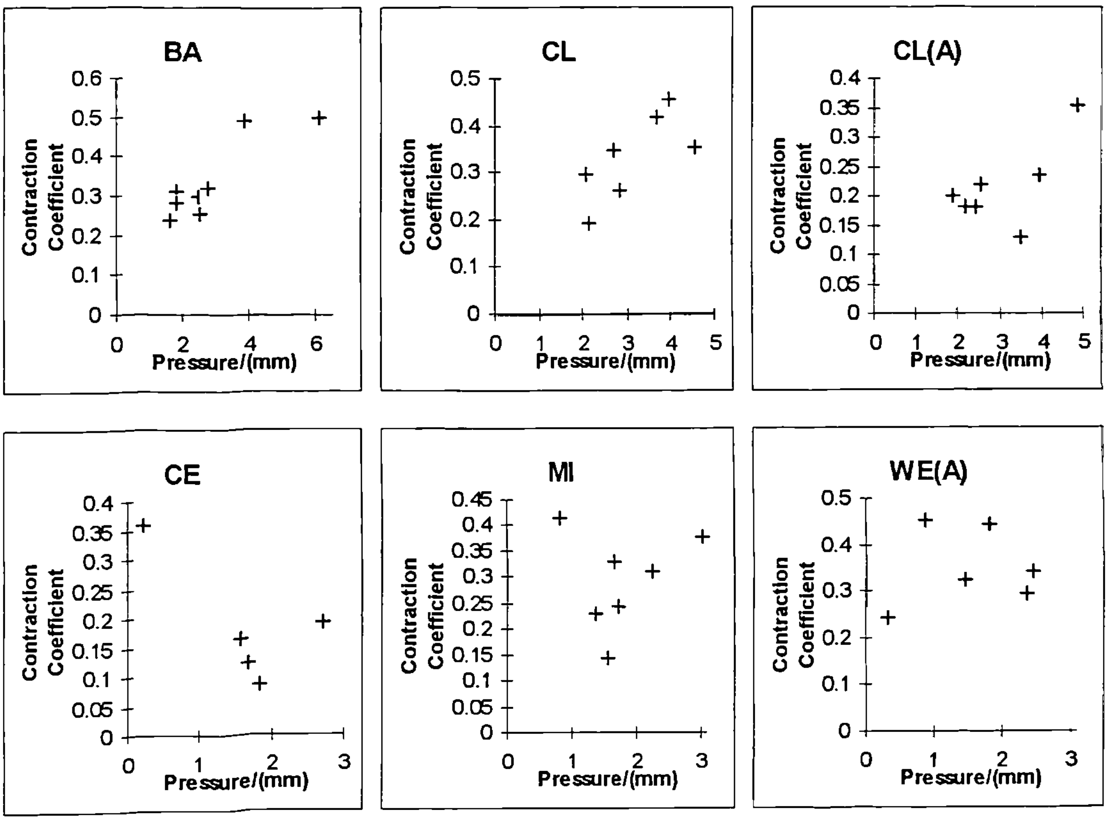


Fig 10.5 These plots show cavity pressure change versus contraction coefficient.

There appears to be some clustering of the data points. This would suggest that, in some cases, the contraction coefficient is at least partly dependent upon the cavity pressure increase. It should be obvious that if the stiffness of the left ventricle was an order of magnitude greater, as suggested by Lundin [61], then this dependence would be vastly reduced.

Below are plots of the contraction coefficient against percentage change in cavity volume.

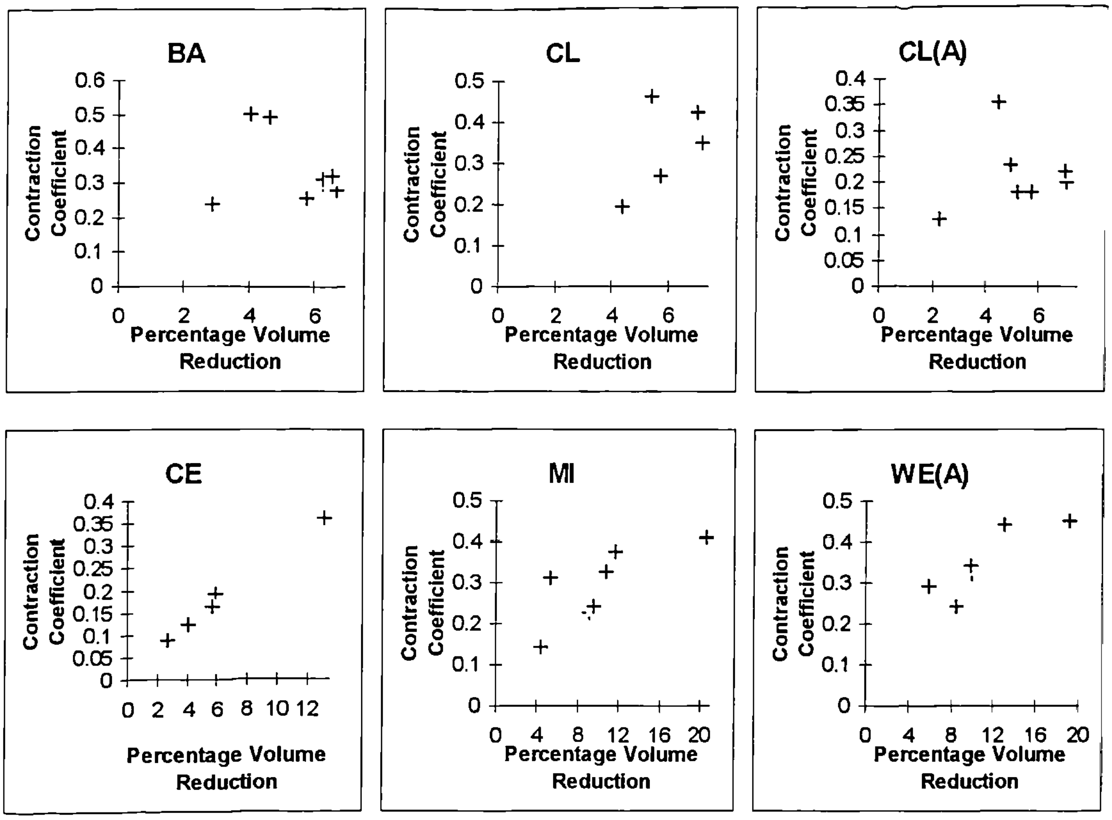


Fig 10.6 These plots show percentage change in cavity volume versus contraction coefficient.

It appears that, in some cases, the contraction coefficient is strongly dependent upon the percentage volume change. The last three graphs show a quite strong dependence upon the volume change and their dependence upon the cavity pressure is much more tenuous. It is thus likely that the contraction coefficient in general is partly formed as a result of the pressure increase and partly as a result of the volume change. If however the model was taken as being significantly stiffer then the contribution from the cavity pressure would in all cases become negligible. The contraction coefficient would then be almost totally dependent upon the volume change.

10.8 Summary

The method proposed to simulate the end of systole can provide a realistic simulation of performance.

When using an elastic modulus which is approximately the average diastolic value the contraction coefficient appears to have a dependence on both the volume change and the cavity pressure change. If however the stiffness of the model was increased by an order of magnitude as suggested by Lundin [61] then the dependence on the cavity pressure would be greatly reduced.

The fibre angle at the endocardium and epicardium can greatly affect the deformation of the model. However, the way in which the model deforms appears to have as much to do with the shape as any other factor.

Some patients could not have their length change sufficiently controlled, by varying the fibre angle, to simulate the observed length change. This may in part be due to the data/reconstruction giving rise to a model shape which will not perform in a realistic way.

Chapter 11

11. Factors that can Affect the Left Ventricle

This section will mainly be concerned with the effect of simulated pathologic disorders upon the left ventricle model. There will also be some investigation as to the effect of an uneven distribution of contractile force. As well as this an examination will be made as to how different model configurations affect the strain within the model and how these values compare with those known to occur within the human left ventricle.

11.1 Local Wall Strain

It is known that the sarcomeres which make up the muscle fibres are only able to contract by about 10 to 15% (Gibson [83]) *depending upon whether they are loaded or* not. A loaded sarcomere can only contract by as much as 10%. It would thus be of great interest to see if our model produces strains which are consistent with this. On the face of it, it would seem unlikely as the contraction coefficient values used in the model would indicate a value much higher than this.

The computer program, as part of its output, produces the strains in the fibre direction for each finite element. This gives rise to a large number of values which give the fibre strain for each corner node of each element. The number of values this generates is four hundred and eighty. The most obvious thing to do is to find the mean value of these. For frame fourteen of "BA" the mean strain value was found to be -4%. This is obviously too high as it is only for about one tenth of overall systole. The combined value for the whole of the analysed region of systole is approximately -50%. The question arises as to why this value is not only much larger than -10% but is also very much larger than would be expected due to simple geometrical considerations.

If it is correct that a sarcomere can contract by only 10%, in the loaded situation, the question becomes is the observed volume change consistent with this value? In any three-dimensional structure the volume and hence the volume change will be related to the cube of the linear dimensions. Thus if there is a 10% reduction in the length of the sarcomeres we would expect a 27% reduction in volume. In the left ventricle however the volume reduction may be in excess of 40% which implies a greater than 15% length reduction. The contradiction between the length reduction of the sarcomeres and the cavity volume change seems to be paradoxical. The way in which the left ventricle overcomes the geometrical considerations and achieves such a high output is far from clear. Two of the important factors are likely to be the change in ventricle shape and the twisting of the ventricle. By the ventricle changing shape the cavity volume can be altered without changing the surface area and hence the average sarcomere length. Likewise twisting of the ventricle may result in the contracting sarcomeres reducing the cavity volume more than would be achieved otherwise.

It is thus obvious that the change in the shape of the ventricle and the twisting of the epicardium relative to the endocardium are important factors to be considered.

11.2 Local Wall Strain and Fibre Slippage

The question has to be why does our model produce such high fibre strains? The logical place to start is to consider how our model differs from the real left ventricle. The most obvious difference is that the real left ventricle is composed f loosely bound muscle fibres. These fibres are also continuous along the length of the ventricle. Our model does not have this level of fibre independence. The fibres are modelled as solid blocks of anisotropic material with the direction of anisotropicity varying through the thickness of the block.

Several different arrangements for the muscle fibres were tried in order to assess which aspect had the greatest effect upon the average fibre strain. The table below shows the average fibre strain and the contraction coefficient for three of these different fibre configurations.

	Varying -85° to 85°	Fixed -45° & 45°	Fixed 0° & 0°
C.C.	0.238	0.565	0.059
A.F.S.	-0.043	-0.105	-0.023

Table 11.1 This table shows the contraction coefficient (C.C.) and the average fibre strain (A.F.S.) for various fibre configurations; a linearly varying fibre direction through the wall thickness and a fixed fibre direction within the epicardial and endocardial layers.

The above results were obtained using the data for patient "BA" at frame 14 and 15. From the above results it can be seen that when the endocardial and epicardial fibre direction is kept constant through the elements the fibre strain and contraction coefficient more than double. This shows that the interface between elements may be a significant contributor to the high values for the contraction coefficient and fibre strain. When the fibre direction is kept fixed through all the elements at an angle of zero degrees to the horizontal there is a quartering of the contraction coefficient and a halving of the average fibre strain. This value is much closer to what would be considered a reasonable value for the mean fibre strain.

The conclusion from this is that the model is inefficient at transferring the forces generated by the fibre strains into work done in reducing the cavity volume. This appears to be partly as a result of the simulated muscle fibres not being free to slide over each other. A model based on the same principles as this one, but using a loosely tethered set of concentric shells, previously proposed by Streeter *et al.* (1970 [21]) for a diastolic investigation, is likely to give far more realistic fibre stresses. That said, if one is only interested in global deformations the present model would seem to be adequate.

11.3 Greater Epicardial than Endocardial Contraction

It is known that the fibre strains in the endocardial layer are less than those in the epicardial layer (Gibson [83]). This may just be a result of geometry. It is however relatively straight forward to check to see whether there is any benefit to our model in having a lower endocardial contraction coefficient. The finite element analysis was performed for the last pair of systolic frames for each of the patients with the contraction coefficient of the endocardial layer being half that in the epicardial layer. The effect this had upon the model is given in the table below.

	Uniform C.C.			Half Inner C.C.		
Patient	C.C.	A.F.S.	%L.R.	C.C.	A.F.S.	%L.R
BA	0.238	-0.043	2.44	0.270	-0.040	2.79
CE	0.360	-0.065	0.99	0.601	-0.117	-5.05
CL	0.420	-0.084	-1.02	0.428	-0.067	-0.28
CL(A)	0.220	-0.026	-0.40	0.267	-0.025	-0.85
MI	0.413	-0.078	2.67	0.564	-0.078	4.62
WE(A)	0.293	-0.025	2.19	0.379	-0.025	3.04

Table 11.2 This table shows how having a higher epicardial than endocardial contraction coefficient (C.C.) affects the average fibre strain (A.F.S.) and the percentage length reduction (% L.R.). Note: the contraction coefficient given in the half inner test is the epicardial value.

There is some effect upon both the percentage length reduction and on the average fibre strain. This is relatively small in most cases and is certainly less significant than the previously investigated model parameters. The exceptions to this are patients “CE” and “MI” which show quite significant changes. The effect of uneven contraction in these cases is not consistent between the patients and this as before is most likely due to ventricular shape. It is certainly the case that partial and progressive contraction are important factors in the performance of the left ventricle. It is, in most cases, of only minor significance when considering global deformation using the present model.

11.4 Ischaemia and Scar Tissue

Ischaemia and scarring of necrotic myocardium are major conditions which affect the performance of the left ventricle. Ischaemia is depression of function by a loss or partial loss of blood supply to part of the myocardium. Since the muscle does not have adequate oxygen it is unable to generate the force which the unaffected myocardium can. If the ischaemia is prolonged then the affected myocardium will die and over a period of several weeks will be replaced by a stiff fibrous and non-contracting scar tissue. When blood supply is restricted the deeper (endocardial) region is affected first and this then spreads to the higher levels of the myocardium.

In order to model the effect of the above processes upon the myocardium two areas of the left ventricle were chosen for investigation. These two areas of the left ventricle are taken to be the anterior free wall and the posterior free wall. To model the effect of ischaemia the myocardium in the affected area is taken to be non-active and to have an elastic modulus the same as the rest of the ventricle and also the same test is done with an elastic modulus a tenth of this. The reason for taking two values is that, depending on the severity of the restriction in blood flow, the myocardium will either be unable to contract or may loose some of its stiffness.

Since scar tissue is stiff and non-contracting it is modelled as an area of increased stiffness. In the present study it is taken as being ten times stiffer than the normal myocardium.

Thus both conditions will be characterised by a non-functioning rea of the myocardium. The passive stiffness in the scarred area will be stiffer than the normal myocardium. In the ischaemic area the stiffness is likely to be reduced or of a similar value to the rest of the myocardium.

Using our model we will investigate the effect of simulating a large area of ischaemia and scarring upon the volume change of the model during systole. The simulation will be achieved by altering the material properties of the elements in these two regions. The size of the areas investigated will be approximately a third of the entire wall of the model. The results presented here were obtained using the last pair of frames and the active and passive material properties used in the previous chapter.

The analysis was performed when only the endocardial region was affected and also when the entire thickness of the wall was affected. The two tables below show the effect when just the endocardial layer is affected (Table 11.3) and when the full thickness of the myocardium is affected (Table 11.4)

	Starting	Normal	Anterior Free Wall			Posterior Free wall		
Patient	Volume	Volume	EM/10	EM	10 EM	EM/10	EM	10 EM
BA	107.19	104.08	106.34	105.00	104.62	105.24	106.58	107.18
CE	155.28	134.92	135.48	135.20	137.72	136.35	144.18	153.81
CL	122.72	114.28	118.84	117.05	111.38	117.86	115.28	115.60
CL(A)	150.03	139.58	142.92	141.94	140.92	140.60	142.28	144.88
MI	56.21	44.62	45.80	46.82	48.44	50.03	49.07	50.81
WE(A)	58.40	54.99	56.95	59.37	55.90	58.40	56.74	56.49

Table 11.3 This table shows the resulting cavity volumes for simulated ischaemia (EM/10 and EM) and scarring (10 EM), when only the inner half of the myocardium is affected. (Note: EM is the elastic modulus used for each patient and is given in Table 10.1)

Before any interpretation of the results is made there is an important point which must be clarified. The volume changes are only an indication of how greatly the performance of the model is affected by the non-functioning area. The reason for this is that, in reality, the pressure increase is a response to the volume reduction nd not an independent variable as it is treated in this analysis. Thus this value gives an indication of how greatly the model is affected and not actual volume changes which might be experienced by a real left ventricle.

It is certainly interesting that the effect upon the volume of taking an area with different relative stiffness appears to vary between patients and regions. It is therefore most likely that it is the *shape* of an affected area which is likely to determine how greatly the performance of the left ventricle is impaired by ischaemia or scarring. This is likely to be the reason why the area of the left ventricle which is affected by coronary artery disease will determine the prognosis for that patient, Gibson [83].

Some of the results in the above table show that performance is less impaired by a less stiff rather than a stiffer region. This is because for those patients the volume change is of greater significance than the pressure change. In this case the epicardial layer is still able to overcome the cavity pressure and thus contribute to the volume reduction.

In other cases the volume change is less affected by the affected area being stiffer. In this case the reduced contraction force in that area is unable to contribute to the volume reduction. This area will tend to bulge out and as a result this area's contribution may be a volume increase. In some of the above cases this volume increase is dwarfed by the overall volume reduction. In other cases there is an overall increase in volume.

The simulation of scarring of the anterior free wall of patient "CL" is interesting since this results in a greater reduction in volume than in the normal case. This would suggest that the anterior free wall for this patient is particularly flexible and during the normal simulation tends to bulge out. This may be due to that part of the wall being slightly thinner in the model than in the real left ventricle or it may be due to the elastic modulus, used for this patient, being too low. Whichever of these two reasons is correct is immaterial since it is obviously a spurious result. That is there is no reason why a left ventricle would perform better when part of it was not functioning. It was found that if a higher value of the elastic modulus was taken then this unexpected result was not repeated.

It is difficult to say whether ischaemia and scarring have the most detrimental effect when in the anterior or posterior region. The simulation of scarring however, appears to be slightly more detrimental in the posterior region than in the anterior region. The effect of the simulation of ischaemia appears to be far more evenly distributed between the two regions. As a result it is harder to pinpoint which region is more greatly affected.

	Starting	Normal	Anterior Free Wall			Posterior Free wall		
Patient	Volume	Volume	EM/10	EM	10 EM	EM/10	EM	10 EM
BA	107.19	104.08	112.51	105.52	104.48	112.40	108.84	107.82
CE	155.28	134.92	135.13	133.62	136.48	143.28	149.66	153.33
CL	122.72	114.04	135.27	117.58	110.11	141.94	121.64	119.92
CL(A)	150.03	139.58	161.16	145.57	141.52	157.15	146.87	146.85
MI	56.21	44.62	46.73	47.17	49.05	55.91	54.05	52.64
WE(A)	58.40	54.99	67.60	57.23	54.94	69.53	60.89	57.57

Table 11.4 This table shows the resulting cavity volumes for simulated ischaemia (EM/10 and EM) and scarring (10 EM), when the full depth of the myocardium is affected. (Note: EM is the elastic modulus used for each patient and is given in Table 10.1)

The above table shows the effect of simulating ischaemia and scarring through the full depth of the anterior and posterior wall. Unlike the previous case we do not have the complication of half of the wall depth still being active. This has resulted in far greater congruity between the results obtained for the different patients. All the patients, with the exception of "MI", behave consistently. That is their performance is less impaired by a stiff region than a less stiff region. On the whole it would seem that the posterior free wall is a more sensitive area as regards performance of the left ventricle. Again patient "CL" shows a greater volume reduction with a stiff anterior free wall than a normal contracting one. This adds further weight the assertion made above about the cause of this increased volume reduction.

As with the analysis when only the endocardial half of the model wall is affected, patient "MI" shows a smaller volume reduction with a stiff anterior fr e wall than with a less stiff one. Since in this case the full depth of the affected part of the wall is non-contracting the previously used explanation does not hold. The volume differences are however very small and thus are likely to be more greatly affected by computational approximation. This may be the reason for this unexpected phenomenon. There is however another possible explanation. Since the volume changes are small there may be another factor which could bring about this strange result. The model when systole is simulated experiences some rotation (Chapter 7) and a very stiff region would impair this rotation. It may be that when the rotation is impaired the volume reduction is prevented to some extent. This could give rise to the result of a less stiff area being not as damaging to the volume reduction as a stiff region.

11.5 Summary

There are a number of conclusions which can be made from the results of this section. Firstly, the model as used at present does not appear to be able to accurately model the local strains. This may in part be due to the coarse approximation which has been made for the varying fibre angle through the wall thickness. It may also be in part due to the lack of the ability of the simulated muscle fibres to slide over each other. A third reason for this may even be due to the way these "fibre strains" are calculated. The "fibre strains" in our model are calculated for each corner node of each element. In the real left ventricle the fibres are continuous and not a series of short fibres as they are treated in the strain calculation. It may be necessary to calculate the strain which would result in a fibre which traverses the entire length of the ventricle. It has been shown that the "optimal" coarse for greatest efficiency of fibre contraction is for the fibres to lie along geodesics (Horowitz *et al.* 1993 [85]). That is they follow the shortest course possible on some surface, at the epicardium and endocardium it will be these surfaces. It may be that consideration of this would be necessary for the correct calculation of the fibre strains.

Secondly, there appears, in most cases, to be little effect upon the global deformation of the model in taking a lower contraction coefficient for the endocardial layer of elements. That is not to say that an uneven distribution of contractile force is not important to the real left ventricle. It is just that this model is not very sensitive to these differences in contraction.

Thirdly, geometry of the left ventricle once again shows that it is important to the function of the left ventricle. In this case it has been demonstrated that the shape and position of an area of ischaemic or scarred left ventricle may be fundamental to how impaired the performance of the left vertical could become.

Chapter 12

12.1 Conclusions

There are a number of conclusions that may be drawn from this work.

(1) The first conclusion is that the two programs "XL1" and "HEART" can be combined into a single package which can be run upon desktop computer systems. It is also the case that the calculation of the passive and active material properties can be automated to provide results without constant user input. The result of this is that large numbers of patients can be analysed and will only require user input for interpretation of the results.

(2) The proposed method of volume matching known as "target volume matching" can be used to analyse the diastolic properties of the myocardium. This method does not have a significant effect upon the results obtained when compared with the results obtained from "smooth volume matching". This new method does however seem to make the model less stiff and there appears to be greater variation in the values obtained for the elastic moduli. Another interesting aspect raised by the introduction of this method was that, due to the lack of divergence in the results obtained by the two methods, the method of "smooth volume matching" was reasonably accurate over the diastolic period.

(3) The modulus ratio can have a significant effect upon the global deformation of the left ventricle during diastole. This effect although significant is not great enough to explain the ventricular length changes that are observed from the X-ray frames during diastole. Whether this inability to account for the observed length changes is due to either some other property of the myocardium or some external force is as yet unclear. It is difficult to deduce much concerning the length and length changes of the left ventricle since the accuracy of the length measurements must be somewhat suspect. The reason

for this is that when the original data was prepared it was not envisaged that the length information would be required and thus less precision would have been required during digitisation. It is also the case that length and displacement information is affected more by small errors than is the volume and hence the cavity volume would always provide greater certainty in regards to its accuracy.

(4) From the analysis of diastole it is clear that the shear modulus is likely to have a significant effect upon the way in which the left ventricle deforms in diastole. It also seems likely that in order to match the length of the left ventricle during diastole the assumed relationships between the various material properties must be changed.

(5) The "thermal stressing" analogy as proposed by Rosen [64] can be used to simulate the active contraction of the myocardium. It can also, in some cases, fully explain the global deformation that has been observed from patient data. The relative rotation of the endocardial and epicardial surfaces is also shown to occur in this model.

(6) The passive material properties of the myocardium do not greatly affect the global deformation of the myocardium during systole. The exception to this is when the overall stiffness of the model's wall is effected and when the wall stiffness is low. Increases in the wall stiffness do not have a very significant effect. Reductions from the "average" diastolic values do however have a more significant effect upon global deformation, especially if the wall stiffness is low to begin with. It is an unlikely scenario that the passive properties would be less stiff during systole than they are in diastole. The reasoning behind this is twofold, one, if the myocardium was less stiff then the contracting myocardium would have to work harder to over come the effect of the cavity pressure. Two, all the research done on systolic myocardium suggest that it behaves much stiffer than during diastole. It is thus likely that the myocardium will be relatively stiff during systole and that consequently the effect of small changes in myocardial stiffness are unlikely to greatly affect global deformation.

(7) The use of a high endocardial and epicardial fibre angle in the model reduces to negligible the effect on the global deformation of the model of changing the ratio of shear stiffness to elastic stiffness. It is thus likely that the passive material properties have a much reduced effect if the particular left ventricle can be shown to have a high fibre angle.

(8) If the myocardium can be shown to be stiffer during systole than during diastole then the effect of the cavity pressure will be negligible. Thus systole can be modelled using geometric data alone. This is important as pressure data is difficult to come by as it must be obtained by invasive means. The result of this is that data suitable for the investigation of systolic left ventricular properties could be more easily obtained.

(9) It is clear that shape is probably the single most important factor responsible for the way in which the left ventricle deforms during systole. Other factors can affect this deformation to some extent though the extent of this is also shape dependent.

(10) The boundary conditions for the model are also a major factor which can contribute to the deformation. The significance of this is of a similar order to that of the material properties.

(11) The use of the shape index as suggested by Grewal [2] has been less than successful. An improvement to this shape index might be to consider some form of envelope function rather than the volume and surface area.

(12) The lack of ability of the "muscle fibres" in this model to "slide over each other" is likely to be responsible for the high values of the contraction coefficients required by the model. The other factor to contribute to this very high value is likely to be the coarseness of the approximation of the varying fibre angle through the wall. The first of these problems may be overcome by the inclusion of the ability of the layers which make up the thickness of the wall to slide freely over each other. The second of these problems may be overcome by increasing the number of elements through the depth of the wall.

(13) If the total length change in the model over systole is adequately simulated by this model then it appears that the frame to frame length changes are also well represented. This means that the model can provide a good simulation of what occurs to the shape of the left ventricle during late systole.

(14) The work of Horowitz *et al.* [85] has shown that the course of the muscle fibres of the myocardium lies on geodesics. That is they run along the shortest path from one point of the myocardium to another while remaining on some "surface" defined within the myocardium. For the endocardium and epicardium this will be the actual surface. Thus in order to consider the fibre strain in the myocardium the change in length of an imaginary fibre following such a path must be considered. This is easily accomplished for idealised ventricular shapes but is a much more complicated task when considering a non-idealised finite element model.

(15) The uneven distribution of contraction force within the myocardium of the left ventricle in general would not appear to greatly alter the global deformation of our model. It is however quite likely to affect the local distribution of stress and strain within the myocardium, especially through the depth of the ventricular wall.

(16) It is also the case that this model shows that the magnitude of the effect of ischaemia and scarring upon the function of the myocardium is not restricted merely to the properties of the affected area but is also dependent upon the shape of the ventricle in the affected area. This means that the position of a damaged area of myocardium is likely to be important as well as its size.

(17) Finally, it has been shown that the use of the thermal stressing analogy for myocardial contraction is applicable to the simulation of a systolic left ventricle.

12.2 Further Work

There are several areas which still require investigation and improvements which can be made and these are described below:

(1) It is vitally important that more data becomes available for the simulation of late systole. At present only six patients have been analysed and this is a very small number from which to draw any great conclusions.

(2) The data used in this analysis is quite old and newer imaging methods provide greater clarity and increased accuracy. What is not generally available are the pressure readings from within the left ventricle. If it is the case that the cavity pressure is not of great importance during systole then geometric data alone could be used to analyse late systole. The most suitable data for modelling purposes would be that which incorporates not only long axis information by short axis cross-sections also.

(3) A more sophisticated method of calculating muscle fibre strains must be introduced. This must be calculated considering the fibres to lie on geodesics.

(4) A comparison should be made between the deformation obtained with this model compared to that of other models produced using more sophisticated constitutive equations. This will allow this model to be rated against those with a far higher computational overhead and see whether there is a need for other more complex models.

(5) If use was to be made of Magnetic Resonance Imaging data then a method for limiting the distortion of the finite element mesh would be required. This is because the temporal resolution of that method is far lower than that of X-ray angiography. The result of this is that the simulation would have to be performed over a longer period before a new mesh could be formed from the next set of data. The solution is to use some form of adaptive meshing scheme similar to that in the finite difference scheme, the “Adaptive Grid Method”. This involves generating the “optimum” discretisation of the

solution space as the problem is solved. In our problem a similar thing could be achieved by remeshing the FE mesh at predefined intervals. These intervals could be defined in terms of time, maximum allowed volume reduction, as a maximum allowable contraction coefficient or when the distortion of the finite elements reaches some tolerance This would not only allow for the use of data with poor temporal resolution but would also allow the presently used geometric data to be kept throughout the analysis of systole

References

[1] Analysis of Stress and Deformation in the Human Left Ventricle PhD Thesis By C.A.Vinson. Department of Mechanical Engineering Brunel University June 1977

[2] The Mechanical Behaviour of the Left Ventricle of the Human Heart in Diastole PhD Thesis By B.S.Grewal. Department of Mechanical Engineering Brunel University May 1988.

[3] Analysis of Left Ventricular Behaviour in Diastole By Means of Finite Element Method. By C.A.Vinson, D.G.Gibson and A.L Yettram. British Heart Journal Volume XLI, No. 1, pages 60-67, January 1979.

[4] Geometric Modelling of the Human Left Ventricle. By A.L Yettram and C.A.Vinson. Journal of Biomechanical Engineering August 1979 Vol. 101 pp221-223

[5] Computer Modelling of the Human Left Ventricle. By A.L.Yettram, C A.Vinson and D.G.Gibson. Transactions of the ASME Vol. 104, May 1982 pp 148-152.

[6] 1993 Mortality Statistics Cause England and Wales. OPCS HMSO 1993.

[7] Three-Dimensional Finite Element Analysis of Anterior-Posterior Variations in Local Sarcomere Length and Active Fibre Stress During Left Ventricular Ejection. By J.M.Guccione, A.D.McCulloch, W.C.Hunter. Advances in Bioengineering ASME 1993 BED-vol. 26 pp 571-574.

[8] Left Ventricular Stresses in the Human Heart. By I.Mirsky. Biophysical Journal Vol. 9 pp 189-208 1969.

[9] Fiber Orientation in the Canine Left Ventricle During Diastole and Systole. By D.D.Streeter, Jr., S.M.Henry, M.Spotnitz, D.P.Patel, J.Ross Jr. and E.H.Sonnenblick. Circulation Research Vol. XXIV 1969.

[10] An Account of the Structures concerned in the Production of the Jugular Pulse By A.Keith MD. Journal of Anatomy and Physiology Vol. XLII pp 1-25 1907

[11] Principles of Physiology. Edited by R.M.Berne and M.N.Levy. Wolfe Publishing Limited 1996.

[12] Gray's Anatomy Descriptive and Applied 29th Edition. Edited by T.B.Johnston and J.Whillis. Longmans, Green and Co. 1947.

[13] The collagen matrix of the heart. By T.K.Borg and J.B.Caulfield. Federation Proceedings Vol. 40 pp 2037-2041 1981.

[14] Structural Three-dimensional Constitutive Law for passive Myocardium. By A.Horowitz, Y.Lanir, F.C.P.Yin, M.Perl, I.Sheinman and R.K.Strumpf. Transactions of the ASME Journal of Biomedical Engineering Vol. 110/201 pp 200-207 August 1988.

[15] The Ventricle: Basic Clinical Aspects. Edited by H.J.Levine and W.H.Gaasch. Martinus Nijhoff Publishing 1985.

[16] Certain Mechanical Peculiarities of the Human Cardiac Pump in Normal and Diseased States. By G.E.Burch, C.T.Ray and J.A.Cronvich. American Heart Journal Vol. 54 Part 6 pp 801-810 1957.

[17] Looking at Size and Shape relationship to Ventricular Performance. By A.C.Burton. American Heart Journal Vol. 5 Part 6 pp 801-810 1957.

[18] Left Ventricular Tension and Stress in Man. By H.Sandler and H.T.Dodge. Circulation Research Vol. 13 Part 2 pp 91-104 1963.

[19] Stress Distribution Within the Left Ventricular Wall Approximated as a Thick Ellipsoidal Shell. By A.Y.K.Wong and P.M.Rautaharju. American Heart Journal Vol. 75 par 5 pp 649-662 1968.

[20] An Analytic Elastic-Viscoelastic Model For the Shape and Forces in the Left Ventricle. By D.N.Ghista and H.Sandler. Journal of Biomechanics Vol. 2 pp 34-47 1969.

[21] Stress Distribution in the Canine Left Ventricle During Diastole and Systole By D.D.Streeter, J.R.Ramesh, N.Vaishnav, D.J.Patel, H.M.Spotnitz, J Ross and E.H.Sonnenblick. Biophysical Journal Vol. 10 Part 4pp 345-363 1970.

[22] An Indirect Determination of the Oxygen Utilisation of the Human Left Ventricle. By D.N.Ghista and H.Sandler. Journal of Biomechanics Vol. 3 pp 161-174 1970.

[23] Myocardial Mechanics: Application of Sliding-Filament Theory to Isovolumic Contraction of the Left Ventricle. By A.Y.K.Wong. Journal of Biomechanics Vol 6 Part 5 pp 565-581 1973.

[24] Effect of Chamber Eccentricity on Equatorial Fiber Stress During Systole By R.F.Janz. Federation Proccedings Vol. 39 Part 2 pp 183-187 1980.

[25] Estimation of Local Myocardial stress. By R.F.Janz. American Journal of Physiology Vol. 242 (Heart Circ. Physiol. 11) pp H875-H881 1982.

[26] Static Analysis of the Left Ventricle. By A.Tözeren. Journal of Biomechanical Engineering Vol. 105 pp 39-46 1983.

[27] Myocardial Material Mechanics: Characteristic Variation of the Circumferential and Longitudinal Systolic Moduli in Left Ventricular Dysfunction. By C.A.Phillips and J.S.Petrofsky. Journal of Biomechanics Vol. 17 Part 8 pp 561-568 1984.

[28] Estimation of Local Cardiac Wall Deformation and Regional Wall Stress from Biplane Coronary Cineangiograms. By H.C.Kim, B.G.Min, M.M.Lee, J.D.Seo, Y.W Lee and M.C.Han. IEEE Transactions on Biomedical Engineering Vol. BME-32 Part 7 1985.

[29] Comparison of Models Used to Calculate Left Ventricular Wall Force By R.M.Huisman, P.Sipkema, N.Westerhof and G.Elzinga. Medical and Biological Engineering and Computing Vol. 18 pp 133-144 1980.

[30] Finite Element Model for the Mechanical Behaviour of the Left Ventricle. Prediction of Deformation in the Potassium-Arrested Rat Heart. By R.F.Janz and A.F.Grimm. Circulation Research Vol. 30 1972.

[31] The Finite Element in Engineering Science. By O.C.Zienhiewicz. McGraw-Hill 1971.

[32] Finite Element Analysis of Human Cardiac Structures. M.S.Hamid and D.N.Ghista. from Finite Element Methods in Engineering. Published By The University of New South Wales pp 337-348 1974.

[33] Plane-Strain Finite-Element Analysis of Reconstructed Diastolic Left Ventricular Cross Section. By Y.C.Pao, R.A.Robb and E.L.Ritman. Annals of Biomedical Engineering Vol. 4 pp 232-249 1976.

[34] 3-D Finite Element Reconstruction of Left Ventricular Geometry From Cross-Sectional Echocardiagraphic Recordings. By P.E.Nikravesh, K.B.Chandran, D.J.Skorton, N.Pandian and R.E.Keber. ASME AMD Vol. 3 pp 15-18 1981.

[35] Comprehensive Model for the Simulation of Left Ventricular Mechanics: Part 1 and Part 2. By A.Horowitz, M.Perl, S.Sideman and E.Ritman. Medical and Biological Engineering and Computing Vol. 24 pp 145-156 1986.

[36] Finite Element Analysis of Myocardial Diastolic Function Using Three-Dimensional Echocardiographic Reconstructions: Application of a New Method for Study of Acute Ischemia in Dogs. By D.D.McPherson, D.J.Skorton, S.Kodiyalam, L.Petree, M.P.Noel R.Kieso, R.E.Keber, S.M.Collins, and K B.Chandran. Circulation Research Vol. 60 pp674-682 1987.

[37] A Two-Phase Finite Element Model of the Diastolic Left Ventricle. By J.M.Huyghe, D.H.Campen, T.Arts, R.M.Heethaar. Journal of Biomechanics 1991 vol. 24 no. 7 pp 527-538.

[38] The Mechanics of the Ischemic Left Ventricle During the Cardiac Cycle. By D.H.M.Bovendeerd, T.Arts, J.M.Huyghe, D.H.Campen, R.S.Reneman. Advances in Bioengineering ASME 1991 vol. 26 pp 651-654.

[39] Effect of Pericardial Pressure on the Left Ventricular Expansion in Diastasis: A Simulation Using Finite Element Analysis. By G.Han, K.B.Chandran, M.Vonesh, N.Gotteiner, D.McPherson. American Society of Mechanical Engineers 1991 vol. 20 pp 219-222.

[40] Regional Epicardial and Endocardial Two-Dimensional Finite Deformations in Canine Left Ventricle. By J.I.Fann, G.E.Sarris, N.B.Ingels, J.R.Marek, A.Niczyporuk, K.L.Yun, G.T.Daughters II, G.C.Derby, D.C.Miller. American Physological Society 1991 pp H1402-H1410.

[41] Non-Linear Passive Elastic Properties of the Left Ventricular Myocardium: Application of Finite Element Analysis to 3-D Geometry Obtained from Ultrasound Images. By G.Han, K.B.Chandran, M.Vonesh, A.Joob, R.Greene, G.Lanza, F.Kasho, D.McPherson. Mechanics Computing in 1990's & Beyond ASCE Eng. Spec. Conf. Proc. 20-22 May 1991 pp 519-523.

[42] Inverse Problems in Biomechanics. By U.Roy, G.Ray. Engineering Mechanics. Proc. 9th Conf. Held College Station, Texas, May 24-27th 1992 pp 980-983.

[43] Dependence of Local Left Ventricular Wall Mechanics on Myocardial fiber Orientation: A Model Study. By D.H.M.Bovendeerd, T.Arts, J.M.Huyghe, D.H.Campen, R.S.Reneman. Journal of Biomechanics 1992 vol. 25 no. 10 pp 1129-1140.

[44] Three-Dimensional Finite Element Analysis of Regional Ventricular Function. By A.D.McCulloch J.M.Guccione. Proc. Annual Int. Conf. of the IEEE Enginneering in Medicine and Biology Soc. 15th international Conference 1993 vol. 2 pp 584-585.

[45] Application of Finite-Element Analysis with Optimization to Assess the In-Vivo Nonlinear Myocardial Material Properties using Echocardiographic Imaging By G.J.Han, R.B.Chandran, N.L.Gotteiner, M.J.Vonesh, A.W.Joob, R.Greene, G M Lanza, D.D.McPherson. Medical & Biological Enginieering & Computing 1993 vol. 31 no 5 pp 459-467.

[46] Finite-Element Assessment of Change in Regional Ventricular Wall Muscle Stiffness Due to Coronary Occlusion. By Y.C.Pao, Robert.D.Simari and Erik L.Ritman BED-Vol. 26, 1993 Advances in Bioengineering ASME 1993.

[47] An Experimental and Finite Element Study of Ventricular Wall Stress Models By A.W.Eberhardt and J.A.Sanford. BED-Vol, 1993 Advances in Bioengineering ASME 1993.

[48] Influence of Endocardial-Epicardial Crossover of Muscle Fibers on Left Ventricular Wall Mechanics. By D.H.M.Bovendeerd, T.Arts, J.M.Huyghe, D.H.Campen, R.S.Reneman. Journal of Biomechanics 1994 vol. 27 no. 7 pp 941-951.

[49] Biomechanics of the Heart Muscle. By D.H.Campen, J.M.Huyghe, D.H.M.Bovendeerd, T.Arts. European Journal of Mechanics, A/Solids, Vol. 13 pt 4 1994 pp 19-41.

[50] Nonhomogeneous Analysis of Epicardial Strain Distributions During Acute Myocardial Ischemia in the Dog. By A.R.Hashima, A.A.Joung, A.D.McCulloch, L.K.Waldman. Journal of Biomechanics 1994 vol. 27 no. 7 pp 941-951.

[51] Mathematical Three-Dimensional Solid Modelling of Biventricular Geometry. By J.S.Pirolo, S.J.Bresina, L.L.Creswell, R.W.Myers, B.A.Szabo, M.W.Vannier, M.K.Pasque. Annals of Biomedical Engineering 1993 vol. 21 no. 3 pp 199-219.

[52] Human Heart: Tagging with MR Imaging - A Method for Noninvasive Assessment of Myocardial Motion. E.A.Zerhouni, D.M.Parish, W.J.Rogers, A.Yang and E.P.Shapiro. Cardiac Radiology Vol.172 pp 349-350 1988.

[53] Heart Wall Motion: Improved Method of Spatial Modulation of Magnetization for MR Imaging. By L.Axel and L.Dougherty. Cardiac Radiology Vol.169 pp 59-63 1989.

[54] Three Dimensional Motion and Deformation of the Heart Wall: Estimation with Spatial Modulation of Magnetisation - A Model Based Approach. By A.A.Young and L.Axel. Radiology Vol. 185 pp 241-247 1992.

[55] Implementation of Displacement Field Fitting for Calculating 3D Myocardial Deformations from Parallel-Tagged MR Images. By W.G.O'Dell, E.R.McVeigh, C.C.Moore and E.A.Zerhouni. Proceedings of the 16th Annual International Conference of the IEEE Engineering in Medicine and Biological Society pp 551-552 1994

[56] A Recursive Filter for Phase Velocity Assisted Shape-Based Tracking of Cardiac Non-ridged Motion. By J.C.McEachen II, F.G.Meyer, R.T.Constable, A.Nehorai and J.S.Duncan. Proccedings Fifth International Conference on Computer Vision Vol. 7 pp 653-658 1995.

[57] Automatic Detection of Left Ventricular Contour Using Hough Transform with Weighted Model From 2D Echocardiograms. By J.H.Cho and M.N.Kim. Proccedings of the SICE Annual Conference Society of Instrumentation and Control Engineers pp 997-1002 1994.

[58] A Knowledge-Based approach to Volumetric Medical Image Segmentation. By C.W.Chen, J.Luo. K.J.Parker and T.S.Huang. Proccedings of IEEE International Conference on Image Proccessing Vol.3 pp 493-497 1994.

[59] CT Volumetric Data-Based Left Ventricle Motion Estimation: An Integrated Approach. By C.W.Chen, J.Luo. K.J.Parker and T.S.Huang. Computerised Medical Imaging and Graphics Vol. 19 Part 1 pp 85-100 1995.

[60] Left Ventricular Boundary Detection From Spatio-Temporal Volumetric Computed Tomography Images. H.K.Tu, A.Matheny, D.B.Goldgof and H.Bunke. Computerised Medical Imaging and Graphics Vol. 19 Part 1 pp 27-46 1995.

[61] Mechanical Properties of Cardiac Muscle. By G.Lundin. ACTA Physiologica Scandinavica Vol.7 Supplementum XX 1944.

[62] Passive Biaxial Mechanical Properties of Isolated Canine Myocardium. By L.L.Demer and F.C.P.Yin. Journal of Physiology Vol. 339 pp 615-630 1983.

[63] Quantification of the Mechanical Properties of Noncontracting Canine Myocardium Under Simultaneous Biaxial Loading. By F.C.P.Yin, R.K.Strumpf, P.H.Chew and S.L.Zeger. Journal of Biomechanics Vol. 20 Part 6 pp 577-589 1987.

[64] A Constituative Description of Contracting Papillary Muscle and its Implications to the Dynamics of the Intact Heart. By J.G.Pinto. Transactions of ASME Journal of Biomechanical Engineering Vol. 109 pp 181-191 1987.

[65] Determination of a Constitutive Relation for Passive Myocardium: 1. A New Functional Form. By J.D.Humphrey, R.K.Strumpf, F.C.P.Yin. Transactions of ASME Journal of Biomechanical Engineering Vol. 112 pp 333-346 1990.

[66] Regional Mechaical-Properties of Passive Myocardium. By V.P.Novak, F.C.P.Yin and J.D.Humphrey. Journal of Biomechanics No. 4 pp 403-412 1994.

[67] Finite Element in Cardiomechanics - A Selective Review. By Y.C.Pao. Conference on Finite Elements in Biomechanics, Tucson, Arizona pp 43-68 1980.

[68] SimplePlot: A Two-Dimensional Drawing Package Mark 2, Primer. Bradford University software services Ltd, 1991.

[69] SimplePlot: A Two-Dimensional Drawing Package Mark 2, Reference Manual. Bradford University software services Ltd, 1991.

[70] The NAG Fortran Library Manual, Mark 16. National Algorithms Group. 13 Banbury Road, Oxford, 1993.

[71] Strength of Biological Materials. By H.Yamada. Williams and Wilkins 1970.

[72] Deformation of the human left ventricle : Material and geometric effects. By C.W.Neckyfarow and A.B.Perlman. Proceedings of the Fourth New England Bioengineering Conference May 7-8, 1976.

[73] GRP in Structural Engineering. By M.Holmes and D.J.Just. Applied Science 1982.

[74] A Theoretical Model for the Analysis of Stresses and Strains in Muscle, Including Active Contraction. By B.W.Rosen. Twenty first ACEMB, Houston Texas pp 41.3 1968.

[75] A General Three-Dimensional Quasi-Static Solid Analysis Computer Program. By B.M.Kwak, C.J.Chen, K.Rim and H.L.Falsetti. University of Iowa, Iowa City 1977.

[76] Formulas of Stress and Strain 5rd Edition. By R.J.Roark and W.C.Young. McGraw-Hill Kogakusha, Ltd 1975 pp 504.

[77] Non-Invasive Quantification of Left Ventricular Rotational deformation in Normal Humans Using Magnetic Resonance Imaging Myocardial Tagging. By M.B.Buchalter, J.L.Weiss, W.J.Rogers, E.A.Zerhouni, M.L.Weisfeldt, R.Beyar, E.P.Shapiro. Circulation Vol. 81 pp 1236-1244 1990.

[78] The Measurement of Left Ventricular Systolic Deformations in Humans and in the Canine Model Using Magnetic Resonance Imaging. MD Thesis By M.B.Buchalter. The Johns Hoppkins University, Department of Cardiology 1992.

[79] A Model for an Active Left Ventricle Deformation - Formulation of a Nonlinear, Quasi-Steady Finite Element Analysis for Orthotropic, Three Dimensional Myocardium. By C.J.Chen, B.M.Kwak, K.Rim and H.L.Falsetti. International Conference Proceedings: Finite Elements in Biomechanics pp 639-655 1980.

[80] Continuous assessment of left ventricular shape in man. By D.G.Gibson and D.J.Brown. British Heart Journal Vol 37 pp 904-910 1975.

[81] Simulation of an Active LV by a Two Layer Orthotropic Model. By C.J.Chen, H.L.Falsetti, C.H.Chen and K.Rim. Thirty First ACEMB pp 315 1978.

[82] Compressibility of Perfused Passive Myocardium. By F.C.P.Yin, C.C.H.Chen and R.M.Judd. American Journal of Physiology-Heart and Circulatory Physiology Vol. 40 Part 5 pp H1864-H1870 1996.

[83] Disscussions with Dr. D.Gibson MD. Department of Cardiology, Royal Brompton Hospital, London, England 1996-1997.

[84] Regional Mechanical Properties of Passive Myocardium. By V.P.Novak, F.C.P.Yin and J.D.Humphrey. Journal of Biomechanics Vol. 27 No. 4 pp 403-412 1994.

[85] Geodesics as a Mechanically Optimal Fiber Geometry for the Left-Ventricle. By A.Horowitz, M.Perl and S.Sideman. Basic Research in Cardiology Vol. 88 No S2 pp 67-74 1993.

Appendix I

AI. Testing and Validation

AI.1 Some Test Data Files For the Inclusion of Muscle Fibre Force

Included here are the data files for some of the test models used to check that the inclusion of the thermal stressing was performing correctly.

AI.1.1 Fibres Running at a Constant Angle of 30°

The following two data files are for a cube of side length 2 and the fibres running parallel to one side and offset at an angle of -30° to the adjacent side. Due to the difference in the way that PAFEC^{TM} and "XL1" define the angular offset the sign of the offset is reversed from one package to the other. The elements are subjected to a 0.1 fibre strain and a -0.05 cross-fibre strain. The elements are not pressure loaded. The bottom face is restrained such that it is free to expand but the centre of that face remains fixed.

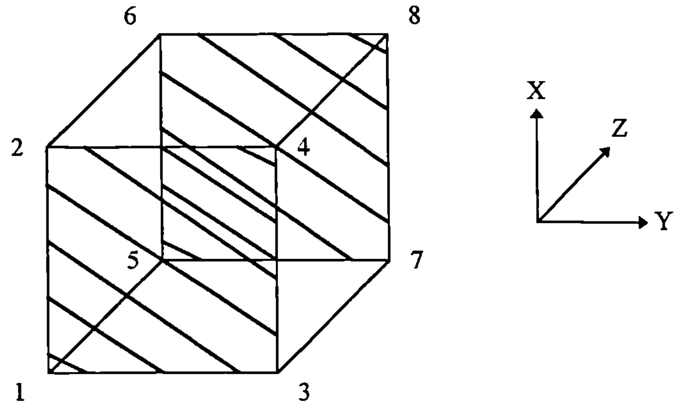


Fig AI.1 This diagram shows the test element used. The corner nodes are numbered and the course of the fibres is represented (-30° to the horizontal). The element is a cube of side length 2 and is centred about the origin.

The PAFEC™ Data File:

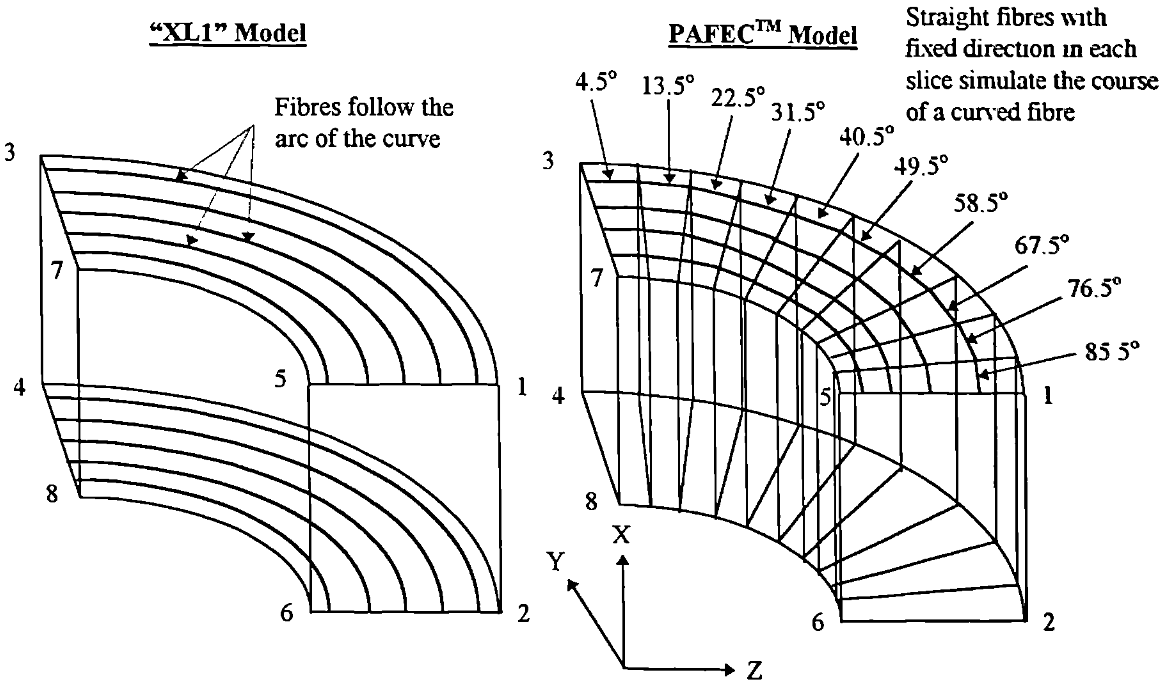
```
CONTROL
PHASE=1
PHASE=2
PHASE=4
PHASE=6
PHASE=7
SKIP.CHECK
STRESS
REACTIONS
FULL.CONTROL
CONTROL.END
TITLE cube with 30 degree angle
NODES
X Y Z
-1 -1 -1
1 -1 -1
-1 1 -1
1 1 -1
-1 -1 1
1 -1 1
-1 1 1
1 1 1
C
0 -1 -1
-1 0 -1
1 0 -1
0 1 -1
-1 -1 0
1 -1 0
-1 1 0
1 1 0
0 -1 1
-1 0 1
1 0 1
0 1 1
ELEM
NUMBER ELEM PROP TOPO
1 37115 11 1 2 3 4 5 6 7 8 9 10 11 12 13
14 15 16 17 18 19 20
LAMINATES
STORE-1
NUMBER ORTHOTROPIC.MATERIAL.NUMBER ANG1
ANG2 ANG3 STORE
11 12 30 0 0 1
ORTHOTROPIC.MATERIAL
NUMBER SXX SYY SZZ SXY SYZ SZX SHXY SHYZ
SHZX ALX ALY ALZ
12 5E-03 1E-03 5E-03 -4.5E-04 -4.5E-04 -
22.5E-04 10E-03 10E-03
*2.90E-03 -0.05 0.1 -0.05
TEMPERATURE
LOAD.CASE TEMP START FINISH STEP
1 -1 1 20 1
RESTRAINTS
NODE DIRE
1 1
3 1
5 1
7 1
10 12
13 13
15 13
18 12
STRAIN
LOAD START FINISH STEP
1 1 1 1
EXTERNAL
LOAD START FINISH STEP
1 1 1 1
END.OF.DATA
```

The "XL1" Data File:

```
Constant 30 degree angle through a cube of side length 2
1,20,0,1,8,19,1
1,1,0,1
     200.0, 1000.0, 200.0
 0.4500, 0.4500, 0.0900
     100.0, 100.0, 344.8
    -0.05000     0.10000    -0.05000
   -1.000000000000,  -1.000000000000,  -1.000000000000,  -1.000000000000
    1.000000000000,  -1.000000000000,  -1.000000000000,  -1.000000000000
   -1.000000000000,   1.000000000000,  -1.000000000000,  -1.000000000000
    1.000000000000,   1.000000000000,  -1.000000000000,  -1.000000000000
   -1.000000000000,  -1.000000000000,   1.000000000000,  -1.000000000000
    1.000000000000,  -1.000000000000,   1.000000000000,  -1.000000000000
   -1.000000000000,   1.000000000000,   1.000000000000,  -1.000000000000
    1.000000000000,   1.000000000000,   1.000000000000,  -1.000000000000
    0.000000000000,  -1.000000000000,  -1.000000000000,  -1.000000000000
   -1.000000000000,   0.000000000000,  -1.000000000000,  -1.000000000000
    1.000000000000,   0.000000000000,  -1.000000000000,  -1.000000000000
    0.000000000000,   1.000000000000,  -1.000000000000,  -1.000000000000
   -1.000000000000,  -1.000000000000,   0.000000000000,  -1.000000000000
    1.000000000000,  -1.000000000000,   0.000000000000,  -1.0000000000 0
   -1.000000000000,   1.000000000000,   0.000000000000,  -1.000000000000
    1.000000000000,   1.000000000000,   0.000000000000,  -1.000000000000
    0.000000000000,  -1.000000000000,   1.000000000000,  -1.0000 0    000
   -1.000000000000,   0.000000000000,   1.000000000000,  -1.00000     00
    1.000000000000,   0.000000000000,   1.000000000000,  -1.000000000000
    0.000000000000,   1.000000000000,   1.000000000000,  -1.000000000000
   1    1    1    1  -30  -30    0    0    0    0    0    0    0   20    0
   1  13    5  10  18    3   15    7    9   17   12   20    2   14    6   11   19    4   16    8
  1,1,0,0
  3,1,0,0
  5,1,0,0
  7,1,0,0
 10,1,1,0
 13,1,0,1
 15,1,0,1
 18,1,1,0
 1.,1.,0.,0.,0.00000
```

AI.1.2 A Quarter of an Annulus

This test is for an annulus of height 1 and outer radius 2 and inner radius 1 5 The fibre strain is 0.1 and the cross-fibre strain is -0.05. The "XL1" model is formed from a single element with the fibre direction along the arc of the annulus. The PAFEC™ model is defined in terms of ten elements which form the arc each having its own angular offset to simulate the fibres following the arc. Both models are restrained so that the top face cannot move in the X-direction and the model can contract freely in the radial direction.



"XL1" Model

PAFEC™ Model

Fibres follow the arc of the curve
Straight fibres with fixed direction in each slice simulate the course of a curved fibre
4.5°
13.5°
22.5°
31.5°
40.5°
49.5°
58.5°
67.5°
76.5°
85 5°
3
7
4
8
5
1
6
2
3
7
4
8
5
1
6
2
X
Y
Z

Fig AI.2 The first of the two diagrams (left) shows the single element for the "XL1" model with the fibres in the local coordinates following the arc of the curve. The second diagram (right) shows the PAFEC™ approximation made from ten elements, along the arc, to simulate the curved course of the fibre. Both models were subjected to a 0.1 fibre strain and a -0.05 cross fibre strain.

The PAFEC^{TM} Data File:

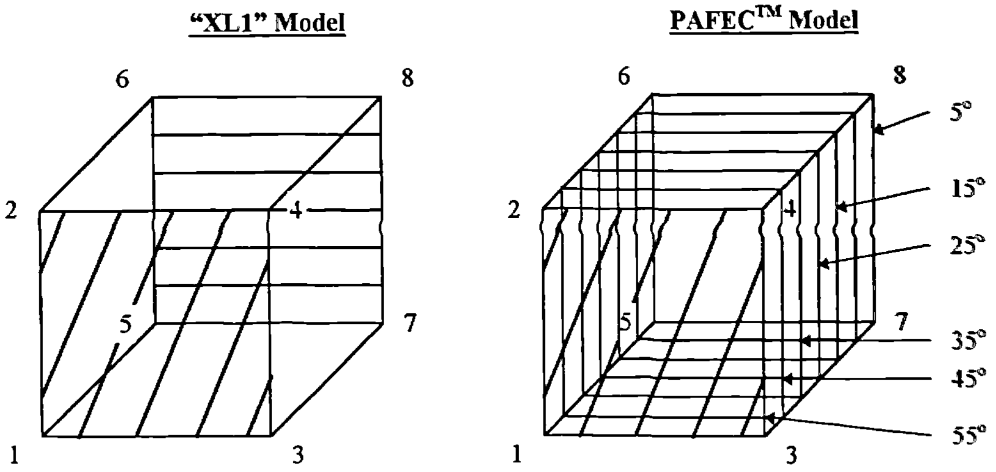
```
CONTROL
PHASE=1
PHASE=4
PHASE=6
PHASE=7
PHASE=9
SKIP.CHECK
STRESS
REACTIONS
FULL.CONTROL
CONTROL.END
TITLE MIKES TEST MIKE5 CURVED BLOCK 5 JAN 1996
NODES
NODE X Y Z
 1 1.0000 0.0000 2.0000 // 2 0.0000 0.0000 2.0000
 3 1.0000 2.0000 0.0000 // 4 0.0000 2.0000 0.0000
 5 1.0000 0.0000 1.5000 // 6 0.0000 0.0000 1.5000
 7 1.0000 1.5000 0.0000 // 8 0.0000 1.5000 0.0000
 9 0.5000 0.0000 2.0000 // 10 1.0000 1.4142 1.4142
 11 0.0000 1.4142 1.4142 // 12 0.5000 2.0000 0.0000
 13 1.0000 0.0000 1.7500 // 14 0.0000 0.0000 1.7500
 15 1.0000 1.7500 0.0000 // 16 0.0000 1.7500 0.0000
 17 0.5000 0.0000 1.5000 // 18 1.0000 1.0607 1.0607
 19 0.0000 1.0607 1.0607 // 20 0.5000 1.5000 0.0000
 21 0.0000 0.1177 1.4954 // 22 0.0000 0.1569 1.9938
 23 1.0000 0.1177 1.4954 // 24 1.0000 0.1569 1.9938
 25 0.0000 0.2347 1.4815 // 26 0.0000 0.2738 1.7285
 27 0.0000 0.3129 1.9754 // 28 0.5000 0.2347 1.4815
 29 0.5000 0.3129 1.9754 // 30 1.0000 0.2347 1.4815
 31 1.0000 0.2738 1.7285 // 32 1.0000 0.3129 1.9754
 33 0.0000 0.3502 1.4586 // 34 0.0000 0.4669 1.9447
 35 1.0000 0.3502 1.4586 // 36 1.0000 0.4669 1.9447
 37 0.0000 0.4635 1.4266 // 38 0.0000 0.5408 1.6643
 39 0.0000 0.6180 1.9021 // 40 0.5000 0.4635 1.4266
 41 0.5000 0.6180 1.9021 // 42 1.0000 0.4635 1.4266
 43 1.0000 0.5408 1.6643 // 44 1.0000 0.6180 1.9021
 45 0.0000 0.5740 1.3858 // 46 0.0000 0.7654 1.8478
 47 1.0000 0.5740 1.3858 // 48 1.0000 0.7654 1.8478
 49 0.0000 0.6810 1.3365 // 50 0.0000 0.7945 1.5593
 51 0.0000 0.9080 1.7820 // 52 0.5000 0.6810 1.3365
 53 0.5000 0.9080 1.7820 // 54 1.0000 0.6810 1.3365
 55 1.0000 0.7945 1.5593 // 56 1.0000 0.9080 1.7820
 57 0.0000 0.7837 1.2790 // 58 0.0000 1.0450 1.7053
 59 1.0000 0.7837 1.2790 // 60 1.0000 1.0450 1.7053
 61 0.0000 0.8817 1.2135 // 62 0.0000 1.0286 1.4158
 63 0.0000 1.1756 1.6180 // 64 0.5000 0.8817 1.2135
 65 0.5000 1.1756 1.6180 // 66 1.0000 0.8817 1.2135
 67 1.0000 1.0286 1.4158 // 68 1.0000 1.1756 1.6180
 69 0.0000 0.9742 1.1406 // 70 0.0000 1.2989 1.5208
 71 1.0000 0.9742 1.1406 // 72 1.0000 1.2989 1.5208
 73 0.0000 1.2374 1.2374 // 74 0.5000 1.0607 1.0607
 75 0.5000 1.4142 1.4142 // 76 1.0000 1.2374 1.2374
 77 0.0000 1.1406 0.9742 // 78 0.0000 1.5208 1.2989
 79 1.0000 1.1406 0.9742 // 80 1.0000 1.5208 1.2989
 81 0.0000 1.2135 0.8817 // 82 0.0000 1.4158 1.0286
 83 0.0000 1.6180 1.1756 // 84 0.5000 1.2135 0.8817
 85 0.5000 1.6180 1.1756 // 86 1.0000 1.2135 0.8817
 87 1.0000 1.4158 1.0286 // 88 1.0000 1.6180 1.1756
 89 0.0000 1.2790 0.7837 // 90 0.0000 1.7053 1.0450
 91 1.0000 1.2790 0.7837 // 92 1.0000 1.7053 1.0450
 93 0.0000 1.3365 0.6810 // 94 0.0000 1.5593 0.7945
 95 0.0000 1.7820 0.9080 // 96 0.5000 1.3365 0.6810
 97 0.5000 1.7820 0.9080 // 98 1.0000 1.3365 0.6810
 99 1.0000 1.5593 0.7945 // 100 1.0000 1.7820 0.9080
 101 0.0000 1.3858 0.5740 // 102 0.0000 1.8478 0.7654
 103 1.0000 1.3858 0.5740 // 104 1.0000 1.8478 0.7654
 105 0.0000 1.4266 0.4635 // 106 0.0000 1.6643 0.5408
 107 0.0000 1.9021 0.6180 // 108 0.5000 1.4266 0.4635
 109 0.5000 1.9021 0.6180 // 110 1.0000 1.4266 0.4635
 111 1.0000 1.6643 0.5408 // 112 1.0000 1.9021 0.6180
 113 0.0000 1.4586 0.3502 // 114 0.0000 1.9447 0.4669
 115 1.0000 1.4586 0.3502 // 116 1.0000 1.9447 0.4669
 117 0.0000 1.4815 0.2347 // 118 0.0000 1.7285 0.2738
 119 0.0000 1.9754 0.3129 // 120 0.5000 1.4815 0.2347
 121 0.5000 1.9754 0.3129 // 122 1.0000 1.4815 0.2347
 123 1.0000 1.7285 0.2738 // 124 1.0000 1.9754 0.3129
 125 0.0000 1.4954 0.1177 // 126 0.0000 1.9938 0.1569
 127 1.0000 1.4954 0.1177 // 128 1.0000 1.9938 0.1569
 C
 C
 C
 ELEM
 NUMB GROUP ELEM PROP TOPO
 1 1 37115 11 6 2 5 1 25 27 30 32 14
+ 17 9 13 21 22 23 24 26 28
+ 29 31
 2 1 37115 12 25 27 30 32 37 39 42 44 26
+ 28 29 31 33 34 35 36 38 40
+ 41 43
 3 1 37115 13 37 39 42 44 49 51 54 56 38
+ 40 41 43 45 46 47 48 50 52
+ 53 55
 4 1 37115 14 49 51 54 56 61 63 66 68 50
+ 52 53 55 57 58 59 60 62 64
+ 65 67
 5 1 37115 15 61 63 66 68 19 11 18 10 62
+ 64 65 67 69 70 71 72 73 74
+ 75 76
 6 1 37115 16 19 11 18 10 81 83 86 88 73
+ 74 75 76 77 78 79 80 82 84
+ 85 87
 7 1 37115 17 81 83 86 88 93 95 98 100 82
+ 84 85 87 89 90 91 92 94 96
+ 97 99
 8 1 37115 18 93 95 98 100 105 107 110 112 94
+ 96 97 99 101 102 103 104 106 108
+ 109 111
 9 1 37115 19 105 107 110 112 117 119 122 124 106
+ 108 109 111 113 114 115 116 118 120
+ 121 123
 10 1 37115 20 117 119 122 124 8 4 7 3 118
+ 120 121 123 125 126 127 128 16 20
+ 12 15
LAMINATES
ORTHO=12
AXIS=1
ANG2=0
ANG1=0
NUMBER ANG3
11 -4.5
12 -13.5
13 -22.5
14 -31.5
15 -40.5
16 -49.5
17 -58.5
18 -67.5
19 -76.5
20 -85.5
ORTHOTROPIC.MATERIAL
NUMBER SXX SYY SZZ SXY SYZ SZX SHXY SHYZ SHZX ALX ALY ALZ
12 1E-03 1E-03 1E-03 -4.5E-04 -4.5E-04 -4.5E-04 2.90E-03
2.90E-03
+2.90E-03 -0.05 0.1 -0.05
TEMPERATURE
LOAD.CASE TEMP START FINISH STEP
1 -1 1 127 1
RESTRAINTS
NODE PLANE DIRE
1 1 1
1 0 2
13 0 2
5 0 2
3 0 3
15 0 3
7 0 3
 EXTERNAL
 LOAD START FINISH STEP
 1 1 10 1
 END.OF.DATA
```

The “XL1” Data File:

```
curved element
1,20,0,1,8,19,1
1,1,0,1
     1000.0, 1000.0, 1000.0
 0.4500, 0.4500, 0.4500
     344.8, 344.8, 344.8
    -0.05000      0.10000     -0.05000
    1.000000000000,   0.000000000000,   2.000000000000,  -1.000000000000
    0.000000000000,   0.000000000000,   2.000000000000,  -1.000000000000
    1.000000000000,   2.000000000000,   0.000000000000,  -1.000000000000
    0.000000000000,   2.000000000000,   0.000000000000,  -1.000000000000
    1.000000000000,   0.000000000000,   1.500000000000,  -1.000000000000
    0.000000000000,   0.000000000000,   1.500000000000,  -1.000000000000
    1.000000000000,   1.500000000000,   0.000000000000,  -1.000000000000
    0.000000000000,   1.500000000000,   0.000000000000,  -1.000000000000
    0.500000000000,   0.000000000000,   2.000000000000,  -1.000000000000
    1.000000000000,   1.414213600000,   1.414213600000,  -1.000000000000
    0.000000000000,   1.414213600000,   1.414213600000,  -1.000000000000
    0.500000000000,   2.000000000000,   0.000000000000,  -1.000000000000
    1.000000000000,   0.000000000000,   1.750000000000,  -1.000000000000
    0.000000000000,   0.000000000000,   1.750000000000,  -1.000000000000
    1.000000000000,   1.750000000000,   0.000000000000,  -1.000000000000
    0.000000000000,   1.750000000000,   0.000000000000,  -1.000000000000
    0.500000000000,   0.000000000000,   1.500000000000,  -1.000000000000
    1.000000000000,   1.060660200000,   1.060660200000,  -1.000000000000
    0.000000000000,   1.060660200000,   1.060660200000,  -1.000000000000
    0.500000000000,   1.500000000000,   0.000000000000,  -1.000000000000
   1    1    1    1    0    0    0    0    0    0    0    0    0   20    0
   1   13    5   10   18    3   15    7    9   17   12   20    2   14    6   11   19    4   16    8
  1,1,1,0
 13,1,1,0
  5,1,1,0
 10,1,0,0
 18,1,0,0
  3,1,0,1
 15,1,0,1
  7,1,0,1
 1.,1.,0.,0.,0.00000
```

AI.1.3 For the Fibre Angle Varying Through the Depth of a Cube

The following two data files are for a cube of side length 2 and the fibre direction varying linearly from 60° on one side to 0° on the opposing side. The PAFEC™ model is defined in terms of six slices through the thickness of the element. The orientation of the fibres within each slice are such as to simulate the varying fibre angle. The cubes are subjected to a 0.1 fibre strain and a -0.05 cross-fibre strain. The elements are not pressure loaded.



"XL1" Model

Fibre direction varying
linearly through the block

PAFEC™ Model

The fixed fibre directions
for each PAFEC™ slice

Fig AI.3 The first of these two diagrams (left) shows the "XL1" model consisting of one element with the fibre direction varying linearly from 60° on one face to 0° on the opposing face. The second model (right) is for PAFEC™ and to simulate the varying fibre angle six slices (elements) are used. Each slice has a slightly different fibre angle to the previous one.

The PAFEC™ Data File:

```
CONTROL
PHASE=1
SKIP.CHECK
REACTIONS
CONTROL.END
TITLE  CUBE  0-60 21 DEC 95
NODES
NODE X Y Z
1 2.0000 2.0000 2.0000 // 2 2.0000 2.0000
1.0000
3 2.0000 2.0000 0.0000 // 4 2.0000 1.0000
2.0000
5 2.0000 1.0000 0.0000 // 6 2.0000 0.0000
2.0000
7 2.0000 0.0000 1.0000 // 8 2.0000 0.0000
0.0000
9 1.0000 2.0000 2.0000 // 10 1.0000 2.0000
0.0000
11 1.0000 0.0000 2.0000 // 12 1.0000
0.0000 0.0000
13 0.0000 2.0000 2.0000 // 14 0.0000
2.0000 1.0000
15 0.0000 2.0000 0.0000 // 16 0.0000
1.0000 2.0000
17 0.0000 1.0000 0.0000 // 18 0.0000
0.0000 2.0000
19 0.0000 0.0000 1.0000 // 20 0.0000
0.0000 0.0000
21 0.0000 0.0000 0.1667 // 22 0.0000
2.0000 0.1667
23 0.0000 0.0000 0.3333 // 24 0.0000
1.0000 0.3333
25 0.0000 2.0000 0.3333 // 26 0.0000
0.0000 0.5000
27 0.0000 2.0000 0.5000 // 28 0.0000
0.0000 0.6667
29 0.0000 1.0000 0.6667 // 30 0.0000
2.0000 0.6667
31 0.0000 0.0000 0.8333 // 32 0.0000
2.0000 0.8333
33 0.0000 1.0000 1.0000 // 34 0.0000
0.0000 1.1667
35 0.0000 2.0000 1.1667 // 36 0.0000
0.0000 1.3333
37 0.0000 1.0000 1.3333 // 38 0.0000
2.0000 1.3333
39 0.0000 0.0000 1.5000 // 40 0.0000
2.0000 1.5000
41 0.0000 0.0000 1.6667 // 42 0.0000
1.0000 1.6667
43 0.0000 2.0000 1.6667 // 44 0.0000
0.0000 1.8333
45 0.0000 2.0000 1.8333 // 46 1.0000
0.0000 0.3333
47 1.0000 2.0000 0.3333 // 48 1.0000
0.0000 0.6667
49 1.0000 2.0000 0.6667 // 50 1.0000
0.0000 1.0000
51 1.0000 2.0000 1.0000 // 52 1.0000
0.0000 1.3333
53 1.0000 2.0000 1.3333 // 54 1.0000
0.0000 1.6667
55 1.0000 2.0000 1.6667 // 56 2.0000
0.0000 0.1667
57 2.0000 2.0000 0.1667 // 58 2.0000
0.0000 0.3333
59 2.0000 1.0000 0.3333 // 60 2.0000
2.0000 0.3333
61 2.0000  .0000 0.5000 // 62 2.0000
2.0000 0.5000
63 2.0000 0.0000 0.6667 // 64 2.0000
1.0000 0.6667
65 2.0000 2.0000 0.6667 // 66 2.0000
0.0000 0.8333
67 2.0000 2.0000 0.8333 // 68 2.0000
1.0000 1.0000
69 2.0000 0.0000 1.1667 // 70 2.0000
2.0000 1.1667
71 2.0000 0.0000 1.3333 // 72 2.0000
1.0000 1.3333
73 2.0000 2.0000 1.3333 // 74 2.0000
0.0000 1.5000
```

```
75 2.0000 2.0000 1.5000 // 76 2.0000
0.0000 1.6667
77 2.0000 1.0000 1.6667 // 78 2.0000
2.0000 1.6667
79 2.0000 0.0000 1.8333 // 80 2.0000
2.0000 1.8333
ELEMENTS
ELEM=37115
NUMBER PROP TOPOLOGY
1 11 20 15 23 25 8 3 58 60 17
*21 22 24 12 10 46 47 5 56
*57 59
2 12 23 25 28 30 58 60 63 65 24
*26 27 29 46 47 48 49 59 61
*62 64
3 13 28 30 19 14 63 65 7 2 29
*31 32 33 48 49 50 51 64 66
*67 68
4 14 19 14 36 38 7 2 71 73 33
*34 35 37 50 51 52 53 68 69
*70 72
5 15 36 38 41 43 71 73 76 78 37
*39 40 42 52 53 54 55 72 74
*75 77
6 16 41 43 18 13 76 78 6 1 42
*44 45 16 54 55 11 9 77 79
*80 4
LAMIN
NUMBER ORTHO AXIS.SET ANG1
11 11 1 -5
12 11 1 -15
13 11 1 -25
14 11 1 -35
15 11 1 -45
16 11 1 -55
ORTHO
NUMBER SXX SYY SZZ SXY SYZ SZX SHXY SHYZ
SHZX ALX ALY ALZ
11 .005 .001 .005 -.00045 -.00045 -.00225
.01 .01 .0029 -0.05 0.1 -0.05
REST
NODE PLANE DIRE
3 2 2
13 0 1
14 0 1
15 0 1
22 0 1
25 0 1
27 0 1
30 0 1
32 0 1
35 0 1
38 0 1
40 0 1
43 0 1
45 0 1
2 0 3
14 0 3
SURFACE
PRESS=0
LOAD NODE PLANE
1 20 2
TEMP
LOAD TEMP START FINISH STEP
2 -1 1 79 1
EXTERNAL
LOAD START FINISH STEP
1 1 6 1
2 1 6 1
STRAIN
LOAD START FINISH STEP
1 1 6 1
2 1 6 1
END.OF.DATA
```

The “XL1” Data File:

```
Fibre angle varying linearly through a cube from 60 drgrees to 0 degrees
1,20,0,1,6,19,1
1,1,0,1
     200.0, 1000.0, 200.0
 0.4500, 0.4500, 0.0900
     100.0, 100.0, 344.8
    -0.05000     0.10000    -0.05000
    2.000000000000,   2.000000000000,   2.000000000000,  -1.000000000000
    2.000000000000,   2.000000000000,   1.000000000000,  -1.000000000000
    2.000000000000,   2.000000000000,   0.000000000000,  -1.000000000000
    2.000000000000,   1.000000000000,   2.000000000000,  -1.000000000000
    2.000000000000,   1.000000000000,   0.000000000000,  -1.000000000000
    2.000000000000,   0.000000000000,   2.000000000000,  -1.000000000000
    2.000000000000,   0.000000000000,   1.000000000000,  -1.000000000000
    2.000000000000,   0.000000000000,   0.000000000000,  -1.000000000000
    1.000000000000,   2.000000000000,   2.000000000000,  -1.000000000000
    1.000000000000,   2.000000000000,   0.000000000000,  -1.000000000000
    1.000000000000,   0.000000000000,   2.000000000000,  -1.000000000000
    1.000000000000,   0.000000000000,   0.000000000000,  -1.000000000000
    0.000000000000,   2.000000000000,   2.000000000000,  -1.000000000000
    0.000000000000,   2.000000000000,   1.000000000000,  -1.000000000000
    0.000000000000,   2.000000000000,   0.000000000000,  -1.000000000000
    0.000000000000,   1.000000000000,   2.000000000000,  -1.000000000000
    0.000000000000,   1.000000000000,   0.000000000000,  -1.000000000000
    0.000000000000,   0.000000000000,   2.000000000000,  -1.000000000000
    0.000000000000,   0.000000000000,   1.000000000000,  -1.000000000000
    0.000000000000,   0.000000000000,   0.000000000000,  -1.000000000000
   1   1   1   1  60   0   0   0   0   0   0   0   0  20   0
   1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17  18  19  20
  1,0,1,0
  2,0,1,0
  5,0,1,1
  6,0,1,1
  9,1,1,0
 17,1,1,1
1.,1.,0.,0.,0.00000
```

AI.2 The Nodal Displacements Obtained for Various Tests

Presented here are the nodal displacements for some of the tests that were performed for the various model configurations described above.

AI.2.1 Fibres Running at a Constant Angle of 30°

Results obtained from PAFEC™:

```
---------------------------------------------------------------------
CASE  1           TRANSLATIONS                   SCALED COORDINATES
 NODE          MULTIPLIED BY 1E 3                MULTIPLIED BY 1E 0
NUMBER       UX         UY         UZ                X        Y       Z
---------------------------------------------------------------------

    1         *      62.500   -50.000              -1.00   -1.00   -1.00
    2     25.000     322.31   -50.000               1.00   -1.00   -1.00
    3         *     -62.500   -50.000              -1.00    1.00   -1.00
    4     25.000     197.31   -50.000               1.00    1.00   -1.00
    5         *      62.500    50.000              -1.00   -1.00    1.00
    6     25.000     322.31    50.000               1.00   -1.00    1.00
    7         *     -62.500    50.000              -1.00    1.00    1.00
    8     25.000     197.31    50.000               1.00    1.00    1.00
    9     12.500     192.40   -50.000               0.00   -1.00   -1.00
   10         *          *    -50.000              -1.00    0.00   -1.00
   11     25.000     259.81   -50.000               1.00    0.00   -1.00
   12     12.500     67.404   -50.000               0.00    1.00   -1.00
   13         *      62.500         *              -1.00   -1.00    0.00
   14     25.000     322.31     0.000               1.00   -1.00    0.00
   15         *     -62.500         *              -1.00    1.00    0.00
   16     25.000     197.31    -0.000               1.00    1.00    0.00
   17     12.500     192.40    50.000               0.00   -1.00    1.00
   18         *          *     50.000              -1.00    0.00    1.00
   19     25.000     259.81    50.000               1.00    0.00    1.00
   20     12.500     67.404    50.000               0.00    1.00    1.00
```

Results Obtained From "XL1":

```
nodal displacements
node            dx             dy            dz
  1        0.0000000E+00 0.6250000E-01-0.5000000E-01
  2        0.2499998E-01 0.3223076E+00-0.5000001E-01
  3        0.0000000E+00-0.6250000E-01-0.5000000E-01
  4        0.2500001E-01 0.1973076E+00-0.4999999E-01
  5        0.0000000E+00 0.6250000E-01 0.5000000E-01
  6        0.2499998E-01 0.3223076E+00 0.5000001E-01
  7        0.0000000E+00-0.6250000E-01 0.5000000E-01
  8        0.2500001E-01 0.1973076E+00 0.4999999E-01
  9        0.1249999E-01 0.1924038E+00-0.5000000E-01
 10        0.0000000E+00 0.0000000E+00-0.5000000E-01
 11        0.2500000E-01 0.2598076E+00-0.5000000E-01
 12        0.1250001E-01 0.6740380E-01-0.5000000E-01
 13        0.0000000E+00 0.6250000E-01 0.0000000E+00
 14        0.2499999E-01 0.3223076E+00-0.4465178E-12
 15        0.0000000E+00-0.6249999E-01 0.0000000E+00
 16        0.2500001E-01 0.1973076E+00-0.3058213E-12
 17        0.1249999E-01 0.1924038E+00 0.5000000E-01
 18        0.0000000E+00 0.0000000E+00 0.5000000E-01
 19        0.2500000E-01 0.2598076E+00 0.5000000E-01
 20        0.1250001E-01 0.6740380E-01 0.5000000E-01
```

Since both packages are capable of performing this test in the same way the results obtained are the identical for each package.

AI.2.2 A Quarter of an Annulus

Results obtained from PAFEC^{TM}:

```
------------------------------------------------------------------
CASE  1            TRANSLATIONS                 SCALED COORDINATES
 NODE           MULTIPLIED BY 1E 3              MULTIPLIED BY 1E 0
NUMBER         UX        UY        UZ               X       Y      Z
------------------------------------------------------------------

   1            *         *     -144.54           1.00    0.00   2.00
   2       -47.553    27.271   -115.98           0.00    0.00   2.00
   3            *    -144.54        *            1.00    2.00   0.00
   4       -47.553   -115.99    27.271           0.00    2.00   0.00
   5            *         *     -167.26           1.00    0.00   1.50
   6       -55.882   -16.180   -139.68           0.00    0.00   1.50
   7            *    -167.26        *            1.00    1.50   0.00
   8       -55.882   -139.68   -16.180           0.00    1.50   0.00
   9       -22.824    20.483   -129.90           0.50    0.00   2.00
  10            *    -124.62   -124.62           1.00    1.41   1.41
  11       -46.488   -120.21   -120.21           0.00    1.41   1.41
  12       -22.824   -129.90    20.483           0.50    2.00   0.00
  13            *         *     -153.21           1.00    0.00   1.75
  14       -51.691     5.794   -129.41           0.00    0.00   1.75
  15            *    -153.21        *            1.00    1.75   0.00
  16       -51.691   -129.41     5.794           0.00    1.75   0.00
  17       -29.644   -15.376   -154.12           0.50    0.00   1.50
  18            *    -141.91   -141.91           1.00    1.06   1.06
  19       -53.911   -136.87   -136.87           0.00    1.06   1.06
  20       -29.644   -154.12   -15.375           0.50    1.50   0.00
```

Results Obtained From *"XL1"*:

```
nodal displacements
node         dx              dy              dz
  1        0.0000000E+00 0.0000000E+00-0.1570446E+00
  2       -0.4599906E-01 0.1532293E-01-0.1375260E+00
  3        0.0000000E+00-0.1570446E+00 0.0000000E+00
  4       -0.4599906E-01-0.1375260E+00 0.1532293E-01
  5        0.0000000E+00 0.0000000E+00-0.1834324E+00
  6       -0.6244982E-01 0.1376562E-02-0.1642944E+00
  7        0.0000000E+00-0.1834324E+00 0.0000000E+00
  8       -0.6244982E-01-0.1642944E+00 0.1376562E-02
  9       -0.2299758E-01 0.8154051E-02-0.1519521E+00
 10        0.0000000E+00-0.1171542E+00-0.1171542E+00
 11       -0.4124055E-01-0.1089619E+00-0.1089619E+00
 12       -0.2299758E-01-0.1519521E+00 0.8154051E-02
 13        0.0000000E-00 0.0000000E+00-0.1687634E+00
 14       -0.5414388E-01 0.8923325E-02-0.1501854E+00
 15        0.0000000E+00-0.1687634E+00 0.0000000E+00
 16       -0.5414388E-01-0.1501854E+00 0.8923325E-02
 17       -0.3156896E-01-0.2950129E-02-0.1764619E+00
 18        0.0000000E+00-0.1338273E+00-0.1338273E+00
 19       -0.5534062E-01-0.1243493E+00-0.1243493E+00
 20       -0.3156896E-01-0.1764619E+00-0.2950129E-02
```

Note: that the displacements are not expected to be identical since the two test elements are not identical in their formation. The "XL1" model has the fibres following the arc of the curve while the PAFEC^{TM} model is composed of ten slices each with a fixed fibre angle. The fibre direction within each of these slices is set so as to simulate the fibre direction following the arc of the annulus.

AI.2.3 For the Fibre Angle Varying Through the Depth of a Cube

Results obtained from PAFEC^{TM}:

```
------------------------------------------------------------------------
CASE  2          TRANSLATIONS                   SCALED COORDINATES
 NODE         MULTIPLIED BY 1E 3                MULTIPLIED BY 1E 0
NUMBER      UX         UY         UZ              X        Y       Z
------------------------------------------------------------------------

    1     -64.757       *       49.174          2.00     2.00    2.00
    2      -3.837       *         *             2.00     2.00    1.00
    3      87.358       *      -33.222          2.00     2.00    0.00
    4      69.580   -17.963     16.910          2.00     1.00    2.00
    5      154.13    129.94    -87.575          2.00     1.00    0.00
    6      191.70   -34.153    -83.404          2.00     0.00    2.00
    7      222.81    110.30    -147.10          2.00     0.00    1.00
    8      216.75    242.72    -204.85          2.00     0.00    0.00
    9     -18.945       *       30.077          1.00     2.00    2.00
   10      31.523       *      -76.789          1.00     2.00    0.00
   11      246.53    -1.705    -85.207          1.00     0.00    2.00
   12      158.29    211.14    -171.27          1.00     0.00    0.00
   13         *         *       51.066          0.00     2.00    2.00
   14         *         *         *             0.00     2.00    1.00
   15         *         *      -77.768          0.00     2.00    0.00
   16      170.01    24.987     48.983          0.00     1.00    2.00
   17      38.568    88.105    -62.824          0.00     1.00    0.00
   18      305.41    25.655    -15.601          0.00     0.00    2.00
   19      207.24    135.17    -49.115          0.00     0.00    1.00
   20      91.402    189.94    -86.229          0.00     0.00    0.00
```

Results Obtained From "XL1":

```
nodal displacements
 node           dx              dy               dz
   1       -0.6331387E-01 0.0000000E+00 0.4105950E-01
   2        0.2019027E-02 0.0000000E+00 0.0000000E+00
   3        0.8046768E-01 0.0000000E+00-0.4057331E-01
   4        0.5544898E-01-0.2206060E-01 0.4703019E-02
   5        0.1499698E+00 0.1313555E+00-0.9817559E-01
   6        0.1829604E+00-0.2699318E-01-0.8862164E-01
   7        0.2181579E+00 0.1008680E+00-0.1613995E+00
   8        0.2051786E+00 0.2582215E+00-0.2389028E+00
   9       -0.2338099E-01 0.0000000E+00 0.2645702E-01
  10        0.2479055E-01 0.0000000E+00-0.7848239E-01
  11        0.2461036E+00-0.8384851E-02-0.9981860E-01
  12        0.1560327E+00 0.2259085E+00-0.1912030E+00
  13        0.0000000E+00 0.0000000E+00 0.5171827E-01
  14        0.0000000E+00 0.0000000E+00 0.0000000E+00
  15        0.0000000E+00 0.0000000E+00-0.6721945E-01
  16        0.1708081E+00 0.1866603E-01 0.3343597E-01
  17        0.3014826E-01 0.9471537E-01-0.7337212E-01
  18        0.3134240E+00 0.1816597E-01-0.2696558E-01
  19        0.1940148E+00 0.1211384E+00-0.7167281E-01
  20        0.8825100E-01 0.1976231E+00-0.1036557E+00
```

Note: that the displacements are not expected to be identical since ıe two test elements are not identical in their formation. The "XL1" model has the fibre angle varying through the depth of a cube from 60° to 0° while the PAFEC^{TM} model is composed of six slices each with a set fibre angle. The displacements are however extremely close considering the differences between the two models.

Appendix II

AII. Input and Output for the Computer Program

Here a sample data file is presented and its structure is explained. An output log generated by the computer program during the matching of systolic cavity and myocardial volumes is also presented.

AII.1 A Sample Data File "XL1-data"

```
7  ba
72,401,0,36,12,112,2
1,1,0,1
```

Title and model parameters

```
   50.0000,     100.0000,      50.0000
 0.470, 0.470, 0.235
   34.0136,      34.0136,      17.0068
   -0.09000       0.20000      -0.09000
400000.0000,400000.0000,400000.0000
 0.470, 0.470, 0.470
136054.4219,136054.4219,136054.4219
    0.00000       0.00000       0.00000
```

Passive and active material properties

```
   0.074437777200,    0.023273280798,    0.062022186360,   -1.000000000000
          .                   .                   .                   .
          .                   .                   .                   .
          .                   .                   .                   .
   0.005457347876,   -0.000056022012,    0.001777471408,   -1.000000000000
```

There are 401 lines of FE data giving x, y and z nodal positions plus the contraction potential.

```
  1   1   1   1 -60   0   0   0   0   0   0   0   0  20   1
  1   2   3   6   7   9  10  11  49  50  52  53  67  68  69  72  73  75  76  77
  2   1   2   1   0  60   0   0   0   0   0   0   0  20   1
  3   4   5   7   8  11  12  13  50  51  53  54  69  70  71  73  74  77  78  79
  3   1   3   1 -60   0   0   0   0   0   0   0   0  20   1
  9  10  11  14  15  17  18  19  52  53  55  56  75  76  77  80  81  83  84  85
  4   1   4   1   0  60   0   0   0   0   0   0   0  20   1
 11  12  13  15  16  19  20  21  53  54  56  57  77  78  79  81  82  85  86  87
  5   1   5   1 -60   0   0   0   0   0   0   0   0  20   1
 17  18  19  22  23  25  26  27  55  56  58  59  83  84  85  88  89  91  92  93
  6   1   6   1   0  60   0   0   0   0   0   0   0  20   1
 19  20  21  23  24  27  28  29  56  57  59  60  85  86  87  89  90  93  94  95
  7   1   7   1 -60   0   0   0   0   0   0   0   0  20   1
 25  26  27  30  31  33  34  35  58  59  61  62  91  92  93  96  97  99 100 101
  8   1   8   1   0  60   0   0   0   0   0   0   0  20   1
 27  28  29  31  32  35  36  37  59  60  62  63  93  94  95  97  98 101 102 103
  9   1   9   1 -60   0   0   0   0   0   0   0   0  20   1
 33  34  35  38  39  41  42  43  61  62  64  65  99 100 101 104 105 107 108 109
 10   1  10   1   0  60   0   0   0   0   0   0   0  20   1
 35  36  37  39  40  43  44  45  62  63  65  66 101 102 103 105 106 109 110 111
 11   1  11   1 -60   0   0   0   0   0   0   0   0  20   0
 41  42  43  46  47   1   2   3  64  65  49  50 107 108 109 112 113  67  68  69
 12   1  12   1   0  60   0   0   0   0   0   0   0  20   0
 43  44  45  47  48   3   4   5  65  66  50  51 109 110 111 113 114  69  70  71
 13   1  13   1 -60   0   0   0   0   0   0   0   0  20   1
 67  68  69  72  73  75  76  77 115 116 118 119 133 134 135 138 139 141 142 143
 14   1  14   1   0  60   0   0   0   0   0   0   0  20   1
 69  70  71  73  74  77  78  79 116 117 119 120 135 136 137 139 140 143 144 145
 15   1  15   1 -60   0   0   0   0   0   0   0   0  20   1
 75  76  77  80  81  83  84  85 118 119 121 122 141 142 143 146 147 149 150 151
 16   1  16   1   0  60   0   0   0   0   0   0   0  20   1
 77  78  79  81  82  85  86  87 119 120 122 123 143 144 145 147 148 151 152 153
 17   1  17   1 -60   0   0   0   0   0   0   0   0  20   1
 83  84  85  88  89  91  92  93 121 122 124 125 149 150 151 154 155 157 158 159
 18   1  18   1   0  60   0   0   0   0   0   0   0  20   1
 85  86  87  89  90  93  94  95 122 123 125 126 151 152 153 155 156 159 160 161
 19   1  19   1 -60   0   0   0   0   0   0   0   0  20   1
 91  92  93  96  97  99 100 101 124 125 127 128 157 158 159 162 163 165 166 167
 20   1  20   1   0  60   0   0   0   0   0   0   0  20   1
 93  94  95  97  98 101 102 103 125 126 128 129 159 160 161 163 164 167 168 169
 21   1  21   1 -60   0   0   0   0   0   0   0   0  20   1
 99 100 101 104 105 107 108 109 127 128 130 131 165 166 167 170 171 173 174 175
```

The description of the "brick" shaped elements

```
 22   1  22   1   0  60   0   0   0   0   0   0   0  20   1
101 102 103 105 106 109 110 111 128 129 131 132 167 168 169 171 172 175 176 177
 23   1  23   1 -60   0   0   0   0   0   0   0   0  20   0
107 108 109 112 113  67  68  69 130 131 115 116 173 174 175 178 179 133 134 135
 24   1  24   1   0  60   0   0   0   0   0   0   0  20   0
109 110 111 113 114  69  70  71 131 132 116 117 175 176 177 179 180 135 136 137
 25   1  25   1 -60   0   0   0   0   0   0   0   0  20   1
133 134 135 138 139 141 142 143 181 182 184 185 199 200 201 204 205 207 208 209
 26   1  26   1   0  60   0   0   0   0   0   0   0  20   1
135 136 137 139 140 143 144 145 182 183 185 186 201 202 203 205 206 209 210 211
 27   1  27   1 -60   0   0   0   0   0   0   0   0  20   1
141 142 143 146 147 149 150 151 184 185 187 188 207 208 209 212 213 215 216 217
 28   1  28   1   0  60   0   0   0   0   0   0   0  20   1
143 144 145 147 148 151 152 153 185 186 188 189 209 210 211 213 214 217 218 219
 29   1  29   1 -60   0   0   0   0   0   0   0   0  20   1
149 150 151 154 155 157 158 159 187 188 190 191 215 216 217 220 221 223 224 225
 30   1  30   1   0  60   0   0   0   0   0   0   0  20   1
151 152 153 155 156 159 160 161 188 189 191 192 217 218 219 221 222 225 226 227
 31   1  31   1 -60   0   0   0   0   0   0   0   0  20   1
157 158 159 162 163 165 166 167 190 191 193 194 223 224 225 228 229 231 232 233
 32   1  32   1   0  60   0   0   0   0   0   0   0  20   1
159 160 161 163 164 167 168 169 191 192 194 195 225 226 227 229 230 233 234 235
 33   1  33   1 -60   0   0   0   0   0   0   0   0  20   1
165 166 167 170 171 173 174 175 193 194 196 197 231 232 233 236 237 239 240 241
 34   1  34   1   0  60   0   0   0   0   0   0   0  20   1
167 168 169 171 172 175 176 177 194 195 197 198 233 234 235 237 238 241 242 243
 35   1  35   1 -60   0   0   0   0   0   0   0   0  20   0
173 174 175 178 179 133 134 135 196 197 181 182 239 240 241 244 245 199 200 201
 36   1  36   1   0  60   0   0   0   0   0   0   0  20   0
175 176 177 179 180 135 136 137 197 198 182 183 241 242 243 245 246 201 202 203
 37   1  37   1 -60   0   0   0   0   0   0   0   0  20   1
199 200 201 204 205 207 208 209 247 248 250 251 265 266 267 270 271 273 274 275
 38   1  38   1   0  60   0   0   0   0   0   0   0  20   1
201 202 203 205 206 209 210 211 248 249 251 252 267 268 269 271 272 275 276 277
 39   1  39   1 -60   0   0   0   0   0   0   0   0  20   1
207 208 209 212 213 215 216 217 250 251 253 254 273 274 275 278 279 281 282 283
 40   1  40   1   0  60   0   0   0   0   0   0   0  20   1
209 210 211 213 214 217 218 219 251 252 254 255 275 276 277 279 280 283 284 285
 41   1  41   1 -60   0   0   0   0   0   0   0   0  20   1
215 216 217 220 221 223 224 225 253 254 256 257 281 282 283 286 287 289 290 291
 42   1  42   1   0  60   0   0   0   0   0   0   0  20   1
217 218 219 221 222 225 226 227 254 255 257 258 283 284 285 287 288 291 292 293
 43   1  43   1 -60   0   0   0   0   0   0   0   0  20   1
223 224 225 228 229 231 232 233 256 257 259 260 289 290 291 294 295 297 298 299
 44   1  44   1   0  60   0   0   0   0   0   0   0  20   1
225 226 227 229 230 233 234 235 257 258 260 261 291 292 293 295 296 299 300 301
 45   1  45   1 -60   0   0   0   0   0   0   0   0  20   1
231 232 233 236 237 239 240 241 259 260 262 263 297 298 299 302 303 305 306 307
 46   1  46   1   0  60   0   0   0   0   0   0   0  20   1
233 234 235 237 238 241 242 243 260 261 263 264 299 300 301 303 304 307 308 309
 47   1  47   1 -60   0   0   0   0   0   0   0   0  20   0
239 240 241 244 245 199 200 201 262 263 247 248 305 306 307 310 311 265 266 267
 48   1  48   1   0  60   0   0   0   0   0   0   0  20   0
241 242 243 245 246 201 202 203 263 264 248 249 307 308 309 311 312 267 268 269
 49   1  49   1 -60   0   0   0   0   0   0   0   0  20   1
265 266 267 270 271 273 274 275 313 314 316 317 331 332 333 336 337 339 340 341
 50   1  50   1   0  60   0   0   0   0   0   0   0  20   1
267 268 269 271 272 275 276 277 314 315 317 318 333 334 335 337 338 341 342 343
 51   1  51   1 -60   0   0   0   0   0   0   0   0  20   1
273 274 275 278 279 281 282 283 316 317 319 320 339 340 341 344 345 347 348 349
 52   1  52   1   0  60   0   0   0   0   0   0   0  20   1
275 276 277 279 280 283 284 285 317 318 320 321 341 342 343 345 346 349 350 351
 53   1  53   1 -60   0   0   0   0   0   0   0   0  20   1
281 282 283 286 287 289 290 291 319 320 322 323 347 348 349 352 353 355 356 357
 54   1  54   1   0  60   0   0   0   0   0   0   0  20   1
283 284 285 287 288 291 292 293 320 321 323 324 349 350 351 353 354 357 358 359
 55   1  55   1 -60   0   0   0   0   0   0   0   0  20   1
289 290 291 294 295 297 298 299 322 323 325 326 355 356 357 360 361 363 364 365
 56   1  56   1   0  60   0   0   0   0   0   0   0  20   1
291 292 293 295 296 299 300 301 323 324 326 327 357 358 359 361 362 365 366 367
 57   1  57   1 -60   0   0   0   0   0   0   0   0  20   1
297 298 299 302 303 305 306 307 325 326 328 329 363 364 365 368 369 371 372 373
 58   1  58   1   0  60   0   0   0   0   0   0   0  20   1
299 300 301 303 304 307 308 309 326 327 329 330 365 366 367 369 370 373 374 375
 59   1  59   1 -60   0   0   0   0   0   0   0   0  20   0
305 306 307 310 311 265 266 267 328 329 313 314 371 372 373 376 377 331 332 333
 60   1  60   1   0  60   0   0   0   0   0   0   0  20   0
307 308 309 311 312 267 268 269 329 330 314 315 373 374 375 377 378 333 334 335
 61   2  61   1   0   0   0   0   0   0   0   0   0  15   0
397 339 331 399 341 333 382 336 379 398 340 332 383 337 380
 62   2  62   1   0   0   0   0   0   0   0   0   0  15   0
399 341 333 401 343 335 383 337 380 400 342 334 384 338 381
 63   2  63   1   0   0   0   0   0   0   0   0   0  15   0
397 347 339 399 349 341 385 344 382 398 348 340 386 345 383
 64   2  64   1   0   0   0   0   0   0   0   0   0  15   0
399 349 341 401 351 343 386 345 383 400 350 342 387 346 384
```

```
 65   2  65   1   0   0   0   0   0   0   0   0   0  15   0
397 355 347 399 357 349 388 352 385 398 356 348 389 353 386
 66   2  66   1   0   0   0   0   0   0   0   0   0  15   0
399 357 349 401 359 351 389 353 386 400 358 350 390 354 387
 67   2  67   1   0   0   0   0   0   0   0   0   0  15   0
397 363 355 399 365 357 391 360 388 398 364 356 392 361 389
 68   2  68   1   0   0   0   0   0   0   0   0   0  15   0
399 365 357 401 367 359 392 361 389 400 366 358 393 362 390
 69   2  69   1   0   0   0   0   0   0   0   0   0  15   0
397 371 363 399 373 365 394 368 391 398 372 364 395 369 392
 70   2  70   1   0   0   0   0   0   0   0   0   0  15   0
399 373 365 401 375 367 395 369 392 400 374 366 396 370 393
 71   2  71   1   0   0   0   0   0   0   0   0   0  15   0
397 331 371 399 333 373 379 376 394 398 332 372 380 377 395
 72   2  72   1   0   0   0   0   0   0   0   0   0  15   0
399 333 373 401 335 375 380 377 395 400 334 374 381 378 396
```

The description of the "wedge" shaped elements

```
 5,1,1,1
 8,1,1,1
13,1,1,1
16,1,1,1
21,1,1,1
24,1,1,1
29,1,1,1
32,1,1,1
37,1,1,1
40,1,1,1
45,1,1,1
48,1,1,1
```

The restrained nodes

```
 2.,0.,0.,1.,0.80984
 4.,0.,0.,1.,0.80984
 6.,0.,0.,1.,0.80984
 8.,0.,0.,1.,0.80984
10.,0.,0.,1.,0.80984
12.,0.,0.,1.,0.80984
14.,0.,0.,1.,0.80984
16.,0.,0.,1.,0.80984
18.,0.,0.,1.,0.80984
20.,0.,0.,1.,0.80984
22.,0.,0.,1.,0.80984
24.,0.,0.,1.,0.80984
26.,0.,0.,1.,0.80984
28.,0.,0.,1.,0.80984
30.,0.,0.,1.,0.80984
32.,0.,0.,1.,0.80984
34.,0.,0.,1.,0.80984
36.,0.,0.,1.,0.80984
38.,0.,0.,1.,0.80984
40.,0.,0.,1.,0.80984
42.,0.,0.,1.,0.80984
44.,0.,0.,1.,0.80984
46.,0.,0.,1.,0.80984
48.,0.,0.,1.,0.80984
50.,0.,0.,1.,0.80984
52.,0.,0.,1.,0.80984
54.,0.,0.,1.,0.80984
56.,0.,0.,1.,0.80984
58.,0.,0.,1.,0.80984
60.,0.,0.,1.,0.80984
62.,0.,0.,1.,0.80984
64.,0.,0.,1.,0.80984
66.,0.,0.,1.,0.80984
68.,0.,0.,1.,0.80984
70.,0.,0.,1.,0.80984
72.,0.,0.,1.,0.80984
```

Pressure loaded elements plus the pressure and the face that is loaded.

The purpose of each part of the data file will now be explained in brief

Title and Model Parameters:

The title, which is automatically generated by the computer program and is formed from the frame number and the name of the patient data file
Total number of elements
Total number of nodal points
Number of point loads to be applied (not presently implemented)
Number of pressure loaded faces
Number of constrained nodes
Maximum node number difference in each element
Number of material types

Material properties

The elastic moduli in the X, Y and Z-directions (E_x, E_y, E_z)
Poisson's ratios in the YX, ZX and ZY-planes (v_{yx}, v_{zx}, v_{zy})
The modulus of rigidity in the XY, YZ and ZX-directions (G_{xy}, G_{yz}, G_{zx}) (See *)
The contraction (thermal expansion) coefficients in the X, Y and Z directions
(Note: the above set of values may be repeated depending on the number of material types specified)

Coordinate data

The X, Y and Z coordinates and the contraction potential

The description of the elements

For each element there are two lines of data.

The first line:
The element number
The number of the material from which it is formed
The orthotropic code number (0, 1, 2 or 3)
The 9 orthotropic angles (See **)
The number of nodes per element (20 for a brick 15 for a wedge)
The sequential node number indicator (1 for increasing, 0 otherwise)

The second line:
This gives the nodes which form the element. These are numbered such that one first moves in the positive local Z-direction then the positive local Y-direction, then in the positive local X-direction.

The constrained nodes

Node number, 1 if constrained 0 if unconstrained in the X-direction, the same then follows for the Y and Z-directions

Pressure loaded faces

Element number, +1 if the positive X-face is pressure loaded, -1 if the negative X-face is loaded and 0 if unloaded. The Y and Z-faces are also specified in the same way. The 15-noded wedge shaped elements can only be loaded on the Z faces. Next is the pressure to be applied.

* To enable greater ease of application of the modelling process there are assumed relationships between the elastic and shear moduli.

$$\mathbf{G}_{xy} = \mathbf{G}_{yz} = \mathbf{r}\mathbf{G}_{zx} = \frac{\mathbf{E}_y}{2(1 + v_{yx})}$$

Where,

$$\mathbf{r} = \frac{\mathbf{E}_y}{\mathbf{E}_x}$$

Note also that $v_{yx}=v_{zx}=rv_{zy}=0.47$ if $r < 1$ otherwise $v_{yx}/r=v_{zx}/r=v_{zy}=0.47$ and normally $r=2$.

** The importance of the orthotropic code numbers and the orthotropic angles is now explained.

Code 0: No angles are required as the material axes are the element local axes

Code 1: The first angle is the offset in the local X,Y-plane for the face Z=-1 and the second is the offset at Z=+1. The material axes are then taken to vary linearly between the two faces.

Code 2: The material axes are in the global directions and thus no numbers are required.

Code 3: The numbers in degrees specify the directional cosines for E_x in the X, Y, Z-directions then likewise for E_y and E_z.

AII.2 A Sample Output Log

This is a sample output log from the computer program. It is for patient "BA" over end systole. The myocardial volume is kept constant while the target volume is matched. The contraction coefficient and contraction coefficient ratio are refined until the errors in the required cavity and myocardial volumes are less than 0.1% of the total. The next frame for analysis is then selected, until all frames have been analysed.

```
Patient ba

 Frame   7
Iteration  1
Work done =      0.00008734360611
Target Base to Apex Length= 79.343mms
Target volume is       155.780
Calculated volume is  163.357
Calculated Base to Apex Length= 81.494mms
Target Shape Index         0.903
Calculated Shape Index    0.932
Target Myocardial Volume       113.908
Calculated Myocardial Volume  114.880
 bounds lower   0.  0.
 new ratio    0.442324
 bounds upper    0.450000    114.880
 Step is      1.00000
Iteration  2
Work done =      0.00008700949608
Target Base to Apex Length= 79.343mms
Target volume is       155.780
Calculated volume is  162.990
Calculated Base to Apex Length= 81.417mms
Target Shape Index         0.903
Calculated Shape Index    0.932
Target Myocardial Volume       113.908
Calculated Myocardial Volume  114.533
 bounds lower   0.  0.
 new ratio    0.437470
 bounds upper    0.442324    114.533
 Step is      1.00000
Iteration  3
Work done =      0.00008684676709
Target Base to Apex Length= 79.343mms
Target volume is       155.780
Calculated volume is  162.763
Calculated Base to Apex Length= 81.368mms
Target Shape Index         0.903
Calculated Shape Index    0.932
Target Myocardial Volume       113.908
Calculated Myocardial Volume  114.315
 bounds lower   0.  0.
 new ratio    0.434341
 bounds upper    0.437470    114.315
 Step is      1.00000
Iteration  4
Work done =      0.00008676208059
Target Base to Apex Length= 79.343mms
Target volume is       155.780
Calculated volume is  162.613
Calculated Base to Apex Length= 81.337mms
Target Shape Index         0.903
Calculated Shape Index    0.932
Target Myocardial Volume       113.908
Calculated Myocardial Volume  114.176
 bounds lower   0.  0.
 new ratio    0.432295
 bounds upper    0.434341    114.176
 Step is      1.00000
Iteration  5
Work done =      0.00008671435273
Target Base to Apex Length= 79.343mms
Target volume is       155.780
Calculated volume is  162.515
Calculated Base to Apex Length= 81.316mms
```

```
Target Shape Index         0.903
Calculated Shape Index    0.933
Target Myocardial Volume       113.908
Calculated Myocardial Volume  114.085
 bounds lower   0.  0.
 new ratio    0.430956
 bounds upper    0.432295    114.085
 Step is      1.00000
Iteration  6
Work done =      0.00008668651638
Target Base to Apex Length= 79.343mms
Target volume is       155.780
Calculated volume is  162.450
Calculated Base to Apex Length= 81.303mms
Target Shape Index         0.903
Calculated Shape Index    0.933
Target Myocardial Volume       113.908
Calculated Myocardial Volume  114.024
 bounds lower   0.  0.
 new ratio    0.430079
 bounds upper    0.430956    114.024
 Step is      1.00000
Iteration  7
Work done =      0.00008667046702
Target Base to Apex Length= 79.343mms
Target volume is       155.780
Calculated volume is  162.409
Calculated Base to Apex Length= 81.294mms
Target Shape Index         0.903
Calculated Shape Index    0.933
Target Myocardial Volume       113.908
Calculated Myocardial Volume  113.986
Surface Area=15081.40mm2
Shape Index=0.932551
Volume of Myocardium = 113.986
Contraction Coefficient = 0.20000
Contraction Coefficient Ratio = 0.43008

 Frame   8
Iteration  1
Work done =      0.00009450791430
Target Base to Apex Length= 80.324mms
Target volume is       148.530
Calculated volume is  150.198
Calculated Base to Apex Length= 81.085mms
Target Shape Index         0.896
Calculated Shape Inde     0.908
Target Myocardial Vol me       121.427
Calculated Myocardial Volume  121.089
 bounds lower    0.430079    121.089
 new ratio    0.432472
 bounds upper   0.  0.
 Step is      1.00000
Iteration  2
Work done =      0. 0009453129230
Target Base to Apex Length- 80.324mms
Target volume is       148.530
Calculated volume is  150.309
Calculated Base to Apex Length= 81.1 9mms
Target Shape Index         0.896
Calculated Shape Index    0.908
Target Myocardial Volume       121.427
Calculated Myocardial Volume  121.2 [illegible]
 bounds lower    0.432472    121.2 2
 new ratio    0.434 76
```

```
   bounds upper   0.  0.
   Step is     1.00000
  Iteration  3
  Work done =     0.00009455329343
  Target Base to Apex Length= 80.324mms
  Target volume is      148.530
  Calculated volume is  150.391
  Calculated Base to Apex Length= 81.126mms
  Target Shape Index        0.896
  Calculated Shape Index    0.908
  Target Myocardial Volume      121.427
  Calculated Myocardial Volume  121.281
   bounds lower    0.434 76    121.281
   new ratio     .435120
   bounds upper   0.   .
   Step is     1.00 00
  Iteration  4
  Work done -     0.00  9456887258
  Target Base to Apex Length= 80.324mms
  Target volume is      148.530
  Calculated volume is  150.442
  Calculated Base to Apex Length= 81.136mms
  Target Shape Index        0.896
  Calculated Shape Index    0.9 8
  Target My cardial Volume      121.427
  Calculated Myocardial Volume  121.329
  Surface Area-14586.72mm2
  Shape Index  .9 8152
  V lume  f My cardium = 121.329
  C ntracti n C efficıent = 0.200 0
  Contraction C efficient Ratio =  .43512

   Frame   9
  Iterati n  1
  W rk d ne -      .    96 999246
  Target Base t  Apex Length= 78.227mms
  Target v lume is      139.2
  Calculated v lume is  138.284
  Calculated Base t  Apex Length= 8 .75 mms
  Target Shape Index        .897
  Calculated Shape Index    .868
  Target My cardıal V lume      122.98
  Calculated My cardıal V lume  121.952
   b unds l wer     .43512     121.952
   new rati     .44_396
   b unds upper    .  .
   Step is     1.
  Iterati n  2
  W rk d ne -      .    9674881968
  Target Base to Apex Length  78.227mms
  Target v lume is      139.2
  Calculated v lume ıs  138.59
  Calculated Base to Apex Length= 80.829mms
  Target Shape Index        .897
  Calculated Shape Index    .868
  Target Myocardial Volume      122.980
  Calculated Myocardial V lume  122.298
   bounds l wer     .442396    122.298
   new ratio     .447299
   bounds upper    .  0.
   Step is     1.0  0
  Iteration  3
  W rk d ne =     0.0 0 967643 587
  Target Base t  Apex Length= 78.227mms
  Target volume is      139.20
  Calculated volume is  138.791
  Calculated Base to Apex Length= 80.882mms
  Target Shape Index        .897
  Calculated Shape Index    0.868
  Target My cardial V lume      122.980
  Calculated Myocardial V lume  122.531
   bounds lower    0.447299    122.531
   new ratio    0.45 561
   bounds upper    .   .
   Step is     1.
  Iteration  4
  Work done =      . 0  9679662705
  Target Base to Apex Length= 78.227mms
  Target v lume is      139.2 0
  Calculated v lume is  138.925
  Calculated Base to Apex Length- 8 .918mms
  Target Shape Index        .897
  Calculated Shape Index    .868
  Target My caıdial V lume      122.980
  Calculated Myocardial V lume  12 .686
   bounds lower    0.45 561    1 2.686
```

```
   new ratio    0.452712
   bounds upper   0.  0.
   Step is     1.00000
  Iteration  5
  Work done =     0.00009682768876
  Target Base to Apex Length= 78.227mms
  Target volume is      139.200
  Calculated volume is  139.014
  Calculated Base to Apex Length= 80.941mms
  Target Shape Index        0.897
  Calculated Shape Index    0.868
  Target Myocardial Volume      122.980
  Calculated Myocardial Volume  122.789
   bounds lower    0.452712    122.789
   new ratio    0.454119
   bounds upper   0.  0.
   Step is     1.00000
  Iteration  6
  Work done =     0.00009685204781
  Target Base to Apex Length= 78.227mms
  Target volume is      139.200
  Calculated volume is  139.072
  Calculated Base to Apex Length= 80.956mms
  Target Shape Index        0.897
  Calculated Shape Index    0.868
  Target Myocardial Volume      122.980
  Calculated Myocardial Volume  122.855
   bounds lower    0.454119    122.855
   new ratio    0.455037
   bounds upper   0.  0.
   Step is     1.00000
  Iteration  7
  Work done =     0.00009687043900
  Target Base to Apex Length= 78.227mms
  Target volume is      139.200
  Calculated volume is  139.111
  Calculated Base to Apex Length= 80.966mms
  Target Shape Index        0.897
  Calculated Shape Index    0.868
  Target Myocardial Volume      122.980
  Calculated Myocardial Volume  122.901
  Surface Area=14271.04mm2
  Shape Index=0.867767
  V lume of Myocardium = 122.901
  C ntraction Coefficient = 0.20000
  Contraction Coefficient Ratio = 0.45504

   Frame   10
  Iteration  1
  Work done =     0.00007966579328
  Target Base to Apex Length= 76.467mms
  Target volume is      130.480
  Calculated volume is  130.252
  Calculated Base to Apex Length= 78.124mms
  Target Shape Index        0.904
  Calculated Shape Index    0.880
  Target Myocardial Volume      105.337
  Calculated Myocardial Volume  106.109
   bounds lower   0.  0.
   new ratio    0.448361
   bounds upper    0.455037    106.109
   Step is     1.00000
  Iteration  2
  Work done =     0.00007953520882
  Target Base to Apex Length= 76.467mms
  Target volume is      130.480
  Calculated volume is  129.983
  Calculated Base to Apex Length= 78.053mms
  Target Shape Index        0.904
  Calculated Shape Index    0.880
  Target Myocardial Volume      105.337
  Calculated Myocardial Volume  105.830
   bounds lower   0.  0.
   new ratio    0.444162
   bounds upper    0.448361    105.830
   Step is     1.00000
  Iteration  3
  Work done =     0.00007948642604
  Target Base to Apex Length= 76.467mms
  Target volume is      130.480
  Calculated volume is  129.815
  Calculated Base to Apex Length= 78.008mms
  Target Shape Index        0.904
  Calculated Shape Index    0.880
  Target Myocardial Volume      105.337
  Calculated Myocardial Volume  105.655
```

```
 bounds lower   0.  0.
 new ratio    0.441477
 bounds upper    0.444162    105.655
 Step is     1.00000
Iteration  4
Work done =     0.00007946876402
Target Base to Apex Length= 76.467mms
Target volume is      130.480
Calculated volume is  129.709
Calculated Base to Apex Length= 77.980mms
Target Shape Index        0.904
Calculated Shape Index    0.880
Target Myocardial Volume      105.337
Calculated Myocardial Volume  105.545
 bounds lower   0.  0.
 new ratio    0.439733
 bounds upper    0.441477    105.545
 Step is     1.00000
Iteration  5
Work done =     0.00007946266096
Target Base to Apex Length= 76.467mms
Target volume is      130.480
Calculated volume is  129.639
Calculated Base to Apex Length= 77.961mms
Target Shape Index        0.904
Calculated Shape Index    0.880
Target Myocardial Volume      105.337
Calculated Myocardial Volume  105.472
 bounds lower   0.  0.
 new ratio    0.438603
 bounds upper    0.439733    105.472
 Step is     1.00000
Iteration  6
Work done =     0.00007946105916
Target Base to Apex Length= 76.467mms
Target volume is      130.480
Calculated volume is  129.593
Calculated Base to Apex Length= 77.949mms
Target Shape Index        0.904
Calculated Shape Index    0.880
Target Myocardial Volume      105.337
Calculated Myocardial Volume  105.424
Surface Area=13481.37mm2
Shape Index=0.880454
Volume of Myocardium = 105.424
Contraction Coefficient = 0.20000
Contraction Coefficient Ratio = 0.43860

 Frame   11
Iteration  1
Work done =     0.00009023811988
Target Base to Apex Length= 76.217mms
Target volume is      121.740
Calculated volume is  119.860
Calculated Base to Apex Length= 77.429mms
Target Shape Index        0.857
Calculated Shape Index    0.881
Target Myocardial Volume      109.326
Calculated Myocardial Volume  108.608
 bounds lower    0.438603    108.608
 new ratio    0.444363
 bounds upper   0.  0.
 Step is     1.00000
Iteration  2
Work done =     0.00009026945061
Target Base to Apex Length= 76.217mms
Target volume is      121.740
Calculated volume is  120.054
Calculated Base to Apex Length= 77.488mms
Target Shape Index        0.857
Calculated Shape Index    0.881
Target Myocardial Volume      109.326
Calculat d Myocardial Volume  108.856
 bounds lower    0.444363    108.856
 new ratio    0.448187
 bounds upper   0.  0.
 Step is     1.00000
Iteration  3
Work done =     0.00009031818126
Target Base to Apex Length= 76.217mms
Target volume is      121.740
Calculated volume is  120.192
Calculated Base to Apex Length= 77.528mms
Target Shape Index        0.857
Calculated Shape Index    0.880
Target Myocardial Volume      109.326
```

```
Calculated Myocardial Volume  109.022
 bounds lower    0.448187    109.022
 new ratio    0.450682
 bounds upper   0.  0.
 Step is     1.00000
Iteration  4
Work done =     0.00009036174544
Target Base to Apex Length= 76.217mms
Target volume is      121.740
Calculated volume is  120.282
Calculated Base to Apex Length= 77.553mms
Target Shape Index        0.857
Calculated Shape Index    0.880
Target Myocardial Volume      109.326
Calculated Myocardial Volume  109.129
 bounds lower    0.450682    109.129
 new ratio    0.452302
 bounds upper   0.  0.
 Step is     1.00000
Iteration  5
Work done =     0.00009039455240
Target Base to Apex Length= 76.217mms
Target volume is      121.740
Calculated volume is  120.339
Calculated Base to Apex Length= 77.570mms
Target Shape Index        0.857
Calculated Shape Index    0.880
Target Myocardial Volume      109.326
Calculated Myocardial Volume  109.199
 bounds lower    0.452302    109.199
 new ratio    0.453355
 bounds upper   0.  0.
 Step is     1.00000
Iteration  6
Work done =     0.00009041816484
Target Base to Apex Length= 76.217mms
Target volume is      121.740
Calculated volume is  120.377
Calculated Base to Apex Length= 77.580mms
Target Shape Index        0.857
Calculated Shape Index    0.880
Target Myocardial Volume      109.326
Calculated Myocardial Volume  109.244
Surface Area=12835.67mm2
Shape Index=0.880324
Volume of Myocardium = 109.244
Contraction Coefficient = 0.20000
Contraction Coefficient Ratio = 0.45336

 Frame   12
Iteration  1
Work done =     0.00008020624438
Target Base to Apex Length= 77.105mms
Target volume is      114.730
Calculated volume is  112.698
Calculated Base to Apex Length= 78.594mms
Target Shape Index        0.851
Calculated Shape Index    0.828
Target Myocardial Volume      103.001
Calculated Myocardial Volume  104.228
 bounds lower   0.  0.
 new ratio    0.442556
 bounds upper    0.453355   104.2279
 Step is     1.00000
Iteration  2
Work done =     0.00008001608464
Target Base to Apex Length= 77.105mms
Target volume is      14.730
Calculated volume is  112.322
Calculated Base to Apex Length= 78.476mms
Target Shape Index        0.851
Calculated Shape Index    0.829
Target Myocardial Volume      103.0 1
Calculated Myocardial Volume  103.777
 bounds lower   0.  0.
 new ratio    0.435886
 bounds upper    0.442556   103.7772
 Step is     1.000 0
Iteration  3
Work done =     0.00007998116207
Target Base to Apex Length= 77.105mms
Target volume is      114.730
Calculated volume is  112.098
Calculated Base to Apex Length- 78.4 4mms
Target Shape Index          .851
Calculated Shape Index      .829
```

```
Target Myocardial Volume      103.001
Calculated Myocardial Volume  103.500
 bounds lower   0.  0.
 new ratio    0.431660
 bounds upper    0.435886   103.5004
 Step is     1.00000
Iteration  4
Work done =     0.00 07999165717
Target Base to Apex Length= 77.105mms
Target volume is       114.730
Calculated volume is  111.951
Calculated Base to Apex Length= 78.358mms
Target Shape Index         0.851
Calculated Shape Index    0.829
Target Myocardial Volume      103.001
Calculated Myocardial Volume  103.324
 bounds lower   0.  0.
 new ratio    0.428955
 bounds upper     .431660   103.3238
 Step is     1.0 000
Iterati n  5
Work done =       . 0 08  1160342
Target Base to Apex Length= 77.105mms
Target v lume is       114.730
Calculated v lume is  111.857
Calculated Base t  Apex Length= 78.328mms
Target Shape Index         0.851
Calculated Shape Index    0.829
Target My cardial V lume      1 3.001
Calculated My cardial Volume  1 3.212
 bounds l wer     .  0.
 new ratio     .4272 1
 b unds upper     .428955   1 3.2117
 Step is     1.
Iterati n  6
W rk d ne          .     8  3  4673
Target Base t  Apex Length- 77.1 5mms
Target v lume is       114.73
Calculated v lume is  111.79
Calculated Base t  Apex Length  78.3 9mms
Target Shape Index          .851
Calculated Shape Index     .829
Target My cardial V lume      1 3.0 1
Calculated My cardial Volume  1 3.139
 b unds l wer     .   .
 new rati      .42o 56
 b unds upper     .42 2 1   1 3.1391
 Step is     1.
Iterati n  7
W rk d ne -        .     8  44532 8
Target Base to Apex Length= 77.1 5mms
Target v lume is       114.730
Calculated v lume is  111.757
Calculated Base t  Apex Length= 78.297mms
Target Shape Index          .851
Calculated Shape Index    0.829
Target Myocardial V lume      1 3.0 1
Calculated Myocardial Volume  103.091
Surface Area=12714.03mm2
Shape Index=0.829 41
V lume of Myocardium = 103.091
Contracti n Coefficient = 0.2 000
Contraction C efficient Ratio = 0.42606

 Frame   13
Iteration  1
Work done -      0.    8 36180793
Target Base to Apex Length= 76.016mms
Target volume is       107.190
Calculated volume is  1 6.535
Calculated Base to Apex Length= 76.253mms
Target Shape Index         0.852
Calculated Shape Index    0.838
Target My cardial Volume      108.922
Calculated My cardial V lume  107.818
 vol error          .611273837136 82
 bounds l wer      .4 6 56    1 7.818
 new ratio     .43469o
 bounds upper      .    .
 Step is     1. 0
Iteration  2
Work done            .     8  87789 2
Target Base to Apex Length- 76.016mms
Target volume is       1   .19
Calculated v lume is  1 6.8 1
Calculated Base t  Apex Length   6.337mms
```

```
Target Shape Index         0.852
Calculated Shape Index    0.838
Target Myocardial Volume      108.922
Calculated Myocardial Volume  108.186
 bounds lower    0.434696    108.186
 new ratio    0.440574
 bounds upper   0.  0.
 Step is     1.00000
Iteration  3
Work done =      0.00008230118273
Target Base to Apex Length= 76.016mms
Target volume is       107.190
Calculated volume is  107.009
Calculated Base to Apex Length= 76.394mms
Target Shape Index         0.852
Calculated Shape Index    0.838
Target Myocardial Volume      108.922
Calculated Myocardial Volume  108.435
 bounds lower    0.440574    108.435
 new ratio    0.444514
 bounds upper   0.  0.
 Step is     1.00000
Iteration  4
Work done =      0.00008233917224
Target Base to Apex Length= 76.016mms
Target volume is       107.190
Calculated volume is  107.140
Calculated Base to Apex Length= 76.432mms
Target Shape Index         0.852
Calculated Shape Index    0.838
Target Myocardial Volume      108.922
Calculated Myocardial Volume  108.604
 bounds lower    0.444514    108.604
 new ratio    0.447114
 bounds upper   0.  0.
 Step is     1.00000
Iteration  5
Work done =      0.00008237692769
Target Base to Apex Length= 76.016mms
Target volume is       107.190
Calculated volume is  107.226
Calculated Base to Apex Length= 76.458mms
Target Shape Index         0.852
Calculated Shape Index    0.838
Target Myocardial Volume      108.922
Calculated Myocardial Volume  108.715
 bounds lower    0.447114    108.715
 new ratio    0.448816
 bounds upper   0.  0.
 Step is     1.00000
Iteration  6
Work done =      0.00008240707530
Target Base to Apex Length= 76.016mms
Target volume is       107.190
Calculated volume is  107.283
Calculated Base to Apex Length= 76.474mms
Target Shape Index         0.852
Calculated Shape Index    0.838
Target Myocardial Volume      108.922
Calculated Myocardial Volume  108.788
 bounds lower    0.448816    108.788
 new ratio    0.449927
 bounds upper   0.  0.
 Step is     1.00000
Iteration  7
Work done =      0.00008242991605
Target Base to Apex Length= 76.016mms
Target volume is       107.190
Calculated volume is  107.321
Calculated Base to Apex Length= 76.485mms
Target Shape Index         0.852
Calculated Shape Index    0.838
Target Myocardial Volume      108.922
Calculated Myocardial Volume  108.837
Surface Area=12285.07mm2
Shape Index=0.838193
Volume of Myocardium = 108.837
Contraction Coefficient = 0.20000
Contraction Coefficient Ratio = 0.44993

 Frame   14
Iteration  1
Work done =      0.00007504197295
Target Base to Apex Length= 75.460mms
Target volume is       1 4.080
Calculated volume is  1   .890
Calculated Base to Apex Length= 74.757mms
Target Shape Index          0.836
Calculated Shape Index      0.841
Target Myocardial Volume          96.421
Calculated Myocardial Volume    96.880
 bounds lower   0.  0.
 new ratio    0.445648
 bounds upper    0.449927    96.8797
 Step is     1.00000
Iteration  2
Work done =       0.00007497528035
Target Base to Apex Length= 75.460mms
Target volume is        104.080
Calculated volume is  100.755
Calculated Base to Apex Length= 74.716mms
Target Shape Index          0.836
Calculated Shape Index      0.841
Target Myocardial Volume          96.421
Calculated Myocardial Volume    96.716
 bounds lower   0.  0.
 new ratio    0.442922
 bounds upper    0.445648    96.7160
 Step is     1.00000
Iteration  3
Work done =       0.00007494553878
Target Base to Apex Length= 75.460mms
Target volume is        104.080
Calculated volume is  100.665
Calculated Base to Apex Length= 74.690mms
Target Shape Index          0.836
Calculated Shape Index      0.841
Target Myocardial Volume          96.421
Calculated Myocardial Volume    96.611
```

```
 bounds lower   0.  0.
 new ratio    0.441174
 bounds upper    0.442922    96.6114
 Step is     1.00000
Iteration  4
Work done =       0.00007493185553
Target Base to Apex Length= 75.460mms
Target volume is        104.080
Calculated volume is  100.607
Calculated Base to Apex Length= 74.673mms
Target Shape Index          0.836
Calculated Shape Index      0.841
Target Myocardial Volume          96.421
Calculated Myocardial Volume    96.545
 bounds lower   0.  0.
 new ratio    0.440041
 bounds upper    0.441174    96.5450
 Step is     1.00000
Iteration  5
Work done =       0.00007492534502
Target Base to Apex Length= 75.460mms
Target volume is        104.080
Calculated volume is  100.571
Calculated Base to Apex Length= 74.663mms
Target Shape Index          0.836
Calculated Shape Index      0.841
Target Myocardial Volume          96.421
Calculated Myocardial Volume    96.503
Surface Area=11736.46mm2
Shape Index=0.841186
Volume of Myocardium =  96.503
Contraction Coefficient = 0.20000
Contraction Coefficient Ratio = 0.44004
```

Appendix III

AIII. The Computer Program Listing

```
      program heart
C*****************************************************
C                                                    *
C THIS IS PROGRAM HEART+. THE PROGRAM IS CONSTRUCTED SO *
C AS TO ENABLE THE USER TO CARRY OUT AN ANALYSIS OF A   *
C PATIENT'S LEFT VENTRICLE VIA THE FINITE ELEMENT      *
C METHOD ON A INTERACTIVE OR NONINTERACTIVE LEVEL. A   *
C COMPLETE GUIDE TO ITS FORM AND COMPOSITION CAN BE    *
C FOUND IN CHAPTER SEVEN OF B.S.GREWAL'S THESIS WITH   *
C SUPPLIMENTARY INFORMATION IN CHAPTER 8 OF THIS ONE.  *
C                                                    *
C*****************************************************
c
c     last modified on 20th August 1996
c
      character part

1     call intro(part)

      if(part.eq.'1') then
         call part1
      elseif(part.eq.'2') then
         call part2
      elseif(part.eq.'3') then
         call part3
      elseif(part.eq.'4') then
         call part4
      endif

      if((part.ne.'Q').and.(part.ne.'q')) goto 1

      stop
      end


      subroutine intro(part)
c
c     It is called from the main program
c
c     It gives a menu of the main options available
c
c     It makes no routine calls
c
      character part

1     write(6,9)
      read(5,'(a1)') part

9     format(//'     The following program is composed of
4 Parts:'/
     +'    -------------------------------------------
'//
     +'    Part 1 : production of volume data plus a new
datafile'/
     +'    for a particular patient for analysis in Part
4.'//
     +'    Part 2 : analyses of results from Part4 plus'/
     +'    production of new datafiles for subsequent
analyses.'//
     +'    Part 3 : visual display of major cross-
sectional '/
     +'    planes before and after deformation, 3D display
of the'/
     +'    finite element mesh and graphs of ventricular'/
     +'    properties v time'//
     +'    Part 4 : automated control of the finite
element analysis'/
     +'    for the diastolic and systolic phase (This
negates the use'/
     +'    of Part 2 and the mesh generation part of Part
1)'//
     +'    Q to quit the program'///
     +'    Which part do you require? ',$)

       return
       end

c*****************************************************
c
c      part 1 analysis of original patient data
c
c*****************************************************


       subroutine part1
c
c      It is called by the main program
c
c      It controls the volume calcutations, smoothing
through
c      volumes, ploting of pressure and volume graphs and
c      production of data files.
c
c      it makes call to routines readfilvenvol femvol
graphs
c
      integer ixrn(151,151),iyrn(151,151 ,ixlm(151,151 ,

+iyln(151,151),ixrp(151,151),iyrp(151,151 ,ixlp 151,151 ,
     +iylp(151,151),nrn(151),nln(151),nrp(151),nlp(151
      real preses(151),volu(151),volum(151)
      integer nf,i,ijk,iresp,nlm,number,ifrmno
      character
filnam*80,filenam*80,answer,patr*6,namer*20,
     +comtr*20,cframe*2
      common/person/ filenam,cframe

C-----Patient data read in here for first time.
      write(6,1)
      read(5,'(a80)') filenam

      call readfil(filenam,patr,namer,comtr,nf,ixrn,iyrn,

+nrn,ixln,iyln,nln,ixrp,iyrp,nrp,ixlp,iylp,nlp,preses

C-----Pressures sent to file
      filnam='pressure.'//filenam

open(7,file=filnam,access='sequential',form='formatted'
      write(7,240) nf
      write(7,230) (preses(i),i=1,nf)
      close(7)

C-----Option to view 3D reconstruction and/or obtain
volume info via
C     VENVOL
15    write(6,73)
      read(*,*) answer
      if(answer.eq.'Y'.or.answer.eq.'y') then
        call venvol(volu)
      elseif(answer.ne.'N'.and.answer.ne.'n') then
        goto 15
      endif

C-----Option to obtain volume info for finite element
model
C     representation
75    write(6,83) nf
      read(*,*) answer
      if(answer.eq.'Y'.or.answer.eq.'y') then
        ijk=0
        iresp=0
        call femvol(volum,ijk,nlm,iresp)
      elseif(answer.ne.'N'.and.answer.ne.'n') then
        goto 75
      endif

C-----Graphical and smoothing optional call made here
85    write(6,209)
      read(*,*) answer
      if(answer.eq.'Y'.or.answer.eq.'y') then
212     write(6,19)
        read(*,*) number
        if((number.lt.1).or.(number.gt.9)) goto 211
        call
graphs(preses,volu,volum,nf,number,patr,namer,comtr)
        goto 212
      elseif(answer.ne.'N'.and.answer.ne.'n') then
        goto 85
      endif

C-----Option to create datafile for FE analysis
211    write(6,217)
       read(*,*) answer
       if(answer.eq.'Y'.or.answer.eq.'y') then
         write(6,213)
         read(*,*) ifrmno
         iresp=0
         call femvol(volum,ifrmno,nlm,iresp)
         ifrmno=0
       elseif(answer.ne.'N'.and.answer.ne.'n') then
         goto 211
       endif

19     format(////'     Choose from the options below :'/
      +'     -------------------------------'//
      +'     Pressure & Volume (from VOLUME) v Time   :
1'/
      +'     Pressure & Volume (from FE Mesh) v Time  :
2'/
      +'     Pressure & Volumes (from 1 & 2) v Time   :
3'/
      +'     Pressure - Volume loop (from VOLUME)     :
4'/
      +'     Pressure - Volume loop (from FE Mesh)    :
5'/
      +'     Pressure - Volume (loops from 4 & 5)     :
6'/
+'   Smoothened Volume (from VOLUME) v Time   :
```

```
7'/
    +'      Smoothened Volume (from FE Mesh) v Time  :
8'/
    +'      Smoothened V [from VOLUME]          '/
    +'                             & Target FE Volumes  :
9'///
    +' Enter number for graph required (0 exits) ',$)
1     format(////
    +' Enter the filnam containing all the patient''s
data from '/
    +' the Brompton Hospital'/'         ',$)
1002  format(5a2)
240   format(1x,i3)
230   format(f7.3)
1003  format(5(2i6,3x))
1004  format(10(f7.2))
73    format(////
    +' Do you wish to obtain volume information via
program VOLUME?'//
    +' (yes or no) ',$)
83    format(////
    +' Do you wish to obtain volume information from the
finite'/
    +' element generated mesh for the',i3,' frames? ',$)
209   format(////
    +' Do you wish to view any results up to now
graphically?'//
    +' (NOTE: Smoothing through routines are included)
',$)
217   format(////
    +' Do you wish to produce a datafile for subsequent
FE Analysis'/
    +' for this patient? '$)
213   format(/' Frame number? ',$)

      return
      end


      subroutine
readfil(filenam,patr,namer,comtr,nf,jxrn,

+jyrn,m,jxln,jyln,mm,jxrp,jyrp,mmm,jxlp,jylp,mmmm,preses)
c
c     It is called by part1 venvol femvol and afemvol
c
c     It read the data from a patient datafile
c
c     It makes no subroutine calls
c
      integer ixrn(151,151),iyrn(151,151),ixln(151,151),

+iyln(151,151),ixrp(151,151),iyrp(151,151),ixlp(151,151),
     +iylp(151,151),nrn(151),nln(151),nrp(151),nlp(151),
     +jxrn(151,151),mmm(151),mmmm(151),

+jyrn(151,151),jxln(151,151),jyln(151,151),jxrp(151,151),

+jyrp(151,151),jxlp(151,151),jylp(151,151),m(151),mm(151)
      real preses(151)
      integer nf,l,j,i,k,nfr,nfl
      character filenam*80,patr*6,namer*20,
     +comtr*20,cpress*10


C-----Patient data read in here for first time.

open(7,file=filenam,access='sequential',form='formatted')
      read(7,2) patr,namer,comtr,nf,nfr,nfl
2     format(a6,1x,2(a20,1x),3i6)

C-----Rao endo co-ordinates
      do 3 l=1,nfr
        read(7,*) nrn(l)
3       read(7,*) (ixrn(l,j),iyrn(l,j),j=1,nrn(l))

C-----Lao endo co-ordinates
      do 4 l=1,nfl
        read(7,*) nln(l)
4       read(7,*) (ixln(l,j),iyln(l,j),j=1,nln(l))

C-----Lao epi co-ordintes
      do 5 l=1,nfr
        read(7,*) nrp(l)
5       read(7,*) (ixrp(l,j),iyrp(l,j),j=1,nrp(l))

C-----Rao epi co-ordinates
      do 6 l=1,nfl
        read(7,*) nlp(l)
6       read(7,*) (ixlp(l,j),iylp(l,j),j=1,nlp(l))

C-----Pressures read in
      read(7,'(a10)') cpress
      read(7,*) (preses(i),i=1,nf)
      close(7)

C-----Finding and removing duplicate readings
      do 184 i=1,nf
        k=0
        do 185 j=1,nrn(i)

if(ixrn(i,j).eq.ixrn(i,j+1).and.iyrn(i,j).eq.iyrn(i,j+1))
     +goto 185
          k=k+1
          jxrn(i,k)=ixrn(i,j)
          jyrn(i,k)=iyrn(i,j)
185     continue
```

```
        m(i)=k
184   continue

      do 187 i=1,nf
        k=0
        do 188 j=1,nln(i)

if(ixln(i,j).eq.ixln(i,j+1).and.iyln(i,j).eq.iyln i,j+1 )
     +goto 188
          k=k+1
          jxln(i,k)=ixln(i,j)
          jyln(i,k)=iyln(i,j)
188     continue
        mm(i)=k
187   continue

      do 190 i=1,nf
        k=0
        do 191 j=1,nrp(i)

if(ixrp(i,j).eq.ixrp(i,j+1).and.iyrp(i,j).eq.iyrp(i,j+1))
     +goto 191
          k=k+1
          jxrp(i,k)=ixrp(i,j)
          jyrp(i,k)=iyrp(i,j)
191     continue
        mmm(i)=k
190   continue

      do 193 i=1,nf
        k=0
        do 194 j=1,nlp(i)

if(ixlp(i,j).eq.ixlp(i,j+1).and.iylp(i,j).eq.iylp(i,j+1 )
     +goto 194
          k=k+1
          jxlp(i,k)=ixlp(i,j)
          jylp(i,k)=iylp(i,j)
194     continue
        mmmm(i)=k
193   continue

      return
      end


      subroutine rotclk(n,rix,riy,rcos,rsin)
      integer i,n
      double precision xd,rcos,rsin,rix(n),riy(n)
C-----Rotates clockwise

      do 10 i=1,n
        xd=rix(i)*rcos-riy(i)*rsin
        riy(i)=riy(i)*rcos+rix(i)*rsin
10      rix(i)=xd

      return
      end


      subroutine
slienv(patr,namer,comtr,env,ax,ay,az,ijkl,ifs,tla,
     +vol,n1,n2,n3,n4)
      double precision
env(50,4,3),xxx,bottx,vol,slen,aly,tenv(4,2),vy

+,vz,tmy(9),tmz(9),uy,uz,proy(40),proz(40),area,tla(3,3),
ax,ay,az
      integer ifss,ifs,i,ik,ijkl,n1,n2,n3,n4
      character*20 patr*6,namer,comtr

C-----Calculates the co-ordinates for the 3d shape
C-----Find lowest point of top ends
      ifss=1
      xxx=env(n1,1,1)

      if(env(n2,2,1).lt.xxx) xxx=env(n2,2,1)
      if(env(n3,3,1).lt.xxx) xxx=env(n3,3,1)
      if(env(n4,4,1).lt.xxx) xxx=env(n4,4,1)
      xxx=xxx-.00001

C-----Calculate constants
      bottx=-1234.
      vol=0.

      do 5002 i=1,4
5002    if(env(1,i,1).gt.bottx) bottx=env(1,i,1)

C-----Calc slice thickness
      slen=((xxx)-(bottx+.00002))/49.
      if(ifs.eq.1) call
title(patr,namer,comtr,ijkl,ax,ay,az)

C-----Loop for each slice
      do 10 i=1,50
        aly=(xxx)-((i-1)*slen)
        if(i.eq.50) aly=.00001+bottx

C-----Endocardium
        call boxsiz(env,tenv,aly)
        call drwarc(2,1,3,tenv,vy,vz,tmy,tmz,uy,uz)
        proy(1)=vy
        proz(1)=vz
        do 3000 ik=1,9
          proy(ik+1)=tmy(ik)
3000      proz(ik+1)=tmz(ik)
        proy(11)=uy
      proz(11)=uz
      call drwarc(1,3,4,tenv,vy,vz,tmy,tmz,uy,uz)
      do 3001 ik=1,9
        proy(ik+11)=tmy(ik)
3001    proz(ik+11)=tmz(ik)
      proy(21)=uy
      proz(21)=uz
      call drwarc(3,4,2,tenv,vy,vz,tmy,tmz,uy,uz)
      do 3002 ik=1,9
        proy(ik+21)=tmy(ik)
3002    proz(ik+21)=tmz(ik)
      proy(31)=uy
      proz(31)=uz
      call drwarc(4,2,1,tenv,vy,vz,tmy,tmz,uy,uz)
      do 3003 ik=1,9
        proy(ik+31)=tmy(ik)
3003    proz(ik+31)=tmz(ik)
      call larea(40,proy,proz,area)
      if(ifs.eq.0) goto 7777
      if(ifss.eq.1) call draws(40,proy,proz,aly,tla)
7777  continue

      vol=vol+area*slen
10    continue

      vol=vol*1000000.

      return
      end


      subroutine movshp(nn,rixn,riyn)
      integer i,mapex,nn
      double precision
zapex,raox,raoy,dx,dy,dd,rixn(151),riyn(151)

C-----Calculates position of apex as being furthest point
from mid
C     aortic root and moves origin to apex in (x) and
lowest point in
C     (y) (lowest point not always the apex) Last changed
19-04-94

      zapex=0.

C-----Calc mid aortic root
      raox=(rixn(1)+rixn(nn))/2.
      raoy=(riyn(1)+riyn(nn))/2.

C-----Find apex
      do 20 i=1,nn
        dx=rixn(i)-raox
        dy=riyn(i)-raoy
        dd=dx*dx+dy*dy
        if(dd.gt.zapex) then
          zapex=dd
          mapex=i
        endif
20    continue

C-----Re-define coords
      dx=rixn(mapex)

      dy=riyn(1)
      do 30 i=2,nn
        if(riyn(i).lt.dy) dy=riyn(i)
30    continue

C-----Endocardium
      do 40 i=1,nn
        rixn(i)=rixn(i)-dx
40      riyn(i)=riyn(i)-dy

      return
      end


      subroutine rotenv(x,y,z,env)
C-----Last changed 30-01-78
      double precision
t1(3,3),t2(3,3),envt(200,3),enva(200,3),t3(3,3)
     +,x,env(50,4,3),cos1,sin1,cos2,sin2,hyp1,hyp2,y,z
      integer i,j,loop,k

      do 10 i=1,3
        do 10 j=1,3
          t1(i,j)=0.
          t2(i,j)=0.
10        t3(i,j)=0.

C-----Find angles
      hyp1=dsqrt(z*z+y*y)
      cos1=y/hyp1
      sin1=z/hyp1
      hyp2=dsqrt(x*x+hyp1*hyp1)
      cos2=x/hyp2
      sin2=hyp1/hyp2

C-----Fill transform matrices
      t1(1,1)=1.
      t1(2,2)=cos1
      t1(3,3)=cos1
      t1(3,2)=sin1
      t1(2,3)=sin1*(-1.)
      t2(1,1)=cos2
      t2(1,2)=sin2*(-1./
      t2(2,1)=sin2
      t2(2,2)=cos2
      t2(3,3)=1.

C-----Find overall transform
      call mmlt(t3,t1,t2,3,3,3

C-----Transform envelope
      do 20 i=1,50
        do 20 j=1,4
          do 20 k=1,3
            loop=((i*4)-4)+j
20          envt(loop,k)=env(i,j,k

      call mmlt(enva,envt,t3,200,3,3)

      do 30 i=1,50
        do 30 j=1,4
          do 30 k=1,3
            loop=((i*4)-4)+j
30          env(i,j,k)=enva(loop,k

      return
      end


      subroutine boxsiz(env,tenv,aly)
C-----Finds corner points of box for horizontal slice
C-----Last changed on 09-03-78
      double precision
y1,y2,y3,y4,z1,z2,z3,z4,aly,env(50,4,3 ,
     +tenv(4,2)
      integer i

      do 1000 i=1,50
1000    if(aly.le.env(i,1,1 ) goto 1510

1510  z1=(((aly-env(i-1,1,1))/(env(i,1,1)-env(i-
1,1,1)))*(env(i,1,3)-
     + env(i-1,1,3)))+env(i-1,1,3)
      y1=(((aly-env(i-1,1,1)   env(i,1,1)-env(i-
1,1,1)))*(env(i,1,2)-
     + env(i-1,1,2)))+env(i-1,1,2)

      do 2000 i=1,50
2000    if(aly.le.env(i,2,1 ) goto 2010

2010  z2=(((aly-env(i-1,2,1))/(env(i,2,1)-env(i-
1,2,1)))*(env(i,2,3)-
     + env(i-1,2,3)))+env(i-1,2,3)
      y2=(((aly-env(i-1,2,1)) (env(i,2,1)-env(i-
1,2,1)))*(env(i,2,2)-
     + env(i-1,2,2)))+env(i-1,2,2)

      do 3000 i=1,50
3000    if(aly.le.env(i,3,1)) goto 3010

3010  y3=(((aly-env(i-1,3,1 )/(env(i,3,1)-env(i-
1,3,1)))*(env(i,3,2)-
     + env(i-1,3,2)))+env(i-1,3,2)
      z3=(((aly-env(i-1,3,1))/(env(i,3,1)-env(i-
1,3,1)))*(env(i,3,3)-
     + env(i-1,3,3)))+env(i-1,3,3)

      do 4000 i=1,50
4000    if(aly.le.env(i,4,1)) goto 4010

4010  y4=(((aly-env(i-1,4,1))/(env(i,4,1)-env(i-
1,4,1)))*(env(i,4,2)-
     + env(i-1,4,2)))+env(i-1,4,2)
      z4=(((aly-env(i-1,4,1))/(env(i,4,1)-env(i-
1,4,1)))*(env(i,4,3)-
     + env(i-1,4,3)))+env(i-1,4,3)

      tenv(1,1)=y1
      tenv(1,2 =z1
      tenv 2,1)=y2
      tenv 2,2)=z2
      tenv(3,1 =y3
      tenv 3,2)=z3
      tenv(4,1)=y4
      tenv 4,2)=z4

      return
      end


      subroutine
drwarc(jj,kk,ll,tenv,vy,vz,tmy,tmz,uy,uz)
C-----Fills in envelope boxes with an arc in each quarter
      double precision
vy,vz,uy,uz,thete,tenv 4,2 ,tmy(9),tmz 9),theta

+,tant,sint,rm,c,wy,wz,rn,d,ro,e,yy,xy,zy,yz,xz,zz,lwz,lw
v,cost
      integer jj,kk,ll,i

C-----Find points u and v
      uy= tenv(kk,1)+tenv ll,1  /2.
      uz= tenv kk,2)+tenv ll,2   2.
      vy= tenv kk,1 +tenv jj,1)  2.
      vz= tenv kk,2 +tenv jj,2 ) 2.

C-----Find eq'n u-v
      rm= vz-uz   vy-uy

C-----Change for 90 deg.
      rm= 1. rm)* -1.
C-----Find mid chord w
      wy=(uy+vy  2.
      wz=(uz+vz  2.

C-----Find eq'n constant
      c=wz-rm*wy

C-----Find line perp to 1-2 and v
      if temv(jj,1 .eq.temv kk,1)) goto 1000
      if temv jj,2).eq.temv kk,2))) goto 1000

      rn=(temv jj,2 -temv(kk,2)))/ temv jj,1 -temv kk,1 )
      rn=(1. rn))*(-1.)
      d=vz-rn*vy
1000  continue

C-----Find line perp to 1-3 and u
      if temv(kk,1 .eq.temv ll,1))) goto 2000
      if temv((kk,2 .eq.temv(ll,2))) goto 2000

      ro=((temv(kk,2)-temv ll,2 ))/((temv kk,1 -temv ll,1 )
      ro=((1./ro)*(-1.))
      e=uz-ro*uy
2000  continue

C-----Find points of equality
      if((temv(kk,1) .eq.temv(jj,1 )) yy=(vz-c /rm
      if temv kk,2).eq.temv((jj,2 ) yy=vy

if(temv kk,1)).ne.temv(jj,1)).and.temv((kk,2)).ne.temv((jj,2)))
     +yy= d-c)/(rn-rm))
      if temv kk,1 .eq.temv((ll,1 )) xy=(uz-c)//rm
      if temv kk,2)).eq.temv ll,2))) xy=uy

if((temv kk,1)).ne.temv((ll,1)).and.temv kk,2)).ne.temv((ll,2)))

     +xy=(e-c)//((rm-ro))
      zy=(yy+xy))//2.
      yz=rm*yy+c
      xz=rm*xy+c
      zz= yz+xz) 2.

C-----Find half angle subtended
      lwz=dsqrt(((zz-wz))*((zz-wz)+((zy-wy))*(zy-wy )
      lwv=dsqrt(((vz-wz *((vz-wz + vy-wy * vy-wy)))
      tant=lwv//lwz
      theta=.2*((datan((tant )

      do 3000 i=1,9
        thete=theta*i
        sint=dsin thete
        cost=dcos thete)
C-----Calc points on arc for data
        tmy i)=( vy-zy *cost+ vz-zz *sint +zy
        tmz(i)=( vz-zz)*cost- vy-zy *sint +zz
3000  continue

      return
      end


      subroutine larea n,x,y,area)
C-----Loop integral from David to calculate loop integral
of
C     co-ordinate pairs  x,y), each of length 'n'
      double precision x n),y(n),a,x1,y1,x2,y2,area
      integer i,m

      a=0.
      x1=x n
      y1=y n

      do 20 i=1,n
        x2=x i)
        y2=y(i)
        a=a+ x2-x1)*(y2+y1
        x1=x2
20      y1=y2

      area=.5*dabs(a)

      return
      end


      subroutine draws ii,ymat,zmat,aly,t1)
C-----Subroutine for drawing 3-d ventricle shape
      double precision
t1(3,3),zmat 40 ,ymat(40 ,cor 40,3 ,cop 40,3),
     +aly,axx
      real x,y
      integer ii,ix,iy,i

      axx=aly*6000.

      do 100 i=1,ii
        cop(i,1)=axx
        cop(i,3)=(zmat(i)*6000.
100     cop(i,2)=(ymat(i)*6000.

      call mmlt(cor,cop,t1,40,3,3)

C-----Control of picture position on screen or paper done
here
      call break
      do 200 i=1,ii
```

```
        ix=idint cor i,2  +500)
        x=float ix)
        iy=idint cor i,1)  +100)
        y=float iy
        if i.ne.11  call plot(x,y,2)
200     if i.eq.11  call plot(x,y,3)

      return
      end


      subroutine mmlt(matp,matl,matr,m1,m2,m3)
C-----Matrix multiplication
      double precision
matp m1,m3 ,matl((m1,m2 ,matr(m2,m3
      integer m1,m2,m3,i,j,k

      do 20 i=1,m1
        do 20 j=1,m3
          matp i,j =0.
          do 20 k=1,m2
20          matp i,j)=matp i,j +matl i,k) *matr(k,j)

      return
      end


      subroutine title patr,namer,contr,num,ax,ay,az)
C-----Puts caption in left hand side corner of 3D
illustration
      double precision ax,ay,az
      integer num
      character*20 patr*6,namer,contr,cnum*3

      call knumb num,cnum)
      call axis ax,ay,az)
      call cp7pt((5.,870.,16,patr)
      call cp7pt 5.,840.,16,namer))
      call cp7pt 5.,810.,16,contr)
      call cp7pt 5.,780.,16,"FRAME NUMBER"))
      call cp7pt 260.,780.,16,cnum)

      return
      end


      subroutine axis ax,ay,az))
C-----Puts captions and angles in on 3D illustration
      double precision ax,ay,az
      real axa,aya,aza
      character*30 caxa,caya,caza

      axa=sngl ax*180./3.14159))
      aya=sngl ay*180. 3.14159
      aza=sngl az*180. 3.14159
      call kreal axa,caxa
      call kreal aya,caya)
      call kreal(aza,caza)

      call plot(25.,300.,3)
      call plot(100.,300.,2
      call cp7pt(105.,300.,16,"Y "
      call plot 25.,300.,3
      call plot 25.,375.,2
      call cp7pt 30.,375.,16,"X ")
      call plot 25.,300.,3
      call plot 65.,340.,2)
      call cp7pt 70.,340.,16,"Z ")
      call plot(25.,230.,3)
      call plot(100.,230.,2)
      call cp7pt(25.,200.,16,"1 CM")
      call cp7pt(0.,130.,16,"ROTATION")
      call cp7pt(0.,100.,16,"ABOUT    DEGREES"
      call cp7pt(25.,70.,16,"X ")
      call cp7pt 125.,70.,16,caxa)
      call cp7pt 25.,40.,16,"Y ")
      call cp7pt(125.,40.,16,caya)
      call cp7pt 25.,10.,16,"Z ")
      call cp7pt(125.,10.,16,caza)

      return
      end


      subroutine chkpnt ix,iy m
C-----Checks for coincident points
      integer ix 151 ,iy 151 ,m,n1,n,off

      off=0
      do 100 n=1,m-1
        n1=n+1
        if ix n).eq.ix n1 .and.iy n).eq.iy n1   off=off+1
        ix(n1-off =ix n1
        iy n1-off)=iy n1
100   continue

      m=m-off

      return
      end


      subroutine plot x,y,ipen
C-----Pen plots if ipen=2 or 4: not when 3 or 5
      real x,y
      integer ipen
      if(ipen.eq.2.or.ipen.eq.4) then
        call brknpt(x,y,0)
      else if(ipen.eq.3.or.ipen.eq.5) then
        call break
        call brknpt(x,y,0)
      end if

      return
      end


      subroutine venvol(volu)
C-----Constructs co-ordinates for 3D visualisation and/or
volume
      integer
nrnr(151),nlnl(151),nrpr(151),nrn,nln,nlpl(151),

+ixrn(151,151),iyrn(151,151),ixln(151,151),iyln(151,151),

+ixrp(151,151),iyrp(151,151),ixlp(151,151),iylp(151,151),
     +nf,jyrn(151),jxln(151),

+jyln(151),jxrn(151),ijkl,ifs,ifrmno,i,j,lowye,lowze,k,ij
,n1,n2,
     +n3,n4
      double precision
rxrn(151),ryrn(151),rxln(151),ryln(151),

+temx(200),temz(200),t1a(3,3),t2a(3,3),t3a(3,3),t4a(3,3),

+env(50,4,3),anx,any,anz,ax,ay,az,fact,xaom,yaom,hyp,
     +rcosr,rsinr,ht,high,yelow,plowl,zelow,plow,slice,

+widthe,x,y,z,x2,axc,axs,ayc,ays,azc,azs,vol,rcosl,rsinl
      real volu(151),preses(151)
      character
answer,patr*6,namer*20,comtr*20,filenam*80
     +,cframe*2
      common/person/ filenam,cframe

      call readfil(filenam,patr,namer,comtr,nf,ixrn,iyrn,

+nrnr,ixln,iyln,nlnl,ixrp,iyrp,nrpr,ixlp,iylp,nlpl,preses
)

9     write(6,8)
8     format(////
     +' Do you wish to by-pass complete viewing of the
ventricle at'/
     +' selected frames'//
     +' i.e. only requiring volume info.instead? ',$)
      read(*,*) answer

      if(answer.eq.'Y'.or.answer.eq.'y') then
        ifs=0
      elseif(answer.eq.'N'.or.answer.eq.'n') then
        ifs=1
      else
        goto 9
      endif

10    if(ifs.eq.0) then
        anx=0.
        any=0.
        anz=0.
        ifrmno=0
      elseif(ifs.eq.1) then
        write(6,14)
14      format(////' Frame number for viewing? (type 0 to
exit) ',$)
        read(*,*) ifrmno
        if(ifrmno.eq.0) goto 80
        write(6,60)
60    format(////
     +' Input the 3 viewing angles for the X,Y and Z axes
resp.
     + (in degs. from 0 to 360):')
        write(6,61)
61    format(' ax= ',$)
        read(*,*) anx
        write(6,76)
76    format(' ay= ',$)
        read(*,*) any
        write(6,77)
77    format(' az= ',$)
        read(*,*) anz

          call initsp
        call page(20.8,29.5)
        call picsiz(19.,28.)
        call newpic
        call eqscal(0.0,900.0,0.0,900.0,0,0)

      endif

      do 9001 ijkl=1,nf
        if((ijkl.eq.ifrmno).or.(ifs.eq.0)) then
          fact=3.141592653/180.
          ax=anx*fact
          ay=any*fact
          az=anz*fact

C-----Full 3d transform
          axc=dcos(ax)
          axs=dsin(ax)
          ayc=dcos(ay)
          ays=dsin(ay)
          azc=dcos(az)
          azs=dsin(az)

          t1a(1,1)=1.
          t1a(1,2)=0.
          t1a(1,3)=0.
          t1a(2,1)=0.
          t1a(2,2)=axc
          t1a(2,3)=axs*(-1.)
          t1a(3,1)=0.
          t1a(3,2)=axs
          t1a(3,3)=axc

          t2a(1,1)=ayc
          t2a(1,2)=0.
          t2a(1,3)=ays
          t2a(2,1)=0.
          t2a(2,2)=1.
          t2a(2,3)=0.
          t2a(3,1)=ays*(-1.)
          t2a(3,2)=0.
          t2a(3,3)=ayc

          t3a(1,1)=azc
          t3a(1,2)=azs*(-1.)
          t3a(1,3)=0.
          t3a(2,1)=azs
          t3a(2,2)=azc
          t3a(2,3)=0.
          t3a(3,1)=0.
          t3a(3,2)=0.
          t3a(3,3)=1.

          call mmlt(t4a,t1a,t2a,3,3,3)
          call mmlt(t1a,t4a,t3a,3,3,3)

C-----Data read in sequentially
          nrn=nrnr(ijkl)
          nln=nlnl(ijkl)
          do 15 i=1,nrn
            jxrn(i)=ixrn(ijkl,i)
15          jyrn(i)=iyrn(ijkl,i)
          do 16 i=1,nln
            jxln(i)=ixln(ijkl,i)
16          jyln(i)=iyln(ijkl,i)

          call chkpnt(jxrn,jyrn,nrn)
          call chkpnt(jxln,jyln,nln)

C-----Float integer values to metres in double precision
          do 7000 i=1,100
            rxrn(i)=dble(jxrn(i))*(.0001)
            ryrn(i)=dble(jyrn(i))*(.0001)
            rxln(i)=dble(jxln(i))*(.0001)
7000        ryln(i)=dble(jyln(i))*(.0001)


C-----Move shapes to convenient origin at apex (defined
as being
C     farthest from mid-aortic root) for x and then
lowest point on
C     endocardium for y
          call movshp(nrn,rxrn,ryrn)
          call movshp(nln,rxln,ryln)

C-----Move Rao to ensure we can always fit a box. this
effects the
c     result by less than 0.01%
          do 1001 i=1,nrn
            if(ryrn(i).lt.0.0001D00) ryrn(i)=0.0001D00
1001      continue

C-----Moves shape up for kink removal
          xaom=(rxln(1)+rxln(nln))/2.
          yaom=(ryln(1)+ryln(nln))/2.
          hyp=dsqrt(xaom*xaom+yaom*yaom)
          rcosl=yaom/hyp
          rsinl=xaom/hyp
          call rotclk(nln,rxln,ryln,rcosl,rsinl)
          call noloop(nln,rxln,ryln)
          rsinl=-rsinl
          call rotclk(nln,rxln,ryln,rcosl,rsinl)

          xaom=(rxrn(1)+rxrn(nrn))/2.
          yaom=(ryrn(1)+ryrn(nrn )/2.
          hyp=dsqrt(xaom*xaom+yaom*yaom)
          rcosr=yaom/hyp
          rsinr=xaom/hyp
          call rotclk(nrn,rxrn,ryrn,rcosr,rsinr)
          call noloop(nrn,rxrn,ryrn)
          rsinr=-rsinr
          call rotclk(nrn,rxrn,ryrn,rcosr,rsinr)
          rsinr=-rsinr

C-----Draw aorta on lao if low compared with rao
          ht=ryrn(1)
          do 515 i=2,nrn
            if(ryrn(i).gt.ht) ht=ryrn(i)
515       continue

          if(ryln(nln).lt.ht) then
            nln=nln+1
            ryln(nln)=ht
            rxln(nln)=rxln(nln-1)
          endif

          if(ryln(1).lt.ht) then
```

```
         do 1006 i=1,nln
           j=nln+1-i
           rxln j+1)=rxln j
1006       ryln(j+1)=ryln j)
         rxln 1 =rxln 1
         ryln 1)=ht
         nln=nln+1
       endif

       call rotclk(nrn,rxrn,ryrn,rcosr,rsinr)

C-----Reverse sign x rao
       do 1511 i=1,nrn
1511     rxrn(i =rxrn(i)*(-1.)

c------find highest end point on endocardium
       high=ryrn(1)
       if high.lt.ryrn(nrn)) high=ryrn(nrn)
       if high.lt.ryln(1)) high=ryln(1)
       if(high.lt.ryln(nln)) high=ryln(nln)

C-----Find lowest point on the endocardinm
       yelow=0.
       plowl=12345678.
       lowye=0

       do 1510 i=1,nln
         if(ryln(i).le.plowl) then
           yelow=rxln(i)
           plowl=ryln(i)
           lowye=i
         endif
1510   continue

       zelow=0.
       plow=12345678.
       lowze=0

       do 1020 i=1,nrn
         if(ryrn(i).le.plow) then
           zelow=rxrn(i
           plow=ryrn(i)
           lowze=i
         endif
1020   continue

       if(plowl.gt.plow) plow=plowl

C-----Slice model horizontally at fifty levels to produce
envelope
C-----where nothing to slice place 0.
C-----First endocardium
       widthe=(high-plow)/50.
       slice=plow

     do 1040 i=1,50
       slice=slice+widthe
       do 1090 j=1,lowze
         if(slice.gt.ryrn 1)) then
           env(i,1,3)=0.
           goto 1112
         endif
         if(slice.gt.ryrn(j)) then
           env(i,1,3)=rxrn(j)+(slice-ryrn(j))/(ryrn(j-
1)-
    +ryrn(j))*(rxrn(j-1)-rxrn(j))
           goto 1112
         endif
1090   continue

1112   env(i,2,3)=env(i,1,3)

       do 1120 j=lowze,nrn
         k=nrn-j+lowze
         if(slice.gt.ryrn(nrn)) then
           env(i,3,3)=0.
           goto 1132
         endif
         if(slice.gt.ryrn(k)) then
           env(i,3,3)=rxrn(k)+(slice-
ryrn(k))/(ryrn(k+1)-
    +ryrn(k))*(rxrn(k+1)-rxrn(k))
           goto 1132
         endif
1120   continue

1132   env(i,4,3)=env(i,3,3)

       do 1040 j=1,4
         ij=((i-1)*4)+j
         temx(ij)=slice
         temz(ij)=env(i,j,3)
         env(i,j,1)=slice
1040   continue

C-----Now rotate rao points back and use x to slice lao
       call rotclk(200,temz,temx,rcosr,rsinr)

       do 1042 i=1,50
         do 1042 j=1,4
           ij=((i-1)*4)+j
           if(env(i,j,3).eq.0.) temz(ij)=0.
           env(i,j,3)=temz(ij)
1042       env(i,j,1)=temx(ij)

       do 1041 i=1,50
         do 1050 j=1,lowye
```

```
           if(env i,2,1 .gt.ryln 1   then
             env i,2,2 =0.
             goto 1062
           endif
             if env(i,2,1 .gt.ryln j ) then
               env i,2,2 =rxln(j)+ env i,,2,1 -
ryln(j))/(ryln(j-1)
     +-ryln(j))*(rxln j-1 -rxln(j))
             goto 1062
           endif
1050     continue
1062     do 1051 j=1,lowye
           if(env i,4,1 .gt.ryln 1 )) then
             env(i,4,2 =0.
             goto 1065
           endif
           if(env(i,4,1 .gt.ryln(j ) then
             env(i,4,2 =rxln(j) + env((i,,4,,1)-
ryln(j))/(ryln(j-1)-
     +ryln(j))* rxln j-1 -rxln(j))
             goto 1065
           endif
1051     continue
1065     do 1070 j=lowye,nln
           k=nln-j+lowye
           if(env(i,1,1 .gt.ryln nln)) then
             env(i,1,2)=0.
             goto 1082
           endif
           if(env(i,1,1).gt.ryln(k)   then
             env i,1,2 =rxln k)+ env i,,1,,1)-
ryln(k))/(ryln(k+1)
     +-ryln(k))*(rxln(k+1)-rxln(k))
             goto 1082
           endif
1070     continue
1082     do 1071 j=lowye,nln
           k=nln-j+lowye
           if env(i,3,1).gt.ryln(nln)) then
             env(i,3,2)=0.
             goto 1041
           endif
           if(env(i,3,1).gt.ryln(k)) then
             env(i,3,2)=rxln(k)+(env(i,3,,1)-
ryln(k))/(ryln(k+1)
     +-ryln(k))*(rxln(k+1)-rxln(k )
             goto 1041
           endif
1071       continue
1041     continue

C-----Rotate actual rao back
         call rotclk(nrn,rxrn,ryrn,rcosr,rsinr)

C-----Find lowest xyz for end points
         do 3060 i=1,50
           j=50-i+1
3060
if(dabs(env(j,1,3)).gt.0..and.dabs env j,1,2)).gt.0.)
     +goto 3070
         goto 2999
3070     n1=j
         do 3080 i=1,50
           j=50-i+1
3080
if(dabs(env(j,2,3)).gt.0..and.dabs(env(j,2,2)).gt.0.)
     +goto 3090
         goto 2999
3090     n2=j
         do 4000 i=1,50
           j=50-i+1
4000
if(dabs(env(j,3,3)).gt.0..and.dabs(env(j,3,2)).gt.0.)
     +goto 4010
         goto 2999
4010     n3=j
         do 4020 i=1,50
           j=50-i+1
4020
if(dabs(env(j,4,3)).gt.0..and.dabs(env(j,4,2)).gt.0.)
     +goto 4030
         goto 2999
4030     n4=j

C----- Rotate envelope in three dimensions so that mid-
aortic line
C----- is vertical
         x=(ryln(1)+ryln(nln))/2.
         y=(rxln(1)+rxln nln))/2.
         z=(rxrn(1)+rxrn(nrn))/2.
         x2=(ryrn(1)+ryrn(nrn))/2.
         z=z*(x/x2)
         call rotenv(x,y,z,env)

C-----Slice remaining ventricle horizontally
         call
slienv(patr,namer,comtr,env,ax,ay,az,ijkl,ifs,tla,vol,
     +n1,n2,n3,n4)
         volu(ijkl)=sngl(vol)
         write 6,74  ijkl,volu(ijkl)
74       format(' Volume frame',i2,' ='f6.2,'cm3')

C-----simpleplot call to terminate plotting
         if(ifs.eq.1) call endplt
         endif
9001  continue
```

```
      if(ifs.eq.1) goto 10
80    goto 75
2999  print*,' Run stopped--end of envelope not found'
      stop

75    print 100
100   format(/' Would you like to re-ruń segment VOLUME?
',$)
      read(*,*) answer
      if(answer.eq.'Y'.or.answer.eq.'y') then
        goto 9
      elseif(answer.ne.'N'.and.answer.ne.'n') then
        goto 75
      endif

151   return
      end


      subroutine femvol(volum,ifrmno,nlm,iresp)
C-----Construction of finite element from the digitised
co-ordinates
      integer
jxrn(151),jyrn(151),jxln(151),jyln(151),jxrp(151),
     +jyrp(151),jxlp(151),jylp(151),ixrn(151,151),

+iyrn(151,151),ixrp(151,151),iyrp(151,151),ixln(151,151),
     +iyln(151,151),ixlp(151,151),iylp(151,151),
     +nrnr(151),nrpr(151),nlnl(151),nlpl(151),lowy,lowz,

+lowye,lowze,n1,n2,n3,n4,n1p,n2p,n3p,n4p,nlm,nf,ifrmno,ij
kl,ij,
     +i,nrn,nrp,nln,nlp,j,k,iresp
      double precision
onod(255,3),x2,plow,xaom,yaom,hyp,rsinl,rcosl,

+rsinr,rcosr,val,temx(200),temz(200),rxrn(151),ryrn(151),

+rxrp(151),ryrp(151),rxln(151),ryln(151),rxlp(151),rylp(1
51),

+ylow,zlow,width,widthe,high,slice,env(50,4,3),envp(50,4,
3),

+geom(3,255),yelow,zelow,x,y,z,geoma(3,255),t3(3,3),ht,
     +plowl,th1,th2
      real preses(151),volum(151)
      character
patr*6,namer*20,comtr*20,filenam*80,cframe*2
      common/person/ filenam,cframe

      call readfil(filenam,patr,namer,comtr,nf,ixrn,iyrn,

+nrnr,ixln,iyln,nlnl,ixrp,iyrp,nrpr,ixlp,iylp,nlpl,preses
)

      if(ifrmno.ne.0) nf=ifrmno

      do 9002 ijkl=1,nf
        if(ifrmno.ne.0.and.ijkl.ne.nf) goto 9002
        write(6,99) ijkl
99      format(//' Frame no. ',I2)

        nrn=nrnr(ijkl)
        nrp=nrpr(ijkl)
        nln=nlnl(ijkl)
        nlp=nlpl(ijkl)
        do 86 i=1,151
          jxrn(i)=ixrn(ijkl,i)
86        jyrn(i)=iyrn(ijkl,i)
        do 87 i=1,151
          jxrp(i)=ixrp(ijkl,i)
87        jyrp(i)=iyrp(ijkl,i)
        do 88 i=1,151
          jxln(i)=ixln(ijkl,i)
88        jyln(i)=iyln(ijkl,i)
        do 89 i=1,151
          jxlp(i)=ixlp(ijkl,i)
89        jylp(i)=iylp(ijkl,i)

          call chkpnt(jxrn,jyrn,nrn)
          call chkpnt(jxln,jyln,nln)
          call chkpnt(jxrp,jyrp,nrp)
          call chkpnt(jxlp,jylp,nlp)

C-----Float integer values to metres
        do 7000 i=1,151
          rxrn(i)=dble(jxrn(i))*(.0001)
          ryrn(i)=dble(jyrn(i))*(.0001)
          rxrp(i)=dble(jxrp(i)}*(.0001)
          ryrp(i)=dble(jyrp(i))*(.0001)
          rxln(i)=dble(jxln(i))*(.0001)
          ryln(i)=dble(jyln(i))*(.0001)
          rxlp(i)=dble(jxlp(i))*(.0001)
7000      rylp(i)=dble(jylp(i))*(.0001)

C-----Move shapes to convenient origin at apex (defined
as being
C     farthest from mid-aortic root) for x and then
lowest point on
C     endocardium for y
        call moveshp(nrn,rxrn,ryrn,nrp,rxrp,ryrp)
        call moveshp(nln,rxln,ryln,nlp,rxlp,rylp)

C-----Move Rao to ensure we can always fit a box. this
has a
c     negligable effect upon the volume
```

```
      do 1001 i=1,nrn
        if(ryrn(i).lt.0.0001d0) ryrn(i)=0.0001d0
1001  continue


C-----Moves shape up for sketch
        xaom=(rxln(1)+rxln(nln))/2.
        yaom=(ryln(1)+ryln(nln))/2.
        hyp=dsqrt(xaom*xaom+yaom*yaom)
        rcosl=yaom/hyp
        rsinl=xaom/hyp
        call rotclk(nln,rxln,ryln,rcosl,rsinl)
        call rotclk(nlp,rxlp,rylp,rcosl,rsinl)
        xaom=(rxrn(1)+rxrn(nrn))/2.
        yaom=(ryrn(1)+ryrn(nrn))/2.
        hyp=dsqrt(xaom*xaom+yaom*yaom)
        rcosr=yaom/hyp
        rsinr=xaom/hyp
        call rotclk(nrn,rxrn,ryrn,rcosr,rsinr)
        call rotclk(nrp,rxrp,ryrp,rcosr,rsinr)
c     Level off base area
c       ht=ryrn(1)
c       do 5 i=2,nrn
c         if(ryrn(i).gt.ht) ht=ryrn(i)
c 5     continue
c       if(ryrn(nrn).lt.ht) then
c         nrn=nrn+1
c         ryrn(nrn)=ht
c         rxrn(nrn)=rxrn(nrn-1)
c       endif
c       if(ryrn(1).lt.ht) then
c         do 25 i=nrn,1,-1
c           rxrn(i+1)=rxrn(i)
c 25        ryrn(i+1)=ryrn(i)
c         rxrn(1)=rxrn(1)
c         ryrn(1)=ht
c         nrn=nrn+1
c        endif

        call noloop(nln,rxln,ryln)
        call noloop(nrn,rxrn,ryrn)

C-----Sketch in pericardium
        ij=0
        call
sketch(nrp,nrn,rxrp,ryrp,rxrn,ryrn,ij,0.,th1,th2)
          call
tidyup(rxrn,ryrn,rxrp,ryrp,nrn,nrp,th1,th2)
        rsinr=(-1.)*rsinr
        call rotclk(nrp,rxrp,ryrp,rcosr,rsinr)
        call rotclk(nrn,rxrn,ryrn,rcosr,rsinr)

        val=ryrp(1)
        do 1012 i=2,nrp
1012      if(ryrp(i).lt.val) val=ryrp(i)

        jj=1

        call
sketch(nlp,nln,rxlp,rylp,rxln,ryln,jj,val,th1,th2)
        rsinl=(-1.)*rsinl
        call rotclk(nln,rxln,ryln,rcosl,rsinl)
        call rotclk(nlp,rxlp,rylp,rcosl,rsinl)
          call
tidyup(rxln,ryln,rxlp,rylp,nln,nlp,th1,th2)

c         call
tao(nln,nlp,nrn,nrp,rxrn,ryrn,rxrp,ryrp,rxln,
c     +ryln,rxlp,rylp)

C-----Move Rao to ensure we can always fit a box. this
has a
c     negligable effect upon the volume
c          do 1001 i=1,nrn
c            ryrn(i)=ryrn(i)+0.001
c1001      continue
c
c     Reposition points on Rao Pericardium if they are
below
c     or close to the lowest on the Lao
c
c          do 1002 i=1,nrp
c            ryrp(i)=ryrp(i)+0.001
c1002      continue

          val=rylp(1)
          do 518 i=2,nlp
          if(rylp(i).lt.val) val=rylp(i)
518     continue
          val=val+0.0001D0
          do 519 i=1,nrp
          if(ryrp(i).lt.val) ryrp(i)=val
519     continue

c         call
tao(nln,nlp,nrn,nrp,rxrn,ryrn,rxrp,ryrp,rxln,
c     +ryln,rxlp,rylp)


c
c     check for over flow in the arrays
c

if((nrn.gt.150).or.(nln.gt.150).or.(nrp.gt.150).or.
     +(nlp.gt.150)) then
          print*,' ERROR !! array is not large enough '
          stop
        endif
```

```
C-----Draw aorta on lao if low compared with rao
      ht=ryrn(1)
      do 515 i=2,nrn
        if(ryrn(i).gt.ht) ht=ryrn i
515   continue

      if(ryln(nln).lt.ht) then
        nln=nln+1
        ryln(nln)=ht
        rxln(nln)=rxln(nln-1)
      endif

      if(ryln(1).lt.ht) then
        do 1006 i=1,nln
          j=nln+1-i
          rxln(j+1)=rxln(j)
1006      ryln(j+1)=ryln(j)
        rxln(1)=rxln(1)
        ryln(1)=ht
        nln=nln+1
      endif

      ht=ryrp(1)
      do 525 i=2,nrp
        if(ryrp(i).gt.ht) ht=ryrp(i)
525   continue

      if(rylp nlp).lt.ht) then
        nlp=nlp+1
        rylp(nlp)=ht
        rxlp(nlp)=rxlp(nlp-1)
      endif

      if(rylp(1).lt.ht) then
        do 2006 i=1,nlp
          j=nlp+1-i
          rxlp(j+1)=rxlp(j)
2006      rylp(j+1)=rylp(j)
        rxlp(1)=rxlp(1)
        rylp(1)=ht
        nlp=nlp+1
      endif

      rsinr=(-1.)*rsinr
      call rotclk(nrp,rxrp,ryrp,rcosr,rsinr)
      call rotclk(nrn,rxrn,ryrn,rcosr,rsinr)

C-----Reverse sign x rao
      do 1011 i=1,100
        rxrn i)=rxrn(i)*(-1.)
1011    rxrp(i)=rxrp(i)*(-1.)

C-----Find highest end point on endocardium
      high=ryrn(1)
      if high.lt.ryrn(nrn)) high=ryrn nrn)
      if(high.lt.ryln(1)) high=ryln(1
      if high.lt.ryln(nln)) high-ryln(nln)

C-----Find lowest point on the endocardium
      yelow=0.
      plowl=12345678.
      lowye=0
      do 1010 i=1,nln
        if ryln(i).le.plowl) then
          yelow=rxln(i)
          plowl=ryln(i)
          lowye=i
        endif
1010  continue
      zelow=0.
      plow=12345678.
      lowze=0
      do 1020 i=1,nrn
        if(ryrn(i).le.plow) then
          zelow=rxrn(i)
          plow=ryrn(i)
          lowze=i
        endif
1020  continue

      if(plowl.gt.plow) plow=plowl

c         call
tao(nln,nlp,nrn,nrp,rxrn,ryrn,rxrp,ryrp,rxln,
c     +ryln,rxlp,rylp)


C-----Slice model horizontally at fifty levels to produce
envelope .
C-----where ..othing to slice place 0.
C-----First endocardium
c
        widthe= high-plow) 50.
        slice=plow

        do 1040 i=1,50

          slice=slice+widthe
          do 1090 j=1,lowze
            if (slice.gt.ryrn(1)) then
              env(i,1,3)-0.
              goto 1112
            endif
            if  slice.gt.ryrn j ) then
              env(i,1,3)=rxrn j)+ slice-ryrn j)
     +(ryrn j-1)-ryrn(j))* rxrn j-1)-rxrn j
              goto 1112
            endif
1090      continue
1112      env i,2,3 =env i,1,3

          do 1120 j=lowze,nrn
            k=nrn-j+lowze
            if slice.gt.ryrn(nrn)) then
              env i,3,3 =0.
              goto 1132
            endif
            if slice.gt.ryrn(k)) then
              env i,3,3 =rxrn(k + slice-ryrn k
     +(ryrn(k+1 -ryrn k  * rxrn k+1 -rxrn k
              goto 1132
            endif
1120      continue
1132      env(i,4,3 =env i,3,3)

          do 1040 j=1,4
            ij=((i-1)*4 +j
            temx(ij)=slice
            temz(ij)=env(i,j,3)
1040        env(i,j,1 =slice

C-----Now rotate rao points back and use x to slice lao
        call rotclk(200,temz,temx,rcosr,rsinr)

        do 1042 i=1,50
          do 1042 j=1,4
            ij=((i-1 *4)+j
            if(env i,j,3).eq.0.) temz ij =0.
            env i,j,3 =temz ij
1042        env(i,j,1)=temx ij

        do 1041 i=1,50

          do 1050 j=1,lowye
            if(env(i,2,1).gt.ryln(1)) then
              env(i,2,2)=0.
              goto 1062
            endif
            if(env i,2,1).gt.ryln j)) then
              env(i,2,2)=rxln(j)+ env(i,2,1)-ryln(j )
     +(ryln(j-1)-ryln(j))*(rxln j-1)-rxln(j))
              goto 1062
            endif
1050      continue
1062      do 1051 j=1,lowye
            if(env i,4,1).gt.ryln(1)) then
              env(i,4,2 =0.
              goto 1065
            endif
            if(env i,4,1 .gt.ryln(j)  then
              env(i,4,2 =rxln(j)+(env(i,4,1)-ryln    /
     +(ryln(j-1)-ryln(j )* rxln(j-1)-rxln j
              goto 1065
            endif
1051      continue

1065      do 1070 j=lowye,nln
            k=nln-j+lowye
            if(env(i,1,1 .gt.ryln(nln)) then
              env(i,1,2)=0.
              goto 1082
            endif
            if (env(i,1,1).gt.ryln(k)) then
              env(i,1,2)=rxln(k)+(env(i,1,1)-ryln(k))/
     +(ryln(k+1)-ryln(k))*(rxln(k+1)-rxln k))
              goto 1082
            endif
1070      continue
1082      do 1071 j=lowye,nln
            k=nln-j+lowye
            if env i, ,1 .gt.ryln nln   then
              env ı,3,2 =0.
              goto 1041
            endif
            if(env(i,3,1).gt.ryln(k)) then
              env(i,3,2)=rxln(k)+(env(i,3,2)-ryln(k   /
     +(ryln(k+1)-ryln(k))*(rxln k+1)-rxln(k )
              goto 1041
            endif
1071      continue

1041    continue

C-----Repeat for pericardium
C-----Find highest end point on pericardium
        high=ryrp(1)
        if(high.lt.ryrp nrp)) high=ryrp(nrp)
        if(high.lt.rylp 1 ) high=rylp 1
        if(high.lt.rylp nlp   high=rylp(nlp

C-----Find lowest point on the pericardium
        ylow-0.
        plowl=12345678.
        lowy-0

        do 8010 i=1,nlp
          if(rylp ı .lt.plowl  then
            ylow-rxlp ı
            plowi=rylp ı
            lowy=i
          endif
8010    c ntınue
```

```
        zlow=0.
        plow=12345678.
        lowz=0
        do 8020 i=1,nrp
          if(ryrp(i).lt.plow) then
            zlow=rxrp(i)
            plow=ryrp(i)
            lowz=i
          endif
8020    continue

        if(plowl.gt.plow) plow=plowl

        width=(high-plow)/50.
        slice=plow

        do 2040 i=1,50
          slice=slice+width
          do 2090 j=1,lowz
            if (slice.gt.ryrp(1)) then
              envp(i,1,3)=0.
              goto 2112
            endif
            if (slice.gt.ryrp(j)) then
              envp(i,1,3)=rxrp(j)+(slice-ryrp(j))/
     +(ryrp(j-1)-ryrp(j))*(rxrp(j-1)-rxrp(j))
              goto 2112
            endif
2090      continue
2112      envp(i,2,3)=envp(i,1,3)
          do 2120 j=lowz,nrp
            k=nrp-j+lowz
            if (slice.gt.ryrp(nrp)) then
              envp(i,3,3)=0.
              goto 2132
            endif
            if(slice.gt.ryrp(k)) then
              envp(i,3,3)=rxrp(k)+(slice-ryrp(k))/
     +(ryrp(k+1)-ryrp(k))*(rxrp(k+1)-rxrp(k))
              goto 2132
            endif
2120      continue
2132      envp(i,4,3)=envp i,3,3)

          do 2040 j=1,4
            ij=((i-1)*4)+j
            temx(ij)=slice
            temz(ij)=envp i,j,3)
2040        envp(i,j,1)=slıce

C-----Now rotate rao points back and use x to slice lao
        call rotclk(200,temz,temx,rcosr,rsinr)

        do 2042 i=1,50
          do 2042 j=1,4
            ij=((i-1)*4)+j
            if(envp(i,j,3).eq.0.) temz ij)=0.
            envp(i,j,3 =temz ij)
2042        envp(i,j,1)=temx ij)

        do 2041 i=1,50

          do 2050 j=1,lowy
            if (envp(i,2,1).gt.rylp 1)  then
              envp(i,2,2)=0.
              goto 2062
            endif
            if (envp(i,2,1).gt.rylp j)) then
              envp(ı,2,2)=rxlp(j +(envp i,2,1)-rylp(j))/
     +(rylp(j-1)-rylp(j))*(rxlp(j-1)-rxlp(j))
              goto 2062
            endif
2050      continue
2062      do 2051 j=1,lowy
            if(envp(i,4,1).gt.rylp(1)) then
              envp(i,4,2)=0.
              goto 2065
            endif
            if(envp(i,4,1).gt.rylp(j)) then
              envp(i,4,2 =rxlp(j)+(envp(i,4,1)-rylp(j))/
     +(rylp(j-1)-rylp(j))*(rxlp(j-1)-rxlp(j))
              goto 2065
            endif
2051      continue

2065      do 2070 j=lowy,nlp
            k=nlp-j+lowy
            if(envp(i,1,1).gt.rylp(nlp)) then
              envp(i,1,2)=0.
              goto 2082
            endif
            if(envp(i,1,1).gt.rylp(k)) then
              envp(i,1,2)=rxlp(k)+(envp(i,1,1)-rylp(k))/
     +(rylp(k+1)-rylp(k))*(rxlp(k+1)-rxlp(k))
              goto 2082
            endif
2070      continue
2082      do 2071 j=lowy,nlp
            k=nlp-j+lowy
            if(envp(i,3,1).gt.rylp nlp)) then
              envp(i,3,2)=0.
              goto 2041
            endif
            if(envp(i,3,1).gt.rylp(k)) then
              envp(i,3,2)=rxlp(k)+(envp(i,3,1)-rylp(k))/
     +(rylp(k+1)-rylp(k))*(rxlp(k+1)-rxlp(k))
              goto 2041
            endif
2071      continue

2041    continue

C-----Rotate actual rao back
        call rotclk(nrn,rxrn,ryrn,rcosr,rsinr)
        call rotclk(nrp,rxrp,ryrp,rcosr,rsinr)

C-----Find lowest xyz for end points
        do 3060 j=50,1,-1
3060
if(dabs(env(j,1,3)).gt.0..and.dabs(env(j,1,2)).gt.0.)
     +goto 3070
        goto 2999
3070    n1=j
        do 3080 j=50,1,-1
3080
if(dabs(env(j,2,3)).gt.0..and.dabs(env(j,2,2)).gt.0.)
     +goto 3090
        goto 2999
3090    n2=j
        do 4000 j=50,1,-1
4000
if(dabs(env(j,3,3)).gt.0..and.dabs(env(j,3,2)).gt.0.)
     +goto 4010
        goto 2999
4010    n3=j
        do 4020 j=50,1,-1
4020
if(dabs(env(j,4,3)).gt.0..and.dabs(env(j,4,2)).gt.0.)
     +goto 4030
        goto 2999
4030    n4=j

c------epi
        do 4060 j=50,1,-1
4060
if(dabs(envp(j,1,3)).gt.0..and.dabs(envp(j,1,2)).gt.0.)
     +goto 4070
        goto 2999
4070    n1p=j
        do 4080 j=50,1,-1
4080
if(dabs(envp(j,2,3)).gt.0..and.dabs(envp(j,2,2)).gt.0.)
     +goto 4090
        goto 2999
4090    n2p=j
        do 5000 j=50,1,-1
5000
if(dabs(envp(j,3,3)).gt.0..and.dabs(envp(j,3,2)).gt.0.)
     +goto 5010
        goto 2999
5010    n3p=j
        do 5020 j=50,1,-1
5020
if(dabs(envp(j,4,3)).gt.0..and.dabs(envp(j,4,2)).gt.0.)
     +goto 5030
        goto 2999
5030    n4p=j

C----- Rotate envelope in three dimensions so that mid-
aortic line
C-----is vertical
        x=(ryln(1)+ryln(nln))/2.
        y=(rxln(1)+rxln(nln))/2.
        z=(rxrn(1)+rxrn(nrn))/2.
        x2=(ryrn(1)+ryrn(nrn))/2.
        z=z*(x/x2)

c       call
tpenv(env,envp,n1,n2,n3,n4,n1p,n2p,n3p,n4p)

        call rotatenv(x,y,z,env,envp,t3)

        call
arcbas(env,envp,geom,n1,n2,n3,n4,n1p,n2p,n3p,n4p)

C----- Slice remaining ventricle horizontally
        call slicenv(env,envp,geom,n1,n2,n3,n4,n1p,
     +n2p,n3p,n4p)
        call fillup(geom)

        call mmlt(geoma,t3,geom,3,3,255)

        do 6000 i=1,255
          onod(i,1)=geoma(1,i)
          onod(i,2)=geoma(2,i)
6000      onod(i,3)=geoma(3,i)

        goto 9999
2999    print*,'Run stopped-end of envelope not found'
        goto 9002
9999    continue

        if((nlm.eq.0).and.(ifrmno.ne.0)) call
checkr(geoma,patr)
        call
split(onod,volum,ijkl,ifrmno,patr,namer,comtr,iresp,nlm)

9002  continue

85    return
      end


      subroutine noloop(np,x,y)
```

```
c
c     it is called by afemvol and femvol
c
c     this subroutine removes any loops that may have
been
c     generated by the sketch routine and that the
outline
c     is monotonically increasing to the right of the
apex
c     and     "       decreasing to the left of the apex
c
c     It makes no subroutine calls
c
      double precision x(151),y(151),low,px1,py1,px2,py2,
     +qx1,qy1,qx2,qy2,dis1,dis2
      integer np,i,j,ilow,count,int,k

      low=y(1)
      do 5 i=2,np
          if(y(i).lt.low) then
            low=y(i)
            ilow=i
          endif
5     continue
c
c     Remove loops
c
c
c     for lhs (gotos used to simulate while loops)
c
      i=ilow
40    continue
        j=i+2
45      continue
          px1=x(i)
          py1=y(i)
          px2=x(i+1)
          py2=y(i+1)
          qx1=x(j)
          qy1=y(j)
          qx2=x(j+1)
          qy2=y(j+1)
          int=insect(px1,py1,px2,py2,qx1,qy1,qx2,qy2)
          if(int.eq.1) then
            np=np-j+i
            do 50 k=i+1,np
              x(k)=x(k+j-i)
              y(k)=y(k+j-i)
50          continue
          endif
        j=j+1
          if(j.le.np-1) goto 45
        i=i+1
      if(i.le.np-3) goto 40
c
c     for rhs (gotos used to simulate while loops)
c
      i=ilow
55    continue
        j=i-2
60      continue
          px1=x i)
          py1=y i)
          px2=x(i-1)
          py2=y(i-1)
          qx1=x(j)
          qy1=y(j)
          qx2=x(j-1)
          qy2=y(j-1)
          int=insect(px1,py1,px2,py2,qx1,qy1,qx2,qy2)
          if(int.eq.1) then
            np=np+j-i
            do 65 k=j,np
              x(k)=x(k+i-j)
              y(k)=y(k+i-j)
65          continue
          endif
        j=j-1
          if(j.ge.2) goto 60
        i=i-1
      if(i.ge.4) goto 55
c
c     ensure that montonically increasing/decreasing in y
on
c     respective sides
c
      low=y(1)
      do 70 i=2,np
          if(y(i).lt.low) then
            l w=y(i)
            ılow=i
          endif
70     continue

c
c     simulated while loop
c
      i=ilow
90    continue
          count=0
          if y i-1).le.y(i)) then
            do 95 j=i-1,1,-1
              if(y j).le.y(i)) count=count+1
95        continue
        endif
        if(count.ne.0) then
          ilow=ilow-count
          np=np-count
          do 100 j=i-count,np
            x(j =x j+count
            y(j =y j+count
100       continue
          endif
        i=i-1-count
      if(i.ge.2) goto 90

      i=ilow
10    continue
          count=0
        dis1=((x(i)-x(i-1) **2.+(y(i)-y i-1))**2.) 100.
        if(i.eq.2) then
          dis2=1000.
        else
          dis2= x i -x(i-2) **2.+ y i)-y i-2  **2.
        endif
        if(dis1.lt.dis2) then
            if y(i-1).le.y(i ) then
              do 15 j=i-1,1,-1
                if y(j).le.y(i)) count count+1
15          continue
            endif
        else
          count=1
        endif
        if(count.ne.0) then
          ilow=ilow-count
          np=np-count
          do 20 j=i-count,np
            x(j)=x(j+count
            y(j)=y(j+count
20        continue
          endif
        i=i-1-count
      if(i.ge.2) goto 10
c
c     simulated while loop
c
      i=ilow
155   continue
          count=0
          if(y(i+1).le.y(i)  then
            do 160 j=i+1,np
              if(y(j).le.y(i)) count=count+1
160       continue
          endif
        if(count.ne.0) then
          np=np-count
          do 165 j=i+1,np
            x(j)=x(j+count
            y(j)=y(j+count)
165       continue
        endif
        i=i+1
      if(i.lt.np) goto 15
      i=ilow
75    continue
          count=0
        dis1=((x(i)-x i+1))**2.+(y(i)-y(i+1))**2.)/100.
        if(i.eq.(np-1)) then
          dis2=1000.
        else
          dis2= x(i)-x(i+2))**2.+(y i -y i+2 )**2.
        endif
        if dis1.lt.dis2) then
            if(y(i+1).le.y(i)) then
              do 80 j=i+1,np
                if(y(j).le.y(i)) count=count+1
80          continue
            endif
        else
          count=1
        endif
        if(count.ne.0) then
          np=np-count
          do 85 j=i+1,np
            x(j)=x(j+count)
            y(j)=y(j+count)
85        continue
        endif
        i=i+1
      if(i.lt.np) goto 75

      return
      end


      integer function
insect(px1,py1,px2,py2,qx1,qy1,qx2,qy2)
      double precision
px1,py1,px2,py2,qx1,qy1,qx2,qy2,m1,
     +m2,c1,c2,xbar,xx
      integer insect

      insect=0

      if(px1.eq.px2) then
        m1=10E+34
        c1=0.
      else
        m1= py1-py2)  px1-px2
        c1-py1-m1*px1
      endif

      ıf(qx1.eq.qx2  then
        m2=10E+34
        c2=0.
      else
        m2=(qy1-qy2)/(qx1-qx2)
        c2=qy1-m2*qx1
      endif

      if(m1.ne.m2) then
        xbar=(c1-c2)/(m2-m1)

         if(qx1.gt.qx2) then
           xx=qx1
           qx1=qx2
           qx2=xx
         endif

         if(px1.gt.px2) then
           xx=px1
           px1=px2
           px2=xx
         endif

         if((xbar.gt.qx1).and.(xbar.le.qx2).and.
     +(xbar.gt.px1).and.(xbar.le.px2)) insect=1
       endif

       return
       end



         subroutine
tao(nln,nlp,nrn,nrp,rxrn,ryrn,rxrp,ryrp,rxln,
     +ryln,rxlp,rylp)
c
c     Plotting routine used for diagnostic purposes only
c
         integer nln,nlp,nrn,nrp,l
         double precision
rxrn(151),ryrn(151),rxrp(151),ryrp(151),
     +rxln(151),ryln(151),rxlp(151),rylp(151)
        real xp,yp

         call devno(1)
         call scales(-.15,.2,1,-.2,.17,1)
c        call scales(-.1,.15,1,-.05,.15,1)
c        call scales(-.007,.007,1,-.007,.007,1)
         call newpic
           xp=sngl(rxrn(1))
           yp=sngl(ryrn(1))
         call markpt(xp,yp,1)
         call break
           xp=sngl(rxrp(1))
           yp=sngl(ryrp(1))
         call markpt(xp,yp,1)
         call break
           xp=sngl(rxln(1))
           yp=sngl(ryln(1))-.15
         call markpt(xp,yp,1)
         call break
           xp=sngl(rxlp(1))
           yp=sngl(rylp(1))-.15
         call markpt(xp,yp,1)
         call break

         do 1274 l=1,nrn
           xp=sngl(rxrn(l))
           yp=sngl(ryrn(l))
1274      call brknpt(xp,yp,0)
         call break
         do 1275 l=1,nrp
             xp=sngl(rxrp(l))
             yp=sngl(ryrp(l))
1275      call brknpt(xp,yp,1)
         call break

         do 1334 l=1,nln
           xp=sngl(rxln(l))
           yp=sngl(ryln(l))-.15
1334      call brknpt(xp,yp,0)
         call break
         do 1335 l=1,nlp
             xp=sngl(rxlp(l))
             yp=sngl(rylp(l))-.15
1335      call brknpt(xp,yp,1)
         call break

         call endplt

         return
         end


         subroutine
tpenv(env,envp,n1,n2,n3,n4,n1p,n2p,n3p,n4p)
c
c     Plotting routine used for diagnostic purposes only
c
         integer n1,n2,n3,n4,n1p,n2p,n3p,n4p,l
         double precision env(50,4,3),envp(50,4,3)
       real xp,yp

         call devno(1)
       call scales(-.35,.15,1,-.1,.2,1)
       call newpic
         call markpt(0.,0.,1)
         call break
```

```
         do 1274 l=1,n3
           xp=sngl(env(l,3,3))
           yp=sngl(env(l,3,1))
1274      call brknpt(xp,yp,0)
         call break
         do 1275 l=1,n1
             xp=sngl(env(l,1,3))
             yp=sngl(env(l,1,1))
1275      call brknpt(xp,yp,0)
         call break
         do 1334 l=1,n3p
             xp=sngl(envp(l,3,3))
             yp=sngl(envp(l,3,1))
1334      call brknpt(xp,yp,1)
         call break
         do 1335 l=1,n1p
             xp=sngl(envp(l,1,3))
             yp=sngl(envp(l,1,1))
1335      call brknpt(xp,yp,1)
         call break


         do 9000 l=1,50
           xp=sngl(env(l,3,3))-.2
           yp=sngl(env(l,3,1))
9000      call brknpt(xp,yp,0)
         call break
         do 9001 l=1,50
             xp=sngl(env(l,1,3))-.2
             yp=sngl(env(l,1,1))
9001      call brknpt(xp,yp,0)
         call break
         do 9002 l=1,50
             xp=sngl(envp(l,3,3))-.2
             yp=sngl(envp(l,3,1))
9002      call brknpt(xp,yp,1)
         call break
         do 9003 l=1,50
             xp=sngl(envp(l,1,3))-.2
             yp=sngl(envp(l,1,1))
9003      call brknpt(xp,yp,1)
         call break

         call endplt

         call devno(1)
       call scales(-.35,.15,1,-.1,.2,1)
       call newpic
         call markpt(0.,0.,1)
         call break

         do 2274 l=1,n4
           xp=sngl(env(l,4,3))
           yp=sngl(env(l,4,1))
2274      call brknpt(xp,yp,0)
         call break
         do 2275 l=1,n2
             xp=sngl(env(l,2,3))
             yp=sngl(env(l,2,1))
2275      call brknpt(xp,yp,0)
         call break
         do 2334 l=1,n4p
             xp=sngl(envp(l,4,3))
             yp=sngl(envp(l,4,1))
2334      call brknpt(xp,yp,1)
         call break
         do 2335 l=1,n2p
             xp=sngl(envp(l,2,3))
             yp=sngl(envp(l,2,1))
2335      call brknpt(xp,yp,1)
         call break


         do 9004 l=1,50
           xp=sngl(env(l,4,3))-.2
           yp=sngl(env(l,4,1))
9004      call brknpt(xp,yp,0)
         call break
         do 9005 l=1,50
             xp=sngl(env(l,2,3))-.2
             yp=sngl(env(l,2,1))
9005      call brknpt(xp,yp,0)
         call break
         do 9006 l=1,50
             xp=sngl(envp(l,4,3))-.2
             yp=sngl(envp(l,4,1))
9006      call brknpt(xp,yp,1)
         call break
         do 9007 l=1,50
             xp=sngl(envp(l,2,3))-.2
             yp=sngl(envp(l,2,1))
9007      call brknpt(xp,yp,1)
         call break

         call endplt

         call devno(1)
       call scales(-.35,.15,1,-.1,.2,1)
       call newpic
         call markpt(0.,0.,1)
         call break

         do 3274 l=1,n3
           xp=sngl(env(l,3,2 )
           yp=sngl(env(l,3,1))
3274      call brknpt(xp,yp,0)
```

```
         call break
         do 3275 l=1,n4
            xp=sngl(env(l,4,2))
            yp=sngl(env(l,4,1))
3275      call brknpt(xp,yp,0)
         call break
         do 3334 l=1,n3p
            xp=sngl(envp(l,3,2))
            yp=sngl(envp(l,3,1))
3334      call brknpt(xp,yp,1)
         call break
         do 3335 l=1,n4p
            xp=sngl(envp(l,4,2))
            yp=sngl(envp(l,4,1))
3335      call brknpt(xp,yp,1)
         call break

         do 9008 l=1,50
           xp=sngl(env(l,3,2))-.2
           yp=sngl(env(l,3,1))
9008      call brknpt(xp,yp,0)
         call break
         do 9009 l=1,50
            xp=sngl(env(l,4,2))-.2
            yp=sngl(env(l,4,1))
9009      call brknpt(xp,yp,0)
         call break
         do 9010 l=1,50
            xp=sngl(envp(l,3,2))-.2
            yp=sngl(envp(l,3,1))
9010      call brknpt(xp,yp,1)
         call break
         do 9011 l=1,50
            xp=sngl(envp(l,4,2))-.2
            yp=sngl(envp(l,4,1))
9011      call brknpt(xp,yp,1)
         call break


         call endplt

         call devno(1)
       call scales(-.35,.15,1,-.1,.2,1)
       call newpic
         call markpt(0.,0.,1)
         call break

1        do 4274 l=1,n2
           xp=sngl(env(l,2,2))
           yp=sngl(env(l,2,1))
4274      call brknpt(xp,yp,0)
         call break
         do 4275 l=1,n1
            xp=sngl(env(l,1,2))
            yp=sngl(env(l,1,1))
4275      call brknpt(xp,yp,0)
         call break
         do 4334 l=1,n2p
            xp=sngl(envp(l,2,2))
            yp=sngl(envp(l,2,1))
4334      call brknpt(xp,yp,1)
         call break
         do 4335 l=1,n1p
            xp=sngl(envp(l,1,2))
            yp=sngl(envp(l,1,1))
4335      call brknpt(xp,yp,1)
         call break

         do 9012 l=1,50
           xp=sngl(env(l,2,2))-.2
           yp=sngl(env(l,2,1))
9012      call brknpt(xp,yp,0)
         call break
         do 9014 l=1,50
            xp=sngl(env(l,1,2))-.2
            yp=sngl(env(l,1,1))
9014      call brknpt(xp,yp,0)
         call break
         do 9015 l=1,50
            xp=sngl(envp(l,2,2))-.2
            yp=sngl(envp(l,2,1))
9015      call brknpt(xp,yp,1)
         call break
         do 9016 l=1,50
            xp=sngl(envp(l,1,2))-.2
            yp=sngl(envp(l,1,1))
9016      call brknpt(xp,yp,1)
         call break

99        call endplt

       return
         end


      subroutine tidyup(xn,yn,xp,yp,nn,np,th1,th2)
c
c     this subroutine is called by subroutine femvol and
c     afemvol it make no rootine of function calls
c
c     tidys up the end points of the sketched pericardium
c     and prevents overlapping of the peri and
endocardium
c     outlines
c
c     It makes no subroutine calls
c
      double precision xn(151),yn(151),xp(151),yp(151),
     +th1,th2
      integer nn,np,i,stat,kk,trip

c     for begining of pericardium
      kk=0
      trip=0
      do 5 i=1,np/2
        if((xp(i).gt.xn(1)).and.(trip.eq.0)) then
          stat=i
          trip=1
        endif
5     continue

c     create new end point if necessary

      trip=0
      do 15 i=stat,np/2
        if((th1*0.7.le.dabs(xn(1)-
xp(i))).and.(trip.eq.0)) then
          kk=i
          trip=1
          endif
15    continue

      if((kk.gt.1).and.(th1*0.7.ne.dabs(xn(1)-xp(kk))))
then
        yp(kk-1)=dabs((xp(kk)-xn(1)-th1)/(xp(kk)-xp(kk-
1)))*
     +(yp(kk-1)-yp(kk))+yp(kk)
        if(yn(1).gt.yp(kk-1)) yp(kk-1)=yn(1)
          xp(kk-1)=xn(1)+th1*0.7
          kk=kk-1
      endif

c     reorder points if necessary

      if(kk.gt.1) then
        np=np-kk+1
        do 20 i=1,np
          xp(i)=xp(i+kk-1)
          yp(i)=yp(i+kk-1)
20      continue
      elseif((kk.eq.0).and.(stat.gt.1)) then
        np=np-stat+1
        do 25 i=1,np
          xp(i)=xp(i+stat-1)
          yp(i)=yp(i+stat-1)
25      continue
      endif

c     for end of pericardium
      kk=0
      trip=0
      stat=np
      do 30 i=np,np/2,-1
        if((xp(i).lt.xn(nn)).and.(trip.eq.0).and.
     +(xp(i).lt.0.)) then
          stat=i
          trip=1
        endif
30    continue

c     create new end point if necessary
      if(trip.eq.1) then
        trip=0
        do 35 i=stat,np/2,-1
          if((th2*0.5.lt.dabs(xn(nn)-
xp(i))).and.(trip.eq.0)) then
            kk=i
            trip=1
            endif
35      continue

        if(trip.eq.0) kk=stat

        if((kk.lt.np).and.(th2*0.5.ne.dabs(xp(kk)-
xn(np)))) then
          yp(kk+1)=yp(kk)+(xn(nn)-th2-xp(kk))/(xp(kk+1)-
xp(kk))*
     +(yp(kk+1)-yp(kk))
          if(yn(nn).gt.yp(kk+1)) yp(kk+1)=yn(nn)
            xp(kk+1)=xn(nn)-th2*0.5
            kk=kk+1
        endif
        np=kk

      endif

c     add in aorta pericardium if missed
      if(dabs(xp(np)-xn(nn)).gt.th2) th2=dabs(xp(np)-
xn(nn))
      if(yn(nn).gt.yp(np)) then
        np=np+1
        yp(np)=yn(nn)
          xp(np)=xn(nn)-th2
      endif

      if(dabs(xp(1)-xn(1)).gt.th1) th1=dabs(xp(1)-xn(1))
      if(yn 1).gt.yp(1)) then
        np=np+1
        do 40 i=np,2,-1
          yp(i)=yp i-1)
            xp(i)=xp(i-1)
40      continue
        yp(1)=yn(1)
          xp(1)=xn(1)+th1
      endif

      return
      end


      subroutine
sketch(nlp,nln,rixlp,riylp,rixln,riyln,jj,val
     +,th1,th2)
C-----Estimates unknown pericardial outline
C-----Bases estimated thickness at last digitized pt
C-----and 2mm apex backfills on first pts thickness
C-----Last changed 06-02-77
      double precision
rixlp 151),riylp(151),rixln(151 ,riyln 151 ,

+xl,yl,sav,xl,yl,x2,y2,hyp,thick,x3,y3,t3,opp,adj,rlow,x,
rx,rxl,

+fact1,fact2,rx2,tempx(151),tempy(151 ,val,low,xx,cc,aa,b
b,
     +y20,y02,x20,x02,x20x,x02x,frac,th1,th2
      integer
i,j,ij,nln,nlp,k,miss,l,kdum,knlp,mm,km,id,isav1,isav2,
     +ii,ilow,p(-2:2),nn,onp,side,jj

      onp=nlp
      nn=0

      low=riyln(1)
      do 5 i=2,nln
        if(riyln(i).lt.low  then
          low=riyln(i)
            xx=rixln(i)
          ilow=i
        endif
5     continue

      side=1
      if(riylp 1).lt.riyln nln ) side=2

      if(riylp nlp .lt.riyln(nln   then
C-----Find endo pt opp last peri pt
C-----and hence define thickness
        xl-rixlp nlp
        yl=riylp nlp
        sav=12345678.

        do 10 i=1,nln
          x1=rixln(i)
          y1=riyln(i)
          x2=xl-x1
          y2=yl-y1
          hyp=asqrt x2*x2+y2*y2
           if hyp.lt.sav) sav=hyp
          if hyp.eq.sav) k=i+1
 10     continue
        thick=sav
          th2=thick

 C-----Fillin pericardium
        miss=0
        do 50 i=k,nln
          x1=rixln(i-1)
          x2=rixln(i
          y1=riyln(i-1)
          y2=riyln(i)
          x3=x2-x1
          y3=y1-y2
          if  x3.ne.0.).and. y3.ne.0.)) then
            t3=x3/y3
            t3=dabs(t3)
            t3=datan(t3)
            t3=dabs t3)
            opp=dsin(t3)*thick
            adj=dcos t3 *thick
            l=i+nlp+1-k-miss
            if  i.gt.ilow+1 .or. i.lt.ilow-1 ) then
              if(x1.gt.x2.and.y1.gt.y2) then
                rixlp(l)=x2+adj
                riylp(l)=y2-opp
              elseif x1.gt.x2.and.y1.lt.y2) then
                rixlp l =x2-adj
                riylp l)=y2-opp
              elseif x1.lt.x2.and.y1.lt.y2  then
                rixlp l =x2-adj
                riylp l)=y2+opp
              elseif(x1.lt.x2.and.y1.gt.y2  then
                rixlp l)=x2+adj
                riylp(l)=y2+opp
              endif
            else
              rixlp l =-1000.
              riylp l =-10 0.
            endif
          else
            print*,'CVERFLOW - point missed k -> nln'
            miss=miss+1
          endif
 50       continue
          nlp=1
        endif

        if riylp 1 .lt.riyln 1   then

 C-----Backfill
          dc 1    i=1,151
            tempx i =rixlp i
1000        tempy i =riylp i

C-----Find endo pt opp first peri pt
C-----and hence find thickness
        sav=12345678.
        xl=rixlp(1)
        yl=riylp 1)

        do 1010 i=1,nln
          x1=rixln i
          y1=riyln i
          x2=xl-x1
          y2=yl-y1
          hyp=dsqrt x2*x2+y2*y2)
          if hyp.lt.sav) sav=hyp
          if hyp.eq.sav) k=i-1
1010    continue

        if(k.ge.2) then
          thick=sav
            th1=thick
          miss=0

          do 1050 i=k,1,-1
            x1=rixln(i+1)
            x2=rixln i)
            y1=riyln i+1)
            y2=riyln(i)
            x3=x2-x1
            y3=y1-y2
            if((x3.ne.0.).and. y3.ne.0.)) then
              t3=x3 y3
              t3=dabs(t3)
              t3=datan(t3)
              opp=dsin(t3)*thick
              adj=dcos(t3)*thick
              l=i+miss
              if((i.gt.ilow+1).or.(i.lt.ilow-1)) then
                if(x1.gt.x2.and.y1.gt.y2) then
                  rixlp(l)=x2-adj
                  riylp(l)=y2+opp
                elseif(x1.gt.x2.and.y1.lt.y2  then
                  rixlp(l)=x2+adj
                  riylp(l)=y2+opp
                elseif(x1.lt.x2.and.y1.lt.y2  tnen
                  rixlp(l)=x2+adj
                  riylp(l)=y2-opp
                elseif(x1.lt.x2.and.y1.gt.y2) then
                  rixlp(l)=x2-adj
                  riylp(l)=y2-opp
                endif
              else
                rixlp(l)=-1000.
                riylp(l)=-1000.
              endif
            else
              print*,'overflow - point missed k -> 1'
              miss=miss+1
            endif
1050      continue

          kdum=k+nlp
          if kdum.ge.148) k=148-nlp
          knlp=k+nlp-miss
            if knlp.gt.148) then
            print*,' ERROR !! array is not large enough '
            stop
          endif

C-----Move forward to accomodate misses
          if(miss.ne.0) then
            mm=miss+1
            do 1052 ii=mm,k
              ij=ii+.-mm
              riylp ij -riylp(ii)
1052          rixlp ij =rixlp ii
          endif

          km=k-miss

          do 1080 i=km,knlp
            id=i-km+1
            rixlp(i+1 =tempx
            riylp i+1 =tempy
1080      continue
          nlp=nlp+km
        endif
      endif

      onp=onp+k-miss
c
c     Add in apex points using the quadratic equation
c                           Y=AAX 2+kYX+cc
      if side.eq.1  then
        nn=1
        do 15 i=nlp,1,-1
          if (rixlp(i .eq.-1000.).and. riylp i).eq.-
1000.)) then
            p(nn)-i
            nn=nn-1
          endif
15      continue
        if nn.lt.1  then
          p    p 1 -i
          p -1  p 1 -.
          nn-10
          endif
        else
          nn=-1
          do 25 i=1,nlp
            if((rixlp(i).eq.-1000.).and.(riylp(i).eq.-
1000.)) then
              p(nn)=i
              nn=nn+1
            endif
 25       continue
          if(nn.gt.-1) then
            p(1)=p(-1)+2
            p(0)=p(-1)+1
            nn=10
          endif
        endif

        if(nn.eq.10) then

          p(-2)=p(-1)-1
          ilow=p(-2)
          do 30 i=ilow,1,-1
            if(riylp(i).lt.riylp(p(-2))) p(-2)=i
 30       continue

          p(2)=p(1)+1
          ilow=p(2)
          do 35 i=ilow,nlp
            if(riylp(i).lt.riylp(p(2))) p(2)=i
 35       continue

          rixlp(p(0))=xx
          if(side.eq.2) then
            riylp(p(0))=low-th1
          else
            riylp(p(0))=low-th2
          endif

          y20=riylp(p(2))-riylp(p(0))
            y02=riylp(p(-2))-riylp(p(0))
            x20=rixlp(p(2))-rixlp p 0))
            x02=rixlp(p(-2))-rixlp(p 0 )
            x20x=rixlp(p(2))*rixlp p 2 )-
rixlp(p(0))*rixlp(p(0))
            x02x=rixlp p(-2))*rixlp p(-2))-
rixlp(p(0))*rixlp(p(0))

          aa=(x20*y02-y20*x02   x02x*x20-x20x*x02)
            bb=(y02-aa*x02x  x02
            cc=riylp p 0) -aa*rixlp p 0 )*rixlp p(0 )-
bb*rixlp p 0))

          do 40 i=p 0 -1,p -2 +1,-1
            frac=dble  fl at i-p     float p -2)-p 0))))
            rixlp i =rixlp p 0  + rixlp p -2 )-
rixlp(p 0 ))*frac
            riylp i)=aa*rixlp i *rixlp i +bb*rixlp i)+cc
40        continue

          do 45 i=p(0)+1,p 2 -1
            frac=dble (float i-p 0   float p 2 -p(0 )))
            rixlp i)=rixlp p 0  + rixlp p 2  -
rixlp p(0)))*frac
            riylp(i)=aa*rixlp i)*rixlp i +bb*rixlp(i)+cc
45        continue
        endif

c         call
tao(nln,nlp,nln,nlp,rixln,riyln,rixlp,riylp,rixln,
c     +riyln,rixlp,riylp)

        call noloop(nlp,rixlp,riylp


C-----Flatten apex to 2mm for rao and corespondig
thickess for lao
        rlow=12345678.

        do 69 i=1,nln
          if(riyln(i).lt.rlow) then
            x=rixln(i)
            rlow=riyln(i)
          endif
69      continue

        rlow=rlow-.002
        if(jj.eq.1) rlow=val

        do 80 i=1,nlp
80        if(riylp(i).lt..005) goto 90

90      isav1=i-1


        do 81 j=i,nlp
81        if(riylp(j).gt..005) goto 91

91      isav2=j


        rx=rixlp(isav1)-x
        fact1=rx*rx/(riylp(isav1 +dabs(rlow))
        rx2=rixlp(isav2)-x
        fact2=rx2*rx2/(riylp(isav2)+dabs(rlow))

        do 99 i=isav1,isav2
          rx1=rixlp(i)-x
```

```
          if(rx1.ge.0.) riylp(i)=(rx1*rx1/fact1)-dabs(rlow)
99        if(rx1.lt.0.) riylp(i)=(rx1*rx1/fact2)-dabs(rlow)

        return
        end


        subroutine fillup(geom)
        double precision geom(3,255)
        integer i,j,k,id,idd

C-----Fills in mid wall points as halfway between epi and
endo
C-----Last changed 9-08-77

        id=0
        do 10 i=1,6
          do 20 j=1,3
            do 20 k=2,27,5
              idd=id+k
20            geom(j,idd)=(geom(j,idd+1)+geom(j,idd-
1))/2.d0
10        id=id+42

        return
        end


        subroutine plane(i,j,k,l,ccx,ccy,ccz,geom)
C-----Last changed 22-05-78
        double precision
ccx,ccy,ccz,x4,x5,y4,y5,z4,z5,c,a,b,geom(3,255)
        integer j,k,l,i


        x4=geom(1,i)-ccx
        x5=geom(1,l)-geom(1,i)
        y4=geom(2,i)-ccy
        y5=geom(2,l)-geom(2,i)
        z4=geom(3,i)-ccz
        z5=geom(3,l)-geom(3,i)
        b=(x4-((y4*x5)/y5))/(z4-((z5*y4)/y5))
        a=(x4- b*z4))/y4
        c=ccx-(a*ccy)-(b*ccz)
        geom(1,j)=a*geom(2,j)+b*geom(3,j)+c
        geom(1,k)=a*geom(2,k)+b*geom(3,k)+c

        return
        end


        subroutine
drawarc(jj,kk,ll,tenv,vy,vz,by,bz,cy,cz,uy,uz)
C-----Fills in envlope boxes with an arc in each quarter
C-----Last changed 09-03-78
        double precision
vy,vz,by,bz,cy,cz,uy,uz,tenv(4,2),theta,tant,

+rm,c,wy,wz,rn,d,ro,e,yy,xy,zy,yz,xz,zz,lwz,lwv,cost,sint
        integer jj,kk,ll

C-----Find points u & v
        uy=(tenv(kk,1)+tenv(ll,1))/2.
        uz=(tenv(kk,2)+tenv(ll,2))/2.
        vy=(tenv(kk,1)+tenv(jj,1))/2.
        vz=(tenv(kk,2)+tenv(jj,2))/2.

C-----Find eq'n u-v
        rm=(vz-uz)/(vy-uy)

C-----Change for 90 deg
        rm=(1./rm)*(-1.)

C-----Find mid-chord w
        wy=(uy+vy)/2.
        wz=(uz+vz)/2.

C-----Find eq'n constant
        c=wz-rm*wy

C-----Find line perp' to 1-2 @ v

if((tenv(jj,1).ne.tenv(kk,1)).and.(tenv(jj,2).ne.tenv(kk,
2))) then
          rn=(tenv(jj,2)-tenv(kk,2))/(tenv(jj,1)-
tenv(kk,1))
          rn=(1./rn)*(-1.)
          d=vz-rn*vy
        endif

C-----Find line perp' to 1-3 @ u

if((tenv(kk,1).ne.tenv(ll,1)).and.(tenv(kk,2).ne.tenv(ll,
2))) then
          ro=(tenv(kk,2)-tenv(ll,2))/(tenv(kk,1)-
tenv(ll,1))
          ro=(1./ro)*(-1.)
          e=uz-ro*uy
        endif

C-----Find points of equality
        if(tenv(kk,1).eq.tenv(jj,1)) yy=(vz-c)/rm
        if(tenv(kk,2).eq.tenv(jj,2)) yy=vy

if(tenv(kk,1).ne.tenv(jj,1).and.tenv(kk,2).ne.tenv(jj,2))
      +yy=(d-c  (rm-rn)
        if(tenv(kk,1).eq.tenv(ll,1)) xy=(uz-c)/rm
      if(tenv(kk,2).eq.tenv(ll,2)) xy=uy

if(tenv(kk,1).ne.tenv(ll,1).and.tenv(kk,2).ne.tenv(ll,2))
     +xy=(e-c)/(rm-ro)

      zy=(yy+xy)/2.
      yz=rm*yy+c
      xz=rm*xy+c
      zz=(yz+xz)/2.

C-----Find angle subtended
      lwz=dsqrt((zz-wz)*(zz-wz)+(zy-wy)*(zy-wy))
      lwv=dsqrt((vz-wz)*(vz-wz)+(vy-wy)*(vy-wy))
      tant=lwv/lwz
      theta=.666667*(datan(tant))
      sint=dsin(theta)
      cost=dcos(theta)

C-----Calc points on arc for data
      by=((vy-zy)*cost+(vz-zz)*sint)+zy
      bz=((vz-zz)*cost-(vy-zy)*sint)+zz
      sint=dsin(theta*2.)
      cost=dcos(theta*2.)
      cy=((vy-zy)*cost+(vz-zz)*sint)+zy
      cz=((vz-zz)*cost-(vy-zy)*sint)+zz

      return
      end


      subroutine rotatenv(x,y,z,env,envp,t3)
      double precision
t1(3,3),t2(3,3),envt(200,3),enva(200,3),

+hyp1,cos1,sin1,hyp2,cos2,sin2,env(50,4,3),envp(50,4,3),
     +x,y,z,t3(3,3)
      integer i,j,loop,k

      do 10 i=1,3
        do 10 j=1,3
          t1(i,j)=0.
          t2(i,j)=0.
10        t3(i,j)=0.

C-----Find angles
      hyp1=dsqrt(z*z+y*y)
      cos1=y/hyp1
      sin1=z/hyp1
      hyp2=dsqrt(x*x+hyp1*hyp1)
      cos2=x/hyp2
      sin2=hyp1/hyp2

C-----Fill transform matrices
      t1(1,1)=1.
      t1(2,2)=cos1
      t1(3,3)=cos1
      t1(3,2)=sin1
      t1(2,3)=sin1*(-1.)

      t2(1,1)=cos2
      t2(1,2)=sin2*(-1.)
      t2(2,1)=sin2
      t2(2,2)=cos2
      t2(3,3)=1.

C-----Find overall transform
      call mmlt(t3,t1,t2,3,3,3)

C-----Transform envelope
      do 20 i=1,50
        do 20 j=1,4
          do 20 k=1,3
            loop=((i*4)-4)+j
20          envt(loop,k)=env(i,j,k)

      call mmlt(enva,envt,t3,200,3,3)

      do 30 i=1,50
        do 30 j=1,4
          do 30 k=1,3
            loop=((i*4)-4)+j
30          env(i,j,k)=enva(loop,k)

      do 40 i=1,50
        do 40 j=1,4
          do 40 k=1,3
            loop=((i*4)-4)+j
40          envt(loop,k)=envp(i,j,k)

      call mmlt(enva,envt,t3,200,3,3)

      do 50 i=1,50
        do 50 j=1,4
          do 50 k=1,3
            loop=((i*4)-4)+j
50          envp(i,j,k)=enva(loop,k)

      return
      end


      subroutine
arcbas(env,envp,geom,n1,n2,n3,n4,n1p,n2p,n3p,n4p)
C-----Last changed 22-05-78
      double precision
vy,vz,by,bz,cy,cz,uy,uz,cbasex,cbasey,cbasez,
+cbaspx,cbaspy,cbaspz,tenv(4,2),env(50,4,3),envp(50,4,3),
     +geom(3,255)
      integer n1,n2,n3,n4,n1p,n2p,n3p,n4p

C-----Calc box size
      tenv(1,1)=env(n1,1,2)
      tenv(1,2)=env(n1,1,3)
      tenv(2,1)=env(n2,2,2)
      tenv(2,2)=env(n2,2,3)
      tenv(3,1)=env(n3,3,2)
      tenv(3,2)=env(n3,3,3)
      tenv(4,1)=env(n4,4,2)
      tenv(4,2)=env(n4,4,3)

      call drawarc(2,1,3,tenv,vy,vz,by,bz,cy,cz,uy,uz)

      geom(2,3)=vy
      geom(3,3)=vz
      geom(2,5)=by
      geom(3,5)=bz
      geom(2,8)=cy
      geom(3,8)=cz
      geom(2,10)=uy
      geom(3,10)=uz

      call drawarc(1,3,4,tenv,vy,vz,by,bz,cy,cz,uy,uz)

      geom(2,13)=by
      geom(3,13)=bz
      geom(2,15)=cy
      geom(3,15)=cz
      geom(2,18)=uy
      geom(3,18)=uz

      call drawarc(3,4,2,tenv,vy,vz,by,bz,cy,cz,uy,uz)

      geom(2,20)=by
      geom(3,20)=bz
      geom(2,23)=cy
      geom(3,23)=cz
      geom(2,25)=uy
      geom(3,25)=uz

      call drawarc(4,2,1,tenv,vy,vz,by,bz,cy,cz,uy,uz)

      geom(2,28)=by
      geom(3,28)=bz
      geom(2,30)=cy
      geom(3,30)=cz

C-----Epi
      tenv(1,1)=envp(n1p,1,2)
      tenv(1,2)=envp(n1p,1,3)
      tenv(2,1)=envp(n2p,2,2)
      tenv(2,2)=envp(n2p,2,3)
      tenv(3,1)=envp(n3p,3,2)
      tenv(3,2)=envp(n3p,3,3)
      tenv(4,1)=envp(n4p,4,2)
      tenv(4,2)=envp(n4p,4,3)

      call drawarc(2,1,3,tenv,vy,vz,by,bz,cy,cz,uy,uz)

      geom(2,1)=vy
      geom(3,1)=vz
      geom(2,4)=by
      geom(3,4)=bz
      geom(2,6)=cy
      geom(3,6)=cz
      geom(2,9)=uy
      geom(3,9)=uz

      call drawarc(1,3,4,tenv,vy,vz,by,bz,cy,cz,uy,uz)

      geom(2,11)=by
      geom(3,11)=bz
      geom(2,14)=cy
      geom(3,14)=cz
      geom(2,16)=uy
      geom(3,16)=uz

      call drawarc(3,4,2,tenv,vy,vz,by,bz,cy,cz,uy,uz)

      geom(2,19)=by
      geom(3,19)=bz
      geom(2,21)=cy
      geom(3,21)=cz
      geom(2,24)=uy
      geom(3,24)=uz

      call drawarc(4,2,1,tenv,vy,vz,by,bz,cy,cz,uy,uz)

      geom(2,26)=by
      geom(3,26)=bz
      geom(2,29)=cy
      geom(3,29)=cz

C-----Define x at mid points
      geom(1,3)=(env(n2,2,1)+env(n1,1,1))/2.
      geom(1,10)=(env(n3,3,1)+env(n1,1,1))/2.
      geom(1,18)=(env(n3,3,1)+env(n4,4,1))/2.
      geom(1,25)=(env(n2,2,1)+env(n4,4,1))/2.
      geom(1,1)=(envp(n2p,2,1)+envp(n1p,1,1))/2.
      geom(1,9)=(envp(n3p,3,1)+envp(n1p,1,1))/2.
      geom(1,16)=(envp(n3p,3,1)+envp(n4p,4,1))/2.
      geom(1,24)=(envp(n2p,2,1)+envp(n4p,4,1))/2.
```

```
C-----Find centre base

cbasex=(geom(1,3)+geom(1,10)+geom(1,18)+geom(1,25))/4.

cbasey=(geom(2,3)+geom(2,10)+geom(2,18)+geom(2,25))/4.

cbasez=(geom(3,3)+geom(3,10)+geom(3,18)+geom(3,25))/4.

cbaspx=(geom(1,1)+geom(1,9)+geom(1,16)+geom(1,24))/4.

cbaspy=(geom(2,1)+geom(2,9)+geom(2,16)+geom(2,24))/4.

cbaspz=(geom(3,1)+geom(3,9)+geom(3,16)+geom(3,24))/4.

C-----Find planes and Fill in for x
      call plane(3,5,8,10,cbasex,cbasey,cbasez,geom)
      call plane(10,13,15,18,cbasex,cbasey,cbasez,geom)
      call plane(18,20,23,25,cbasex,cbasey,cbasez,geom)
      call plane(25,28,30,3,cbasex,cbasey,cbasez,geom)
      call plane 1,4,6,9,cbaspx,cbaspy,cbaspz,geom)
      call plane(9,11,14,16,cbaspx,cbaspy,cbaspz,geom)
      call plane(16,19,21,24,cbaspx,cbaspy,cbaspz,geom)
      call plane(24,26,29,1,cbaspx,cbaspy,cbaspz,geom)

      return
      end


      subroutine checkr(geoma,patr)
C-----Program CHECKR which checks form of data in geom
prepared by
C     envelope for f.e. analysis
C-----last changed 22-05-94
      double precision geoma(3,255)
      real geom(3,255),x,y,z,min1,min2
      integer iseq(64),iapex(17)

+,isl1(97),isl2(31),isl3(97),isl4(31),isl5(97),isl6(31),
     +i,ipen,k,j
      character answer,patr*6,ascii*7

      data iseq/1,4,6,9,11,14,16,19,21,24,26,29,1,
     + 2,3,5,8,10,13,15,18,20,23,25,28,30,3,
     + 5,8,7,6,9,11,12,13,15,18,17,16,19,21,
     + 22,23,25,28,27,26,29,1,
     + 31,33,35,37,39,41,31,32,34,36,
     + 38,40,42,32,31/
      data iapex/241,243,245,247,249,251,241,
     + 242,244,246,248,250,252,242,253,254,255/
      data
isl1/1,2,3,32,45,74,87,116,129,158,171,200,213,242,255,24
8,

+228,206,186,164,144,122,102,80,60,38,18,17,16,37,58,59,

+60,59,58,79,100,101,102,101,100,121,142,143,144,143,142,
163,

+184,185,186,185,184,205,226,227,228,227,226,247,253,254,
255,

+254,253,241,211,212,213,212,211,199,169,170,171,170,169,
157,127,

+128,129,128,127,115,85,86,87,86,85,73,43,44,45,44,43,31,
1/
      data
isl2/4,5,47,89,131,173,215,255,230,188,146,104,62,20,19,6
1,
     +
103,145,187,229,253,254,255,254,253,214,172,130,88,46,4/
      data
isl3/6,7,8,34,50,76,92,118,134,160,176,202,218,244,255,25
0,

+233,208,191,166,149,124,107,82,65,40,23,22,21,39,63,64,

+65,64,63,81,105,106,107,106,105,123,147,148,149,148,147,
165,

+189,190,191,190,189,207,231,232,233,232,231,249,253,254,
255,254,

+253,243,216,217,218,217,216,201,174,175,176,175,174,159,
132,133,

+134,133,132,117,90,91,92,91,90,75,48,49,50,49,48,33,6/
      data
isl4/9,10,52,94,136,178,220,255,235,193,151,109,67,25,24,

+66,108,150,192,234,253,254,255,254,253,219,177,135,93,51
,9/
      data
isl5/11,12,13,36,55,78,97,120,139,162,181,204,223,246,255
,

+252,238,210,196,168,154,126,112,84,70,42,28,27,26,41,68,
69,

+70,69,68,83,110,111,112,111,110,125,152,153,154,153,152,
167,

+194,195,196,195,194,209,236,237,238,237,236,251,253,254,
255,254,

+253,245,221,222,223,222,221,203,179,180,181,180,179,161,
137,138,
```

```
+139,138,137,119,95,96,97,96,95,77,53,54,55,54,53,35,11/
      data
isl6/14,15,57,99,141,183,225,255,240,198,156,114,72,30,29
,

+71,113,155,197,239,253,254,255,254,253,224,182,140,98,56
,14/

903   write(6,902)
902   format(///
     +' Do you wish to check the form of the 255 co-
ordinates as'/
     +' found by ENVLOP before continuing? ',$)
      read(*,*) answer

      if(answer.eq.'Y'.or.answer.eq.'y') then

904     do 1 i=1,3
          do 1 j=1,255
1           geom(i,j)=sngl(geoma(i,j))

          call initsp
        call page 29.,20.)
        call picsiz(27.,18.)
        call newpic

c
c     Set scale for the plotting of the element meshes
c
        call scales(0.0,0.65,1,-0.453,-0.02,1)

          min1=-123.
          min2=-123.
        ipen=3
        do 11 i=1,5

225     ipen=3
        do 10 j=1,49
          k=iseq(j)+(i-1)*42
          y=geom(2,k)+(.055)
          x=geom(3,k)+(.09*(i))
            if(y.gt.min1) min1=y
          call plot(x,-y,ipen)
10        ipen=2

        ipen=3
        do 12 j=50,64
          k=iseq(j)+(i-1)*42
          y=geom(2,k)+.155
          x=geom(3,k)+(.09*i)
            if(y.gt.min2) min2=y
          call plot(x,-y,ipen)
12        ipen=2
        ipen=3

11      continue

235     ipen=3
        do 20 j=1,49
          k=iseq(j)+210
          y=geom(2,k)+(.055)
          x=geom(3,k)+.54
          call plot(x,-y,ipen)
20        ipen=2

        ipen=3
        do 30 i=1,17
          k=iapex(i)
          y=geom(2,k)+.155
          x=geom(3,k)+.54
          call plot(x,-y,ipen)
30        ipen=2

        ipen=3
        do 110 j=1,97
          i=isl1(j)
          y=geom(1,i)+.25
          x=geom(3,i)+.09
          call plot(x,-y,ipen)
110       ipen=2

        ipen=3
        do 120 j=1,31
          i=isl2(j)
          y=geom(1,i)+.25
          x=geom(3,i)+.18
          call plot(x,-y,ipen)
120       ipen=2

        ipen=3
        do 130 j=1,97
          i=isl3(j)
          y=geom(1,i)+.25
          x=geom 3,i)+.27
          call plot(x,-y,ipen)
130       ipen=2

        ipen=3
        do 140 j=1,31
          i=isl4(j)
          y=geom 1,i +.25
          x=geom 3,i)+.36
          call plot(x,-y,ipen)
140       ipen=2

        ipen=3
        do 150 j=1,97
          i=isl5(j)
          y=geom(1,i)+.25
          x=geom(3,i)+.45
          call plot(x,-y,ipen)
150       ipen=2

        ipen=3
        do 160 j=1,31
          i=isl6(j)
          y=geom(1,i)+.25
          x=geom(3,i)+.54
          call plot(x,-y,ipen)
160       ipen=2

        ipen=3
        do 210 j=1,97
          i=isl1(j)
          y=geom(1,i)+.35
          x=geom(2,i)+.09
          call plot(x,-y,ipen)
210       ipen=2

        ipen=3
        do 220 j=1,31
          i=isl2(j)
          y=geom(1,i)+.35
          x=geom(2,i)+.18
          call plot(x,-y,ipen)
220       ipen=2

        ipen=3
        do 230 j=1,97
          i=isl3(j)
          y=geom(1,i)+.35
          x=geom(2,i)+.27
          call plot(x,-y,ipen)
230      ipen=2

        ipen=3
        do 240 j=1,31
          i=isl4(j)
          y=geom(1,i)+.35
          x=geom(2,i)+.36
          call plot(x,-y,ipen)
240      ipen=2

        ipen=3
        do 250 j=1,97
          i=isl5(j)
          y=geom(1,i)+.35
          x=geom(2,i)+.45
          call plot(x,-y,ipen)
250       ipen=2

        ipen=3
        do 260 j=1,31
          i=isl6(j)
          y=geom(1,i)+.35
          x=geom(2,i)+.54
          call plot(x,-y,ipen)
260       ipen=2
         ipen=3
c
c      Add Text to picture
c
          min1=1030.-min1/.43*1000.
          min2=1030.-min2/.43*1000.

         call scales(0.0,1500.0,1,0.0,1000.0,1)


         call cp7pt(0.,1000.,16,patr)
         call cp7pt(0.,980.,16,'Nodal points joined by
straight lines for
      + diagnostic purposes only')

         call cp7pt(0.,min1,16,'level')
         call cp7pt(0.,650.,16,'y-z')
         call cp7pt(0.,min2,16,'level')
         call cp7pt(0.,380.,16,'x-z')
         call cp7pt(0.,130.,16,'x-y')

           call figfmt('F',1,4)

           do 5 i=1,5
           k=i+(i-1)*42
           z=140.+float(i*180)
             call kreal(geom(1,k),ascii)
           call cp7pt(z,min1,16,ascii)
           k=k+30
             call kreal(geom(1,k),ascii)
           call cp7pt(z,min2,16,ascii)
5          continue

           call kreal(geom 1,211),ascii)
         call cp7pt(1210.,min1,16,ascii)

         call endplt

       elseif(answer.ne.'N'.and.answer.ne.'n') then
         goto 903
       endif

       return
       end
```

```
      subroutine
split(onod,volum,ijkl,ifrmno,patr,namer,comtr,
     +iresp,nlm)
      real volum(151)
      double precision onod(255,3),wnod(401,3),vo,smovol
      integer
ijkl,ii,jj,i,j,k,iresp,nlm,ifrmno,iswtch,isw
      character
answer,filnam*80,filenam*80,patr*6,namer*20,
     +comtr*20,cframe*2
      common/person/  filenam,cframe

C-----Subroutine splits into 2 layer model
C     Program to Change 36 element data to 72
      ii=0
      jj=0

      do 10 i=1,6
        do 20 j=1,6
          do 30 k=1,3
            wnod(ii+1,k)=onod(jj+1,k)
            wnod(ii+3,k)=onod(jj+2,k)
            wnod(ii+5,k)=onod(jj+3,k)
            wnod(ii+2,k)=(wnod(ii+1,k)+wnod(ii+3,k))/2.
            wnod(ii+4,k)=(wnod(ii+3,k)+wnod(ii+5,k))/2.
            wnod(ii+6,k)=onod(jj+4,k)
            wnod(ii+8,k)=onod(jj+5,k)
30          wnod(ii+7,k)=(wnod(ii+6,k)+wnod(ii+8,k))/2.
          jj=jj+5
20        ii=ii+8
        do 40 j=1,6
          do 50 k=1,3
            wnod(ii+1,k)=onod(jj+1,k)
            wnod(ii+3,k)=onod(jj+2,k)
50          wnod(ii+2,k)=(wnod(ii+1,k)+wnod(ii+3,k))/2.
          jj=jj+2
40        ii=ii+3
10     continue

       do 80 k=1,3
         wnod(397,k)=onod(253,k)
         wnod(399,k)=onod(254,k)
         wnod(401,k)=onod(255,k)
         wnod(398,k)=(wnod(397,k)+wnod(399,k))/2.
80       wnod(400,k)=(wnod(399,k)+wnod(401,k))/2.


       if((nlm.eq.0).and.(ifrmno.ne.0)) then
5        if(ifrmno.ne.0) write(6,1) ifrmno
         read(*,*) answer
         if(answer.eq.'Y'.or.answer.eq.'y') then
           call knumb(ifrmno,cframe)
           filnam='coords'//cframe//'.'//filenam

open(9,file=filnam,access='sequential',form='formatted')
           write(9,3) patr,namer,comtr,ifrmno
           write(9,4)
(wnod(i,1),wnod(i,2),wnod(i,3),i,i=1,401)
           close(9)
         elseif(answer.ne.'N'.and.answer.ne.'n') then
           goto 5
         end if
         call data(wnod,ifrmno,0)
       else
         call
fevolume(wnod,vo,smovol,ijkl,iresp,iswtch,isw)
         volum(ijkl)=sngl(vo)
       endif

1      format(////
     +' Do you wish to create a datafile containing the
401'/
     +' co-ordinates for frame ',i2,'? ',$)
3      format(a6,1x,2(a20,1x)/' CO-ORDINATES FOR FRAME NO.
',i2)
4      format(3f18.12,i5)
12     format(a80)

       return
       end


       subroutine data wnod,ifrmno)
C-----Datafile for FE analysis constructed here
       double precision wnod(401,3)
       real
modrat,mat(10,9),presss,prelem(72),aaa(72),bbb(72),
      +ccc(72),presve(72)
       integer
ifrmno,nelem,nonop,mannd,lo,nplc,co,nomat,matno(72),

+elemty(72),codeno(72),angle1(72),angle2(72),angle3(72),

+angle4(72),angle5(72),angle6(72),angle7(72),angle8(72),

+angle9(72),nonode(72),seqnon(72),az,by,A(72 ,B(72),C 72)
,D(72),

+E(72),F(72),G(72),H(72),U 72),V(72),W(72),X 72),Y(72),Z(
72),

+O(72),P(60),Q 60),R(60),S(60),T(60),aa(72),bb(72),cc(72)
,no,

+dd 72),i,j,first,last,matype,n2,n3,ik,icount,ii,isw1,isw
2,isw3
```

```
      character answer,title*80

      data
A/1,3,9,11,17,19,25,27,33,35,41,43,67,69,75,77,83,85,91,9
3,

+99,101,107,109,133,135,141,143,149,151,157,159,165,167,1
73,175,

+199,201,207,209,215,217,223,225,231,233,239,241,265,267,
273,275,

+281,283,289,291,297,299,305,307,397,399,397,399,397,399,
397,399,
     +397,399,397,399/
      data
B/2,4,10,12,18,20,26,28,34,36,42,44,68,70,76,78,84,86,92,

+94,100,102,108,110,134,136,142,144,150,152,158,160,166,1
68,174,

+176,200,202,208,210,216,218,224,226,232,234,240,242,266,
268,274,

+276,282,284,290,292,298,300,306,308,339,341,347,349,355,
357,363,
     +365,371,373,331,333/
      data
C/3,5,11,13,19,21,27,29,35,37,43,45,69,71,77,79,85,87,93,

+95,101,103,109,111,135,137,143,145,151,153,159,161,167,1
69,175,

+177,201,203,209,211,217,219,225,227,233,235,241,243,267,
269,275,

+277,283,285,291,293,299,301,307,309,331,333,339,341,347,
349,355,
     +357,363,365,371,373/
      data
D/6,7,14,15,22,23,30,31,38,39,46,47,72,73,80,81,88,89,96,

+97,104,105,112,113,138,139,146,147,154,155,162,163,170,1
71,178,

+179,204,205,212,213,220,221,228,229,236,237,244,245,270,
271,278,

+279,286,287,294,295,302,303,310,311,399,401,399,401,399,
401,399,
     +401,399,401,399,401/
      data
E/7,8,15,16,23,24,31,32,39,40,47,48,73,74,81,82,89,90,97,

+98,105,106,113,114,139,140,147,148,155,156,163,164,171,1
72,179,

+180,205,206,213,214,221,222,229,230,237,238,245,246,271,
272,279,

+280,287,288,295,296,303,304,311,312,341,343,349,351,357,
359,365,
     +367,373,375,333,335/
      data
F/9,11,17,19,25,27,33,35,41,43,1,3,75,77,83,85,91,93,99,

+101,107,109,67,69,141,143,149,151,157,159,165,167,173,17
5,133,

+135,207,209,215,217,223,225,231,233,239,241,199,201,273,
275,281,

+283,289,291,297,299,305,307,265,267,333,335,341,343,349,
351,357,
     +359,365,367,373,375/
      data
G/10,12,18,20,26,28,34,36,42,44,2,4,76,78,84,86,92,94,100
,

+102,108,110,68,70,142,144,150,152,158,160,166,168,174,17
6,134,

+136,208,210,216,218,224,226,232,234,240,242,200,202,274,
276,282,

+284,290,292,298,300,306,308,266,268,382,383,385,386,388,
389,391,
     +392,394,395,379,380/
      data
H/11,13,19,21,27,29,35,37,43,45,3,5,77,79,85,87,93,95,101
,

+103,109,111,69,71,143,145,151,153,159,161,167,169,175,17
7,135,

+137,209,211,217,219,225,227,233,235,241,243,201,203,275,
277,283,

+285,291,293,299,301,307,309,267,269,336,337,344,345,352,
353,360,
     +361,368,369,376,377/
      data
U/49,50,52,53,55,56,58,59,61,62,64,65,115,116,118,119,121
,

+122,124,125,127,128,130,131,181,182,184,185,187,188,190,
191,193,
```

```
+194,196,197,247,248,250,251,253,254,256,257,259,260,262,
263,313,

+314,316,317,319,320,322,323,325,326,328,329,379,380,382,
383,385,
     +386,388,389,391,392,394,395
      data
V/50,51,53,54,56,57,59,60,62,63,65,66,116,117,119,120,122
,

+123,125,126,128,129,131,132,182,183,185,186,188,189,191,
192,194,

+195,197,198,248,249,251,252,254,255,257,258,260,261,263,
264,314,

+315,317,318,320,321,323,324,326,327,329,330,398,400,398,
400,398,
     +400,398,400,398,400,398,400
      data
W/52,53,55,56,58,59,61,62,64,65,49,50,118,119,121,122,124
,

+125,127,128,130,131,115,116,184,185,187,188,190,191,193,
194,196,

+197,181,182,250,251,253,254,256,257,259,260,262,263,247,
248,316,

+317,319,320,322,323,325,326,328,329,313,314,340,342,348,
350,356,
     +358,364,366,372,374,332,334
      data
X/53,54,56,57,59,60,62,63,65,66,50,51,119,120,122,123,125
,

+126,128,129,131,132,116,117,185,186,188,189,191,192,194,
195,197,

+198,182,183,251,252,254,255,257,258,260,261,263,264,248,
249,317,

+318,320,321,323,324,326,327,329,330,314,315,332,334,340,
342,348,
     +350,356,358,364,366,372,374/
      data
Y/67,69,75,77,83,85,91,93,99,101,107,109,133,135,141,143,

+149,151,157,159,165,167,173,175,199,201,207,209,215,217,
223,225,

+231,233,239,241,265,267,273,275,281,283,289,291,297,299,
305,307,

+331,333,339,341,347,349,355,357,363,365,371,373,383,384,
386,387,
     +389,390,392,393,395,396,380,381/
      data
Z/68,70,76,78,84,86,92,94,100,102,108,110,134,136,142,144
,

+150,152,158,160,166,168,174,176,200,202,208,210,216,218,
224,226,

+232,234,240,242,266,268,274,276,282,284,290,292,298,300,
306,308,

+332,334,340,342,348,350,356,358,364,366,372,374,337,338,
345,346,
     +353,354,361,362,369,370,377,378/
      data
O/69,71,77,79,85,87,93,95,101,103,109,111,135,137,143,145
,

+151,153,159,161,167,169,175,177,201,203,209,211,217,219,
225,227,

+233,235,241,243,267,269,275,277,283,285,291,293,299,301,
307,

+309,333,335,341,343,349,351,357,359,365,367,373,375,380,
381,383,
     +384,386,387,389,390,392,   ,395,396/
      data
P/72,73,80,81,88,89,96,97,104,105,112,113,138,139,146,147
,

+154,155,162,163,170,171,178,179,204,205,212,213,220,221,
228,229,

+236,237,244,245,270,271,278,279,286,287,294,295,302,303,
310,311,
     +336,337,344,345,352,353,360,361,368,369,376,377/
      data
Q/73,74,81,82,89,90,97,98,105,106,113,114,139,140,147,148
,

+155,156,163,164,171,172,179,180,205,206,213,214,221,222,
229,230,

+237,238,245,246,271,272,279,280,287,288,295,296,303,304,
311,312,
     +337,338,345,346,353,354,361,362,369,370,377,378
      data
R 75,77,83,85,91,93,99,101,107,109,67,69,141,14 ,149,1 .,
+157,159,165,167,173,175,133,135,207,209,215,217,223,225,
231,233,

+239,241,199,201,273,275,281,283,289,291,297,299,305,307,
265,267,
     +339,341,347,349,355,357,363,365,371,373,331,333/
      data
S/76,78,84,86,92,94,100,102,108,110,68,70,142,144,150,152
,

+158,160,166,168,174,176,134,136,208,210,216,218,224,226,
232,234,

+240,242,200,202,274,276,282,284,290,292,298,300,306,308,
266,268,
     +340,342,348,350,356,358,364,366,372,374,332,334/
      data
T/77,79,85,87,93,95,101,103,109,111,69,71,143,145,151,153
,

+159,161,167,169,175,177,135,137,209,211,217,219,225,227,
233,235,

+241,243,201,203,275,277,283,285,291,293,299,301,307,309,
267,269,
     +341,343,349,351,357,359,365,367,373,375,333,335/

120   write(6,54)
      read(*,*) answer

      if(answer.eq.'Y'.or.answer.eq.'y') then

      write(6,2)
      read(*,'(a80)') title

C-----Number of total elements=72
      nelem=72
C-----Number of nodal points=401
      nonop=401

C-----Number of nodes where point loads are applied
      write(6,3)
      read(*,*) lo

C-----Number of presss loaded faces
      write(6,4)
      read(*,*) nplc

C-----Number of nodes where constraints are applied
      write(6,5)
      read(*,*) co

C-----Maximum node number difference in any element=112
      mannd=112

C-----Number of material types
      write(6,6)
      read(*,*) nomat

C-----Material properties for anisotropy
      do 100 i=1,nomat
        write(6,10) i
        write(6,9)
        read(*,*) mat(i,1)
        write(6,7)
        read(*,*) modrat
        write(6,8)
        read(*,*) mat(i,4)
        mat(i,2)=mat(i,1)*modrat
        mat(i,3)=mat(i,1)
        mat(i,5)=mat(i,4)
        mat(i,6)=mat(i,4)/modrat
        mat(i,9)=mat(i,1)/(2.d0*(1.d0+mat(i,4)))
        mat(i,7)=mat(i,9)*modrat
        mat(i,8)=mat(i,7)
100   continue

103   write(6,68)
      read(*,*) answer
      if(answer.eq.'Y'.or.answer.eq.'y') then
        do 13 i=1,nomat
          write(6,14) i
          read(*,*) first,last
          if(first.eq.0.and.last.eq.0) goto 12
          do 16 j=first,last
16          matno(j)=i
13      continue
      elseif(answer.ne.'N'.and.answer.ne.'n') then
        goto 103
      endif

12    if(first.ne.1.or.last.ne.nelem) then
106     write 6,17)
        read *,*) answer
        if answer.eq.'Y'.or.answer.eq.'y') then
          do 19 i=1,nomat
            write(6,20) i
21          read(*,*) matype
            if matype.ne.0) then
              matno(matype)=i
              goto 21
            endif
19        continue
        elseif answer.ne.'N'.and.answer.ne.'n') then
          goto 1
        endif
```

```
      endif

C-----Fibre angle configuration
      write(6,48)
      write 6,23)
      read(*,*  n2
      write(6,24)
      read(*,*) n3
      if(n2.gt.0) then
        call
angl1(n2,angle1,angle2,angle3,angle4,angle5,angle6,
     +angle7,angle8,angle9,1,codeno)
      endif
      if(n3.ne.0) then
        call
angl2(n3,angle1,angle2,angle3,angle4,angle5,angle6,
     +angle7,angle8,angle9,2,codeno)
      endif

      ik=(5*nelem)/6

      do 30 i=1,nelem
        if(i.lt.ik+1) nonode(i)=20
        if(i.gt.ik) nonode(i)=15
30    continue

      do 31 i=1,nelem
        seqnon(i)=1

if((i.eq.11).or.(i.eq.12).or.(i.eq.23).or.(i.eq.24).or.
     +(i.eq.35).or.(i.eq.36).or.(i.eq.47).or.(i.eq.48))
seqnon(i)=0
31    continue

      do 32 i=ik-1,nelem
32      seqnon(i)=0

      do 33 i=1,nelem
33      elemty(i)=i

C-----Constraint data
      write(6,25)

      icount=0
      do 34 i=1,nelem
        icount=icount+1
        read(*,*) az,by
        if(az.eq.0.and.by.eq.0) goto 35
        aa(i)=az
        if(by.eq.1) then
          bb(i)=1
          cc(i)=0
          dd(i)=0
        else if(by.eq.2) then
          bb(i)=0
          cc(i)=1
          dd(i)=0
        else if(by.eq.3) then
          bb(i)=0
          cc(i)=0
          dd(i)=1
        else if(by.eq.4) then
          bb(i)=1
          cc(i)=1
          dd(i)=0
        else if(by.eq.5) then
          bb i)=1
          cc(i)=0
          dd(i)=1
        else if(by.eq.6) then
          bb(i)=0
          cc(i)=1
          dd i)=1
        else if(by.eq.7) then
          bb i)=1
          cc i)=1
          dd i)=1
        end if
34    continue

35    write(6,36)
      read *,*) answer

      if(answer.eq.'Y'.or.answer.eq.'y') then
108     write 6,38)
        read *,*) presss
        presss=presss*9.81*13600./1000000.
        ii=2
        do 39 i=ii,nelem,2
39        presve(i)=presss
      elseif(answer.eq.'N'.or.answer.eq.'n') then
37      write 6,80)
        read *,*) answer
        if answer.eq.'Y'.or.answer.eq.'y') then
          write(6,83)
          read *,*) presss
          presss=presss*9.81*13600./1000000.
          write 6,87)
92        read *,*) first,last
          if first.eq.0.and.last.eq.0  goto 82
          do 84 i=first,last,2
84          presve(i)=presss
          goto 92
        elseif(answer.ne.'N'.and.answer.ne.'n') then
          goto 37
        endif
82      write 6,85
```

```
86       read(*,*) no,presss
         presss-presss*9.81*13600./1000000.
         presve(no)=presss
         if(no.eq.0.and.presss.eq.0) goto 90
         goto 86
        else
         goto 35
        endif

90      continue

        do 40 i=ii,nelem,2
          aaa(i -0
          bbb i)=0.
          ccc i)=1.
40       prelem(i)=i

C-----Switches in XL1
        write(6,69)
        write 6,70)
        read(*,*) isw1
        write(6,71)
        read(*,*) isw2
        write 6,72)
        read(*,*  isw3

C-----Sending info to file for attachment to XL1
        open(77,file='XL1-
data',access='sequential',form='formatted')
        write(77,41)
title,nelem,nonop,lo,nplc,co,mannd,nomat
        write(77,42) isw1,isw2,isw3
        write(77,43)
(mat(i,1),mat(i,2),mat(i,3),mat(i,4),mat(i,5),
     +mat(i,6),mat(i,7),mat(i,8),mat(i,9),i=1,nomat)
        write(77,44)
(wnod(i,1),wnod(i,2),wnod(i,3),i=1,401)
        do 45 i=1,nelem
          write(77,46
i,matno(i),elemty(i),codeno(i),angle1(i ,

+angle2 i),angle3(i),angle4(i),angle5(i ,angle6 i),angle7
(i),
     +angle8(i),angle9 i),nonode(i),seqnon i)
          if(i.lt.ik+1) write(77,47)
A i ,B i),C(i),D i),E(i),F(i),

+G(i),H(i),U(i),V i),W(i),X(i),Y(i),Z(i),O(i),P(i),Q(i),R
(i),
     +S(i),T(i)
          if i.gt.ik  write 77,46)
A i),B i ,C(i),D(i),E(i),F i),
     +G i),H(i),U(i),V(i ,W i),X i),Y(i),Z i),O(i)
45       continue
         write(77,52) (aa(i),bb i),cc(i),dd(i),i=1,icount-
1
         write(77,53)
 prelem(i),aaa(i),bbb(i),ccc(i),presve(i),i=
     +2,72,2)
         close(77)
       elseif(answer.ne.'N'.and.answer.ne.'n'  then
         goto 120
       endif

2      format(// /' TITLE = ',$)
3      format(   /' NUMBER OF POINT LOADS ? = ',$)
4      format(/// ' NUMBER OF PRESSURE LOADED FACES ? =
',$)
5      format(////
      +' NUMBER OF NODES WHERE CONSTRAINTS ARE APPLIED ? =
',$)
6      format(////' NUMBER OF MATERIAL TYPES ? = ',$)
7      format(////
      +' ALONGSIDE/CROSS FIBRE MODULII RATIO
(Ey Ex;Ey/Ez)? = ',$
8      format(////
      +' ALONGSIDE TO CROSS FIBRE POISSON RATIO (Mu-yx;Mu-
zx)? = ',$)
9      format(////
      +' MODULII (Ex,Ez) FOR MINOR AXIS (MERIDIONAL
DIRECTION)'/
      +' in KN/m2 = ',$
10     format(/// //'    Material Properties for Material
No.',i2,/
      +'    --------------------------------------')
14     format( ///
      +' Enter the 1st and last element numbers comprising
of '/
      +' material no.',i2,' TYPE ''0 0'' to exit)'//)
17     format(   /' Are any of the material properties
scattered? ',$)
20     format(  //' Enter element number of material
type',i2,/
      +' (TYPE IN ''0'' to exit '  )
23     format(  / ' No. of elements with fibre orientation
code 1? ',$)
24     format( /  ' No. of elements with fibre orientation
code 2? ',$)
25     format(/// '     Constraint data entered here'/
      +'     ---------------------------'/
      +' Enter 2 numbers(spaced),the first being the node
number'/
      +' and the second being a number from 1 to 7
depending'/
      +' upon the following global conditions :'
      +'      1 : restrained in X direction only'
      +'      _ :     "      in Y    "     "  '
```

```
      +'     3 :     "      in Z     "      "  '
      +'     4 :     "      in X and Y directi ns on_y'
      +'     5 :     "      in X and Z     "       " '
      +'     6 :     "      in Y and Z     "       " '
      +'     7 :     "      in X,Y and Z directions
altogether'///
      +' Put info. for each element on seperate line '
      +' (TYPE ''0 0'' to exit)'/
      +' e.g.5 7 means element 5 is restrained in all 3
axes'// )
36    format(   /'     Pressure load data entered here'
      +'     ------------------------------'
      +' Is the presss uniform throughout the ventricle?
',$)
38    format  ' Pressure in the Ventricle ?  in mmHg
',$)
41
format a80 i2,',',i3,',',i1,',',i2,',',i2,',',i3,',',i1
42    format i1,',',i1,',',i1)
43
format f11.4,',',f11.4,',',f11.4 f6.3,',',f6.3,',',f6.3
     +f11.4,',',f11.4,',',f11.4
44    format f18.12,',',f17.12,',',f17.12
46    format 15i4
47    format 20i4
48    format      '    Fibre angle options available are
:'
      +'    -----------------------------------'
      +' Fibre angle varying through the element:
1'//
      +' Fibre orientation parallel to global axes:    2'
52    format(i2,',',i1,',',i1,',',i1)
53    format(f3.0,',',f2.0,',',f2.0,',',f2.0,',',f7.5
54    format( ///
      +' Preperation of the data file for the finite
element analysis'/
      +' can now be carried out. Data is entered from the
keyboard;'/
      +' NOTE: this only applies to the 2 layer thro''wall
thickness'/
      +' model.'   /
      +' Do you wish to produce this data file? ',$
68    format(////
      +' Are the material properties of some or all or the
elements'/
      +' in an ordered sequential numbering order. ',$
69    format(////
      +' There are 3 switches introduced in FE program :
'/
      +' ISW1,ISW2 & ISW3'///
      +'      ISW1 = 1 FOR STRESS PLOTTING        , ELSE =
0'/
      +'      ISW2 = 2 FOR STRAIN OUTPUT          , ELSE
0'/
      +'      ISW3 = 3 FOR REFLECTION OF DATA  NLY,  LSE =
0'//)
70    format(///'Value for I W1? = ',$)
71    format(///'Value for ISW2? = ',$)
72    format(///'Value for ISW3? = ',$
80    format // /
      +' Even though the pressure is non-uniform,is there
a series'/
      +' of sequentially numbered elements over which the
presss is'
      +' uniform? ',$
83    format(////' Pressure in ventricle  in mmHg  ',$
87    format(////
      +' Enter the 1st & last frame numbers (spaced) over
which'/
      +' pressure is uniform (TYPE ''0 0'' to exit)'
85    format(////
      +' Enter 2 numbers for scattered elements,the 1st
being the'/
      +' element number,the 2nd the pressure  in mmHg '
      +' (''0 0'' to exit)'/ )

      return
      end



      subroutine
angl1(n,a1,a2,a3,a4,a5,a6,a7,a8,a9,jj,codeno)
      integer
n,a1 72 ,a2 72 ,a3 72),a4 72 ,    72 ,a6 72 ,a7 72 ,
     +a8 72 ,a9 72 ,jj,codeno  2),first,last,alpha,beta,j
      character answer
      integer elemno

2     write(6,1) n,jj
      read *,*  answer

      if(answer.eq.'Y'.or.answer.eq.'y'  g t  4
      if answer.eq.'N'.or.answer.eq.'n'  g t  3
      goto 2

4     wri e 6,99
      read(*,*  first,last
      if first.eq.0.and.last.eq.0  goto 3
      write 6,7
7     f rmat   / ' Angle on  ut r wall -z)  f these
elements? ',$
      read *,*  alpha

      do 8 j-first,last,_
        a1 j  alpha
8       codeno
```

```
      write 6,9
9     format ' Angle on inner wall +z) of these elements?
',$)
      read *,*) beta

      do 10 j=first,last,2
        a2(j)=beta
        a3(j)=0
        a4(j)=0
        a5(j)=0
        a6(j)=0
        a7(j)=0
        a8 j)=0
10      a9 j)=0

      goto 4
3     continue
5     write(6,11) n,jj
      read(*,*) answer
      if(answer.eq.'Y'.or.answer.eq.'y') goto 13
      if(answer.eq.'N'.or.answer.eq.'n') goto 12
      goto 5
13    write(6,14)
      read(*,*) elemno
      if(elemno.eq.0) goto 12
      write 6,7)
      read(*,*  alpha
      a1(elemno)=alpha
      codeno(elemno)=jj
      write(6,9)
      read(*,*) beta
      a2(elemno =beta
      a3(elemno)=0
      a4(elemno)=0
      a5(elemno)=0
      a6(elemno)=0
      a7(elemno)=0
      a8(elemno)=0
      a9(elemno)=0
      goto 13
12    continue

99    format    /
     +' Enter the first and last element numbers'/
     +' (''0 0'' to exit '  )
11    format(////
     +' Are the ',i2,' elements(orientation',i2,')
scattered? ',$)
1     format  ///
     +' Are the ',i2,' elements orientation',i2,')
ordered? ',$
14    format(    ' Enter element number  0 to exit) ',$

      return
      end


      subroutine
angl2 n,a1,a2,a3,a4,a5,a6,a7,a8,a9,jj,codeno
      integer
n,a1(72),a2 72 ,a3 72),a4 72),a5(72 ,a6 72),a7 72),
     +a8(72),a9 72 ,jj,codeno 72 ,elemno,first,last,i
      character answer

12    write(6,1  n,jj
1     format(
     +' Are the ',i2,' elements fibre orientation ',i1,'
in a'/
     +' sequentially numbered order? ',$)
      read *,*) answer
      if answer.eq.'Y'.or.answer.eq.'y') goto 11
      if answer.eq.'N'.or.answer.eq.'n') goto 2
      goto 12
11    write 6,3)
3     format(// /
     +' Enter 1st and last element numbers for ordered
elements'
     +' (''0 0'' to exit)')
10    read *,*) first,last
      if first.eq.0.and.last.eq.0) goto 2

      do 4 i=first,last
        codeno(i)-jj
        a1 i =0
        a2 i)=0
        a3(i)=0
        a4 i =0
        a5 i =0
        a6 i  0
        a7(i  0
        a8 i)=0
4       a9 i)=0

      goto 10

2     write(6,5) jj
5     format(////
     +' Are any elements(orientation ',i1,' scattered?
',$)
      read *,*) answer
      if answer.eq.'Y'.or.answer.eq.'y') goto 13
      if answer.eq.'N'.or.answer.eq.'n') goto 9
      goto 2

13    write(6,7)
```

```
7     format     'Enter element number  '')  '' t  exit
',$
8     read *,*) elemn
      if elemn .eq.0  goto 9
      codeno elemn) =jj
      goto 8
9     continue

      return
      end



      subroutine
fevolume(array,vo,sm vol,ijkl,iresp iswtch isw
C-----Surface Area and Shape Index calculated  er
      integer
j,idiv,idivi,incr,iswtch,kl,klm,itmp,ijkl isw ire.p

+KX,KXX,KY,KZ,IX,IY,IZ,JX,JY,IXX IYY,IZZ IXXX,Jk JXX  .
kll,ji
      double precision
array(401,3),CX(109 ,CY 109 ,CZ 1 9  volume
     +vo,smovol,CCX 1 9),CCY 109 ,CCZ 1    thicknes

+a 3 ,b(3),c(3),AA,BB,CC,DD,d,al1 al2 al3 am1 am  am  an
,an2,

+an3,tran(3,3),slope1,slope ,xmid1,xmid ,ymid1,ymid  ref
rx 12),

+xcentre,ycentre,shapeindex,surfarea,radius,  ntr} --

+abcd(401,3),radii 12),thick 12 ,circxc circyc a1,a2 a
      character ans

      call rearrange 5,1,11,1,CX,CY,CZ,array
      call rearrange 8,2,12,1,CX,CY,CZ,array
      call rearrange 51,13,18,2,CX,CY,CZ array
      call rearrange 71,19,29,1,CX,CY,CZ,array
      call rearrange 74,20,30,1,CX,CY,CZ,array
      call rearrange 117,31,36,2,CX,CY,CZ,array
      call rearrange 137,37,47,1,CX,CY,CZ array
      call rearrange 140,38,48,1,CX,CY,CZ,array
      call rearrange 183,49,54,2,CX,CY,CZ,array
      call rearrange 203,55,65,1,CX,CY,CZ,array
      call rearrange 206,56,66,1,CX,CY,CZ,array
      call rearrange 249,67,72,2,CX,CY,CZ,array
      call rearrange 269,73,83,1,CX,CY,CZ,array
      call rearrange 272,74,84,1,CX,CY,CZ,array
      call rearrange 315,85,90,2,CX,CY,CZ,array)
      call rearrange 335,91,101,1,CX,CY,CZ,array
      call rearrange 338,92,102,1,CX,CY,CZ,arra
      call rearrange 381,103,108, ,CX,CY,CZ,arr
      CX(109 =array 401,1
      CY(109)=array 401,
      CZ(109 =array(401,3
      call rearrange 1,1,11,1,CCX,CCY,CCZ,array
      call rearrange 6,2,12,1,CCX,CCY,CCZ,array
      call rearrange 49,13,18,2,CCX,CCY CCZ,array
      call rearrange 67,19,29,1,CCX,CCY,CCZ,array)
      call rearrange 72,20,30,1,CCX,CCY,CCZ,array
      call rearrange(115,31,36,2,CCX,CCY,CCZ,array
      call rearrange 133,37,47,1,CCX,CCY,CCZ,array
      call rearrange 138,38,48,1,CCX,CCY,CCZ,array
      call rearrange 181,49,54,2,CCX,CCY,CCZ,array
      call rearrange(199,55,65,1,CCX,CCY,CCZ,array)
      call rearrange(204,56,66,1,CCX,CCY,CCZ,array
      call rearrange(247,67,72,2,CCX,CCY,CCZ,array
      call rearrange(265,73,83,1,CCX,CCY,CCZ,array
      call rearrange(270,74,84,1,CCX,CCY,CCZ,array
      call rearrange 313,85,90,2,CCX,CCY,CCZ,array
      call rearrange 331,91,101,1,CCX,CCY,CCZ,array
      call rearrange  36,92,102,1,CCX,CCY,CCZ,array)
      call rearrange 379,103,108,2,CCX,CCY,CCZ,array
      CCX(109 =array 401,1)
      CCY 109)=array 401,2)
      CCZ(109)=array 401,3)

      if((iresp.ne.1 .and. iresp.ne.2)  then
C-----Option to skip alternate nodes for layers defined
by 12 nodes
C     but which are in fact  n   defined by 6 nodes ;
answer 1 for
C     this but with 2 if one  ants every layer t  be
defined by 12
C     points

2       write 6,3)
3       format(/
     +' Do you wish to skip interpolating alternate
nodes? ',$
        read(*,*  ans
        if(ans.eq.'Y'.or.ans.eq.'y'  then
          iresp=1
        elseif ans.eq.'N'.or.ans.eq.'n'  then
            iresp=2
        else
          goto 
        endif

      endif

C-----Splitting each layer int  ub-divi i n
```

```
C-----Set idiv to 100 : therefore have 1200 slices thro'
ventricle
      do 1000 idiv=100,100
         volume=0

C-----Top layer
        idivi=idiv
        itmp=0*idiv
        call
subdiv(1,13,3,14,5,15,7,16,9,17,11,18,2,20,4,22,

+6,24,8,26,10,28,12,30,CX,CY,CZ,idivi,0,itmp,2,volume,ire
sp,
     +CCX,CCY,CCZ)

C-----Intermediate layers
        incr=idiv
        itmp=0*idiv+1
        call
subdiv(1,13,3,14,5,15,7,16,9,17,11,18,2,20,4,22,

+6,24,8,26,10,28,12,30,CX,CY,CZ,idiv,0,itmp,1,volume,ires
p,
     +CCX,CCY,CCZ)
        itmp=1*idiv+1
        call
subdiv(13,19,14,21,15,23,16,25,17,27,18,29,2,20,4,22,6,

+24,8,26,10,28,12,30,CX,CY,CZ,idiv,incr,itmp,1,volume,ire
sp,
     +CCX,CCY,CCZ)
        itmp=2*idiv+1
        call
subdiv(19,31,21,32,23,33,25,34,27,35,29,36,20,38,22,40,

+24,42,26,44,28,46,30,48,CX,CY,CZ,idiv,0,itmp,1,volume,ir
esp,
     +CCX,CCY,CCZ)
        itmp=3*idiv+1
        call
subdiv(31,37,32,39,33,41,34,43,35,45,36,47,20,38,22,40,

+24,42,26,44,28,46,30,48,CX,CY,CZ,idiv,incr,itmp,1,volume
,
     +iresp,CCX,CCY,CCZ)
        itmp=4*idiv+1
        call
subdiv(37,49,39,50,41,51,43,52,45,53,47,54,38,56,40,58,

+42,60,44,62,46,64,48,66,CX,CY,CZ,idiv,0,itmp,1,volume,ir
esp,
     +CCX,CCY,CCZ)
        itmp=5*idiv+1
        call
subdiv 49,55,50,57,51,59,52,61,53,63,54,65,38,56,40,58,

+42,60,44,62,46,64,48,66,CX,CY,CZ,idiv,incr,itmp,1,volume
,
     +iresp,CCX,CCY,CCZ)
        itmp=6*idiv+1
        call
subdiv 55,67,57,68,59,69,61,70,63,71,65,72,56,74,58,76,

+60,78,62,80,64,82,66,84,CX,CY,CZ,idiv,0,itmp,1,volume,ir
esp,
     +CCX,CCY,CCZ
        itmp=7*idiv+1
        call
subdiv(67,73,68,75,69,77,70,79,71,81,72,83,56,74,58,76,

+60,78,62,80,64,82,66,84,CX,CY,CZ,idiv,incr,itmp,1,volume
,
     +iresp,CCX,CCY,CCZ)
        itmp=8*idiv+1
        call
subdiv(73,85,75,86,77,87,79,88,81,89,83,90,74,92,76,94,

+78,96,80,98,82,100,84,102,CX,CY,CZ,idiv,0,itmp,1,volume,
     +iresp,CCX,CCY,CCZ)
        itmp=9*idiv+1
        call
subdiv(85,91,86,93,87,95,88,97,89,99,90,101,74,92,76,94,

+78,96,80,98,82,100,84,102,CX,CY,CZ,idiv,incr,itmp,1,volu
me,
     +iresp,CCX,CCY,CCZ)
        itmp=10*idiv+1
        call
subdiv(91,103 93,104,95,105,97,106,99,107,101,108,92,109,

+94,109,96,ı09,98,109,100,109,102,109,CX,CY,CZ,idiv,0,itm
p,
     +1,volume,iresp,CCX,CCY,CCZ)
        itmp=11*idiv+1
        call
subdiv(103,109,104,109,105,109,106,109,107,109,108,109,

+92,109,94,109,96,109,98,109,100,109,102,109,CX,CY,CZ,idi
v,incr,
     +itmp,1,volume,iresp,CCX,ᶜCY,CCZ)
        vo=volume*1000000.D0
        write 6,44) vo,idiv*12
44      tormat /' Volume =',f7.2,'cm3  No.  r layers
=',i4)

1000  continue
```

```
C-----S.Area & S.Index calculation
      do 1 j-1,3
        a j =array 5,j
        b(j)-array 21,j
1       c(j =array 37,j

      call roots(a,b,c,AA,BB,CC,DD)
c
c     This method does not yield the ventricle lenqth
c
c
d=dabs AA*array 401,1 +BB*array 401,2 +CC*array 401,3 +DD
)/
c      +dsqrt(AA*AA+BB*BB+CC*CC


a1=(array(5,1)+array(8,1 +array 13,1)+array 16,1)+array(2
1,1)+

+array(24,1)+array(29,1)+array(32,1)+array 37,1)+array(4
,1 +
     +array 45,1 +array(48,1 ) 12.d0

a2=(array(5,2 +array(8,2 +array(13,2 +array 16,2)+array 2
1,2)+

+array 24,2 +array 29,2 +array 32,2 +array 37,2 +array 4
,2)+
     +array 45,2)+array(48,2 )/12.d0

a3= array 5,3)+array(8,3)+array(13,3)+array 16,3 +array 2
1,3)+

+array(24,3 +array(29,3)+array(32,3)+array(37,3)+array 4
,3 +
     +array(45,3)+array(48,3))/12.d0
      d=((abs(a1-array(401,1)))**3.+(abs a2-
array 401,2)))**3.
     ++ abs(a3-array(401,3 ) **3.)**(1. 3.


C-----Base to apex length
      write(*,9) d*1000.
9     format(' Long Dimension (Base to Apex
Length)=',f7.3,'mms')

C-----Now outer loop for 6 layers with 12 inner nodes to
find centre
c     and radius and average wall thickness of each layer
      KX=1
      KXX=48
      KY=17
      KZ=33
      IX=5
      IY=21
      IZ=37
      JX=1
      JY=5

      do 4 kl=1,11,2
        do 5 j=1,3
          a(j)=array(KX,j)
          b j =array KY,j)
5         c(j =array(KZ,j

        call roots(a,b,c,AA,BB,CC,DD)
        call dcos1(AA,BB,CC,DD,al3,am3,an3)
        call secos2(KX,KY,array,al1,am1,an1)
        call dcos3(al3,am3,an3,al1,am1,an1,al2,am2,an2)
        call
trans(al3,am3,an3,al1,am1,an1,al2,am2,an2,tran)
        call dmmlt1(KX,KXX,array,tran,abcd)

C-----Taking points 5,21,37 - 8,24,40 - 13,29,45 -
16,32,48 to find
C     centre
C-----Finding slope of first line i.e.between 5 and 21
        xcentre=0.D0
        ycentre=0.D0
        circxc=0.D0
        circyc=0.D0
        IXX=IX
        IYY=IY
        IZZ=IZ

        do 6 klm=1,4
          if(klm.eq.2.or.klm.eq.4) then
            IXX-IXX+3
            IYY=IYY+3
            IZZ=IZZ+3
          else if klm.eq.3) then
            IXX=IXX+5
            IYY=IYY+5
            IZZ=IZZ-5
          end if
          slcpel= abcd IYY,) -abcd IXX,) / abcd IYY,1 -
abcd(IXX,1 )

C-----Slope of perp.bi-sector must be negative reciprical
          slopel--1.D  sl pel

C-----Finding midp int
          xmidl= abcd IYY,1 +abcd IXX,1   2.
          ymidl=(abcd IYY,  +abcd IXX,      .

C-----Equation is Y MX+C i.e. an be written a  Y-Y. M X-
X1
```

```
C-----or as Y-MID1=M(X-XMID1)
C-----i.e as Y=(SLOPE1*X)-(SLOPE1*XMID1)+YMID1
C-----in form for solving using subroutine eqsolv is;
C-----(SLOPE1*X)-Y- SLOPE1*XMID1)-YMID1
C-----Finding slope of second line i.e. between 5 and 37
         slope2=(abcd(IZZ,2)-abcd(IXX,2))/(abcd(IZZ,1)-
abcd(IXX,1))
         slope2=-1.D0/slope2
         xmid2=(abcd(IZZ,1)+abcd(IXX,1))/2.
         ymid2=(abcd(IZZ,2)+abcd(IXX,2))/2.
C-----Call to equation solving subroutine
         call
eqsolv(slope1,slope2,xmid1,ymid1,xmid2,ymid2,xcentre,
     +ycentre,2)
         circxc=xcentre+circxc
6      circyc=ycentre+circyc

       centrx(kl)=circxc/4.
       centry(kl)=circyc/4.

C-----Now taking all inner node points to find average
radius
*     i.e 5,8,13,16,21,24,29,32,37,40,45,48 for first
layer
       radius=0.D0
       IXXX=IX

       do 7 ik=1,12

if(ik.eq.2.or.ik.eq.4.or.ik.eq.6.or.ik.eq.8.or.ik.eq.10
     +.or.ik.eq.12) then
           IXXX=IXXX+3
         else
if(ik.eq.3.or.ik.eq.5.or.ik.eq.7.or.ik.eq.9.or.
     +ik.eq.11) then
           IXXX=IXXX+5
         end if
           radius=radius+dsqrt( centrx(kl)-
abcd(IXXX,1 )**2.+
     +(centry(kl)-abcd(IXXX,2))**2.)
7      continue


       radii(kl)=radius/12.
       thickness=0.D0
       JXX=JX
       JYY=JY

       do 8 kll=1,12
         if((kll.eq.2).or.(kll.eq.4 .or.(kll.eq.6).or.
     + kll.eq.8).or.(kll.eq.10 .or.(kll.eq.12)) then
           JXX=JXX+5
           JYY=JYY+3
         else
if(kll.eq.3.or.kll.eq.5.or.kll.eq.7.or.kll.eq.9.or.
     +kll.eq.11) then
           JXX=JXX+3
           JYY=JYY+5
         end if
         thickness=thickness+dsqrt((abcd JXX,1)-
abcd JYY,1) **2.+
     +(abcd(JXX,2)-abcd(JYY,2))**2.)
8      continue

       thick(kl)=thickness/12.

C-----Put thickness into arrays
       IX=IX+66
       IY=IY+66
       IZ=IZ+66
       JX=JX+66
       JY=JY+66
       KX=KX+66
       KXX=KXX+66
       KY=KY+66
4      KZ=KZ+66

      call calsa(array,surfarea)
c      surfarea=radii(1)*2.*d*3.141592654D0
      write(*,15) surfarea*1000000.
15    format(' Surface Area=',f8.2,'mm2')
      shapeindex=6.*dsqrt(3.141592654D0)*vo*1.E-
6/(surfarea**1.5)
      write(*,16) shapeindex
16    format(' Shape Index=',f8.6)

C-----Now for the intermediate layers
      KX=49
      KXX=66
      KY=55
      KZ=61
      IX=51
      IY=57
      IZ=63
      JX=49
      JY=51
      do 10 kl=2,12,2
        do 11 j=1,3
          a(j)=array(KX,j)
          b(j)=array(KY,j)
11        c(j)=array(KZ,j)
        call roots(a,b,c,AA,BB,CC,DD)
        call dcos1(AA,BB,CC,DD,al3,am3,an3)
        call secos2(KX,KY,array,al1,am1,an1)
        call dcos3 al3,am3,an3,al1,am1,an1,al2,am2,an2)
        call
trans(al3,am3,an3,al1,am1,an1,al2,am2,an_,tran)
```

```
        call dmmltl KX,KXX,array,tran,abcd
        xcentre=0.D0
        ycentre=0.D0
        circxc=0.D0
        circyc-0.D0
        IXX=IX
        IYY=IY
        IZZ=IZ
        do 12 klm=1,2
          slope1=(abcd IYY,2)-abcd(IXX,2 )  abcd IYY,1 -
abcd(IXX,1))

C-----Slope of perp.bi-sector must be negative reciprical
          slope1=-1.D0 slope1

C-----Finding midpoint
          xmid1=(abcd(IYY,1)+abcd(IXX,1)  2.
          ymid1=(abcd IYY,2)+abcd(IXX,2)  2.

C-----Equation is Y-MX+C i.e.can be written as Y-Y1 M X-
X1)
C-----or as Y-MID1=M(X-XMID1)
C-----i.e as Y=(SLOPE1*X -(SLOPE1*XMID1 +YMID1
C-----in form for solving using subroutine eqsolv is:
C-----(SLOPE1*X)-Y-(SLOPE1*XMID1)-YMID1
C-----Finding slope of second line i.e. between 5 and 37
          slope2=(abcd IZZ,2)-abcd(IXX,2    abcd IZZ,1 -
abcd(IXX,1))
          slope2=-1.D0 slope2
          xmid2=(abcd(IZZ,1)+abcd IXX,1)  2.
          ymid2=(abcd(IZZ,2)+abcd IXX,2   2.

C-----Call to equation solving subroutine
          call
eqsolv(slope1,slope2,xmid1,ymid1,xmid2,ymid2,xcentre,
     +ycentre,2)
          circxc=xcentre+circxc
          circyc=ycentre+circyc
          IXX=IXX+3
          IYY=IYY+3
12        IZZ=IZZ+3

        centrx(kl)=circxc/2.
        centry(kl)=circyc/2.

C-----Now taking all inner node points to find average
radius
        radius=0.D0
        IXXX=IX
        do 13 ik=1,6
          radius=radius+dsqrt((centrx kl)-
abcd IXXX,1))**2.+
     +(centry(kl)-abcd IXXX,2))**2.)
13        IXXX=IXXX+3
        radii(kl)=radius 6.
        thickness=0.D0
        JXX=JX
        JYY=JY
        do 14 kll=1,6
          thickness=thickness+dsqrt (abcd(JXX,1)-
abcd(JYY,1))**2.+
     +(abcd JXX,2)-abcd(JYY,2 )**2.)
          JXX=JXX+3
14        JYY=JYY+3
        thick(kl)=thickness/6.
        IX=IX+66
        IY=IY+66
        IZ=IZ+66
        JX=JX+66
        JY=JY+66
        KX=KX+66
        KXX=KXX+66
        KY=KY+66
10      KZ=KZ+66

      do 17 ji=1,12
17      write(*,18) ji,radii(ji)*1000.,ji,thick ji(*100 .

18    format(' Radius ',i2,')=',f6.3,'mms',5x,'Wall
Thickness(',i2,')
     +=',f6.3,'mms')

47    return
      end


      subroutine
eqplan(xcord,ycord,zcord,npoint,a,b,c,I,J,K)
      double precision
xcord npoint),ycord npoint ,zcord npoint ,
     +a(3),b(3),c 3
      integer I,J,K,npoint

      a 1 =xcord I
      b(1)=xcord J)
      c(1)=xcord(K)
      a(2)=ycord(I)
      b(2)=ycord(J)
      c(2)=ycord(K
      a 3)=zcord I)
      b 3 =zcord J
      c 3 =zcord K

      return
      end
      subroutine
maxmin(xcord,ycord,zcord,icount,idiv,volume,npoint)
C-----Finds greatest distance between two points on a
plane
      double precision xcord 12),ycord(12),zcord(12),

+cordyy,distce(151),xlenth 151),distx,disty,the,rolling,s
tones,
     +xmin,xxmax,amin,amax,xmax,cordxx,volume
      integer
icount,idiv,npoint,imin,imax,ik,l,m,loop,n,j,i

      amax=1234.D0
      amin=1234.D0
      xmax=-1234.D0
      xmin=1234.D0
      imin=99
      imax=999

      do 1 loop=1,npoint
        cordxx=xcord(loop)
        cordyy=ycord(loop)
        do 2 m=1,npoint
          distx=(cordxx-xcord(m))**2.
          disty=(cordyy-ycord(m))**2.
2         distce(m)=dsqrt(distx+disty
        xxmax=xmax
        do 3 ik=1,npoint
          if(distce(ik).gt.xxmax) xxmax=distce(ik)
3         xlenth(loop)=xxmax
1     continue

      xxmax-xmax

      do 6 n=1,npoint
6       if(xlenth(n).gt.xxmax) xxmax=xlenth(n)

      do 9 l=1,npoint
        cordxx=xcord(l)
        cordyy=ycord l
        do 7 j=1,npoint
          the=(cordxx-xcord(j))**2.
          rolling=(cordyy-ycord(j  **2.
          stones=dsqrt(the+rolling)
          if(dabs(stones-xxmax).lt.amin) then
            amin=daps(stones-xxmax
            imin=l
          end if
7       continue
9     continue

      cordxx=xcord imin)
      cordyy=ycord(imin)
      do 5 i-1,npoint
        the=(cordxx-xcord i))**2.
        rolling= cordyy-ycord(i))**2.
        stones=dsqrt(the+rolling)
        if(dabs stones-xxmax).lt.amax) then
          amax=dabs stones-xxmax)
          imax=i
        end if
5     continue

      xmin=dmin1(xcord(imin ,xcord(imax))
      xmax=dmax1(xcord(imin),xcord(imax )

      do 8 i=1,npoint
        if(xcord i).eq.xmin) imin=i
8       if(xcord(i).eq.xmax) imax=i

      call
tranrot(xcord,ycord,zcord,imin,imax,icount,idiv,volume,
     +npoint)

      return
      end


      subroutine
tranrot xcord,ycord,zcord,imin,imax,icount,idiv,
     +volume,npoint)
C-----Translates and transforms plane to new origin
      double precision
xcord(npoint),ycord(npoint),zcord(npoint),

+txcord(151),tycord(151),rxcord(151),rycord(151),alpha,vo
lume
      integer icount,idiv,npoint,i,imin,imax

      do 1 i 1,npoint
        txc rd i)=xcord(i)-xcord(imin)
1       tycord(i)=ycord(i)-ycord(imin)

      alpha=-datan2 tycord(imax)-
tycord(imin),txcord(imax)-
     +txcord(imin))

      do 2 i=1,npoint
        rxcord(i)=txcord(i)*dcos alpha)+tycord(i)*-
dsin alpha)
2
rycord i)=txcord(i)*dsin(alpha +tycord(i)*dcos(alpha)

      call
area rxcord,rycord,zcord,imin,imax,icount,idiv,volume,
     +npoint)
```

```
      return
      end


      subroutine
area(rxcord,rycord,zcord,imin,imax,i-ount,idiv
     +volume,npoint
      double precision
rxcord(npoint),rycord npoint),zc rd np-int ,
     +volume,avheight(1300 ,cxarea(13 0 ,volone,voltwo
     +radius(13 ,angle 13 ,error,xarea,xmid ymid height
      integer npoint,idiv,icount,i,imin,imax iimax,ifai

C-----Use NAG routine d01gaf to evaluate integral between
points and
C-----hence work out area using polar co-ordinates
C-----Taking minimum and maximum x co-ordinates and
finding halfway
C     point gives a centre for origin of axes
      xmid=(rxcord(imin +rxcord imax   2.
      ymid=(rycord imin +rycord imax   2.

C-----Evaluating radiis-the Y integral
      iimax=imax

      do 1 i=1,npoint
        if iimax.lt.1) iimax-npoint
        radius(i)-( rxcord(iimax -
xmid)**2.+ rycord(iimax)-ymid)**2.
     +*0.5
1       iimax=iimax-1

C-----Last radius same as first
      radius(npoint+1)=radius(1)

C-----Correspondingingly for angles as well
      iimax=imax

      do 2 i=1,npoint
        if(iimax.lt.1) iimax=npoint
        if i.eq.1  angle(i =0.D0
        if iimax.eq.imin  angle i =3.141592653
        if(i.eq.1) goto 2

if(rxcord(iimax).gt.xmid.and.rycord(iimax .gt.ymid)
     +    angle i =datan (rycord iimax)-
ymid)/(rxcord(iimax)-xmid))

if(rxcord(iimax).lt.xmid.and.rycord iimax).gt.ymid)
     +    angle(i)=3.141592653-dabs(datan (ry-ord iimax -
ymid)/
     +(rxcord(iimax)-xmid) )

if(rxcord(iimax).lt.xmid.and.rycord iimax .lt.ymid)
     +    angle(i)=3.141592653+dabs datan (ryc r  i max)-
ymid)/
     +(rxcord iimax)-xmid

if(rxcord iimax .gt.xmid.and.rycord(iimax .lt.ymid)
     +    angle i =6.283185306-dabs(datan( rycord iimax -
ymid)/
     +(rxcord(iimax)-xmid)))
2       iimax=iimax-1

C-----2*PIE for last one
      angle(npoint+1)=6.283185306

C-----Call to Gill and MIller's method
      ifail-1
      call
d01gaf(angle,radius,npoint+1,xarea,error,ifail)

C-----Calculating average height
      height=0.D0

      do 3 i=1,npoint
3       height=height+zcord(i)

      height=height/float(npoint)
      cxarea icount)=xarea
      avheight icount)=height

      if(icount.eq.1  volume=0.D0
      if(icount.eq.1  goto 2

      volone=cxarea(icount-1   avheight(icount-1 -
avheight(icount))
      voltwo=cxarea(icount *(avheight(icount-1 -
avheight icount)
      volume=volume+dabs(volone+voltwo  2.

25    return
      end



      subroutine rearrange(I,J,K,L,CX,CY,CZ,array)
      double precision
CX(109),CY 109 ,CZ 109),array 4 1,3)
      integer I,J,K,L,II,m

      goto (1,- , L
1     II=I

      o  10 m=J,K,2
        CX m  array II,1
        CY m -array II,-)
        CZ(m)=array(II,3)
10      II=II+8

      goto 12
2     II=I

      do 11 m=J,K
        CX(m)=array(II,1)
        CY(m)=array(II,2)
        CZ(m)=array(II,3)
11      II=II+3

12    continue

      return
      end


      subroutine
subdiv(ia,ib,ic,id,ie,if,ig,ih,ii,ij,ik,il,im,in,io,

+ip,iq,ir,is,it,iu,iv,iw,ix,CX,CY,CZ,idiv,incr,icount,klm
,volume,
     +iresp,CCX,CCY,CCZ)
C-----Defines co-ordinates on planes after subdivision
      double precision
CX(109),CY(109),CZ(109),xcord(12),a(3),b(3),

+c(3),ycord(12),zcord(12),volume,al1,am1,an1,al2,am2,an2,
al3,am3,

+an3,tran(3,3),AA,BB,CC,DD,CCX(109),CCY(109),CCZ(109),epi
x(12),
     +epiy(12),epiz(12),looop,divi,fincr
      integer
ia,ib,ic,id,ie,if,ig,ih,ii,ij,ik,il,npoint,loop,

+im,in,io,ip,iq,ir,is,it,iu,iv,iw,ix,idiv,icount,klm,ires
p,incr

       if klm.eq.2) idiv=1
       fincr=dble(float(incr))
       divi=dble(float(idiv))

       do 1 loop=1,idiv
         looop=dble(float(loop))
         if(klm.eq.2) looop=0.
         icount=icount+1
         if(icount.eq.1) goto 3
         if(iresp.eq.2) goto 3
         if(icount.eq.2*idiv+1.and.iresp.eq.1) goto 3
         if(icount.eq.4*idiv+1.and.iresp.eq.1) goto 3
         if(icount.eq.6*idiv+1.and.iresp.eq.1) goto 3
         if(icount.eq.8*idiv+1.and.iresp.eq.1) goto 3
         if(icount.eq.10*idiv+1.and.iresp.eq.1) goto 3
         xcord(1)=CX(ia)+((CX(ib)-CX ia))*looop)/divi
         epix(1)=CCX(ia)+((CCX(ib)-CCX(ia )*looop)/divi
         ycord(1)=CY(ia)+( CY(ib)-CY(ia))*looop)/divi
         epiy(1)=CCY(ia)+((CCY(ib)-CCY(ia) *looop)/divi
         zcord(1)=CZ(ia)+((CZ(ib)-CZ(ia))*looop)/divi
         epiz(1)=CCZ(ia)+((CCZ(ib)-CCZ(ia) *looop)/divi
         xcord(2)=CX(ic)+((CX(id)-CX(ic))*looop /divi
         epix(2)=CCX(ic)+( CCX(id)-CCX(ic))*looop)/divi
         ycord 2)=CY(ic)+((CY(id)-CY ic))*looop /divi
         epiy(2)=CCY(ic)+((CCY(id)-CCY ic))*looop)/divi
         zcord(2)=CZ ic)+((CZ(id)-CZ(ic))*looop)/divi
         epiz(2)=CCZ(ic)+((CCZ(id)-CCZ(ic))*looop)/divi
         xcord(3)=CX(ie)+((CX(if)-CX ie))*looop)/divi
         epix(3)=CCX(ie)+((CCX(if)-CCX ie))*looop)/divi
         ycord(3)=CY(ie)+((CY(if)-CY(ie))*looop)/divi
         epiy(3)=CCY(ie)+((CCY(if)-CCY(ie))*looop)/divi
         zcord(3)=CZ(ie)+((CZ(if)-CZ(ie))*looop)/divi
         epiz(3)=CCZ(ie)+((CCZ(if)-CCZ(ie))*looop)/divi
         xcord(4)=CX(ig)+((CX(ih)-CX(ig))*looop)/divi
         epix(4)=CCX(ig)+((CCX(ih)-CCX(ig))*looop)/divi
         ycord(4)=CY(ig)+((CY(ih)-CY(ig))*looop)/divi
         epiy 4)=CCY(ig)+((CCY(ih)-CCY(ig))*looop)/divi
         zcord(4)=CZ(ig)+((CZ(ih)-CZ(ig))*looop)/divi
         epiz(4)=CCZ(ig)+((CCZ(ih)-CCZ(ig))*looop)/divi
         xcord(5)=CX(ii)+((CX(ij)-CX(ii))*looop)/divi
         epix(5)=CCX(ii)+((CCX(ij)-CCX(ii))*looop)/divi
         ycord(5)=CY(ii)+((CY(ij)-CY(ii))*looop)/divi
         epiy(5)=CCY(ii)+((CCY(ij)-CCY(ii))*looop)/divi
         zcord(5)=CZ ii)+((CZ(ij -CZ(ii))*looop)/divi
         epiz(5)=CCZ ii)+((CCZ(ij)-CCZ(ii))*looop)/divi
         xcord(6)=CX(ik)+((CX(il)-CX ik))*looop)/divi
         epix(6)=CCX(ik)+( CCX(il)-CCX(ik))*looop)/divi
         ycord(6)=CY(ik)+((CY(il)-CY(ik))*looop)/divi
         epiy 6)=CCY(ik)+((CCY(il)-CCY(ik))*looop)/divi
         zcor  6)=CZ(ik)+((CZ(il)-CZ(ik))*looop)/divi
         epi  6 =CCZ(ik)+((CCZ(il)-CCZ(ik))*looop)/divi
         npoint=6
         if icount.eq.12*idiv+1) goto 2
         call eqplan(epix,epiy,epiz,npoint,a,b,c,1,3,5)
         call roots(a,b,c,AA,BB,CC,DD)
         call dcos1(AA,BB,CC,DD,al3,am3,an3)
         call dcos2(1,4,epix,epiy,epiz,al1,am1,an1)
         call dcos3(al3,am3,an3,al1,am1,an1,al2,am2,an2)
         call
 trans(al3,am3,an3,al1,am1,an1,al2,am2,an2,tran)
         call matmly(npoint,xcord,ycord,zcord,tran)
         goto 5
 3       xcord 1)=CX(ia)+((CX(ib -CX(ia))*looop)/divi
         epix 1)-CCX(ia)+((CCX(ib -CCX(ia  *looop) divi
         ycord 1)=CY(ia)+((CY(ib)-CY(ia )*looop)/divi
         epiy 1 =ᒥCY(ia)+((CCY(ib -CCY ia))*looop) divi
         zcord 1)-CZ(ia)+((CZ(ib -CZ(ia))*looop)/divi
```

```
         epiz(1)=CCZ ia +  CCZ ib -CCZ ia )*looop  divi
         xcord(3 =CX ic)+  CX(id -CX ic  *loo p  divi
         epix(3)=CCX ic +  CCX id -CCX ic  *looop  divi
         ycord(3)=CY ic)+  CY(id -CY ic  *looop  divi
         epiy(3)=CCY ic)+  CCY id)-CCY ic  *looop  divi
         zcord(3)-CZ ic +  CZ(id -CZ ic  *looop  divi
         epiz(3)=CCZ ic)+  CCZ(id)-CCZ ic  *looop  divi
         xcord(5 =CX(ie +  CX if)-CX ie  *loo p  divi
         epix 5)=CCX ie + (CCX if)-CCX ie  *loo p  divi
         ycord 5 =CY ie + (CY if)-CY ie  *looop  divi
         epiy(5)=CCY ie +  CCY(if)-CCY ie  *looop  divi
         zcord 5 =CZ ie)+  CZ if)-CZ ie  *looop  divi
         epiz 5 =CCZ ie +  CCZ(if)-CCZ ie  *looop  divi
         xcord 7)=CX ig +  CX ih -CX ig  *looop  divi
         epix 7 =CCX ig +  CCX ih)-CCX ig  *l oop  divi
         ycord 7)=CY ig +( CY(ih -CY(ig) *looop  divi
         epiy 7 =CCY ig +  CCY ih)-CCY ig  *looop  divi
         zcord 7 -CZ ig +( CZ ih -CZ(ig) *looop  divi
         epiz(7 =CCZ ig +((CCZ(ih)-CCZ ig )*looop  divi
         xcord(9 =CX ii +( CX ij -CX ii  *lo op  divi
         epix(9 =CCX ii +  CCX(ij)-CCX ii))*looop  divi
         ycord 9)=CY(ii)+ (CY(ij -CY(ii  *looop) divi
         epiy 9 =CCY(ii)+((CCY(ij -CCY(ii )*looop  divi
         zcord(9)=CZ(ii)+((CZ ij)-CZ ii))*looop  divi
         epiz(9)=CCZ(ii)+  CCZ ij)-CCZ(ii))*looop) divi
         xcord(11)=CX(ik)+( CX il)-CX(ik))*looop) divi
         epix(11)=CCX ik)+  CCX il)-CCX ik))*looop) divi
         ycord(11)=CY(ik)+ (CY il)-CY(ik))*looop)/divi
         epiy(11)=CCY(ik)+( CCY il)-CCY(ik))*looop) divi
         zcord(11)=CZ ik)+( CZ il -CZ(ik))*l oop) divi
         epiz(11)=CCZ(ik)+( CCZ il)-CCZ(ik))*looop) divi
         xcord 2)=CX im)+  CX in -
CX im))*(looop+fincr   (divi*2.)
         epix(2)=CCX im)+( CCX(in -
CCX(im))*(looop+fincr )/(divi*2.)
         ycord(2)=CY im +((CY(in)-
CY(im))*(looop+fincr))/(divi*2.)
         epiy(2)=CCY(im)+((CCY(in)-
CCY(im))*(looop+fincr))/(divi*2.)
         zcord(2)=CZ(im)+((CZ(in)-
CZ(im))*(looop+fincr))/ divi*2.)
         epiz(2)=CCZ(im +((CCZ(in)-
CCZ(im))*(looop+fincr))/(divi*2.)
         xcord(4)=CX(io)+((CX(ip)-
CX(io))*(looop+fincr))/ divi*2.)
         epix(4)=CCX(io)+((CCX(ip)-
CCX(io))*(looop+fincr))/(divi*2.)
         ycord(4)=CY(io)+((CY(ip)-
CY(io))*(looop+fincr))/(divi*2.)
         epiy(4)=CCY io)+((CCY(ip)-
CCY(io))*(looop+fincr))/(divi*2.)
         zcord(4)=CZ io)+((CZ(ip)-
CZ io))*(looop+fincr)) (divi*2.)
         epiz(4)=CCZ(io)+((CCZ(ip)-
CCZ(io))*(looop+fincr))/ divi*2.)
         xcord(6)=CX(iq)+((CX(ir)-
CX(iq))* looop+fincr))/   ' i*2.)
         epix(6)=CCX(iq)+( CCX(ir)-
CCX iq))*(looop+fincr))/(divi*2.)
         ycord(6)=CY(iq)+((CY(ir)-
CY(iq))*(looop+fincr))/(divi*2.)
         epiy(6)=CCY(iq)+((CCY(ir)-
CCY(iq))*(looop+fincr))/(divi*2.)
         zcord(6)=CZ(iq)+((CZ(ir)-
CZ(iq))*(looop+fincr))/(divi*2.)
         epiz(6)=CCZ(iq)+((CCZ(ir)-
CCZ(iq))*(looop+fincr))/(divi*2.)
         xcord(8)=CX(is)+((CX(it)-
CX(is))*(looop+fincr))/(divi*2.)
         epix(8)=CCX(is)+((CCX(it)-
CCX(is))*(looop+fincr))/(divi*2.)
         ycord(8)=CY(is)+((CY(it)-
CY(is))*(looop+fincr))/(divi*2.)
         epiy(8)=CCY(is)+((CCY(it)-
CCY(is))*(looop+fincr  /(divi*2.)
         zcord(8)=CZ(is)+((CZ(it)-
CZ(is))*(looop+fincr))/(divi*2.)
         epiz(8)=CCZ(is +((CCZ(it)-
CCZ(is))*(looop+fincr))/(divi*2.)
         xcord(10)=CX(iu)+((CX iv)-
CX(iu))*(looop+fincr))/ divi*2.
         epix(10)=CCX(iu)+((CCX(iv -
CCX(iu))*(looop+fincr )/(divi*2.)
         ycord 10)=CY iu +  C/  v)-
CY(iu))*(looop+fincr))  divi
         epiy(10)=CCY iu)+((C   iv -
CCY(iu))*(looop+fincr))/ divi*2.)
         zcord(10)=CZ(iu +( CZ iv)-
CZ(iu))*(looop+fincr))/(divi*2.
         epiz(10)=CCZ(iu)+( CCZ(iv)-
CCZ iu))* looop+fincr )/(divi*2.)
         xcord(12 =CX iw)+  CX ix -
CX(iw))*(looop+fincr))/ divi*2.)
         epix(12)=CCX iw)+((CCX ix)-
CCX(iw))*(looop+fincr   (divi*2.)
         ycord(12)=CY(iw +( CY(ix)-
CY(iw))*(looop+fincr)   divi*2.)
         epiy 12)=CCY(iw +((CCY(ix -
CCY(iw))*(looop+fincr) /(divi*2.)
         zcord(12)=CZ iw +( CZ(ix)-
CZ iw))*(looop+fincr    divi*2.)
         epiz 12 =CCZ(iw +  CCZ ix)-
CCZ(iw) *(looop+fincr    divi*2.
         npoint=12

         if(icount.eq.1_*idiv+1  g t  2
         call eqplan epix,epiy,epi-,np int,a,b,c,1, ,9
```

```
        call roots(a,b,c,AA,BB,CC,DD)
        call dcos1(AA,BB,CC,DD,al3,am3,an3)
        call dcos2(1,7,epix,epiy,epiz,al1,am1,an1)
        call dcos3(al3,am3,an3,al1,am1,an1,al2,am2,an2)
        call
trans(al3,am3,an3,al1,am1,an1,al2,am2,an2,tran)
        call matmly(npoint,xcord,ycord,zcord,tran)
5       call
maxmin(xcord,ycord,zcord,icount,idiv,volume,npoint)
1     continue

2     return
      end



      subroutine roots(a,b,c,AA,BB,CC,DD)
C-----Solving for equation of plane
      double precision a(3),b(3),c 3),AA,BB,CC,DD

      AA=a(2)*b(3)-a(2)*c(3)-
a(3)*b(2)+a(3)*c(2)+b(2)*c(3)-c(2)*b(3)
      BB=a(1)*b(3)-a(1)*c(3)-
a(3)*b(1)+a(3)*c(1)+b(1)*c(3)-c(1)*b(3)
      BB=BB*-1.D0
      CC=a(1)*b(2)-a(1)*c(2)-
a(2)*b(1)+a(2)*c(1)+b(1)*c(2)-c(1)*b(2)
      DD=a(1)*b(2)*c(3)-a(1)*c(2)*b(3)-
a(2)*b(1)*c(3)+a(2)*c(1)*b(3)
     ++a(3)*b(1)*c(2)-a(3)*c(1)*b(2)
      DD=DD*-1.D0

      return
      end


      subroutine dcos1(AA,BB,CC,DD,al3,am3,an3)
      double precision AA,BB,CC,DD,al3,am3,an3,hypot1

      hypot1=dsqrt(AA*AA+BB*BB+CC*CC)
      al3=AA/hypot1
      am3=BB/hypot1
      an3=CC/hypot1

      return
      end


      subroutine
dcos2(II,JJ,xcord,ycord,zcord,al1,am1,an1)
      double precision xcord(12),ycord(12),zcord(12),
     +al1,am1,an1,hypot2,x,y,z
      integer II,JJ

      x=xcord(JJ)-xcord(II)
      y=ycord(JJ)-ycord(II)
      z=zcord(JJ)-zcord(II)
      hypot2=dsqrt(x*x+y*y+z*z)
      al1=x/hypot2
      am1=y/hypot2
      an1=z/hypot2

      return
      end


      subroutine
dcos3(al3,am3,an3,al1,am1,an1,al2,am2,an2)
      double precision
al3,am3,an3,al1,am1,an1,al2,am2,an2

      al2=am1*an3-am3*an1
      am2=-(al1*an3-al3*an1)
      an2=al1*am3-al3*am1

      return
      end


      subroutine matmly(npoint,xcord,ycord,zcord,tran)
      double precision xcord(12),ycord(12),zcord(12),
     +tran(3,3),cordx,cordy,cordz
      integer npoint,i

      do 1 i=1,npoint

cordx=tran(1,1)*xcord(i)+tran(1,2 *ycord(i)+tran(1,3)*zco
rd(i)

cordy=tran(2,1)*xcord(i)+tran(2,2)*ycord(i)+tran 2,3)*zco
rd(i)

cordz=tran(3,1)*xcord(i)+tran(3,2)*ycord(i)+tran 3,3)*zco
rd(i)
        xcord(i)=cordx
        ycord(i)=cordy
1       zcord(i)=cordz

      return
      end


      subroutine dmmlt1(i,j,array,tran,wxyz)
      double precision
array(401,3),tran 3,3),wxyz 401,3 ,x,y,z
      integer i,j,k
```

```
      do 1 k=i,j

x=tran(1,1)*array(k,1)+tran(1,2)*array(k,2)+tran(1,3)*
     +array(k,3)

y=tran(2,1)*array(k,1 +tran(2,2)*array(k,2)+tran(2,3)*
     +array(k,3)

z=tran(3,1)*array(k,1 +tran(3,2)*array(k,2)+tran(3,3)*
     +array(k,3)
        wxyz(k,1)=x
        wxyz(k,2)=y
1       wxyz(k,3)=z

      return
      end



      subroutine secos2(i,j,xnod,al1,am1,an1)
      double precision
xnod(401,3),al1,am1,an1,x,y,z,hypot2
      integer i,j

      x=xnod(j,1)-xnod(i,1)
      y=xnod(j,2)-xnod(i,2)
      z=xnod(j,3)-xnod(i,3)
      hypot2=dsqrt(x*x+y*y+z*z)
      al1=x/hypot2
      am1=y/hypot2
      an1=z/hypot2

      return
      end



      subroutine
trans(al3,am3,an3,al1,am1,an1,al2,am2,an2,tran)
      double precision
al3,am3,an3,al1,am1,an1,al2,am2,an2,tran(3,3)

      tran(1,1)=al1
      tran(1,2)=am1
      tran(1,3)=an1
      tran(2,1)=al2
      tran(2,2)=am2
      tran(2,3)=an2
      tran(3,1)=al3
      tran(3,2)=am3
      tran(3,3)=an3

      return
      end


      subroutine
eqsolv(slope1,slope2,xmid1,ymid1,xmid2,ymid2,xcentre,
     +ycentre,n)
      double precision
a(2,2),b(2),x(2),c(2,3),slope1,slope2,xmid1,
     +ymid1,xmid2,ymid2,xcentre,ycentre,t,amax,p
      integer n,i,j,ir,ic,ii,k

C-----Program to solve simultaneous equations and find
centre of base
      a(1,1)=slope1
      a(1,2)=-1.D0
      b(1)=slope1*xmid1-ymid1
      a(2,1)=slope2
      a(2,2)=-1.D0
      b(2)=slope2*xmid2-ymid2
7      format(4x,2hx(,i2,2h)=,1pe16.7/)

      do 10 i=1,n
10      c(i,1)=0.D0

      ii=0
11    amax=-1.D0

      do 16 i=1,n
        if(c(i,1)) 16,12,16
12      do 15 j=1,n
          if(c(j,1)) 15,13,15
13        t=dabs(a(i,j))
          if(t-amax  15,15,14
14        ir=i
          ic=j
          amax=t
15      continue
16    continue

      if(amax) 27,32,17
17    c ic,1)=ir
      if ir-ic)18,20,18

18    do 19 j=1,n
        t=a(ir,j
        a(ir,j)=a ic,j)
19      a(ic,j)=t

      ii=ii+1
      c ii,2 =ic
20    p=a ic,ic)
      a ic,ic =1.D0
      p=1.D0 p
      do 21 j=1,n
21      a(ic,j)=a ic,j)*p

      do 24 i=1,n
        if(i-ic) 22,24,22
22      t=a(i,ic)
        a(i,ic)=0.D0
        do 23 j=1,n
23        a(i,j)=a(i,j)-a(ic,j)*t
24    continue

      goto 11
25    ic=c(ii,2)
      ir=c(ic,1)

      do 26 i=1,n
        t=a(i,ir)
        a(i,ir)=a(i,ic)
26      a(i,ic)=t

      ii=ii-1
27    if(ii) 25,28,25

28    do 29 i=1,n
        x(i)=0.D0
        do 29 k=1,n
29        x(i)=x(i)+a(i,k)*b(k)

C-----x(1)=xcentre and x(2)=ycentre
      xcentre=x(1)
      ycentre=x(2)
      goto 34
32    write(6,33)
33    format(////' ERROR!!. EQUATIONS CANNOT BE SOLVED')

34    return
      end


      subroutine moveshp(nn,rixn,riyn,np,rixp,riyp)
      double precision
 rixn 151),riyn(151),rixp(151 ,riyp(151),
     + zapex,raox,raoy,dx,dy,dd
      integer nn,np,i,mapex

C-----Calculates position of apex as being furthest point
C-----from mid ao root and moves origin to apex
C-----in (x) and lowest point in (y)
C-----last changed 19-04-94

      zapex=0.

C-----Calc mid ao root
      raox= rixn(1)+rixn(nn)) 2.
      raoy=(riyn(1)+riyn(nn)) 2.

C-----Find apex
      do 20 i=1,nn
        dx=rixn(i)-raox
        dy=riyn(i)-raoy
        dd=dx*dx+dy*dy
        if(dd.gt.zapex) then
          zapex=dd
          mapex=i
          endif
20    continue

C-----Redefine coords
      dx=rixn(mapex)

      dy=riyn(1)
      do 30 i=2,nn
        if(riyn(i).lt.dy) dy=riyn(i)
30    continue

C-----Endo
      do 40 i=1,nn
        rixn(i)=rixn(i)-dx
40      riyn(i)=riyn(i)-dy

C-----Epi
      do 50 i=1,np
        rixp(i)=rixp(i)-dx
50      riyp(i)=riyp(i)-dy

      return
      end


      subrou_ine slicenv(env,envp,geom,n1,n2,n3,n4,n1p,
     +n2p,n3p,n4p)
      double precision
 vy,vz,by,bz,cy,cz,uy,uz,xxx,bottx,tenv(4,2),
     +env(50,4,3),envp(50,4,3 ,geom 3,255 ,slen,aly
      integer i,il,j,jd,n1,n2,n3,n4,n1p,n2p,n3p,n4p

C-----Calculates the coordinates for the 3d shape
C-----Last changed 14-05-96

C-----Find lowest point of top ends

      xxx=env(n1,1,1)
      if(env n2,2,1).lt.xxx) xxx=env(n2,2,1)
      if(env(n3,3,1).lt.xxx) xxx=env(n3,3,1)
      if env n4,4,1 .lt.xxx) xxx=env(n4,4,1'
      if(envp(n1p,1,1).lt.xxx  xxx-envp n1p,1,1)
```

```
      if envp n2p,2,1 .lt.xxx  xxx=envp n_p,2,1
      if(envp n3p,3,1 .lt.xxx  xxx=envp(n3p,3,1
      if(envp(n4p,4,1 .lt.xxx  xxx=envp n4p,4,1

C-----Calc constants
      bottx=-1234.

      do 5002 i=1,4
        if env(1,i,1 .gt.bottx) bottx=env 1,i,1
5002    if(envp 1,i,1 .gt.bottx) bottx=envp 1,i,1

C-----Calc slice thickness
c      slen=( xxx)- bottx+.002   11.
      slen=((xxx -(bottx+.004   10.


C-----Loop for each slice
      il=0
      do 10 i=2,12
        if i.eq.3  il=( i-1  2 *42
        if(i.eq.5) il=( i-1 /2 *42
        if(i.eq.7  il=( i-1  2)*42
        if(i.eq.9  il=( i-1) 2)*42
        if(i.eq.11) il=  i-1  2 *42
        aly=(xxx)- dble(float i-1  *slen
c        if(i.eq.12) aly=.001+bottx

        if i.eq.11) aly=0.004+bottx
        if(i.eq.12) aly=0.0005+bottx

C-----Endocardium
        call boxsiz env,tenv,aly

        if((i.ne.2 .and.(i.ne.4).and.(i.ne.6 .and.
     +(i.ne.8 .and. i.ne.10).and.(i.ne.12   then

          do 20 j=1,30
            jd=j+il
20          geom(1,jd)=aly

          call
 drawarc 2,1,3,tenv,vy,vz,by,bz,cy,cz,uy,uz
          geom(2,il+3)=vy
          geom(3,il+3 =vz
          geom(2,il+5)=by
          geom(3,il+5)=bz
          geom 2,il+8 =cy
          geom 3,il+8)=cz
          geom 2,il+10)=uy
          geom 3,il+10 =uz
          call
 drawarc(1,3,4,tenv,vy,vz,by,bz,cy,cz,uy,uz
          geom(2,il+13 =by
          geom(3,il+13 =bz
          geom(2,il+15)=cy
          geom(3,il+15 =cz
          geom(2,il+18  uy
          geom(3,il+18 =uz
          call
 drawarc 3,4,2,tenv,vy,vz,by,bz,cy,cz,uy,uz
          geom 2,il+20 =by
          geom 3,il+20 =bz
          geom 2,il+23 =cy
          geom(3,il+23)=cz
          geom(2,il+25)=uy
          geom(3,il+25 =uz
          call
 drawarc(4,2,1,tenv,vy,vz,by,bz,cy,cz,uy,uz)
          geom(2,il+28)=by
          geom(3,il+28)=bz
          geom(2,il+30)=cy
          geom(3,il+30)=cz
          else
          do 30 j=31,42
            jd=j+il
30          geom 1,jd =aly
          call
 drawarc(2,1,3,tenv,vy,vz,by,bz,cy,cz,uy,uz
          geom(2,il+32)=vy
          geom(3,il+32 =vz
          geom(2,il+34 =cy
          geom(3,il+34)=cz
          call
 drawarc(1,3,4,tenv,vy,vz,by,b  cy,cz,uy,uz)
          geom(2,il+36)=by
          geom(3,il+36)=bz
          geom 2,il-38 =uy
          geom(3,il+38)=uz
          call
 drawarc(3,4,2,tenv,vy,vz,by,bz,cy,cz,uy,uz
          geom 2,il+40 =cy
          geom(3,il+40 =cz
          call
 drawarc 4,2,1,tenv,vy,vz,by,bz,cy,cz,uy,uz
          geom 2,il-42 =by
          geom(3,il+42 =bz
          endif

C-----Epicardium
        call boxsiz envp,tenv,aly


 if  i.ne.2 .and. i.ne.4 .and. i.ne.6 .and. i.ne.
     +.and. i.ne.1  .and. i.ne.1  ) then
          il-il-2
          call
 drawarc  ,1,3,tenv,vy,vz,by,bz,cy,cz uy,uz
```

```
          geom(2,il+3)=vy
          geom(3,il+3)=vz
          geom(2,il+6)=by
          geom(3,il+6)=bz
          geom 2,il+8 =cy
          geom(3,il+8)=cz
          geom(2,il+11)=uy
          geom(3,il+11)=uz
          call
drawarc(1,3,4,tenv,vy,vz,by,bz,cy,cz,uy,uz
          geom(2,il+13)=by
          geom(3,il+13)=bz
          geom(2,il+16 =cy
          geom(3,il+16)-cz
          geom(2,il+18)=uy
          geom(3,il+18)=uz
          call
drawarc(3,4,2,tenv,vy,vz,by,bz,cy,cz,uy,uz)
          geom 2,il+21)=by
          geom(3,il+21)=bz
          geom(2,il+23)=cy
          geom(3,il+23)=cz
          geom 2,il+26)=uy
          geom(3,il+26)=uz
          call
drawarc(4,2,1,tenv,vy,vz,by,bz,cy,cz,uy,uz)
          geom(2,il+28)=by
          geom(3,il+28)=bz
          geom(2,il+31)=cy
          geom 3,il+31)=cz
          il=il+2
          else
          il=il-1
          call
drawarc(2,1,3,tenv,vy,vz,by,bz,cy,cz,uy,uz)
          geom(2,il+32)=vy
          geom(3,il+32)=vz
          geom(2,il+34)=cy
          geom(3,il+34)=cz
          call
drawarc(1,3,4,tenv,vy,vz,by,bz,cy,cz,uy,uz)
          geom 2,il+36)=by
          geom 3,il+36)=bz
          geom 2,il+38 =uy
          geom 3,il+38)=uz
          call
drawarc(3,4,2,tenv,vy,vz,by,bz,cy,cz,uy,uz)
          geom 2,il+40)=cy
          geom 3,il+40)=cz
          call
drawarc(4,2,1,tenv,vy,vz,by,bz,cy,cz,uy,uz)
          geom 2,il+42)=by
          geom 3,il+42)=bz
          il=il+1
          endif
10    continue


geom(1,255 = env 1,1,1)+env(1,2,1 +env 1,3,1 +env 1,4,1 )
/4.
      geom(1,253 =geom(1,255)-.002
      geom 1,254 =(geom(1,255 +geom(1,253)  2.

geom(2,255 = env 1,1,2)+env 1,2,2)+env 1,3,2)+env(1,4,2
/4.
      geom 2,253 =geom(2,255)
      geom(2,254 =geom(2,255)

geom(3,255)=(env 1,1,3)+env 1,2,3 +env(1,3,3 +env 1,4,3)
/4.
      geom(3,253)=geom(3,255)
      geom(3,254)=geom(3,255)

      return
      end


      subroutine
graphs(preses,volu,volum,nf,n,patr,namer,comtr)
      real
preses(nf),volu(nf),volum(nf),anf,time 151),vmax,
     +vvmax,pmax,volmax,mprs
      integer n,isum,isumm,i,nf,
     +icount
      character patr*6,namer*20,comtr*20

      mprs=0.
      do 5 i=1,nf
         if(preses(i).lt.mprs) mprs=preses(i)
5     contin e
      if(mp s.lt.0.) mprs=mprs-1.

      isum=0
      isumm=0
      do 20 i=1,nf
        if(volu i).eq.0.) isum=isum+1
20      if(volum(i).eq.0.) isumm=isumm+1


if((isum.gt.0.and.(n.eq.1.or.n.eq.4.or.n.eq.3.or.n.eq.6.o
r.

+n.eq.7.or.n.eq.9)).or.(isumm.gt.0.and. n.eq.-.or.n.eq.5.
or.
     +n.eq.3.or.n.eq.6. r.n.eq.8  ) then
        write 6,-3)
-3    f rmat(   /
```

```
     +'*Too few points : graph not plotted : check ? -me
data*'/
     +'--------------------------------------------------
------'//)
      else

        if (n.ge.7).and.(n.le.9 ) then
          call
smooth(volu,volum,nf,n,patr,namer,comtr,preses
        else

          anf-float nr)

          call initsp
          call page 18.,24.7
          call picsiz 16.,22.7)

          vmax=-12345678.
          vvmax=vmax
          pmax=vvmax

          do 7 i=1,nf
7           if volu(i .gt.vmax  vmax=volu i

          do 8 i=1,nf
8           if volum i).gt.vvmax) vvmax=volum i

          volmax=max vmax,vvmax)

          do 11 i=1,nf
11          if(preses i).gt.pmax) pmax=preses i

          icount=0
          do 9 i=1,nf
            time i)=icount*0.02
9           icount=icount+1

          if( n.lt.4 .or.(n.gt.6)) then
            call pen 1)
            if(n.eq.1  call
scales(0.,anf*0.02+0.1,1,0.,vmax,1)
            ir n.eq.2  call
scales(0.,anf*0.02+0.01,1,0.,vvmax,1
            if n.eq.3) call
scales 0.,anf*0.02+0.1,1,0.,volmax,1

            call newpic
            call xaxis(0.,anf*0.02,0.,'TIME
(SECONDS ',14

            if(n.eq.1) call yaxis(0.,vmax,0.,
     +'BI-PLANE VOLUME (cm3) (calc.from VOLUME ',4
            if(n.eq.2) call yaxis(0.,vvmax, .
     +'BI-PLANE VOLUME (cm3) (calc from FE Me h ',41
            if n.eq.3) call yaxis 0.,volmax, .
     +'BI-PLANE VOLUMES (cm3) (from VOLUME & FE
Mesh)',46)

            call cvtype(2)

            if n.eq.1) then
              call brkncv time,volu,nf,0
            else if n.eq.2) then
              call pen(2)
              call drawcv(time,volum,nf)
            else if(n.eq.3) then
              call pen(1)
              call drawcv(time,volu,nf)
              call pen(2)
              call drawcv time,volum,nf)
            end if

            call pen(1)

            do 10 i=1,nf,2
              if(n.eq.1  call
numbpt(time(i),volu(i),11,i)
              if(n.eq.2) call pen(2)
10            if(n.eq.2) call
numbpt(time i),volum(i),11,i)

            call pen(3)
            call yscale mprs,pmax,1
            call yaxis mprs,p  ,anf*0.02,'BI-PLANE
PRESSURE (mmHg)',24
            call cvtype(2
            call brkncv(time,preses,nf,-1)

            do 12 i=1,nf,2
12            call numbpt time(i ,preses i),11,i

            call endplt
          endif

          call pen 1

          ir n.eq.4) call scales 0.,vmax,1,mpr ,pmax,1
          if n.eq.5  call scales 0.,vvmax,1,mprs,pmax,1
          if n.eq.6) -all scales(0.,v lmax,1,mprs,pmax,1
          if n.eq.4  -all axes7 'VOLUME   m3 ','PPESSUPE
(mmHg)'
          ir n.eq.5  call axe 7 'V L'ME  cm   fr m FE
Mesh',
     +'PPESSURE  mmHg '
          ir n.eq.6   a-l axe   'V LUMES  cm   fr m
VOLUME & FE Mesn'
     +,'PRESSUPE  mmHg '
```

```
         call cvtype 2

         if n.eq.5  call pen 2
         if(n.eq.6  call pen 3
         if n.eq.4.or.n.eq.6  cal1
drawcv(volu,preses,nf
         if(n.eq.5.or.n.eq.6  call
drawcv(volum,preses,nf

         call pen 1

         if n.eq.5  call pen 2
         if n.eq.6) call pen 3

         do 14 i=1,nf,2
           if n.eq.4( call numb
pt(volu(i ,preses i),11,i
14         if n.eq.5) call numb
pt(volum i),preses i ,11,i

         call endplt

         endif
      endif

      return
      end


      subroutine
smooth volu,volum,m,ij,patr,namer,comtr,preses)
C-----Smoothing through of digitised volumes and
reconstruction of new
C-----targeted volumes using splines
      double precision
ax(151),ay 151),w 151),c 200),work1 151 ,

+ak 200),ifail,ss,flr1,xxx,flm1,a,b,s 151 ,work2(4,200
      double precision press 151),opress
      real
xx(151),yy 151),pp(151),axx(151),ayy 151),volu(151),

+volum 151),tarvol(50),volnex,volume(151 ,preses(151)
     +,apress 151 ,volmax,volmin,volumax,volumin
      integer
iflag,m,ij,ivalue,i,ik,l,ll,nlm,n,jj,ii,ncap,klm,j,
     +ncap3,ncap7,ncap2,j2,mplus2,k,irplus1,ir
      character
answer,filnam*80,filenam*80,patr*6,namer*20,
     +comtr*20,cframe*2
      common/person/ filenam,cframe

54    ivalue=0
      iflag=0
      opress=-123.
      volmax=-1234.
      volmin=1234.
      volumax=volmax
      volumin=volmin

      do 31 i=1,m
        if volu(i .gt.volmax  volmax=volu(i)
        if(volu(i .lt.volmin  volmin=volu i)
        if(volum(i).gt.volumax  volumax=volum i)
31      if(volum(i .lt.volumin  volumin=volum i)

      ik=m
      write(6,8)

c     call
graph volmax,volmin,volumax,volumin,m,l,ik,ij)

      do 26 i=1,m
26      xx(i)=float(i)
      if ij.eq.8) then
        do 27 i=1,m
27        yy(i =volu i)
      else
        do 28 i=1,m
28        yy(i =volum i)
      endif

c     call brkncv(xx,yy,m,-1)
c     do 22 i=1,m,2
c22     call numb pt(xx i),yy(i),3,i)
c     call endplt

      write 6 9)

47    write(6,23
      read(*,*) l
      write(6,32)
      read(*,*) ll
      if(( ll-l .lt.3).and.(ll.ne.l)) then
        write(6,53)
        goto 47
      endif

C-----If same numbers femvol called once
      if(l.eq.ll) then
        nlm=1
        call femvol(volume,l,nlm,0)
      elseif l.lt.ll) then
        n=0
        jj=1
        do 25 ii-l,ll
```

```
           n=n+1
           ax jj =dble xx ii
           ay jj =dble  ; ii
           press jj =dble preses ii
           if press jj .lt.opress  then
             iflag=1
             write 6,1
           endif
           opress=press jj
25         jj=jj+1

38    do 9 9 ncap-1,n

if  ivalue.gt.0.and.ncap.ne.ivalue . iflag.eq.   goto
999
         print*,'Ncap = ',ncap
         ncap2=ncap+2
         ncap3=ncap+3
         ncap7=ncap+7
         if ncap7.lt.ncap+7.or.ncap.gt.m+4  g t    3
C-----Positioning knots evenly on X-axis
         klm=1
         do 30 j=5,ncap3
           ak(j)=press 1 + (press n -
press 1 )*klm float ncap
30         klm=klm+1
         do 1 i=1,n
1          w i)-1.0D0
         w 1 =100.D0

         ifail=1
         call
e02baf n,ncap7,press,ay,w ak work1,work2 - ss ifail
         if ifail.ne.0) g to 2
         j=1
         if(ivalue.eq.0) write 6,3  j c 1
         do 4 j=2,ncap2
           j2=j+2
           if(ivalue.eq.0  write 6 5  j ak j2  c j
4        continue
         if(ivalue.eq.0) write 6,6' ncap3 c ncap2
         if(ivalue.eq.0  write 6,7' ss
2        print*,'ifail=',ifail
         a=ak 4)
         b=ak(ncap+4)
         mplus2=n+2
         flm1=n-1
         k=0
         do 14 irplus1=1,mplus2
           if(irplus1.eq.1.or.irplus1.eq.mplus2  g t  14
           k=k+1
           ir=irplus1-1
           flr1=ir-1
           xxx=press k
           ifail-1
           call e02bbf(ncap7,ak,c,xxx,s k',ifail
           if ifail.ne.0) g    15
           if(ivalue.eq.0  write(6,16) ir,xxx,s k
           if(ivalue.gt.0.and.ij.eq.7) write(8 4   k
     +volu l+k-1 ,s(k)
           if(ivalue.gt.0.and.ij.eq.8) write 8,4   k,
     +volum(l+k-1 ,s(k)
           goto 14
15         if(ivalue.eq.0) write(6,18) ir,xxx
14         continue

C-----Calculating target volumes
           if(ivalue.gt.0) then
             tarvol(1)=volum l
             do 60 i=1,k-1
               volnex=( s(i)+(s(i+1 -s(i) )*volum i+l-
1 )/s(i)
60             tarvol(i+1)=volnex
           end if
           if(ivalue.gt.0.and.ij.eq.9) then
             write(8,61  (i+l-1,volum(i+l-
1),s(i),tarvol(i),
     +i=1,ll+1-l)
           end if
           if(ivalue.gt.0) write 6,44
           if(ivalue.gt.0) goto 70
           call page 29.7,21.
           call picsiz 29.,20.3
           if(ij.eq.7.or.ij.eq.   call scales volmin-
5.,volmax+15.,1,
     +preses(l)-1.,preses(ll   .,1
             if(ij.eq.8  call scales(volumin-
5.,volumax+15.,1,
     +preses(l)-1.,preses ll +1.,1)
           call axes7('VOLUME  cm3 ','PRESSURE  mmHg '
           call cvtype(2
70         do 13 i=1,n
             axx(i)=sngl ax(i
             ayy(i)=sngl(ay(i
13           apress i =sngl press i
           do 19 j=1,k
19           pp j)=sngl(s(j)
           if(ivalue.gt.0) goto 71
           call brkncv ayy,apress,n,-1)
           call brkncv pp,apress,k,0)
           do 20 i=1,n
20           call numbpt(ayy i ,apre s i ,3,i+ l-1)
           call endplt
55         write(6,55)
           read *,*) answer
           if(answer.eq.'Y'. r.answer.eq.'y'  g t  71
           if(answer.eq.'N'. r.answer.eq.'n'  g t  7
       goto 55
71       if(ivalue.gt.0) goto 996
72     write(6,39)
       read(*,* answer
       if(answer.eq.'Y'.or.answer.eq.'y'  goto 73
       if(answer.eq.'N'.or.answer.eq.'n'  goto 36
       goto 72
73     write(6,37)
       read(*,*) ivalue
       filnam='voludata.'//filenam
       open(8,file=filnam,access='sequential',
    +form='formatted')
       write(8,42) patr,namer,comtr,ll-l+1
       if(ij.eq.7) write(8,43)
       if(ij.eq.8) write(8,45)
       if(ij.eq.9) write(8,67)
       goto 38
36     write(6,34)
       read(*,*) answer
       if(answer.eq.'Y'.or.answer.eq.'y'  goto 997
       if(answer.eq.'N'.or.answer.eq.'n'  goto 999
       goto 36
999    continue

       goto 997
998    print*,'Boundary values exceeded--going back to
beginning'
       goto 54
997    continue
996    close(8)

49     write(6,46)
       read(*,*) answer
       if(answer.eq.'Y'.or.answer.eq.'y') then
         ivalue=0
         goto 54
       else if(answer.ne.'N'.and.answer.ne.'n') then
         goto 49
       end if

      endif
48    continue

3     format(1h,i3,20x,e20.5)
5     format 1h,i3,3e20.5)
6     format(1h,i3,20x,e20.5)
7     format ' Residual sum=',e20.5)
8     format( /
     +' The complete VOLUME - FRAME NO. graph is drawn
first'/
     +' to enable choosing of range of frames for
analysis' /)
9     format(///
     +' It is advised that whilst one attempts to find
the best'
     +' smooth curve through selected cardiac frames,one
should'
     +' use the screen as a viewing device before
trasferring'/
     +' volume data for a hard copy.'///
     +' NOTE ALSO:that if one requires finite element
based'/
     +' volumetric and geometric data for simply one
frame'/
     +' anywhere in the cardiac cycle without the
necessity'/
     +' to smooth through a selected number of frames,
this'/
     +' can be done by assigning the same frame number to
the'/
     +' questions asked concerning the starting and end'/
     +' diastolic frame numbers.' //)
10    format(////
     +' WARNING !! You must choose a different range.'
     +' The pressure must always be non-decreasing' /)
16    format(1h,i3,2e20.5)
18    format 1h,i3,e20.5,'Argument outside range')
23    format(// /' Choose starting frame for
diastole/systole? ',$)
32    format(// /' And end diastole/systole frame number?
',$)
34    format(////
     +' Do you wish to quit this plotting routine?'/
     +' (with the option of repeating) ',$)
35    format(////
     +' Is there here what looks like a satisfactory best
curve fit'/
     +' between the points? ',$)
37    format(////' Value for Ncap? ',$)
39    format  //
     +' Do ȷ u then wish to enter a value for Ncap which
gives'/
     +' a good fit and also generate the data file? ',$)
40    format(3x,i2,f23.2,f32.2)
42    format(a6,1x,2(a20,1x  'NO. OF FRAMES SMOOTHENED
THROUGH=',
     +i2)
43    format''FRAME NO.',5x,'VOLUME (cm3  from
VOLUME',5x,
     +'VOLUME (cm3) from smoothened curve')
44    format(//' VOLUME INFORMATION SENT TO FILE - FILE
NOW CLOSED')
45    format('FRAME NO.',5x,'VOLUME (cm3  fr⁻m FEM',5x,
     +'VOLUME from smoothed curve'
46    format(/  /
     +' Do you wish to re-run thro'' this sm.otning
subroutine'/
```

```
     +' segment again? ',$)
53    format // /
     +' More data points than that are requirea f¬r
smoothing'  /)
61    format(3x,i2,11x,f6.2,13x,f6.2,13x,f6.2
67    format('FRAME NO.',2x,'VOL(cm3 --VOLUME ',2x,
     +'VOL(cm3)--SMOOTH ',2x,'VOL cm3)--TARGET'

      return
      end


      subroutine
graph(volmax,volmin,volumax,volumin,m,il,ik,ij)
      real volmax,volmin,volumax,volumin,am bm
      integer m,il,ij,ik

      am-float(il
      bm=float(ik

      call initsp
      call picsiz(15.,17.
      if(ij.eq.7.or.ij.eq.9) call
scales(am,bm+1.,1,volmin-10.,
     +volmax+10.,1
      if(ij.eq.8  call scales(am,bm+1.,1,volumin-
10.,volumax+10.,1)
      call axes('FRAME NUMBER',12,'VOLUME  cm3 ',12)
      call cvtype 2)

      return
      end


c*****************************************************
**********
c
c     part 2 manual volume matching
c
c*****************************************************
**********


      subroutine part2
c
c     It is called from the main program
c
c     controls the execution of the second par‑
c
c     it call subroutine pool
c
      character filenam*80,cframe*2
      common person/ filenam,cframe

      write(6,9)
      read 5,'(a20)') filenam

      call pool

 9    format(  / ///' Which datafile do you wish t  work
on ',$)

      return
      end


      subroutine pool
c
c     Results fr⁻m FE part analysed here for ongoing
analysis;
c     displacements are added and scaled accordingly and
a datafile
c     built up
c
c     This process has been aut mated in Part 4 so this
routine
c     is nolonger used during automated analysis
c
      double precision wmod(401,3),disp(401,3),vo,
     +smovol,ewcord 401,3)
      real
volone(151),voltwo(151 ,volthr(151),preses(151 ,
     +delvol 151
      integer
iframe,nof,int 151 ,nofras,i    es,j,iswtch,iresp,isw,
     +ireply
      character*80
filnam,filenam,title,head1,head2,patr*6,namer*20
     +,comtr*20,cframe*2
      common/person/ filenam,cframe

      filnam='pressure.'  filenam
C-----Pressures read in here from file

open(7,file=filnam,access-'sequential',form 'f rmarted'
      read(7,*) nofras
      read(7,*   preses ī ,i 1,n fras)
      close 7

C-----Inital fram- co-ordinates read in here
      write o,o
      read(5,' a2 '  cframe
      filnam='c crds'  cframe  '.'  filenam

open 9,file filnam,access-' -quential',f rm 'f rmatted'
      read 9,   patr,namer,c mtr,.frame
      read 9,    wmcd i,1 ,wn d i,. ,wn d i,3 ,i 1,4
      close(9)

      filnam='disp' /cframe//'.'//filenam

open(9,file=filnam,access='sequential',form='formatted')

      read(9,7) title,head1,head2
      read(9,8) (disp(i,1),disp(i,2),disp(i,3),i=1,401)
      close 9)

C-----Option to add displacements if scaling factors for
E are unknown
44    write 6,41)
      read(*,*) noyes
      if(noyes.eq.1  then
        do 4 i=1,401
          do 4 j=1,3
4           ewcord(i,j)=wmod(i,j)+disp(i,j)
        iswtch=3
        call
fevolume(ewcord,vo,smovol,iframe+1,iresp,iswtch,isw)
      elseif(noyes.ne.2) then
        goto 44
      endif

C-----Matching volume result with targeted ones tabulated
      write(6,18
      read(*,*) ireply

      filnam='voludata.'//filenam

open(9,file=filnam,access='sequential',form='formatted'

      read(9,11) patr,namer,comtr,nof,title

      do 12 i=1,nof
        if(ireply.eq.1) then
          read(9,*) int(i),volone i),voltwo i
          delvol(int(i))=voltwo(i
        else if(ireply.eq.2  then
          read(9,*) int(i),volone i ,voltwo i ,volthr i
          delvol(int(i))=volthr(i
        end if
12    continue

      close(9)

      do 13 i=1,nof
        if(int(i).eq.iframe+1.and.ireply.eq.1)
smovol=voltwo i)
        if int i).eq.iframe+1.and.ireply.eq.2)
smovol=volthr(i)
13    continue

      if(noyes.eq.1) write(*,14)
iframe+1,vo,iframe+1,smovol

      call
equvol(smovol,wnod,disp,vo,iframe,patr,namer,comtr,
     +iswtch,noyes,preses,delvol)

3     format(a6,1x,2(a20,1x) 28x,i2)
5     format(3f18.12)
6     format(  /' Which frame is to be analysed ? ',$
7     format(a80/a80/a80)
8     format(11x,3e14.7)
11    format(a6,1x,2(a20,1x)/33x,i6 a80
18    format(/
     +' Is the present analysis considering matching the
smoothened'/
     +' VOLUME/FE volumes or the smooth FE TARGET
Volumes:'/
     +' (1=VOLUME/FE  2=TARGET) ',$)
14    format(/
     +' Volume from fevolume for supposed frame
',i2,'=',f7.2,/
     +' Volume from smoothened analysis for frame
',i2,'=',f7.2/)
41    format(/
     +' Do you wish to obtain a new volume figure by
adding the'/
     +' displacements as obtained from the FE Analysis or
do you wish'/
     +' to bypass this stage and proceed directly onto
adding'/
     +' displacements of a known scale factor to obtain a
volume'/
     +' and/or directly create a new datafile for the
next frame ?'///
     +' 1=OBT IN VOLUME 2=PROCEED ',$)

      return
      end



      subroutine
equvol(smovol,wnod,disp,vo,iframe,patr,namer,comtr
     +,iswtch,noyes,preses,delvol)
C-----Scaling and Matching of volumes carried out here
and subsequent
C     datafile for next frame in sequence built up

      double precisi n smovol,wnod(401,3),disp 401,3),vo,
     +xtdisp 401,3),xtfrco(401,3)
      real
Ex,Ey,Ez,Muyx,Muzx,Muzy,Gzx,Gyz,Gxy,ExEz,ratio,Poirat
```

```
      real preses 151 ,delvol 151 ,step
      integer
iframe,iswtch,noyes,krep,irep,lrep,j,k,isw iresp jrep .
      logical torf
      common/person  filenam,cframe
      character filnam*80,answer,patr*6,namer*20,comtr*
     +,cframe*2,filenam*80,cha

      if noyes.eq.2  then
        write(6,55) iframe+1
        read(*,*) krep
      end if

      if(noyes.eq.1  goto 58
      if(krep.eq.2) irep=2
      if(krep.eq.2) goto 26

60    write(6,59)
      read *,*) lrep

      if lrep.eq.1) goto 24
      if(lrep.eq.2) goto 25

      goto 60

58    if vo.lt.smovol  goto 24
      if(vo.gt.smovol) goto 25

24    write(6,32)
      read(*,*) step
      print*,'Step = ',step

      do 2 j=1,401
         do 2 k=1,3
            xtfrco j,k)=disp j,k *step+wn d j,k)
2           xtdisp j,k =disp j,k *step+wnod(j,k)

      isw=4
      call
fevolume(xtdisp,vo,smovol,iframe+1,iresp,iswtch,isw
      write 6,34  vo,smovol
36    write(6,35)
      read(*,*) answer

      if(answer.eq.'Y'.or.answer.eq.'y') goto 3
      if(answer.eq.'N'.or.answer.eq.'n'  goto 24

      goto 36
3     continue
      write 6,30) vo,smovol,step
      goto 26
25    write(6,37)
      read(*,*) step
      print*,'Step = ',step

      do 28 j=1,401
        do 28 k=1,3
          xtfrco(j,k)=(disp  ,k)/step)+wnod(j,k)
28        xtdisp(j,k)=(disp(j,k)/step)+wnod(j,k)

      isw=5
      call
fevolume xtdisp,vo,smovol,iframe+1,iresp,iswtch,isw
      write 6,34  vo,smovol

38    write 6,35
      read(*,*) answer
      if answer.eq.'Y'.or.answer.eq.'y'  goto 29
      if(answer.eq.'N'.or.answer.eq.'n'  goto 25
      goto 38

29    continue
      write(6,30) vo,smovol,step
26    continue

      if(irep.eq.2) then
        write(6,50)
        read(*,*) jrep
        write(6,40)
        read(*,*) step
      end if

      if(jrep.eq.1) then
        do 51 j=1,401
          do 51 k=1,3
51          xtfrco j,k)=disp   k *step+wnod j,k)
          isw=4
      else if(jrep.eq.2) then
          do 52 j=1,401
            do 52 k=1,3
52            xtfrco(j,k =(disp(j,k)/step)+wnod(j,k)
          isw=5
      end if

21    write(6,17  iframe+1
      read(5,*  answer
      if(answer.eq.'Y'.or.answer.eq.'y') then
        filnam='dcoords'//cframe//'.'/ filenam

open 7,file-filnam,access='sequential',f rm 'formatted'
        write(7,19  patr,namer,comtr,iframe+1
        write 7,_0
 xtfrco i,1 ,xtfrco i,2 ,xtfrc  i,3 ,i,i 1,4 1
        close 7)
      elseif answer.ne.'N'.and.answer.ne.'n'  then
        g t  _1
      end if
```

```
C-----Patient Material Results File created here
      write 6,4)
      read *,*) ExEz
      write 6,5)
      read(*,*) ratio
      write 6,6)
      read(*,*) Poirat
      Ex=ExEz
      Ey=Ex*ratio
      Ez=Ex
      Muyx=Poirat
      Muzx=Muyx
      Muzy=Poirat/ratio
      Gzx=ExEz/(2*(1+Poirat))
      Gyz=Gzx*ratio
      Gxy=Gyz

C-----Corrected values calculated here
      if(isw.eq.4) then
        Ex=Ex*(1./step
        Ey=Ex*ratio
        Ez=Ex
        Muyx=Poirat
        Muzx=Muyx
        Muzy=Poirat/ratio
        Gzx=Gzx*(1./step)
        Gyz=Gzx*ratio
        Gxy=Gyz
       elseif(isw.eq.5) then
        Ex=Ex*step
        Ey=Ex*ratio
        Ez=Ex
        Muyx-Poirat
        Muzx=Muyx
        Muzy=Poirat/ratio
        Gzx=Gzx*step
        Gyz=Gzx*ratio
        Gxy=Gyz
       end if

       filnam='matprop'//'.'/ filenam
       inquire file=filnam,exist=torf
       if(torf) then

open(7,file=filnam,access='sequential',form='formatted')
444     read(7,'(al)',end=911) cha
        goto 444
911     write(7,22)
iframe,iframe+1,Ex,Ey,Ez,Muyx,Muzx,Muzy,Gxy,Gyz,
     +Gzx,(preses(iframe+1)-
preses(iframe)) 0.02, delvol(iframe+1)-
     +delvol(iframe)) 0.02,(delvol(iframe+1)-
delvol iframe))/(
     +preses iframe+1)-preses(iframe))
        close(7)
       else

open(7,file=filnam,access='sequential',form='formatted'
        write(7,14)
patr,iframe,iframe+1,Ex,Ey,Ez,Muyx,Muzx,Muzy,Gxy,
     +Gyz,Gzx,(preses iframe+1)-
preses(iframe))/0.02, delvol(iframe+1)
     +-delvol(iframe))/0.02, delvol(iframe+1)-
delvol(iframe))/
     +(preses(iframe+1)-preses(iframe))
      end if

      close(7)
 16   write(6,15)
      read(*,*) answer
      if(answer.eq.'Y'.or.answer.eq.'y') then
        call data(xtfrco,iframe)
      else if(answer.ne.'N'.and.answer.ne.'n') then
        goto 16
      end if

4     format(////
     +' What was the original value for the minor
modulii'/
     +' i.e. Ex,Ez? ',$)
5     format(////
     +' And the long fibre to cross fibre modular ratio?
',$)
6     format(////' Poisson''s ratio (for Mu-yx & Mu-zx)?
',$)
14    format(' MATERIAL PROPERTIES FOR ',a6/' -----------
---------------
     +----'/' Frame
Nos.',3x,'Ex', x,'Ey',5x,'Ez',4x,'Muyx',4x,'Muzx',

+4x,'Muzy',4x,'Gxy',4x,'Gyz',4x,'Gzx',4x,'Dp/Dt',4x,'Dv/D
t',4x,
     +'Dv/Dp'/' ----------    --      ---     --     ----
----    ----
     + ----   ---     ---     -----     -----     -----
'/13x,'KN m2',2x,

+'KN/m2',2x,'KN/m2',25x,'KN/m2',2x,'KN/m2',2x,'KN/m2',2x,
'mmHg/s',
     +4x,'cm3/s',3x,'cm3 mmHg'/'                -----  -----
-----
     +                   -----  -----  -----  ------    ---
--   --------'
     +/i5,'-
',i2,f9.2,f8.2,f7.2,f7.3,f8.3,f8.3,f7.2,f7.2,f7.2,f8.3,
     +f9.3,f9.3)
```

```
15    format
     +' Do you wish to complete preperation of the next
frame''s'
     +' entire datafile for FE analysis ' $
17    format
     +' Do you wish to create part of the next frame''s
datafile'/
     +' which contains the new set of co-ordinates'
     +' i.e. for frame ',i2,'? ',$
18    format  ' Name of file for housing co-ordinates.
',$)
19    format a6,1x,2 a20,1x  ' CO-ORDINATES FOR FRAME NO.
',i2)
20    format 3f18.12,i5
22    format(i5,'-
',i2,f9.2,f8.2,f7.2,f7.3,f8.3,f8.3,f7.2,f7.2,f7.2,
     +f8.3,f9.3,f9.3)
30    format(/
     +' Volume from femesh ='f7.2,' Value from
smoothening =',f7.2,
     +' Value of step =',f3.1
37    format(/
     +' By what factor would you now like to decrease the
original'/
     +' displacements by (so that the volumes can be
matched) ? ',$
34    format(/'Volume from factor increase =
',f6.2,'(cm3)'/
     +'Smoothened Target Volume    = ',f6.2,' cm3)'
32    format(/
     +' By what factor would you now like to increase the
original'/
     +' displacements by  so that the volumes can be
matched) ? ',$)
40    format(/' Value for STEP-the scale factor ? ',$
55    format(/
     +' Do you wish to produce a volume at all or proceed
directly'
     +' onto scaling the displacements as produced by FE
analysis'
     +' for the next frame i.e. frame ',i2,/
     +' 1=PRODUCE VOLUME 2=PROCEED ',$
59    format(/
     +' Are the displacements from FE part being scaled
UP or DOWN'/
     +' ? (1=UP 2=DOWN ',$)
35    format(/' Is this a satisfactory figure ? ',$)
50    format(/
     +' Are you scaling the displacements up or down
?'///
     +' (1=UP 2=DOWN) ',$)

      return
      end


c*****************************************************
c
c     part 3 Visualisation of results from FE analysis
c
c*****************************************************
      subroutine part3
c
c     It is called by the main program
c
c     It is a menu allowing a choice between material
propery graphs
c     and cross-sectional views
c
c     It calls routines planevw propgraph
c
      character answer

 200  write(6,201)
      read(*,'(al)') answer
      if(answer.eq.'1') then
        call planevw
      elseif(answer.eq.'2') then
        call propgraph
      elseif(answer.eq.'3') then
        call viewm
      endif

      if(answer.ne.'q'.and.answer.ne.'Q') then
        goto 200
      end if

 201  format(////
     +'          Choose options '/
     +'          --------------'///
     +' 1       Cross sectional planes' /
     +' 2       Property graphs' /
     +' 3       3-D F.E. Mesh Reconstruction' /
     +' Q       Return to main menu' /
     +'         ',$)

      return
      end


      subroutine viewm
c
c     This routine allows the FE mesh to be viewed using
c     a process similar to that used in Part 1 for
Visualisation
c
c     It calls subroutines readpoints and freehand
c
      real nodes1(401,3),nodes2(401,3),nodes3(401,3)
      character*80 file1,file2,file3,ans*1

      write(6,9)
      read(5,'(a1)') ans

      write(6,19)
      read(5,'(a80)') file1
      call readpoints(nodes1,file1)

      if((ans.eq.'2').or.(ans.eq.'3')) then
        write(6,19)
        read(5,'(a80)') file2
        call readpoints(nodes2,file2)
      endif

      if(ans.eq.'3') then
        write(6,19)
        read(5,'(a80)') file3
        call readpoints(nodes3,file3)
      endif

      call screenhand(nodes1,nodes2,nodes3,ans)

  9   format(//' Do you wish two overlay 1,2 or 3 sets of
data? ',$)
 19   format(//' What is the name of the coordinate data
file? ',$)

      return
      end


      subroutine plotend(tnodes)
c
c     this subroutine is called by screenhand
c
c     It plots the endocardium of the mesh
c
c     It calls only subroutine drwring
c
      real tnodes(401,3)
      integer
ir1(13),ir2(13),ir3 13),ir4(13),ir5(13),ir6(13)

      data ir1/5,8,13,16,21,24,29,32,37,40,45,48,5/
      data
ir2/71,74,79,82,87,90,95,98,103,106,111,114,71/
      data
ir3/137,140,145,148,153,156,161,164,169,172,177,180,137/
      data
ir4/203,206,211,214,219,222,227,230,235,238,243,246,203/
      data
ir5/269,272,277,280,285,288,293,296,301,304,309,312,269/
      data
ir6/335,338,343,346,351,354,359,362,367,370,375,378,335/

      call drwring(tnodes,ir1,ir2,ir3,ir4,ir5,ir6,0)

c     vertical lines
c      do 85 i=1,12,1
c        call brknpt(tnodes(ir1(i),2),tnodes(ir1(i),1),0)
c        call brknpt(tnodes(ir2(i),2),tnodes(ir2(i),1),0)
c        call brknpt(tnodes(ir3(i),2),tnodes(ir3(i),1),0)
c        call brknpt(tnodes(ir4(i),2),tnodes(ir4(i),1),0)
c        call brknpt(tnodes(ir5(i),2),tnodes(ir5(i),1),0)
c        call brknpt(tnodes(ir6(i),2),tnodes(ir6(i),1),0)
c        call break
c 85   continue

c     apex
c      do 70 i=1,12,4
c        call brknpt(tnodes(ir6(i),1),tnodes(ir6(i),1),0)
c        call brknpt(tnodes(401,1),tnodes(401,1),0)
c        call break
c 70   continue

c     mark some points
c      call cp7pt(tnodes(,2),tnodes(,3),15,'')
c      call cp7pt(tnodes(5,2),tnodes(5,1),15,'5')
c      call cp7pt(tnodes(29,2),tnodes(29,1),15,'29')
c      call cp7pt(tnodes(16,2),tnodes(16,1),15,'16')
c      call cp7pt(tnodes(40,2),tnodes(40,1),15,'40')
c      call cp7pt(tnodes(335,2),tnodes(335,1),15,'335')
c      call cp7pt(tnodes(359,2),tnodes(359,1),15,'359')
c      call cp7pt(tnodes(370,2),tnodes(370,1),15,'370')
c      call cp7pt(tnodes(346,2),tnodes(346,1),15,'346')

      return
      end


      subroutine plotepi(tnodes)
c
c     this subroutine is called by screenhand
c
c     It plots the epicardium of the mesh
c
c     It calls only subroutine drwring
c
      real tnodes 401,3)
      integer
or1(13),or2(13 ,or3(13),or4(13),or5(13),or6(13)

      data or1/1,6,9,14,17,22,25,30,33,38,41,46,1/
      data or2/67,72,75,80,83,88,91,96,99,104,107,112,67/
```

```
      data
or3/133,138,141,146,149,154,157,162,165,170,173,178,133
      data
or4 199,204,207,212,215,220,223,228,231,236,239,244,199
      data
or5 265,270,273,278,281,286,289,294,297,302,305,310,265
      data
or6 331,336,339,344,347,352,355,360,363,368,371,376,331

      call drwring tnodes,or1,or2,or3,or4,or5,or6,-1

c     vertical lines
c      do 80 i-1,12,1
c        call brknpt tnodes or1(i ,2),tnodes or1 i),1),1
c        call brknpt tnodes(or2 i ,2 ,tnodes(or2 i ,1 ,1
c        call brknpt tnodes or3(i ,2 ,tnodes(or3 i ,1 ,1
c        call brknpt tnodes(or4 i),2 ,tnodes(or4 i),1 ,1
c        call brknpt tnodes(or5 i),2),tnodes(or5 i),1 ,1
c        call brknpt tnodes(or6 i),2),tnodes(or6(i),1),1
c        call break
c 80   continue

c     apex
c      do 75 i=1,12,4
c        call brknpt(tnodes(or6(i ,1 ,tnodes or6(i),2 ,1
c        call brknpt(tnodes(397,1 ,tnodes 397,2),1)
c        call break
c 75   continue

      return
      end


      subroutine
drwring(tnodes,rr1,rr2,rr3,rr4,rr5,rr6,ltyp
c
c     this subroutine is called by plotepi and plotepi
c
c     It plots the endocardium of the mesh
c
c     It calls only simpleplot routine brkncv
c
      real
tnodes(401,3),x(12),y(12),xx(12),yy(12 ,xm(12),ym(12)
      integer
rr1(12 ,rr2(12),rr3(12),rr4(12),rr5(12),rr6(12),
     + i,j,ltyp

      do 5 i=1,12
        x(i)=tnodes(rr1(i),2)
        y(i)=tnodes(rr1(i),1)
 5    continue
      call brkncv(x,y,12,ltyp)

      do 10 i=1,12
        xx(i)=tnodes(rr2(i),2)
        yy(i)=tnodes(rr2 i ,1)
10    continue
      call brkncv(xx,yy,12,ltyp)

      do 35 i=1,7
        do 40 j=1,12
          xm(j)=(x(j)*float(i)+xx(j)*float(7-i))/7.
          ym(j)=(y(j)*float(i)+yy(j)*float(7-i))/7.
40      continue
        call brkncv(xm,ym,12,ltyp)
35    continue

      do 15 i=1,12
        x(i)=tnodes(rr3(i),2)
        y(i)=tnodes(rr3(i),1)
15    continue
      call brkncv(x,y,12,0)

      do 45 i=1,4
        do 50 j=1,12
          xm(j)=(x(j)*float(i)+xx(j)*float(4-i))/4.
          ym(j)=(y(j)*float(i)+yy(j)*float(4-i))/4.
50      continue
        call brkncv(xm,ym,12,0)
45    continue

      do 20 i=1,12
        xx(i)=tnodes(rr4 i),2
        yy(i)=tnodes(rr4(i),1
20    continue
      call brkncv(xx,yy,12,ltyp)

      do 55 i=1,4
        do 60 j=1,12
          xm(j)=(x(j)*float i)+xx j)*float(4-i )/4.
          ym(j)=(y(j)*float(i +yy(j)*float(4-i))/4.
60      continue
        call brkncv(xm,ym,12,ltyp)
55    continue

      do 25 i=1,12
        x(i)=tnodes(rr5 i),2
        y(i)=tnodes(rr5 i),1)
25    continue
      call brkncv(x,y,12,ltyp

      do 65 i=1,4
        do 70 j 1,12
          xm(j)= x j)*float i +xx j)*fl at 4-i   4.
          ym(j)= y(j)*float i +yy   *fl at 4-i)  4.
70      continue
```

```
        call brkncv(xm,ym,12,ltyp)
65    continue

      do 30 i=1,12
        xx(i)=tnodes(rr6(i),2)
        yy(i)=tnodes(rr6(i),1)
30    continue
      call brkncv(xx,yy,12,ltyp)

      do 75 i=1,4
        do 80 j=1,12
          xm(j =(x(j)*float(i)+xx(j)*float(4-i))/4.
          ym(j)= y(j)*float(i)+yy(j)*float(4-i))/4.
80      continue
        call brkncv(xm,ym,12,ltyp)
75    continue


      return
      end


      subroutine
rotate(nodes,tnodes,anx,any,anz,xmin,xmax,ymin,
     + ymax)
c
c     this subroutine is called by screenhand
c
c     It rotates the FE mesh to a specified orientatıon
c
c     It calls no subroutines
c
      real nodes(401,3),tnodes 401,3),anx,any,anz,
     +
axc,axs,ayc,ays,azc,azs,fact,t1a 3,3 ,t2a 3,3),t3a 3,3 ,
     + t4a(3,3),xmin,xmax,ymin,ymax
      integer i

      fact=3.141592653/180.

      axc=cos((anx)*fact)
      axs=sin( anx)*fact)
      ayc=cos((any)*fact)
      ays=sin((any)*fact)
      azc=cos((anz)*fact)
      azs=sin((anz)*fact)

      t1a(1,1)=1.
      t1a(1,2)=0.
      t1a(1,3)-0.
      t1a(2,1)=0.
      t1a 2,2)=axc
      t1a 2,3)=axs* -1.)
      t1a 3,1 =0.
      t1a 3,2 =axs
      t1a(3,3 -axc

      t2a(1,1)=ayc
      t2a(1,2 =0.
      t2a 1,3 -ays
      t2a(2,1 =0.
      t2a 2,2 =1.
      t2a 2,3 =0.
      t2a 3,1 =ays*(-1.)
      t2a 3,2 =0.
      t2a 3,3 =ayc

      t3a 1,1)=azc
      t3a 1,2)=azs*(-1.)
      t3a 1,3 =0.
      t3a 2,1)=azs
      t3a 2,2)=azc
      t3a 2,3 =0.
      t3a 3,1)-0.
      t3a(3,2)=0.
      t3a(3,3)=1.

      call rmmlt(t4a,t1a,t2a,3,3,3)
      call rmmlt(t1a,t4a,t3a,3,3,3)

      call rmmlt(tnodes,nodes,t1a,401,3,3)

      do 5 i=1,401
        if tnodes(i,1).lt.xmin) xmin=tnodes(i,1)
        if(tnodes(i,1).gt.xmax) xmax=tnodes(i,1)
        if(tnodes(i,2).lt.ymin) ymin=tnodes(i,2)
        if(tnodes(i,2).gt.ymax) ymax=tnodes(i,2)
 5    continue

      return
      end


      subroutine screenhand(nodes1,nodes2,nodes3,ans
c
c     this subroutine is called by viewm
c
c     It handles all the ploting output
c
c     It calls subroutines translate, rotapex, rotate,
c     plotepi, plotendo
c     and simpleplot routines initsp, page, eqscal,
newpic,
c     pen, endplot
c
      real
nodes1(401,3 ,tnodes1 401,3),xmin,xmax,ymin,ymax,
     +
anx,any,anz,nodes2 401,3 ,tnodes2 401,3 ,nodes3 4 1,3 ,
     + tnodes3 401,3
      character ans*1,anw*1

 100  continue

      xmin=0.0
      xmax=0.0
      ymin=0.0
      ymax=0.0

      write 6,60)
60    format( ///
     +' Input the 3 viewing angles for the X,Y and Z axes
resp.
     + (in degs. from 0 to 360):'
      write(6,61
61    format(' ax= ',$
      read(*,*) anx
      write(6,76)
76    format(' ay= ',$)
      read *,*) any
      write 6,77
77    format(' az= ',$
      read(*,*) anz

      call translate(nodes1)
      call rotapex(nodes1)
      call
rotate(nodes1,tnodes1,anx,any,anz,xmin,xmax,ymin,ymax
      if( ans.eq.'2').or.(ans.eq.'3'   then
        call translate(nodes2)
        call rotapex(nodes2)
        call
rotate(nodes2,tnodes2,anx,any,anz,xmin,xmax,ymin,ymax
      endif
      if(ans.eq.'3') then
        call translate(nodes3)
        call rotapex(nodes3)
        call
rotate(nodes3,tnodes3,anx,any,anz,xmin,xmax,ymin,ymax
      endif

      xmin=xmin*1.1
      xmax=xmax*1.1
      ymin=ymin*1.1
      ymax=ymax*1.1

      print*,' X range ',xmin,xmax
      print*,' Y range ',ymin,ymax

c      xmin=-2.7E-02
c      xmax=6.4E-02
c      ymin=-7.1E-03
c      ymax=9.9E-02

      call initsp
      call page(21.0,29.7)
      call eqscal(ymin,ymax,xmin,xmax,0
      call newpic

      call pen(1
      write(6,19)
      read(5,' a1)') anw
      if (anw.eq.'y').or. anw.eq.'Y')  then
        call plotepi tnodes1)
        if((ans.eq.'2').or.(ans.eq.'3')) then
          call pen(2
          call plotepi(tnodes2)
        endif
        if ans.eq.'3') then
          call pen 2
          call plotepi)tnodes1\
        endif
      endif
      write(6,29)
      read(5,' a1'' anw
      if (anw.eq.'y' .or. anw.eq.'Y')  then
        call plotend tnodes1)
        if  ans.eq.'2').or. ans.eq.'3'   then
          call pen 2
          call plotend tnodes2
        endif
        if(ans.eq.'3') then
          call pen 3
          call plotend(tnodes3)
        endif
      endif

      call endplt

      write 6,9
      read(5,' a1 '  anw
      if  anw.eq.'y' .or. anw.eq.'Y'   g to 100

9     format / ' do you wish to repeat. ',$
19    format  /
     +' Do you wish to plot epicardium s . ',$
29    format
     +' Do you wish to plot endocard um s . ',$

      return
      end


      subr utine translate n d-
```

```
c
c     this subroutine is called by screenhand
c
c     It moves the apex of the FE mesh to point zero
c
c     It calls no subroutines
c
      real nodes(401,3),x1,y1,z1
      integer i

      x1=(nodes(5,1)+nodes(8,1)+nodes(13,1 +nodes(16,1)+
     + nodes(21,1)+nodes(24,1)+nodes 29,1 +nodes(32,1)+
     +
nodes(37,1)+nodes(40,1)+nodes(45,1)+nodes 48,1))/12.
      y1=(nodes(5,2)+nodes(8,2 +nodes 13,2)+nodes(16,2)+
     + nodes(21,2)+nodes(24,2)+nodes(29,2)+nodes(32,2)+
     +
nodes(37,2)+nodes(40,2)+nodes(45,2)+nodes(48,2))/12.
      z1=(nodes(5,3)+nodes(8,3)+nodes(13,3 +nodes(16,3)+
     + nodes(21,3)+nodes(24,3)+nodes(29,3 +nodes(32,3)+
     +
nodes(37,3)+nodes(40,3)+nodes(45,3)+nodes(48,3))/12.

      do 5 i=1,401
        nodes(i,1)=nodes(i,1)-x1
        nodes(i,2)=nodes(i,2)-y1
        nodes(i,3)=nodes(i,3)-z1
 5    continue

      return
      end


      subroutine rotapex(nodes)
c
c     this subroutine is called by screenhand
c
c     It rotates the FE mesh to be vertical
c
c     It calls only subroutine rmmlt
c
      real nodes(401,3),nodesa(401,3),t1(3,3 ,t2(3,3),
     + t3(3,3),hyp1,cos1,sin1,hyp2,cos2,sın2,pi
      integer i

      pi=3.141592653

      do 10 i=1,3
        do 10 j=1,3
          t1 i,j)-0.
          t2(i,j =0.
10        t3(i,j)=0.

C-----Find angles

hyp1=sqrt(nodes 399,1)*nodes(399,1)+nodes 399,2 *nodes(39
9,2))
      cos1=nodes 399,1) hyp1
      sın1=nodes 399,2  hyp1
      hyp2=sqrt nodes 399,3)*nodes(399,3 +hyp1*hyp1)
      cos2=hyp1/hyp2
      sin2=nodes(399,3)/hyp2

      sin1=sin(asin sin1)-pi)
      cos1=cos(acos(cos1)-pi)

 C-----Fill transform matrices
      t1(1,1 =cos1
      t1(2,2)=cos1
      t1(3,3)=1.
      t1(1,2 =sin1*(-1.)
      t1(2,1 =sin1
      t2 1,1)=cos2
      t2(1,3)=sin2
      t2(3,1)=sin2*(-1.)
      t2 2,2)=1.
      t2(3,3)=cos2

 C-----Find overall transform
      call rmmlt(t3,t1,t2,3,3,3)

      call rmmlt(nodesa,nodes,t3,401,3,3)

      do 5 i=1,401
        nodes(i,1)=nodesa(i,1
        nodes(i,2)=nodesa(i,2)
        nodes(i,3)=nodesa(i,3)
 5    continue

      return
      end


      subroutine readpoints(nodes,file
 c
 c    this subroutine is called by viewm
 c
 c    It reads the coordinate point from the data file
 c
 c    It calls no subroutines
 c
      double precision dnodes(401,3)
      real nodes(401,3)
      integer i
      character*80 rile,framea*_,namer*_ ,c mtr*_0,patr*b
open(7,file=file,access-'sequential',form='forrat-ed'
      read(7,9  patr,namer,comtr,framea
      read(7,19)
 dnodes i,1),dnodes i,2 ,dnodes i,3 ,ı=1,4 1
      close(7)

      do 5 i=1,401
        nodes i,1 =sngl dnodes i,1)
        nodes(i,2 -sngl dnodes(i,2)
        nodes(i,3 =sngl dnodes i,3)
 5    continue

 9    format(a6,1x,2 a20,1x)/28x,a2)
 19   format 3r18.12

      return
      end


      subroutine rmmlt matp,matl,matr,n1,n2,n3)
c
c     this subroutine is called by rotate and rotapex
c
c     It multipies two real matricies together
c
c     It calls no subroutines
c
      real matp(n1,n3),matl(n1,n2),matr(n2,n3)
      integer n1,n2,n3,i,j,k

      do 20 i=1,n1
        do 20 j=1,n3
          matp(i,j)=0.
          do 20 k=1,n2
20          matp i,j =matp i,j)+matl(i,k)*matr(k,j

      return
      end



      subroutine planevw
c
c     It is called by subroutine propgraph
c
c     It draws the cross-sectional plan views of the
ventricle
c
c     It uses routines planes dcos1 roots secos2 dcos3
trans dmmlt1
c     point1 point2
c
      double precision
xnod(401,3),tgfrco(401,3),tran(3,3),uran(3,3),

+corxyz(401,3),xtfxyz(4 1,3  al1,am1,an1,al2,am2,an2,al3,
am3,an3,

+bl1,bm1,bn1,bl2,bm2,bn2,bl3,bm3,bn3,aaa,bbb,ccc,ddd,eee,
fff,ggg,
     +hhh,a(3),b(3),c(3),d(3),e(3),f(3
      real xmin,xmax,ymin,ymax,pie
      integer j1,i,n
      character
answer,filenam*80,cframe*2,filnam*30,file1*80,patr*6,

+frnum*12,orgpr*16,defpr*20,framea*2,frameb*2,namer*20,co
mtr*20
      common/person/ filenam,cframe

      n=100
      pie=3.1415926

202   write(6,204)
204   format(////
     +' The name patient to be analysed ',$)
      read(*,'(a80)'  filenam
      write(6,205
      read(*,'(a2)') cframe
205   format(////' The number of the original frame? ',$)
      filnam='coords'//cframe//'.'//filenam

open 7,file=filnam,access='se   ntial',form='formatted'
      read(7,206  patr,namer    tr,framea
206   format(a6,1x,2(a20,1x)/  x,a2)
      read 7,207)  xnod i,1 ,xnod(i,2),xnod(i,3),i 1,401)
207   format(3f18.12)
      close(7
      print 208
208   format(////
     +' Filename containıng the 401x3 co-ordinates of the
target'/
     +' frame? ',$)
      read *,'(a80)'  file1

open(7,file=file1,access='sequentıal',f rm='formatted'
      read(7,206  patr,namer,comtr,frameb
      read(7,207
(tgfrco(i,1),tgfrco(ı,2),tgfrco i,3),i 1,401)
      close(7

      frnum-'FRAME NUMBER'
      orgpr='ORIGINAL PPOFILE'
      defpr='DEFLECTED PR FILE'

C-----Set IJ so grapn axes are - nstant
```

```
      jl=0
      xmin=1234.
      xmax=-1234.
      ymin=1234.
      ymax=-1234.
      if(jl.eq.0  goto 213
238   print*
      print*,' Do you wish to view the 1st plane?'
      read *,' a1)') answer
      if(answer.eq.'n'.or.answer.eq.'N'  goto 214

C-----First layer;take nodes 1,17 and 33 to fit equation
of the plane
c213   call planes a,b,c,d,e,f,xnod,tgfrco,1,17,33
c      call roots(a,b,c,aaa,bbb,ccc,ddd)
c      call roots d,e,f,eee,fff,ggg,hhh

C-----Finding direction cosines and new z axis
c      call dcos1'aaa,bbb,ccc,ddd,al3,am3,an3
c      call dcos1(eee,fff,ggg,hhh,bl3,bm3,bn3
c      call secos2(1,17,xnod,al1,am1,an1)
c      call secos2(1,17,tgfrco,bl1,bm1,bn1
c      call dcos3 al3,am3,an3,al1,am1,an1,al2,am2,an2
c      call dcos3(bl3,bm3,bn3,bl1,bm1,bn1,bl2,bm2,bn2
c      call
trans(al3,am3,an3,al1,am1,an1,al2,am2,an2,tran)
c      call
trans(bl3,bm3,bn3,bl1,bm1,bn1,bl2,bm2,bn2,uran)
c      call dmmlt1 1,48,xnod,tran,corxyz
c      call dmmlt1 1,48,tgfrco,uran,xtfxyz
c      call
point1 1,5,6,8,9,13,14,16,17,21,22,24,25,29,30,32,33,37,3
8,
c
+40,41,45,46,48,3,47,1,jl,xmin,xmax,ymin,ymax,corxyz,xtfx
yz,
c     +patr,frnum,framea,orgpr,defpr

C-----First layer;take nodes 1,17 and 33 to fit equation
of the plane
213   call planes a,b,c,d,e,f,xnod,tgfrco,5,21,37
      call roots a,b,c,aaa,bbb,ccc,ddd
      call roots(d,e,f,eee,fff,ggg,hhh

C-----Finding direction cosines and new z axis
      call dcos1 aaa,bbb,ccc,ddd,al3,am3,an3
      call dcos1 eee,fff,ggg,hhh,bl3,bm3,bn3)
      call secos2 5,21,xnod,al1,am1,an1
      call secos2 5,21,tgfrco,bl1,bm1,bn1)
      call dcos3 al3,am3,an3,al1,am1,an1,al2,am2,an2)
      call dcos3 bl3,bm3,bn3,bl1,bm1,bn1,bl2,bm2,bn2
      call
trans al3,am3,an3,al1,am1,an1,al2,am2,an2,tran
      call
trans bl3,bm3,bn3,bl1,bm1,bn1,bl2,bm2,bn2,uran
      call dmmlt1 1,48,xnod,tran,corxyz)
      call dmmlt1 1,48,tgfrco,uran,xtfxyz)
      call
point1(1,5,6,8,9,13,14,16,17,21,22,24,25,29,30,32,33,37,3
8,

+40,41,45,46,48,3,47,1,jl,xmin,xmax,ymin,ymax,corxyz,xtfx
yz,
     +patr,frnum,framea,orgpr,defpr

C---------------------------------------------------------
-------------
      if jl eq.0) goto 215
214   print*
      print*,'Do you wish to view the plane between 1st
and 2nd
     +layers?'
      read *,' a1)'  answer
      if(answer.eq.'n'.or.answer.eq.'N'  goto 216

C-----Intermediate layer between 1st and 2nd;taking nodes
49,55 and 61
*     to fit eq. of plane
215   call planes(a,b,c,d,e,f,xnod,tgfrco,49,55,61
      call roots a,b,c,aaa,bbb,ccc,ddd
      call roots d,e,f,eee,fff,ggg,hhh
      call dcos1 aaa,bbb,ccc,ddd,al3,am3,an3
      call dcos1 eee,fff,ggg,hhh,bl3,bm3,bn3
      call secos2 49,55,xnod,al1,am1,an1
      call secos2 49,55,tgfrco,bl1,bm1,bn1
      call dcos3 al3,am3,an3,al1,am1,an1,al2,am2,an2
      call dcos3 bl3,bm3,bn3,bl1,bm1,bn1,bl2,bm2,bn2
      call
trans al3,am3,an3,al1,am1,an1,al2,am2,an2,tran
      call
trans bl3,bm3,bn3,bl1,bm1,bn1,bl2,bm2,bn2,uran
      call dmmlt1(49,66,xnod,tran,corxyz
      call dmmlt1 49,66,tgfrco,uran,xtfxyz
      call
point2 49,51,52,54,55,57,58,60,61,63,64,66,50,65,1,jl,xmi
n,

+xmax,ymin,ymax,corxyz,xtfxyz,patr,frnum,framea,orgpr,def
pr

C---------------------------------------------------------
-------------
      if jl.eq.0  goto 217
      print*
216   print*,'Do you wish to view the 2nd plane?'
      read(*,' a1 ') answer
      if answer.eq.'n'. r.answer.eq.'n ') goto 218

C-----Second layer;taking nodes 67,83 and 99 to fit eq.
of plane
217   call planes a,b,c,d,e,f,xnod,tgfrco,67,83,99
      call roots a,b,c,aaa,bbb,ccc,ddd
      call roots(d,e,f,eee,fff,ggg,hhh

C-----Finding direction cosines and new z axis
      call dcos1'aaa,bbb,ccc,ddd,al3,am3,an3
      call dcos1(eee,fff,ggg,hhh,bl3,bm3,bn3
      call secos2 67,83,xnod,al1,am1,an1
      call secos2 67,83,tgfrco,bl1,bm1,bn1
      call dcos3 al3,am3,an3,al1,am1,an1,al2 am2 an2
      call dcos3 bl3,bm3,bn3,bl1,bm1,bn1,bl2,bm2,bn2
      call
trans(al3,am3,an3,al1,am1,an1,al2,am2,an2,tran
      call
trans(bl3,bm3,bn3,bl1,bm1,bn1,bl2,bm2,bn2,uran)
      call dmmlt1 67,114,xnod,tran,corxyz
      call dmmlt1 67,114,tgfrco,uran,xtfxyz
      call
point1 67,71,72,74,75,79,80,82,83,87,88,9  91 95,96 98,99
,

+103,104,106,107,111,112,114,69,113,2,jl,xmin,xmax ymin,y
max,
     +corxyz,xtfxyz,patr,frnum,framea,orgpr defpr

C---------------------------------------------------------
-------------
      if(jl.eq.0) goto 219
218   print*
      print*,'Do you wish to view the plane between 2nd
and 3rd
     +layers?'
      read *,' a1)'  answer
      if answer.eq.'N'.or.answer.eq.'n'  goto 220

C-----Intermediate layer between 2nd and 3rd;taking nodes
115,121 and
C     127
219   call planes a,b,c,d,e,f,xnod,tgfrco,115,121,127
      call roots a,b,c,aaa,bbb,ccc,ddd)
      call roots(d,e,f,eee,fff,ggg,hhh

C-----Finding direction cosines and new z axis
      call dcos1(aaa,bbb,ccc,ddd,al3,am3,an3)
      call dcos1 eee,fff,ggg,hhh,bl3,bm3,bn3)
      call secos2 115,121,xnod,al1,am1,an1)
      call secos2 115,121,tgfrco,bl1,bm1,bn1)
      call dcos3 al3,am3,an3,al1,am1,an1,al2,am2,an2
      call dcos3 bl3,bm3,bn3,bl1,bm1,bn1,bl2,bm2,bn2)
      call
trans al3,am3,an3,al1,am1,an1,al2,am2,an2,tran
      call
trans(bl3,bm3,bn3,bl1,bm1,bn1,bl2,bm2,bn2,uran
      call dmmlt1 115,132,.  d,tran,corxyz)
      call dmmlt1 115,132,tgfrco,uran,xtfxyz)
      call
point2 115,117,118,120,121,123,124,126,127,129,130,132,11
6,

+131,2,jl,xmin,xmax,ymin,ymax,corxyz,xtfxyz,patr,frnum,fr
amea,
     +orgpr,defpr

C---------------------------------------------------------
-------------
      if jl.eq.0  goto 221
      print*
220   print*,'Do you wish to view the 3rd plane?'
      read(*,'(a1 ') answer
      if answer.eq.'n'.or.answer.eq.'no'  goto 222

C-----Third layer;taking nodes 133,149 and 165
221   call planes a,b,c,d,e,f,xnod,tgfrco,133,149,165
      call roots(a,b,c,aaa,bbb,ccc,ddd)
      call roots'd,e,f,eee,fff,ggg,hhh)

C-----Finding direction cosines and new z axis
      call dcos1 aaa,bbb,ccc,ddd,al3,am3,an3
      call dcos1 eee,fff,ggg,hhh,bl3,bm3,bn3
      call secos2 133,149,xnod,al1,am1,an1
      call secos2 133,149,tgfr   bl1,bm1,bn1
      call dcos3 al3,am3,an3,a   am1,an1,al2,am2,an2)
      call dcos3 bl3,bm3,bn3,b  ,bm1,bn1,bl2,bm2,bn2
      call
trans al3,am3,an3,al1,am1,an1,al2,am2,an2,tran
      call
trans bl3,bm3,bn3,bl1,bm1,bn1,bl2,bm2,bn2,uran
      call dmmlt1 133,180,xnod,tran,corxyz
      call dmmlt1 133,180,tgfrco,uran,xtfxyz
      call
point1 133,137,138,140,141,145,146,148,149,153,154,156,1
7,

+161,162,164,165,169,170,172,173,177,178,180,135,179,3,jl
,xmin,

+xmax,ymin,ymax,corxyz,xtfxyz,patr,frnum,framea,orgpr,def
pr

C---------------------------------------------------------
-------------
      if jl.eq.0  goto 223
222   print*
```

```
      print*,'Do you wish to view the plane between the
3rd and '
      print*,'4th layers?'
      read(*,'(al)') answer
      if(answer.eq.'n'.or.answer.eq.'no') goto 224

C-----Intermediate layer between 3rd and 4th;taking nodes
181,187 and
C     193
223   call planes(a,b,c,d,e,f,xnod,tgfrco,181,187,193)
      call roots(a,b,c,aaa,bbb,ccc,ddd)
      call roots(d,e,f,eee,fff,ggg,hhh

C-----Finding direction cosines and new z axis
      call dcosl(aaa,bbb,ccc,ddd,al3,am3,an3)
      call dcosl(eee,fff,ggg,hhh,bl3,bm3,bn3)
      call secos2(181,187,xnod,all,aml,anl)
      call secos2(181,187,tgfrco,bll,bml,bnl)
      call dcos3 al3,am3,an3,all,aml,anl,al2,am2,an2)
      call dcos3 bl3,bm3,bn3,bll,bml,bnl,bl2,bm2,bn2)
      call
trans(al3,am3,an3,all,aml,anl,al2,am2,an2,tran)
      call
trans(bl3,bm3,bn3,bll,bml,bnl,bl2,bm2,bn2,uran)
      call dmmltl(181,198,xnod,tran,corxyz)
      call dmmltl 181,198,tgfrco,uran,xtfxyz)
      call
point2(181,183,184,186,187,189,190,192,193,195,196,198,18
2,

+197,3,jl,xmin,xmax,ymin,ymax,corxyz,xtfxyz,patr,frnum,fr
amea,
     +orgpr,defpr)

C---------------------------------------------------------
-------------
      if(jl.eq.0) goto 225
224   print*
      print*,'Do you wish to view the 4th layer?'
      read(*,'(al)') answer
      if(answer.eq.'n'.or.answer.eq.'no') goto 226

C-----Fourth layer;taking nodes 199,215 and 231 for plane
225   call planes(a,b,c,d,e,f,xnod,tgfrco,199,215,231)
      call roots(a,b,c,aaa,bbb,ccc,ddd)
      call roots(d,e,f,eee,fff,ggg,hhh)

C-----Finding direction cosines and new z axis
      call dcosl(aaa,bbb,ccc,ddd,al3,am3,an3)
      call dcosl(eee,fff,ggg,hhh,bl3,bm3,bn3)
      call secos2(199,215,xnod,all,aml,anl)
      call secos2 199,215,tgfrco,bll,bml,bnl)
      call dcos3 al3,am3,an3,all,aml,anl,al2,am2,an2)
      call dcos3 bl3,bm3,bn3,bll,bml,bnl,bl2,bm2,bn2
      call
trans al3,am3,an3,all,aml,anl,al2,am2,an2,tran
      call
trans(bl3,bm3,bn3,bll,bml,bnl,bl2,bm2,bn2,uran
      call dmmltl 199,246,xnod,tran,corxyz)
      call dmmltl 199,246,tgfrco,uran,xtfxyz)
      call
pointl 199,203,204,206,207,211,212,214,215,219,220,222,22
3,

+227,228,230,231,235,236,238,239,243,244,246,201,245,4,jl
,xmin,

+xmax,ymin,ymax,corxyz,xtfxyz,patr,frnum,framea,orgpr,def
pr)

C---------------------------------------------------------
-------------
      if(jl.eq.0) goto 227
226   print*
      print*,'Do you wish to view the plane between the
4th and '
      print*,'5th layers?'
      read(*,'(al)'  answer
      if(answer.eq.'n'.or.answer.eq.'no') goto 228

C-----Intermediate layer between 4th and 5th;taking nodes
247,253 and
C     259
227   call planes(a,b,c,d,e,f,xnod,tgfrco,247,253,259)
      call roots(a,b,c,aaa,bbb,ccc,ddd
      call roots(d,e,f,eee,fff,ggg,hhh)

C-----Finding direction cosines and new z axis
      call dcosl(aaa,bbb,ccc,ddd,al3,am3,an3)
      call dc sl(eee,fff,ggg,hhh,bl3,bm3,bn3
      call s cos2(247,253,xnod,all,aml,anl)
      call secos2(247,253,tgfrco,bll,bml,bnl)
      call dcos3 al3,am3,an3,all,aml,anl,al2,am2,an2)
      call dcos3(bl3,bm3,bn3,bll,bml,bnl,bl2,bm2,bn2)
      call
trans(al3,am3,an3,all,aml,anl,al2,am2,an2,tran)
      call
trans(bl3,bm3,bn3,bll,bml,bnl,bl2,bm2,bn2,uran)
      call dmmltl(247,264,xnod,tran,corxyz)
      call dmmltl 247,264,tgfrco,uran,xtfxyz)
      call
point2 247,249,250,252,253,255,256,258,259,261,262,2o4,24
8,

+263,4,jl,xmin,xmax,ymin,ymax,corxyz,xtfxyz,patr,frnum,fr
amea,
     +orgpr,defpr

C---------------------------------------------------------
-------------
      if(jl.eq.0  goto 229
228   print*
      print*,'Do you wish to view the 5th plane?'
      read(*,' al ') answer
      if(answer.eq.'n'.or.answer.eq.'no'  goto 230

C-----Fifth layer;take nodes 265,281 and 297
229   call planes a,b,c,d,e,f,xnod,tgfrco,265,281,297
      call roots(a,b,c,aaa,bbb,ccc,ddd
      call roots(d,e,f,eee,fff,ggg,hhh

C-----Finding direction cosines and new z axis
      call dcosl(aaa,bbb,ccc,ddd,al3,am3,an3)
      call dcosl(eee,fff,ggg,hhh,bl3,bm3,bn3)
      call secos2 265,281,xnod,all,aml,anl)
      call secos2 265,281,tgfrco,bll,bml,bnl)
      call dcos3 al3,am3,an3,all,aml,anl,al2,am_,an_
      call dcos3 bl3,bm3,bn3,bll,bml,bnl,bl2,bm2,bn_
      call
trans al3,am3,an3,all,aml,anl,al2,am2,an2,tran
      call
trans bl3,bm3,bn3,bll,bml,bnl,bl2,bm2,bn2,uran
      call dmmltl 265,312,xnod,tran,corxyz)
      call dmmltl(265,312,tgfrco,uran,xtfxyz)
      call
pointl 265,269,270,272,273,277,278,280,281,285,_86,288,28
9,

+293,294,296,297,301,302,304,305,309,310,312,267,311,5,jl
,xmin,

+xmax,ymin,ymax,corxyz,xtfxyz,patr,frnum,framea,orgpr,def
pr

C---------------------------------------------------------
-------------
      if(jl.eq.0) goto 231
230   print*
      print*,'Do you wish to view the plane between the
5th and '
      print*,'6th layers?'
      read(*,'(al)') answer
      if(answer.eq.'n'.or.answer.eq.'no') goto 23_

C-----Intermediate layer between 5th and 6th;taking nodes
313,319 and
C     325
231   call planes a,b,c,d,e,f,xnod,tgfrco,313,319,325
      call roots a,b,c,aaa,bbb,ccc,ddd)
      call roots d,e,f,eee,fff,ggg,hhh)

C-----Finding direction cosines and new z axis
      call dcosl aaa,bbb,ccc,ddd,al3,am3,an3
      call dcosl eee,fff,g   hhh,bl3,bm3,bn3
      call secos2 313,319,xnod,all,aml,anl
      call secos2(313,319,tgfrco,bll,bml,bnl)
      call dcos3(al3,am3,an3,all,aml,anl,al2,am2,an2
      call dcos3(bl3,bm3,bn3,bll,bml,bnl,bl2,bm2,bn_
      call
trans(al3,am3,an3,all,aml,anl,al2,am2,an2,tran)
      call
trans(bl3,bm3,bn3,bll,bml,bnl,bl2,bm2,bn2,uran)
      call dmmltl(313,330,xnod,tran,corxyz
      call dmmltl(313,330,tgfrco,uran,xtfxyz)
      call
point2(313,315,316,318,319,321,322,324,325,327,328,330,31
4,

+329,5,jl,xmin,xmax,ymin,ymax,corxyz,xtfxyz,patr,frnum,fr
amea,
     +orgpr,defpr)

C---------------------------------------------------------
-------------
      if(jl.eq.0) goto 233
232   print*
      print*,'Do you wish to view the 6th plane?'
      read(*,'(al)') answer
      if(answer.eq.'n'.or.answer.eq.'no'  goto 234

C-----Sixth layer;take nodes     347 and  63
233   call planes a,b,c,d,e,f    od,tgfrco,331,347,363
      call roots(a,b,c,aaa,bbb ccc,ddd
      call roots d,e,f,eee,fff,ggg,hhh

C-----Finding direction cosines and new z axis
      call dcosl(aaa,bbb,ccc,ddd,al3,am3,an3)
      call dcosl eee,fff,ggg,hhh,bl3,bm3,bn3)
      call secos2 331,347,xnod,all,aml,anl
      call secos2 331,347,tgfrco,bll,bml,bnl)
      call dcos3 al3,am3,an3,all,aml,anl,al2,am2,an_
      call dcos3 bl3,bm3,bn3,bll,bml,bnl,bl2,bm_,bn_
      call
trans(al3,am3,an3,all,aml,anl,al2,am2,an2,tran)
      call
trans bl3,bm3,bn3,bll,bml,bnl,bl2,bm2,bn ,uran)
      call dmmltl 331,378,xnod,tran,c rxyz
      call dmmltl 331,378,tgfrco,uran,xtfxyz)
      call
p intl 331,335,336, 38,3_9,_4_,344,346, 47,351, [illegible] ,_ 4,
5,

+359,360,36_,36_,_67,3o ,37 ,  1,_75,3 o,378, _3,    ,6, 1
,xmin,
+xmax,ymin,ymax,corxyz,xtfxyz,patr,frnum,framea,orgpr,def
pr)

C---------------------------------------------------------
-------------
      if(jl.eq.0) goto 235
234   print*
      print*,'Do you wish to view the plane between the
6th layer '
      print*,'and apex?'
      read(*,'(a1)') answer
      if(answer.eq.'n'.or.answer.eq.'no') goto 236

C-----Intermediate layer between 6th and apex;taking
nodes 379,385 and
C     391
235   call planes(a,b,c,d,e,f,xnod,tgfrco,379,385,391)
      call roots(a,b,c,aaa,bbb,ccc,ddd)
      call roots(d,e,f,eee,fff,ggg,hhh)

C-----Finding direction cosines and new z axis
      call dcos1(aaa,bbb,ccc,ddd,al3,am3,an3)
      call dcos1(eee,fff,ggg,hhh,bl3,bm3,bn3)
      call secos2(379,385,xnod,al1,am1,an1)
      call secos2(379,385,tgfrco,bl1,bm1,bn1)
      call dcos3(al3,am3,an3,al1,am1,an1,al2,am2,an2)
      call dcos3(bl3,bm3,bn3,bl1,bm1,bn1,bl2,bm2,bn2)
      call
trans(al3,am3,an3,al1,am1,an1,al2,am2,an2,tran)
      call
trans(bl3,bm3,bn3,bl1,bm1,bn1,bl2,bm2,bn2,uran)
      call dmmlt1(379,396,xnod,tran,corxyz)
      call dmmlt1(379,396,tgfrco,uran,xtfxyz)
      call
point2(379,381,382,384,385,387,388,390,391,393,394,396,38
0,

+395,6,jl,xmin,xmax,ymin,ymax,corxyz,xtfxyz,patr,frnum,fr
amea,
     +orgpr,defpr)

C---------------------------------------------------------
-------------
236   continue

      if(jl.eq.1) goto 237
        jl=1
        goto 238
237   continue

240   continue
      return
      end


      subroutine propgraph
c
c     It is called by subroutine part3
c
c     It is a menu system for choosing the material
property graphs
c     which are to be produced
c
c     It call routines psing plotall cxview
c
      character ans

2     write(6,9)
      read(5,'(a1)') ans
9     format(////'        Plot Types'/
     +'            ----------'//
     +' 1   Minor E v Frame No.'/
     +' 2   Major E v Frame No.'/
     +' 3   Minor E v Frame No. for all patients'/
     +' 4   Major E v Frame No. for all patients'/
     +' 5   Major E/Minor E v Frame No.'/
     +' 9   cxview'/
     +' Q   Exit to previous menu'//
     +' Enter your chose ? ',$)
      if(ans.eq.'1') then
        call psing(ans)
      elseif(ans.eq.'2') then
        call psing(ans)
      elseif(ans.eq.'3') then
          call plotall(ans)
      elseif(ans.eq.'4') then
          call plotall(ans)
      elseif(ans.eq.'5') then
        call psing(ans)
      el eif(ans.eq.'9') then
        call cxview
      elseif((ans.ne.'Q').and.(ans.ne.'q')) then
        goto 2
      endif

      return
      end


      subroutine
formvals(iv,nof,nofras,volume,smovol,fevolume,
c
c     It is called by psing and plotall
c
c     It returns the data necessary for ploting material
property
```

```
c     graphs.
c
c     It makes no subroutine calls
c

+preses,ex,ey,gxy,gzx,dvdt,dpdt,dvdp,patr,namer,comtr,fil
enam)
      real volume 101 ,smovol(101),fevolume(101),

+preses(101),ex(101),ey 101),cc 101),gxy 101),gzx(101 ,
     +dvdt(101 ,dpdt 101),dvdp(101)
      integer iv 101),ip(101 ,nof,nofras,nfp,i
      character filenam*80,filnam*80,patr*6,atitle*80,
     +namer*20,comtr*20,head5*132,head6*132,
     +head1*132,head2*132,head3*132,head4*132

      filnam='pressure.'//filenam

open(7,file=filnam,access='sequential',form='formatted')
      read(7,*) nofras
      read(7,*)  preses i),i=1,nofras)
      close(7)

      filnam='voludata.'//filenam

open(7,file=filnam,access='sequential',form='formatted')
      read 7,9) patr,namer,comtr,nof,atitle
      read(7,*)
(iv(i),volume(i ,smovol(i),fevolume(i),i 1,nof
      close(7)


      filnam='matprop.'//filenam

open(7,file=filnam,access='sequential',form 'formatted')
      read(7,19) head1,head2,head3,head4,head5,head6

      nfp=0
      do 4 i=1,101
        read(7,29,end=99)
ip(i),ex(i),ey(i),cc(i),gxy(i),gzx(i),
     +dpdt(i),dvdt(i),dvdp(i)
        nfp=nfp+1
4     continue
99    continue
      close(7)

      if(ip(nfp)+1.ne.iv(nof)) then
        write(6,39)
        if(ip(nfp)+1.gt.iv(nof)) then
          nfp=nof-1
        else
            nof=nfp+1
          endif
      endif

      if(ip(1).ne.iv(1 ) then
        write(6,49)
        if(ip 1).gt.iv(1)) then
          nof=nfp+1
          do 5 i=1,nof
            iv(i)=ip(1)+i-1
5           continue
        else
          nfp=nof-1
          do 10 i=1,nfp
            ip(i)=iv(1)+i-1
10          continue
        endif
      endif

9     format(a6,1x,2(a20,1x)/33x,i6/a80)
19    format(a132,5(/a132))
29
format(i5,3x,f9.2,f8.2,7xf7.3,23x,f7.2,7x,f7.2,f8.3,f9.3,
f9.3)
39    format(///
     +' Warning mismatch in volume and pr perty finish
frames')
49    format(//
     +' Warning mismatch in volume and pr perty start
frames')

      return
      end


      subroutine psing ans)
c
c     It is called by subroutine propgraph
c
c     It plots single graphs up n a single page
c
c     It calls subroutines initsp page periph gr up
margin cvtype
c     formvals scales axes7 brkncv cp7pt
c
      real volume 101),smovol(101 ,fev lume 1 1 ,

+preses 101 ,ex(101 ,ey 101 ,gxy(101 ,gzx 1 1 ,rats 1 1 ,
     +dvdt(101 ,dpdt 101 ,dvdp 1 1 ,frame 1 1 ,ymin,ymax
      integer iv 101 ,n f,n fra ,i
      character ans,filenam*  ,patr*6,namer*  ,  mtr*  ,
     +title*20

      write 6,1
1     format
```

```
     +' For which patient do you require material
property graphs? ',$)
      read(*,'(a30)') filenam

      call
formvals(iv,nof,nofras,volume,smovol,fevolume,preses,

+ex,ey,gxy,gzx,dvdt,dpdt,dvdp,patr,namer,comtr,filenam


       if(ans.eq.'5') then
         do 5 i=1,nof-1
           rats(i)=ey(i)/ex(i)
5        continue
         call limexc(rats,nof-1,ymin,ymax)
         ymax=ymax+(ymax-ymin)/7.
         ymin=ymin-(ymax-ymin)/7.
       elseif(ans.eq.'1') then
         call limexc(ex,nof-1,ymin,ymax)
         ymax=ymax+(ymax-ymin)/7.
         ymin=ymin-(ymax-ymin)/7.
         if(ymin.lt.0.0) ymin=0.
       elseif(ans.eq.'2') then
         call limexc(ey,nof-1,ymin,ymax)
         ymax=ymax+(ymax-ymin)/7.
         ymin=ymin-(ymax-ymin)/7.
         if(ymin.lt.0.0) ymin=0.
       endif

       do 10 i=1,nof
           frame(i)=float(iv(i))+0.5
 10    continue

       call initsp
       call page(29.7,21.0)
       call scales(frame(1)-0.5,frame(nof)-
 0.5,1,ymin,ymax,1)
       call axes7('Frame Number','E/(kN/m^2)')
       call cvtype(2)

        if(ans.eq.'1') then
          title='    MINOR MODULAS   '
          call brkncv(frame,ex,nof-1,0)
          call markcv(frame,ex,nof-1,11,1)
        elseif(ans.eq.'2') then
          title='    MAJOR MODULAS   '
          call brkncv(frame,ey,nof-1,0)
          call markcv(frame,ey,nof-1,11,1)
        elseif(ans.eq.'5') then
          title='   MAJOR E/MINOR E  '
          call brkncv(frame,rats,nof-1,0)
          call markcv(frame,rats,nof-1,11,1)
        endif

        call scales(0.,1000.,1,0.,1000.,1)
        call cp7pt(370.,980.,16,title)
        call cp7pt(30.,950.,16,namer)
        call cp7pt(300.,950.,16,patr)
        call cp7pt(30.,920.,16,comtr)

        call endplt

        return
        end


        subroutine plotall(ans)
   c
   c    It is called by subroutine propgraph
   c
   c    It plots multiple graphs upon a single page
   c
   c    It calls subroutines initsp page periph group
   margin cvtype
   c    formvals scales axes7 brkncv cp7pt
   c
         real volume(101),smovol(101),fevolume(101),
        +preses(101),ex(101),ey(101),gxy(101),gzx 101),
        +dvdt(101),dpdt(101),dvdp 101),frame(101),ymin,ymax
         integer iv(101),nof,nofras,i,n
         character ans,filenam(12)*80,patr*6,namer*20,
        +comtr*20,title(12)*5

         data filenam/'andrews','bale','clayton','clayton-
    angina',

    +'cullum','ford','hardy','miller','monckton','rees','west
    ',
        +'west-angina'/

          data
    title/'AN','BA','CL','CL(A)','CU','FO','HA','MI','MO',
        +'RE','WE','WE(A)'/

          call initsp
          call page(21.0 29.7
          call periph(3.
          call group(3,4)
          call margin 1.5)
          call cvtype 2)

          do 5 i=1,12
             call
     formvals(iv,nof,nofras,volume,smovol,fevolume,preses,

     +ex,ey,gxy,gzx,dvdt,dpdt,dvdp,patr,namer,c mtr,filenam i)
     )
```

```
          call limexc ex,nof-1,ymin,ymax
          ymax=ymax+ ymax-ymin  7.
          ymin=ymin- ymax-ymin  7.
          if ymin.lt.0.0  ymin=0.
          do 10 n=1,not
             frame n =float iv n  +0.5
10        continue
          call scales frame 1 -0.5,frame nof)-
0.5,1,ymin,ymax,1
          call axes7 'Frame Number','E  kN m 2 '
c         call axes7 'No.','E')

          if(ans.eq.'3'  then
            call brkncv frame,ex,nof-1,0
          else
            call brkncv frame,ey,nof-1,0)
          endif

          call scales 0.,1000.,1,0.,1000.,1
          call cp7pt 40 .,980.,16,title(i

 5      continue

        call endplt

        return
        end


        subroutine cxview
 c
 c      It is called by subroutine propgraph
 c
 c      It draws the material property graphs
 c
 c      It uses routines g02caf Eg02caf Eg02cbf plote
 limexc scales
 c      axes7 title7 cvtype numbpt brkncv setky linek7
 endplt liminc
 c      polout bothk7 markpt
 c
        integer n
        real pie
        parameter n=151,pie=3.1415926
        double precision dPaoprl(n)
        double precision
 ddpdv(n),result(20 ,dE(n),alogep n),

 +dfevolume(n),pa(n),dpress(n),dsphvol n ,dmidpress n ,

 +dpspmidpdv(n),dspdpdv n),pb n),delldpdv n),dellvol n ,
       +dpellmidpdv(n
        real presss(n ,alp(n ,fevolume(n),v lume n),pie,

 +smovol(n),Eell1(n),Eell2(n),Eell3(n ,Eell4 n ,yy n ,cc n
 ),

 +delta(n),gamma(n),akk n ,elldpdv(n),pellmidpdv n),midfra
 me(n)
         real xmin,xmax,ymin,ymax,ppnew(n),Paoel(n),Pa e2 n)
         real dvdp n),M(n),dpdv n),Eg02caf(n),Eg02cbf n),

 +EPao1(n),Exp(n),press(n ,EPao2(n),p lypresss n ,pdpdv(n)
 ,

 +spress(n),polymidpress(n),pmidpdv n),longdim(n),radius n
 ),

 wallthick(n),midlongdim(n),midrad(n),midwallthick n ,mdv
 olin(n),

 +mdvolout(n),radmidwall n ,cirstl(n),Esph1(n),sphv l n ,E
 sph2(n),

 +pspmidpdv(n),Esph3 n),Esph4 n ,Esph5(n),Esph6 n),cir t2
 n),My(n)

 +,spdpdv(n),bkk n),dd(n),ellvol n ,midshpndex n),shpndex
 n),

 +logep(n),r6,r7,zyx,yxz,nf,bet3,alph4,bet4,yymid,alph1,be
 t1,

 +alph2,bet2,edradius,edpr sss,edwallthick,edl ngdim,ak,x,
 r6exp
         real
 frame(n),midellvol n),coeffs(2 ,Paoprl n ,y,ymid,pk,alph3
 ,
        +beta2,alpha2,alpha1,beta1
         integer
 nof,i,k,n,icount,nofras,ifail,lk,is,j,jj,ii,'c unt,nw
         integer int n),ideg,nl,ij,l,irank 2
         character head1*132,head2*132,head3*13_,head4*1 2
         character head5*132,head6*13
         character filnam*3 ,atitle*80
         character answer,titl*20
         character patr*6,namer*_ ,c mtr*
        +,filenam*8 ,cframe*2
         c mmon/person  filenam,cframe

         write 6,1
  1      f rmat(
        +' For which patient d  y u require material
  pr perty graphs. ',$
          read *,' a   '  filenam
          filnam 'matpr p.'  filenam
```

```
open(7,file=filnam,access='sequential',form='formatted')

C-----Read in E and dvdp
      read(7,3) head1,head2,head3,head4,head5,head6
3     format(a132,5(/a132))

      icount=0
      do 4 i=1,n
        read(7,5,end=99) M(i),dvdp(i)
        icount=icount+1
4     continue
5     format(8x,f9.2,76x,f9.3)

99    continue
      close(7)

      filnam='pressure.'//filenam

open(7,file=filnam,access='sequential',form='formatted')
      read(7,*) nofras
      read(7,*) (presss(i),i=1,nofras)
      close(7)

      filnam='voludata.'//filenam

open(7,file=filnam,access='sequential',form='formatted')
      read(7,19) patr,namer,comtr,nof,atitle
19    format(a6,1x,2(a20,1x)/33x,i6/a80)
      read(7,*)
(int(i),volume(i),smovol(i),fevolume(i),i=1,nof)
      close(7)

C-----Converting fevolumes to double precision
      do 22 i=1,nof

C-----Giving frames a reference
        frame(i)=float(int(i))
22      dfevolume(i)=dble(fevolume(i))

      k=0

      do 6 i=1,icount
        if(M(i).eq.0.) goto 6
        k=k+1
        M(k)=M(i)
        dpdv(k)=1./dvdp(i)
        dE(k)=dble(M(i))
        ddpdv(k)=dble(1./dvdp(i))
6     continue

C-----k is the true number of values there are
C-----Call NAG routines g02caf and g02cbf for regression
relationships
C-----the former with a constant; the latter without
C-----Open file to which all data is sent to
      filnam='results.'//filenam

open(1,file=filnam,access='sequential',form='formatted')
      ifail=0
      call g02caf(k,ddpdv,dE,result,ifail)
      write(1,7) result(6),result(7),result(5)

C-----Saving values for derivation of modulus later
      r6=result(6)
      r7=result(7)

      do 9 i=1,k
9       Eg02caf(i)=result(6)*dpdv(i)+result(7)

7
format('Reg.fit:E=',f12.6,'DP/DV+',f12.6,':Cor=',f12.6)
      call g02cbf(k,ddpdv,dE,result,ifail)
      write(1,10) result(6),result(7),result(5)

      do 11 i=1,k
11      Eg02cbf(i)=result(6)*dpdv(i)

10
format('Reg.fit:E=',f12.6,'DP/DV+',f12.6,':Cor=',f12.6)
      write(1,16)
16    format('RESULTS FROM E=mDP/DV+c and
E=mDP/DV'/10x,'DP/DV',13x,
     +'E',13x,'Eg02caf',13x,'Eg02cbf')
      write(1,12)
(dpdv(i),M(i),Eg02caf(i),Eg02cbf(i),i=1,k)
12    format(4(5x,f12.6))
33    write(6,30)
30    format(////
     +' Wish to view the individual E v Dp/Dv graph? ',$)
      read(*,'(al)') answer
      if(answer.eq.'y'.or.answer.eq.'Y') goto 31
      if(answer.eq.'n'.or.answer.eq.'N') goto 32
      goto 33
31    call plote
      call limexc(dpdv,k,xmin,xmax)
      call limexc(M,k,ymin,ymax)
      call scales(0.,xmax+.1,1,ymin-5.,ymax+5.,1)
      call axes7('CHAMBER STIFFNESS : DP/DV  mmHg/cm3)',
     +'MINOR MODULI : Ex/z (KN/m2)')
      call title7('T','C','MOD (E) v STIFF  DP DV) FOR')
      call cv type(3)

      do 13 i-1,k
13      call numb pt dpdv i),M i ,0,i

      call brkncv(dpdv,Eg02caf,k,0
      call brkncv dpdv,Eg02cbf,k,-1
      call setky 'B','R',2,20
      call linek7 0,'E = DP DV + CONSTANT'
      call linek7 -1,'E = DP DV + 0'
      call endplt

32    continue
      write(6,270
270   format
     +' Wish to view the E v Dp Dv graphs for all the
patients? ',$)
      read(*,' al ') answer
      if answer.eq.'y'.or.answer.eq.'Y'  g to 272
      if answer.eq.'n'.or.answer.eq.'N'  g t  253
      goto 32
272   call plote
      xmin=1234.
      xmax=-1234.
      ymin=xmin
      ymax=xmax
259   write(6,254)
254   format( // 'Enter the patient property filnam :
',$)
      read(*,*) filnam

open(7,file=filnam,access='sequential',form 'f rmatted'
      read(7,3) head1,head2,head3,head4,head5,head6
      icount=0

      do 255 i=1,n
        read(7,5,end=256) M i),dvdp(i
255     icount=icount+1

256   continue
      close(7)
      lk=0

      do 257 i=1,icount
        if(M(i).eq.0.) goto 257
        lk=lk+1
        M(lk)=M(i)
        dpdv(lk)=1./dvdp(i)
257   continue

      call liminc M,lk,ymin,ymax)
      call liminc(dpdv,lk,xmin,xmax)
261   print 258
258   format(////'Another patient property file? ',$)
      read(*,'(al)') answer
      if(answer.eq.'y'.or.answer.eq.'Y') goto 259
      if(answer.eq.'n'.or.answer.eq.'N') got  260
      goto 261
260   call scales xmin,xmax,1,ymin,ymax,1
      call axes7('CHAMBER STIFFNESS : DP DV
(mmHg/cm3)','MINOR MODULI
     +:Ex/z (KN/m. '
      call title7 'T','C','DP DV v Ex z F R ALL
PATIENTS')
      is=0
266   is=is+1
      zyx=1234.
      yxz=-1234.
      print 273
273   format(////' Enter next patient''s filnam : ',$
      read(*,'(a30)') filnam

open(7,file=filnam,access='sequential',form='formatted'
      read(7,3) head1,head2,head3,head4,head5,head6
      icount=0

      do 262 i=1,n
        read(7,5,end=263) M i),dvdp i)
262     icount=icount+1

263   continue
      close(7)
      lk=0

      do 264 i=1,icount
        if M(i).eq.0.  goto 264
        lk=lk+1
        M(lk)-M i)
        dpdv(lk)=1./dvdp i
264   continue

      call cv type 4
      call brkn cv dpdv,M,lk,

      do 813 i 1,lk
        if dpdv i .gt.yxz) jj=i
        if dpdv i .gt.yxz  yxz dpdv i)
        if dpdv i).lt.zyx) ii i
813     if dpdv i .lt.zyx  zyx dpdv i)

      call polout coeffs
      write *,269   coeffs(i ,i 1,
269   format /    ' Constant  ',f12.6,'  C effi ient
=',f12.6
268   write(6,.58
      read *,' al '  answer
      if answer.eq.'y'. r.answer.eq.'Y'  g t  266
      if answer.eq.'n'. r.answer.eq.'N'  g t  26
      g t  268
267   call end plt
253   print*,'P. tting the exp nent al p-v graph a  l g r
v volume'
```

```
C-----Converting to logs
      do 21 i=1,nof
        logep(i)=alog(presss(int(i)))
        alogep(i)=dble(alog(presss(int(i))))
        spress(i)=presss(int(i))
21      dpress(i)=dble(presss(int(i)))

C-----Call to NAG routine
      ifail=0
      call g02caf(nof,dfevolume,alogep,result,ifail)
      write(1,23) result(6),result(7)

C-----Storing values again
      r6exp=result(6)
23    format(////' Slope of logeP v Vol Graph =',f12.6/
     +' Intercept (or logeA)=',f12.6)
      do 24 i=1,nof
      alp(i)=sngl(result(6)*dfevolume(i)+result 7))
24    ppnew(i)=exp((i))

C-----Converting loge of preses
35    print 34
34    format(////' Wish to view the logep v vol graph?
',$)
      read(*,'(a1)') answer
      if(answer.eq.'y'.or.answer.eq.'Y') goto 36
      if(answer.eq.'n'.or.answer.eq.'N') goto 37
      goto 35

C-----Now plot graph
36    call plote
      call limexc(fevolume,nof,xmin,xmax)
      call limexc(alp,nof,ymin,ymax)
      call scales(xmin-1.,xmax+1.,1,ymin-.1,ymax+.1,1)
      call axes7('FEVOLUME (cm3)','LOGe PRESSURE')

      do 25 i=1,nof
        call mark pt(fevolume(i),sngl(alogep(i)),11)
25      call mark pt(fevolume(i),alp(i),0)

      call cv type(3)
      call brkn cv(fevolume,alp,nof,0)
      call cv type(2)
      call brkn cv(fevolume,logep,nof,-1)
      call set ky('B','R',2,8)
      call line k7(0,'EXPECTED')
      call line k7(-1,'ACTUAL')
      call title7('T','C','LOGe P V FEVOL FOR PATIENT')
      call end plt
37    print 250
250   format(////' Wish to view p-v graph if exponential
fit? ',$)
      read(*,'(a1)') answer
      if(answer.eq.'y'.or.answer.eq.'Y') goto 251
      if(answer.eq.'n'.or.answer.eq.'N') goto 252
      goto 37
251   call plote
      xmin-1234.
      xmax=-1234.
      ymin=xmin
      ymax=xmax
      call liminc(volume,nof,xmin,xmax)
      call liminc(smovol,nof,xmin,xmax)
      call liminc(fevolume,nof,xmin,xmax)
      call liminc(spress,nof,ymin,ymax)
      call liminc(ppnew,nof,ymin,ymax)
      call scales(xmin-1.,xmax+1.,1,ymin-1.,ymax+1.,1)
      call axes7('VOLUME (CM3)','PRESSURE (mmHg)')
      call cv type(2)
      call brkn cv(volume,spress,nof,-6)
      call brkn cv(smovol,spress,nof,-4)
      call brkn cv(fevolume,spress,nof,-2)
      call brkn cv(fevolume,ppnew,nof,0)
      call set ky('L','C',4,15)
      call line k7(-6,'VOLUME (VOLUME)')
      call line k7(-4,'SMOVOL (VOLUME)')
      call line k7(-2,'FEVOLUME')
      call line k7(0,'FEVOLUME (PNEW)')
      call title7('T','C','P - V GRAPH FOR EXPONENTIAL v
ACTUAL')
      call end plt
252   print*,'Comparing Modulii between fe and
exponential methods...'
      write 1,39) patr,namer,comtr
39    format(a6,1x,2(a20,1x))
      write(1,128)
128   format(/'Frames',4x'Modulus (fe)',4x,'Modulus
(exp)')
      write(1,38)
(int(i), nt(i+1),M(i),r6*(presss(int(i))+presss(
     +int(i+1)))/2.*r6exp-r7,i=1,k)
38    format(/i3,'-',i3,f12.6,2x,f12.6)

C-----Comparing values as derived for fe model,exponenial
model and
C     Pao model
      do 44 i=1,nof-1

Exp(i)=r6*(presss(int(i))+presss(int(i+1 ))/2.*r6exp-r7
        press(i)-(presss(int(i))+presss(int(i+1)))/2.
44      dmidpress(i)-dble(press(i )

C-----Pao's linear relationship tested
C-----Converting Pressures from mmHg to Dynes cm2
      do 45 i=1,nof-1
        EPao1(i)=press i)*1334.16*24.9-3300 0.
```

```
C-----Converting back to KN m2
        EPao1(i  EPao1 i  100 0.

C-----Now Pao's Quadratic Relation i.e. E=1660.-
139P+320P^2
        EPao2 i =1660.-139*press(i +320.*press i **2.
45      EPao2 i)=EPao2 i  10000.

43    print 40
40    format(    ' Wish to view the E-P curves. ',$
      read(*,'(a1 '  answer
      if answer.eq.'y'.or.answer.eq.'Y'  goto 41
      if(answer.eq.'n'.or.answer.eq.'N'  g to 42
      goto 43
41    call plote
      call limexc(press,nof-1,xmin,xmax
      ymin=1234.
      ymax=-1234.
      call liminc(M,k,ymin,ymax)
      call liminc(Exp,nof-1,ymin,ymax)
      call liminc EPao1,nof-1,ymin,ymax)
      call liminc(EPao2,nof-1,ymin,ymax
      call scales xmin-.5,xmax+.5,1,ymin-1.,ymax+1.,1
      call axes7('PRESSURE  mmHg)','MINOR MODULI
(KN/m2)')
      call cv type 2)
      call brkn cv press,M,k,0

      do 50 i=1,k
50      call numb pt press(i),M(i),0,i

      call brkn cv press,Exp,k,0)

      do 85 i=1,k
85      call numb pt press(i),Exp(i),1,i)

      call brkn cv press,EPao1,k,0)

      do 86 i=1,k
86      call numb pt(press(i ,EPao1 i),4,i)

      call brkn cv press,EPao2,k,0)

      do 87 i=1,k
87      call numb pt(press i),EPao2 i),11,i)

      call set ky('L','C',4,12)
      call both k7(0,0,'E v P  FE)')
      call both k7(0,1,'E v P (EXP)')
      call both k7 0,4,'E v P (PAO1)')
      call both k7 0,11,'E v P (PAO2)')
      call title7('T','C','E - P RELATI NSHIPS : FE & PA
VARIOUS')
      call end plt
42    print 300
300   format(////
     +' Wish to view a 1 E-P curves vs Pao''s
relationships? ',$)
      read(*,'(a1)') answer
      if(answer.eq.'y'.or.answer.eq.'Y') goto 301
      if(answer.eq.'n'.or.answer.eq.'N') goto 302
301   call plote
      xmin=12345678.
      xmax=-12345678.
      ymin-xmin
      ymax=xmax
306   print 254
      read(*,*) filnam

open(7,file=filnam,access='sequential',form 'f rmatted'
      read(7,3) head1,head2,head3,head4,head5,head6
      icount=0

      do 303 i=1,n
        read(7,315,end=304) int(i),M(i)
315     format(i5,3x,f9.2)
303     icount=icount+1

304   continue
      close(7)
      lk=0

      do 305 i=1,ic unt
        if(M(i .eq.0.) g    305
        lk=lk+1
        int(lk =int i)
        M(lk =M i
305   continue

      call liminc M,lk,ymin,ymax

C-----Now the preses
307   print 17
17    format     ' What is the name  f the pres   fi e.
',$
      read *,*  filnam

open(7,file filnam,access 'sequential',f rm 'f rmatted'
      read(7,*) nofras
      read 7,*   presss i),i 1,n fras
      cl se 7

      d  308 i 1,lk
308     press i   presss int i  +presss int i +1     .

      call limin  res ,lk,xmin,xmax
      print 25
      read(*,'(al)') answer
      if answer.eq.'y'.or.answer.eq.'Y'  goto 306
309   call scales(xmin-1.,xmax+1.,1,ymin-1.,ymax+1.,1)
      call axes7('PRESSURE  mmHg)','MINOR MYOCARDIUM
MODULI  Kn m2 ')
      call title7('T','C','E v P - PAO & RESULTS'
      is=0
      jcount 0
314   is=is+1
      print 254
      read(*,*) filnam
      open 7,file=filnam,access='sequential',
     +form='formatted')
      read(7,3) head1,head2,head3,head4,head5,head6
      icount=0

      do 310 i=1,n
        read(7,315,end=311) int(i),M(i
310     icount=icount+1

311   continue
      close(7)
      lk=0

      do 312 i=1,icount
        if(M(i).eq.0.) goto 312
        lk=lk+1
        int lk)=int(i)
        M(lk)-M(i)
312   continue

      print 17
      read(*,*) filnam

open(7,file=filnam,access='sequential',form='formatted')
      read(7,*) nofras
      read 7,*  (presss(i),i=1,nofras)
      close(7)

      do 316 i=1,lk
        jcount=jcount+1
        press i)=(presss(int(i))+presss int(i +1 ) 2.
C-----Store temporarily
        Paopr1 jcount)=press(i)
        dPaopr1(jcount =dble(Paopr1(jcount )
        Paoe1(jcount)=(Paopr1 jcount)*1334.16*24.9-
330000. /10000.
        Paoe2(jcount)=(1660.-139.*Paopr1(jcount +320.*
     +Paopr1(jcount)**2. /10000.
316   continue

      call cv type(2
      call brkn cv(press,M,lk,is)

      do 313 i=1,lk
313     call mark pt(press(i),M(i),is)

      print 258
      if answer.eq.'y'.or.answer.eq.'Y') goto 314
C-----Now plot Pao points via NAG
      ifail=0
      nw=jcount

      print*,' routinue replaced '
c      call m01ajf(dPaopr1,w,ind,indw,jcount,nw,ifail)
      call m01daf(dPaopr1,1,jcount,'A',irank,ifail)

      do 320 i=1,jcount
        Paopr1(i)=sngl(dPaopr1(i))
        Paoe1 i)=(Paopr1(i)*1334.16*24.9-330000.)/10000.
320     Paoe2(i)=(1660.-
139.*Paopr1(i +320.*Paopr1(i)**2. /10000.

      call brkn cv(Paopr1,Paoe1,jcount,0
      call brkn cv(Paopr1,Paoe2,jcount,0)
302   continue

C-----Polynomial fitting as suggested by Mirsky
      write(1,46)
46    format(/'Frames',4x,'Modulus (fe)',4x,'Modulus
(exp)',4x,
     +'Modulus (Pao1)',4x,'Modulus  Pao2)')
      write(1,127)
(int(i),int(i+1),M(i),Exp(i),EPao1(i),EPao2(i),i=1,
     +k
127   format(/i3,'-',i3,f12.6,5x,f12.6,5x,f12.6,5x,f12.6)
52    write(6,51)
51    format(// /
     +' Wish t  fit 4th degree polynomial or other to
presss-volume'
     +' curve? ',$)
      read(*,' al ') answer
      if answer.eq.'y'.or.answer.eq.'Y') goto 53
      if(answer.eq.'n'.or.answer.eq.'N') goto 74
      goto 52
53    write(6,54)
54    format(  //
     +' What degree of polynomial would you like  1
greater than'/
     +' highest power) to fit? ',$)
      read(*,*) ideg

C-----Using NAG routine e02acf and fevolume
      ifail-0
      call e _acf dfevolume,dpress,nof,pa,ideg,ifail

      do 55 i-1,nof
        y-0.
        ymid=0.
        nl=ideg
        do 56 ii=1,ideg-1
          y=y+sngl pa nl *dfevolume i ** nl-1
          ir(i.eq.nof) goto 73

ymid=ymid+sngl(pa nl *  dfevolume(i +dfevolume i+1 ) 2.
     +**(nl-1))
73        continue
56        nl=nl-1
        polypresss i =y+sngl pa 1
        if(i.eq.nof  goto 55
        polymidpress i)=ymid+sngl pa 1
55    continue

58    print 57
57    format  // ' Wish to view graph? ',$
      read(*,'(al)') answer
      if(answer.eq.'y'.or.answer.eq.'Y') goto 59
      if(answer.eq.'n'.or.answer.eq.'N') goto 61
      goto 58
59    call plote
      xmin=1234.
      xmax=-1234.
      call liminc volume,nof,xmin,xmax
      call liminc(smovol,nof,xmin,xmax
      call liminc(fevolume,nof,xmin,xmax)
      call limexc(spress,nof,ymin,ymax
      call scales xmin-1.,xmax+1.,1,ymin-.1,ymax+.1,1
      call axes7 'VOLUME  cm3 ','PRESSURE  mmHg)')
      call cv type(2)
      call brkn cv(volume,spress,nof,-6
      call brkn cv(smovol,spress,nof,-3
      call brkn cv(fevolume,spress,nof,-1
      call brkn cv(fevolume,polypresss,nof,0)

      do 60 i=1,nof
        call mark pt(volume(i),spress i),11)
        call mark pt(smovol(i),spress i),4)
        call mark pt fevolume(i),spress i ,1)
60      call mark pt fevolume(i),polypresss i),11

      call set ky('L','C',4,18)
      call both k7(-6,11,'VOLUME (VOLUME)')
      call both k7(-3,4,'VOLUME  SMOOTH)'
      call both k7(-1,1,'VOLUME (FEVOL)')
      call both k7(0,11,'VOLUME (POLY)')
      call title7('T','C','P - V CURVE FOR PATIENT')
      call end plt

61    write 1,62  ideg
62    format /'Results on smooth p-v curve'/'Degree of
polynomial = ',

+i3/'Pressure',8x,'Polypresss',8x,'Volume',8x,'SV lume',8
x,
     +'Fevolume')
      write(1,63)
(spress(i ,polypresss(i),volume(i ,smovol(i ,fevo
     +lume(i),i=1,nof)
63    format(1x,3f12.2,2f12.3)

      do 64 j=1,ideg
64      print*,'pa(',j,') =',pa j)

68    write(6,65)
65    format(////
     +' Do you wish to continue the polynomial fitting
routine? ',$)
      read(*,'(al)') answer
      if answer.eq.'y'.or.answer.eq.'Y') goto 53
      if(answer.eq.'n'.or.answer.eq.'N') goto 67
      goto 68

67    continue

      do 69 i=1,nof
        y=0.
        ymid=0.
        if(ideg.eq.2) goto 75
        nl=ideg
        do 71 ii=1,ideg-2
          y=y+sngl (nl-1)*pa n   dfevolume i **(nl-2 )
          if(i.eq.nof) got
          ymid=ymid+sngl((nl-
1)*pa(nl *((dfevolume i)+dfevolume
     +(i+1))/2. **(nl-2))
78        continue
71        nl=nl-1
75      continue
        pdpdv i =y+sngl pa(2
        if(i.eq.nof) goto 69
        pmidpdv(i)-ymid+sngl pa 2
69    continue

      write 1,72  ideg
72    format( 'DP DV for fe p-v curve'/' Degree  f
polynomial =',i3

+'Pressure',1 x,'Polypresss',1 x,'FEV lum',1 x,'dvdp',1 x
,'dpdv'
      l-0
      k=0

      do 76 j 1,nof*_-1
        if mod i,2 .eq.   then
```

```
          l=l+1
          write(1,77)
press(l),polymidpress(l),(dfevolume(l)+
     +dfevolume(l+1))/2.,1./pmidpdv(l),pmidpdv(l)
77        format(1x,f8.2,2f12.2,2f12.3)
        else if(mod(j,2).eq.1) then
          k=k+1
          write(1,63)
spress(k),polypresss(k),dfevolume(k),1./pdpdv(k)
     +,pdpdv(k)
        end if
76    continue

74    continue

C-----Mirsky's analysis tested here for sphere and
ellipsoid
82    print 79
79    format(////' Wish to derive values for E from
Mirsky et_al? ',$)
      read(*,'(a1)') answer
      if(answer.eq.'y'.or.answer.eq.'Y') goto 80
      if(answer.eq.'n'.or.answer.eq.'N') goto 81
      goto 82

C-----Calculating stage by stage
80    print*,'If a file needs to be created enter 1 and
input for all'
      print*,'frames the long dimension (base-apex
length),the inner '
      print*,'radius and the wall thickness (in this
order and in mms)
     +'
      print*,'for a chosen plane.However if it already
exists enter
     +2.'
      read(*,'(a1)') answer
      if(answer.eq.'y'.or.answer.eq.'Y') then
        print 96
96      format(/'Give this file a name........',$)
        read(*,*) filnam

open(7,file=filnam,access='sequential',form='formatted')
        print*,'Enter data on next line then...... 0 0 0
when finished
     +)'
        ij=1
129     read *,*)
longdim(ij),radius ij),wallthick ij),shpndex(ij)
        if(longdim ij).eq.0..or.radius(ij).eq.0..or.
     +wallthick ij).eq.0.) goto 130
        ij=ij+1
        goto 129
130     write 7,97
(longdim(i),radius i ,wallthick i),shpndex i),
     +i=1,ij)
        close(7)
      elseif answer.eq.'n'.or.answer.eq.'N') then
        print 98
98      format /'What is the existing filnam ? ',$)
        read(*,*) filnam

open(7,file=filnam,access='sequential',form='formatted')
        ij=0
        do 132 i=1,nof
          read 7,*,end=131)
longdim(i),radius(i),wallthick i),
     +shpndex i)
132       ij=ij+1
131     continue
        close(7)
      end if

C-----Calculating volume for spherical configuration.
      do 83 i=1,ij
      sphvol(i)=4.*pie*radius i)**3./3.
      dsphvol(i)=dble(4.*pie*radius i)**3./3.)

C-----Volumes for ellipsoidal geometry
      ellvol(i)=(4./3.)*pie*((longdim(i)*(7./12.)+1. -
(wallthick(i)/
     +2.))*radius(i)**2.
83
dellvol(i)=dble((4./3.)*pie*((longdim(i)*(7./12.)+1.)-
     +(wallthick(i)/2.))*radius i **2.)

      do 84 i=1,ij-1
        midshpndex(i)=(shpndex i)+shpndex(i+1))/2.
        midframe(i)=(int(i)+int(i+1)  2.

midlongdim(i)=((longdim i)+longdim i+1)) 2.)*(7./12.)+1.
        midrad(i)=((radius(i)+radius(i+1))/2.)

midwallthick(i)=((wallthick(i)+wallthick(i+1))/2.)

C-----Calculating mid-frame volumes
        mdvolin(i)=4.*pie*midrad i)**3./3.

mdvolout(i)=4.*pie*((midrad(i)+midwallthick i))**3.-
     +midrad(i)**3.)/3.
        midellvol(i)=(4. 3.)*pie*(midlongdim(i)-
(midwallthick(i)/2.))*
     +midrad(i)**2.
        radmidwall(i)=midrad(i + midwallthick(i) 2.)

C-----Now calculating stresses for spherical model i.e
based on
```

```
C     stresscir10

cirst1(i)=press(i)*.133416*(mdvolin(i)/mdvolout(i))*(1.+(

+(midrad(i)+midwallthick(i))**3./(2.*radmidwall(i)**3.)))

C-----Now working out dp/dv based on spherical models at
mid points
        spdpdv(i)=(spress(i+1)-spress(i))/(sphvol(i+1)-
sphvol(i))
        dspdpdv(i)=dble(spress(i+1)-
spress(i))/(sphvol(i+1)-sphvol(i))

C-----And working out dp/dv based on ellipsoidal models
at mid points
        elldpdv(i)=(spress(i+1)-spress(i))/(ellvol(i+1)-
ellvol(i))
        delldpdv(i)=dble((spress(i+1)-
spress(i))/(ellvol(i+1)-
     +ellvol(i)))

C-----Now work out E for spherical geometry
84
Esph1(i)=3.*cirst1(i)*(1.+((mdvolout(i)/mdvolin(i))*midra
d(i)*

+*2.)/(midrad(i)**2.+(midrad(i)+midwallthick(i))**2.))*(1
.+
     +(mdvolin(i)/press(i))*spdpdv(i))

C-----Using NAG to calculate dp/dv for polynomial based
on spherical
C     volumes
      print 54
      read(*,*) ideg
      ifail=0
      call e02acf(dsphvol,dpress,ij,pa,ideg,ifail)

C-----Using NAG to calculate dp/dv for polynommial based
on
C     ellipsoidal volumes
      ifail=0
      call e02acf(dellvol,dpress,ij,pb,ideg,ifail)

      do 90 i=1,ij
        ymid=0.
        yymid=0.
        if(ideg.eq.2) goto 91
        nl=ideg
        do 92 ii=1,ideg-2
          if(i.eq.ij) goto 93
          ymid=ymid+sngl((nl-
1)*pa(nl)*((sphvol(i)+sphvol(i+1))/2.)**
     +(nl-2))
          yymid=yymid+sngl((nl-
1)*pb(nl)*((ellvol(i)+ellvol(i+1))/2.)
     +**(nl-2))
93        continue
92        nl=nl-1
91      continue
        if(i.eq.ij) goto 90
        pspmidpdv(i)=ymid+sngl(pa(2))
        pellmidpdv(i)=yymid+sngl(pb(2))
        dpspmidpdv(i)=dble(pspmidpdv(i))
        dpellmidpdv(i)=dble(pellmidpdv(i))
90    continue

C-----Now work out Esph2 for mid poly dp/dv values
      do 94 i=1,ij-1
94
Esph2(i)=3.*cirst1(i)*(1.+((mdvolout(i)/mdvolin(i))*midra
d(i)

+**2.)/(midrad(i)**2.+(midrad(i)+midwallthick(i))**2.))*(
1.+
     +((mdvolin(i))/press(i)*pspmidpdv(i)))

C-----Use NAG routine g02caf to fit regression
relationship for
C     exponential
*     relationship on spherical configuration.
      ifail=0
      call g02caf(ij-1,dmidpress,dspdpdv,result,ifail)
      write(1,103) result(6),result(7)
103   format(/'Alpha on exponential dp/dv curve
(sphere)=',f16.12/
     +'Beta on exponential dp/dv curve (sphere)=',f16.12)
      alph1=sngl(result(6))
      bet1=sngl(result(7))

C-----Use NAG routine for poly now
      ifail=0
      call g02caf(ij-1,dmidpress,dpspmidpdv,result,ifail)
      write(1,104) result(6),result(7)
104   format(/'Alpha on poly exponential sphere
curve=',f16.12/
     +'Beta on poly exponential sphere curve=',f16.12)
      alph2=sngl(result(6))
      bet2=sngl(result(7))

C-----Calculaing Esph3 then
      do 100 i=1,ij-1
100
Esph3(i)=3.*cirst1(i)*(1.+(((mdvolout(i) mdvolin(i))*

+midrad(i)**2.)/(midrad(i)**2.+(midrad(i)+midwallthick(i)
)**2.)))
```

```
     +*(1.+mdvolin(i)*alph1+(mdvolin(i)*bet1/press(i)))

C-----Now working out constants k and c for sphere based
on exp relat.
*-----Need end diastolic presss,radius and wall
thickness.
*  -- Read from file where can store this info.
      print 101
101   format(/'Name of file with end diastolic
presss,radius and wall
     +thickness data ?',$)
      read(*,*) filnam

open(7,file=filnam,access='sequential',form='formatted')
      read(7,*) edradius,edwallthick,edpresss,edlongdim
      close(7)

ak=3.*(1.+alph1*(4./3.)*pie*edradius**3.)*(1.+((4./3.)*pi
e*
     +((edradius+edwallthick)**3.-
edradius**3.)/((4./3.)*pie*edradius*

+*3.)*edradius**2.)/(edradius**2.+(edradius+edwallthick)*
*2.))
      x=3.*bet1*(4./3.)*pie*edradius**3./edpresss

x=x*edpresss*((4./3.)*pie*edradius**3./((4./3.)*pie*((edr
adius+
     +edwallthick)**3.-
edradius**3.)))*(1.+(edradius+edwallthick)**3./
     +(2.*((edradius+(edradius+edwallthick))/2.)**3.))
      x=x*(1.+((4./3.)*pie*((edradius+edwallthick)**3.-
edradius**3.)/

+((4./3.)*pie*edradius**3.)*edradius**2.)/(edradius**2.+
     +(edradius+edwallthick)**2.))

C-----Now work back to find and compare E's for
E=k(sigma)+c
*     and for
.................................E=3(1.+......)*sigma

      do 102 i=1,ij-1
102     Esph4(i)=ak*cirst1(i)+x*.133416

C-----Esph5 for polynomial fit-similar to Esph3
      do 105 i=1,ij-1
105
Esph5(i)=3.*cirst1(i)*(1.+(((mdvolout(i)/mdvolin(i))*midr
ad(i)

+**2.)/(midrad(i)**2.+(midrad(i)+midwallthick(i))**2.)))*
(1.+
     +mdvolin(i)*alph2+(mdvolin(i)*bet2/press(i)))

C-----Now finding k and c for poly fit

pk=3.*(1.+alph2*4.*pie*edradius**3./3.)*(1.+((((4./3.)*pi
e*
     +((edradius+edwallthick)**3.-
edradius**3.)/(4.*pie*edradius**3.))

+*edradius**2.)/(edradius**2.+(edradius+edwallthick)**2.)
))
      y=3.*bet2*(4./3.)*pie*edradius**3./edpresss

y=y*edpresss*((4./3.)*pie*edradius**3./((4./3.)*pie*((edr
adius+
     +edwallthick)**3.-
edradius**3.)))*(1.+(edradius+edwallthick)**3./
     +(2.*((edradius+(edradius+edwallthick))/2.)**3.))
      y=y*(1.+((4./3.)*pie*((edradius+edwallthick)**3.-
edradius**3.)/

+((4./3.)*pie*edradius**3.)*edradius**2.)/(edradius**2.+
     +(edradius+edwallthick)**2.))

C-----Now work back to find and compare E's for
E=k(sigma)+c
*     and for
.................................E=3(1.+......)*sigma

      do 106 i=1,ij-1
106     Esph6(i)=pk*cirst1(i)+y*.133416

C-----Using NAG g02caf to fit regression relation for
ellipsoid
c     configuration
      ifail=0
      call g02caf(ij-1,dmidpress,delldpdv,result,ifail)
      write(1,123) result(6),result(7)
123   format(/'Alpha on ellipsoid dp/dv v p graph
=',f12.6/
     +'Beta on ellipsoid dp/dv v p graph =',f12.6)
      alph3=sngl(result(6)
      bet3=sngl(result(7))

C-----Using routine for the poly fit now
      ifail=0
      call g02caf(ij-
1,dmidpress,dpellmidpdv,result,ifail)
      write(1,124) result(6),result(7)
124   format(/'Alpha on poly ellipsoid dp/dv v p
=',f12.6/
     +'Beta on poly ellipsoid dp/dv v p =',f12.6)
      alph4=sngl(result(6))
      bet4=sngl(result(7))
```

```
C-----Now working out E's based on ellipsoidal geometry
      do 108 j=1,2
        do 107 i=1,ij-1

C-----Working out stress at mid points

cirst2(i)=press(i)*.133416*(radmidwall i  midwallthick i
)*
     +(1.-(midwallthick(i) (2.*radmidwall(i)))-
(radmidwall(i)**2./
     +(2.*midlongdim(i)**2.)))

beta2=(midwallthick(i)/(midrad(i + midwallthick i  2. ) *

+(2.+((midwallthick(i)**2.) (2.* midrad(i)+ midwallthick
i)/2.))*
     +*2.)))+(midwallthick(i)/midlongdim i )

alpha2=(midwallthick(i)/(midrad(i)+ midwallthick(i  2.
*
     +(1.-
(midwallthick(i)/(2.*(midrad(i)+(midwallthick(i  2.) ))+

+(midwallthick(i)**2./(4.*((midrad(i +(midwallthick(i) 2.
))**2.))
     +))+(midwallthick(i)/(2.*midlongdim(i)))*(1.-
(midwallthick(i)/
     +(2.*midlongdim(i))))
          alpha1=3.-
(midwallthick(i)/(midrad(i)+(midwallthick(i) 2.)
     +-
(midwallthick(i)/(2.*midlongdim(i)))+(midwallthick i)**2.
/
     +(2.*(midrad(i)+(midwallthick(i)/2. )**2.))

beta1=(midwallthick(i)/midlongdim(i))+(2.*midwallthick(i)
/
     +(midrad(i)+(midwallthick(i)/2.)))

gamma(i)=((alpha1*(4./3.)*pie*((midrad(i)+midwallthick i
)
     +**3.-midrad(i)**3.))-
(beta1*(4./3.)*pie*midrad(i)**3.))/

+((alpha2*(4./3.)*pie*((midrad(i)+midwallthick i))**3.-
midrad(i)*
     +*3.))+(beta2*(4./3.)*pie*midrad(i)**3.))

delta(i)=((2.+(gamma(i)*midwallthick(i))/(midrad(i)+

+(midwallthick(i)/2.)))*(1.+(midwallthick(i)/(2.*(midrad(
i)+

+(midwallthick(i)/2.))))+(midwallthick(i)**2./(4.* midrad
(i)+

+(midwallthick(i)/2.))**2.))))+((1.+(gamma(i)*midwallthic
k(i))/

+(2.*midlongdim(i)))*(1.+(midwallthick(i)/(2.*midlongdim
i)))))
          cc(i)=1.-
((midrad(i)+(midwallthick(i)/2.))**2.)/(2.*
     +midlongdim(i)**2.)
          yy(i)=1.-
(midwallthick(i)/(2.*(midrad(i)+(midwallthick(i)/
     +2.))*cc(i)))
          if(j.eq.1) then

Eell1(i)=delta(i)*(cirst2(i)/press(i))*(4./3.)*pie*
     +(midlongdim(i)-
(midwallthick(i)/2.))*midrad(i)**2.*elldpdv(i)+
     +(1.+gamma(i))*cirst2(i)/yy(i)

akk(i)=delta(i)*alph3*(4./3.)*pie*((edlongdim*(7./12.)+1.
)
     +-(edwallthick/2.))+((1.+gamma(i))/yy(i))

cc(i)=bet3*delta(i)*(4./3.)*pie*((edlongdim*(7./12.)+1.)-

+(edwallthick/2.))*edpresss*((edradius+(edwallthick/2.))/
     +edwallthick)*(1.-
(edwallthick/(2.*(edradius+(edwa lthick/2.))))-

+((edradius+(edwallthick/2.))**2./(2.*(edlongdim*(7./12.)
+1.))))*
     +.133416/edpresss

C-----Now go back and work out E for ellipsoidal model
using k and c
C      from above
           Eell3(i)=akk(i)*cirst2(i)+cc(i)
          else if(j.eq.2) then

Eell2(i)=delta(i)*(cirst2(i)/press(i))*(4./3.)*pie*
     +(midlongdim(i)-(midwallthick(i)/2.))*midrad(i)**2.*
     +pellmidpdv(i)+(1.+gamma(i))*cirst2(i)/yy(i)

bkk(i)=delta(i)*alph4*(4./3.)*pie*((edlongdim*(7./12.)+1.
)
     +-(edwallthick/2.))+((1.+gamma(i))/yy(i))

dd(i)=bet4*delta(i)*(4./3.)*pie*((edlongdim*(7./12.)+1.)-

+(edwallthick/2.))*edpresss*((edradius+(edwallthick/2.))/
```

```
     +edwallthick)*(1.-
(edwallthick/(2.*(edradius+(edwallthick/2.))))-

+((edradius+(edwallthick/2.))**2./(2.*(edlongdim*(7./12.)
+1.))))*
     +.133416/edpresss

C-----Now go back and work out E for ellipsoidal model
using k and c
C-----from above
            Eell4(i)=bkk(i)*cirst2(i)+dd(i)
          end if
107      continue
108    continue

C-----Plotting graph of various modulii v frames
C-----E(fe)-Esph1-Esph3-Esph4-Eell1-Eell3
136    print 133
133    format(////' Wish to view modulii v frame graph?
',$)
       read(*,'(a1)') answer
       if(answer.eq.'y'.or.answer.eq.'Y') goto 134
       if(answer.eq.'n'.or.answer.eq.'N') goto 137
       goto 136
134    call plote
       ymin=1234.
       xmin=ymin
       ymax=-1234.
       xmax=ymax

       do 139 i=1,ij-1
139      My(i)=M(i)*2.

       call liminc(My,ij-1,ymin,ymax)
       call liminc(Esph1,ij-1,ymin,ymax)
       call liminc(Esph3,ij-1,ymin,ymax)
       call liminc(Esph4,ij-1,ymin,ymax)
       call liminc(Eell1,ij-1,ymin,ymax)
       call liminc(Eell3,ij-1,ymin,ymax)
       call liminc(midframe,ij-1,xmin,xmax)
       call scales(xmin-.5,xmax+.5,1,ymin-1.,ymax+1.,1)
       call axes7('FRAMES','MINOR MODULII (Kn/m2)')
       call cv type(2)
       call brkn cv(midframe,My,ij-1,-6)

       do 143 i=1,ij-1
143      call mark pt(midframe(i),My(i),0)

       call brkn cv(midframe,Esph1,ij-1,2)

       do 140 i=1,ij-1
140      call mark pt(midframe(i),Esph1(i),4)

       call brkn cv(midframe,Esph3,ij-1,4)

       do 141 i=1,ij-1
141      call mark pt(midframe(i),Esph3(i),5)

       call brkn cv(midframe,Esph4,ij-1,-4)

       do 144 i=1,ij-1
144      call mark pt(midframe(i),Esph4(i),1)

       call brkn cv(midframe,Eell1,ij-1,6)

       do 142 i=1,ij-1
142      call mark pt(midframe(i),Eell1(i),6)

       call brkn cv(midframe,Eell3,ij-1,-2)

       do 138 i=1,ij-1
138      call mark pt(midframe(i),Eell3(i),2)

       call set ky('T','L',6,16)
       call both k7(-6,0,'E (FE)')
       call both k7(2,4,'E (SPH:non-exp)')
       call both k7(4,5,'E (SPH:exp))')
       call both k7(-4,1,'E (SPH:exp:ko+c)')
       call both k7(6,6,'E (ELL:exp)')
       call both k7(-2,2,'E (ELL:exp:ko+c)')
       print 990
990    format(/'Name of patient as title? ',$)
       read(*,*) titl
       call title7('T','C',titl)
137    continue
       write(1,126) patr,namer,comtr
126    format(a6,1x,2(a20,1x)/)
       write(1,110)
110    format(//'SPHERICAL GEOMETRY'/)
       write(1,113)
       write(1,111)
(int(i),int(i+1),cirst1(i),M(i)*2.,Esph1(i),
     +Esph3(i),Esph4(i),i=1,ij-1)
113
format('Frames',2x,'Stress(cir)',2x,'E fe)',2x,'E non-
exp)',2x,

+'E exp)',2x,'E exp:ko+c)'/11x,'KN m2',5x,'KN/m2',5x,'KN/
m2',5x,
     +'KN/m2',5x,'KN/m2'/'------',2x,'-----------',2x,'--
---',5x,
     +'-----',5x,'-----',5x,'-----'
111     format i2,'-',i2,f11.2,f10.2,f10.2,f10.2,f10.2
        write 1,116)
116     format(//'ELLIPSOIDAL GEOMETRY'
        write 1,118)
```

```
       write(1,117)
(int(i),int(i+1),cirst2(i),M(i)*2.,Eell1(i),
     +Eell3(i),i=1,ij-1)
118
format('Frames',2x,'Stress(cir)',2x'E(fe)',2x,'E(exp)',2x
,

+'E exp:ko+c)'/11x,'KN/m2',5x,'KN/m2',3x,'KN/m2',5x,'KN/m
2'/
     +'------',2x,'-----------',2x,'-----',3x,'-----
',5x,'-----')
117    format(i2,'-',i2,f11.2,f10.2,f8.2,f10.2)
       close(1)
81     continue
151    print 152
152    format(////
     +' Wish to view all P-V loops i.e.for all patients?
',$)
       read(*,'(a1)') answer
       if(answer.eq.'y'.or.answer.eq.'Y') goto 153
       if(answer.eq.'n'.or.answer.eq.'N') goto 15
       goto 151
153    close(1)
       call plote
       ymax=-1234.
       ymin=1234.
157    print 155
155    format(////' Enter presss filnam for patient : ',$)
       read(*,*) filnam

open(1,file=filnam,access='sequential',form='formatted')
       read(1,*) nf
       read(1,*) (presss(i),i=1,nf)
       close(1)
       call liminc(presss,nf,ymin,ymax)
159    print 156
156    format(////' Another patient presss filnam? ',$)
       read(*,'(a1)') answer
       if(answer.eq.'y'.or.answer.eq.'Y') goto 157
       if(answer.eq.'n'.or.answer.eq.'N') goto 158
       goto 159
158    xmax=-1234.
       xmin=1234.
161    write(6,160)
160    format(////
     +' Enter corresponding volume filnam for patient :
',$)
       read(*,*) filnam

open(1,file=filnam,access='sequential',form='formatted')
       read(1,*) nf
       read(1,1001) (volume(i),i=1,nf)
       close(1)
       call liminc(volume,nf,xmin,xmax)
163    print 162
162    format(////' Another patient volume file? ',$)
       read(*,'(a1)') answer
       if(answer.eq.'y'.or.answer.eq.'Y') goto 161
       if(answer.eq.'n'.or.answer.eq.'N') goto 164
       goto 163
164    call scales(xmin-10.,xmax+10.,1,ymin-1.,ymax+1.,1)
       call axes7('VOLUME (cm3)','PRESSURE (mmHg)')
       call title7('T','C','PRESSURE-VOLUME LOOPS FOR ALL
PATIENTS')
       call cv type(2)
166    print 155
       read(*,*) filnam

open(1,file=filnam,access='sequential',form='formatted')
       read(1,*) nf
       read(1,*) (presss(i),i=1,nf)
       close(1)
       print 160
       read(*,*) filnam

open(1,file=filnam,access='sequential',form='formatted')
       read(1,*) nf
       read(1,1001) (volume(i),i=1,nf)
1001   format 17x,f6.2)
       close(1)
       call brkn cv(volume,presss,nf,0)
       print 156
       read(*,'(a1)') answer
       if(answer.eq.'y'.or.answer.eq.'Y') goto 166
       if(answer.eq.'n'.or.answer.eq.'N') goto 15
       call endplt

15     continue
97     format 4(1x,f7.3))

       return
       end


       subroutine
point1(ia,ib,ic,id,ie,if,ig,ih,ii,ij,ik,il,im,in,io,

+ip,iq,ir,is,it,iu,iv,iw,ix,kk,ll,incr,jl,xmin,xmax,ymin,
ymax,
     +corxyz,xtfxyz,patr,frnum,frame,orgpr,defpr)
C-----Constructs cross-section with twelve points
       double precision corxyz 401,3),xtfxyz(401,3)
       real
tt(13),uu(13),vv(13),ww(13),gg 13),gj(13),hh(13),gk(13),
     +xmin,xmax,ymin,ymax,x,y
       integer
ia,ib,ic,id,ie,if,ig,ih,ii,ij,ik,il,kk,ll,incr,jl,i,im,
      [illegible]
       character
      data [illegible]

       tt 1 =sngl corxyz [illegible]
       uu 1 =sngl xtf yz [illegible] 1
       vv 1 =sngl [illegible] 1
       ww 1 =sngl xtf yz [illegible] 1
       tt 2 =sngl corxyz [illegible] 1
       uu 2 =sngl [illegible] 1
       vv 2 =sngl [illegible]
       ww 2 =sngl [illegible] 1
       tt 3 =sngl corxyz [illegible]
       uu 3 =sngl xtfxyz [illegible]
       vv 3 =sngl cor yz [illegible] 1
       ww 3 =sngl xtfxyz [illegible] 1
       tt 4 =sngl [illegible] 1
       uu 4 =sngl xtf yz [illegible]
       vv 4 =sngl cor yz [illegible] 1
       ww 4 =sngl xtfxyz [illegible]
       tt 5 =sngl [illegible] 1
       uu 5 =sngl [illegible] 1
       vv 5 =sngl corxyz [illegible]
       ww 5 =sngl xtfxyz [illegible] 1
       tt 6 =sngl corxyz [illegible] 1
       uu 6 =sngl xtfxyz [illegible] 1
       vv 6 =sngl corxyz [illegible] 1
       ww 6 =sngl xtf yz [illegible] 1
       tt 7 =sngl corxyz [illegible] 1
       uu 7 =sngl xtfxyz [illegible] 1
       vv 7 =sngl corxyz [illegible]
       ww 7 =sngl xtfxyz [illegible] 1
       tt 8 =sngl corxyz [illegible] 1
       uu 8 =sngl xtfxyz [illegible] 1
       vv 8 =sngl corxyz [illegible] 1
       ww 8 =sngl xtfxyz [illegible] 1
       tt 9 =sngl corxyz [illegible]
       uu 9 =sngl xtfxyz [illegible] 1
       vv 9 =sngl corxyz [illegible] 1
       ww 9 =sngl xtf yz [illegible] 1
       tt 10 =sngl corxyz [illegible] 1
       uu 10 =sngl xtfxyz [illegible] 1
       vv 10 =sngl corxyz [illegible]
       ww 10 =sngl xtfxyz [illegible] 1
       tt 11 =sngl corxyz [illegible] 1
       uu 11 =sngl xtf yz [illegible] 1
       vv 11 =sngl corxyz [illegible]
       ww 11 =sngl xtf yz [illegible] 1
       tt 12 =sngl corxyz [illegible] 1
       uu 12 =sngl xtfxyz [illegible] 1
       vv 12 =sngl [illegible]
       ww 12 =sngl xtfxyz [illegible] 1
       tt 13 =sngl cor yz [illegible]
       uu 13 =sngl xtf [illegible] 1
       vv 13 =sngl corxyz [illegible]
       ww 13 =sngl xtfxyz [illegible] 1
       gg 1 =sngl cor yz [illegible] 1
       hh 1 =sngl [illegible] 1
       gj 1 =sngl corxyz [illegible] 1
       gk 1 =sngl xtf yz [illegible] 1
       gg 2 =sngl corxyz [illegible] 1
       hh 2 =sngl xtf yz [illegible] 1
       gj 2 =sngl corxyz [illegible] 1
       gk 2 =sngl xtf yz [illegible] 1
       gg 3 =sngl cor yz [illegible] 1
       hh 3 =sngl xtfxyz [illegible] 1
       gj 3 =sngl cor yz [illegible] 1
       gk 3 =sngl xtfxyz [illegible] 1
       gg 4 =sngl corxyz [illegible] 1
       hh 4 =sngl xtf yz [illegible] 1
       gj 4 =sngl corxyz [illegible] 1
       gk 4 =sngl xtfxyz [illegible] 1
       gg 5 =sngl corxyz [illegible] 1
       hh 5 =sngl xtfxyz [illegible] 1
       gj 5 =sngl corxyz [illegible] 1
       gk 5 =sngl xtfxyz [illegible] 1
       gg 6 =sngl corxyz [illegible],1
       hh 6 =sngl xtf yz [illegible],1
       gj 6 =sngl corxyz [illegible],1
       gk 6 =sngl xtf yz [illegible] 1
       gg 7 =sngl corxyz [illegible],1
       hh 7 =sngl xtfxyz [illegible] 1
       gj 7 =sngl corxyz [illegible] 1
       gk 7 =sngl xtfxyz [illegible] 1
       gg 8 =sngl corxyz [illegible] 1
       hh 8 =sngl xtf yz [illegible] 1
       gj 8 =sngl corxyz [illegible] 1
       gk 8 =sngl xtfxyz [illegible] 1
       gg 9 =sngl corxyz [illegible] 1
       hh 9 =sngl xtfxyz [illegible] 1
       gj 9 =sngl corxyz [illegible],1
       gk 9 =sngl xtfxyz [illegible],2
       gg 10 =sngl corxyz [illegible] 2
       hh 10 =sngl xtfxyz [illegible],2
       gj 10 =sngl corxyz [illegible],2
       gk 10 =sngl xtfxyz [illegible],2
       gg 11 =sngl corxyz [illegible],2
       hh 11 =sngl xtfxyz [illegible],2
       gj 11 =sngl corxyz [illegible],2
       gk 11 =sngl xtfxyz [illegible],2
       gg 12 =sngl corxyz [illegible]
       hh 12 =sngl xtfxyz [illegible] 2
       gj 12 =sngl corxyz [illegible] 2
       gk 12 =sngl xtfxyz [illegible] 2
       gg 13 =sngl corxyz [illegible] 2
       hh 13 =sngl xtfxyz [illegible],
       gj 13 =sngl [illegible]
```

```
      gk 13 =sngl xtf yz [illegible]

      do 1 [illegible]
        if [illegible] xmax [illegible]
        if [illegible] xmax [illegible]
        if [illegible] xmin [illegible]
        if [illegible] xmin [illegible]
        if [illegible] ymax [illegible]
        if [illegible] ymax [illegible]
        if [illegible] ymin [illegible]
1       if [illegible] ymin [illegible]

C------ [illegible] first time
      if [illegible] eq [illegible]
10    print*
      call [illegible]

      write 6 [illegible]
      read [illegible] ans
      if  ans.eq.'F' .or. ans.eq.'f' [illegible]
        call [illegible]
      else
        call [illegible]
      endif

      call [illegible]
      call scales xmin-.0?5 xmax-. [illegible] ymin-
.0 5 ymax-[illegible] 1
      call axes [illegible]
      call scales [illegible]

      call cp pt 5. 25. [illegible]
      call cp pt 5. 15. [illegible] frame
      call cp pt [illegible] 15. [illegible] frame
      call [illegible]
      call break
      call [illegible] 21.. 16.5 [illegible]

C------Black pen  o.1
      call pen 1
      call [illegible] 21.. 16.5 [illegible]
      call cp pt 20. 15. 1 [illegible]

C------Red pen  No.2
      call pen 2
      call break
      call [illegible] 6.5 [illegible]
      call [illegible] 6.5,-1
      call cp pt 225. 5.,16 [illegible]

C------Back to Black Pen
      call pen 1

      call scales xmin-...5,xmax-. 5 1 ymin-
.015,ymax-.1 5,1
      call cvtype 2
      call break

C------Ensure black pen is in holder no.1 on [illegible] it
being used
      call pen 1
      call draw cv tt gg 13
      call draw cv vv gj 13
      call break
      call join pt tt 1 ,gg 1
      call join pt vv 1 ,gj 1
      call break
      call join pt tt 3 ,gg 3
      call join pt vv 3 ,gj 3
      call break
      call join pt tt 5 ,gg 5
      call join pt vv 5 ,gj 5
      call break
      call join pt tt 7 ,gg 7
      call join pt vv   ,gj 7
      call break
      call join pt tt 9(,gg 9
      call join pt vv 9 ,gj 9
      call break
      call join pt tt 11 ,gg 11
      call join pt vv 11 ,gj 11
      call break
      call join pt tt 13  gg 13
      call join pt vv 13 ,g  1
      call break

C------Red pen called now i.e. no.2
      call pen 2
      call brkn cv u ,hh,13,-1
      call brkn cv ww,gk,13,-1
      call break
      call join pt uu 1 ,hh 1
      call join pt ww 1 ,gk 1
      call break
      call join pt uu 3 ,hh 3
      call join pt ww 3 ,gk 3
      call break
      call join pt uu 5 ,hh 5
      call join pt ww 5 ,gk 5
      call break
      call join pt uu 7 ,hh 7
      call join pt ww 7 ,gk
      call break
      call join pt uu 9 ,hh 9
      call join pt ww 9 ,gk 9
      call break
      call join pt uu 11 ,hh 11
      call join pt ww 11 ,gk 11
      call break
      call join pt uu 13  hh 13
      call join pt ww 13  gk 13
      call break

C-----Printing nodal points with initial pen since
initial co-ord.
      call pen 1

      do 9 i=kk,ll,4
          x=sngl corxyz i,1
          y=sngl corxyz i,_
9       call numbpt x,y 4,i

999      call endplt

 19   format
     +' Do you wish output to a File or the Screen: F or
S ? ',$

100   return
      end


      subroutine
point2 ia,ib,ic,id,ie,if ig,ih,ii,ij,_k,il,kk,ll,

+incr,jl,xmin,xmax,ymin,ymax,corxyz,xtfxyz,patr,frnum,fra
me,
     +orgpr,defpr
C-----Constructs cross-section with six points
      double precision corxyz 401,3 ,xtfxyz 4 1,3
      real
tt(7 ,uu 7),vv 7 ,ww 7 ,gg 7 ,gj 7 ,hh 7 ,gk 7 ,xmin,
     +xmax,ymin,ymax,x,y
      integer
ia,ib,ic,id,ie,if,ig,ih,ii,ij,ik,il,kk,ll,incr,jl,i
      character patr*6,frnum*12,orgpr*16,defpr*20,frame*2

      tt 1 =sngl corxyz ia,1
      uu 1 =sngl xtfxyz ia,1
      vv 1 =sngl corxyz ib,1
      ww 1 =sng_ xtfxyz ib,1
      tt 2 =sngl corxyz ic,1
      uu 2 =sngl xtfxyz ic,1
      vv(2 =sngl corxyz id,1
      ww _ =sngl xtfxyz id,1
      tt 3 =sngl corxyz ie,1
      uu 3 =sngl xtfxyz ie,1
      vv 3 =sngl corxyz if,1
      ww 3 =sngl xtfxyz if,1
      tt 4 =sngl corxyz ig,1
      uu 4 =sngl xtfxyz ig,1
      vv 4 =sngl corxyz ih,1
      ww 4 =sngl xtfxyz ih,1
      tt 5 =sngl corxyz ii,1
      uu 5 -sngl xtfxyz ii,1
      vv 5 =sngl corxyz ij,1
      ww 5 =sngl xtfxyz ij,1
      tt 6 =sngl corxyz ik,1
      uu 6 =sngl xtfxyz ik,1
      vv 6 =sngl corxyz il,1
      ww 6 =sngl xtfxyz il,1
      tt(7 =sngl corxyz ia,1
      uu 7 =sngl xtfxyz(ia,1
      vv(7 =sngl corxyz ib,1
      ww(7 =sngl xtfxyz(ib,1
      gg(1 =sngl corxyz(ia,2)
      hh 1 =sngl xtfxyz ia,2
      gj 1 =sngl corxyz(ib,2
      gk(1 =sngl xtfxyz ib,2
      gg 2 =sngl corxyz ic,2 )
      hh 2 =sngl xtfxyz ic,2
      gj 2 =sngl corxyz id,2
      gk 2 =sngl xtfxyz id,2))
      gg(3)=sngl corxyz(ie,2))
      hh(3)=sngl xtfxyz(ie,2))
      gj(3)=sngl corxyz(if,2
      gk(3)=sngl xtfxyz if,2
      gg 4)=sngl corxyz ig,2)
      hh 4)=sngl xtfxyz(ig,2
      gj 4 =sngl corxyz ih,2
      gk(4)=sngl xtfxyz ih,2
      gg(5)=sngl corxyz ii,2))
      hh(5)=sngl xtfxyz(ii,2))
      gj 5)=sngl corxyz ij,2))
      gk 5 =sngl(xtfxyz ij,2))
      gg(6)=sngl corxyz(ik,2))
      hh(6)=sn l xtfxyz ik,2)
      gj(6)=sngl corxyz il,2)
      gk(6 =sngl xtfxyz il,2
      gg(7 =sngl corxyz ia,2
      hh(7)=sngl xtfxyz(ia,2
      gj(7)=sngl corxyz(ib 2)
      gk(7)=sngl xtfxyz.ib,2))

      do 1 i=1,7
        if(tt(i).gt.xmax) xmax=tt i)
        if(uu(i).gt.xmax) xmax-uu i)
        if(tt(i).lt.xmin) xmin=tt i)
         if(uu i).lt.xmin  xmin=uu i
        if gg(i .gt.ymax) ymax=gg i
        if(hh i .gt.ymax) ymax=hh i)
        if(gg(i .lt.ymin) ymin=gg i)
1       if(hh(i .lt.ymin) ymin=hh i
```

```
C-----Overriding
      if jl.eq..  goto 1 .
.7    print*
      call initsp

      write 6 19
      read 5,' a1 '  ans
      if  ans.eq.'F' ..r. ans.eq.'f'   then
        call devno 4
      else
        call devno 1
      endif

      call page 29.7 21.
      call scales xmin-..05 xmax+.005 1 ymin-
.005,ymax+.0.5 1
      call axes 'X' 1 '.' 1
      call scales 0.,400. 1 .. 400. 1
      call cp~pt 5. 25. .6 patr
      call cp~pt 5. 15.,16,frnum
      call cp~pt 49. 15. 16 frame
      call xlayer incr
      call break
      call brknpt 21...,16.,.

C-----Black pen  No.1
      call pen 1
      call brknpt 233. 16.,0
      call cp~pt 235.,15. 16 orgpr

C-----Red pen  No.2
      call pen 2
      call break
      call brknpt 210. 6.5 0
      call brknpt 233.,6.5 -1
      call cp~pt 235.,5. 16,defpr

      call scales xmin-.005,xmax+.005 1 ymin-
.005,ymax+.005,1
      call cv type 2
      call break

C-----Black pen called
      call pen 1
      call draw cv tt,gg,7
      call draw cv vv,gj,7
      call break
      call join pt tt 1 ,gg 1),
      call join pt vv 1 ,gj 1'
      call break
      call join pt tt 3 ,gg 3 )
      call join pt vv 3 ,gj 3
      call break
      call join pt tt 5 ,gg 5
      call join pt vv 5),gj 5
      call break
      call join pt tt 7),g  7
      call join pt vv(7),gj(7
      call break

C-----Red pen for displaced profile
      call pen(2
      call brkn cv uu,hh,7,-1)
      call brkn cv ww,gk,7,-1
      call break
      call join pt uu(1 ,hh 1
      call join pt(ww 1 ,gk(1))
      call break
      call join pt(uu 3),hh(3 )
      call join pt(ww(3 ,gk(3))
      call break
      call join pt uu 5 ,hh 5 )
      call join pt(ww(5),gk(5 )
      call break
      call join pt(uu   ,hh(7))
      call join pt ww(7),gk(7)
      call break
      call pen(1)

      do 9 i=kk,ll,3
          x=sngl(corxyz(i,1))
          y=sngl(corxyz(i,2)
9       call numb pt x,y,4,i

      call endplt

 19   format //
     +' Do you wish output to a File or the Screen: F  r
S ? ',$)

100   return
      end


      subroutine planes a,b,c,d,e,f,xnod,tgfrco,i, ,k
      double precision
a(3),b(3),c 3),d 3),e 3 ,f(3),xnod(401,3),
     +tgfrco(401,3)
      integer i,j,k,l

      do 1 l=1,3
        a 1 =xnod i,1
        b 1)=xnod j,1
        c 1)=xnod k,.
        d 1)=tgfrco i,.
        e 1)=tgfrc. j,1
1       f 1 =tgfr-- k,1
```

```
      return
      end


      subroutine xlayer(incr)
c
c     It is called by subroutine cxview
c
c     It writes the title for the plan views of the
ventricle
c
c     It calls the simpleplot routine cp7pt
c
      integer incr

      if(incr.eq.1) then
        call cp7pt(5.,390.,16,
     +'VIEW ON MID-PLANE BETWEEN FIRST AND SECOND
LAYERS')
      elseif(incr.eq.2) then
        call cp7pt(5.,390.,16,
     +'VIEW ON MID-PLANE BETWEEN SECOND AND THIRD
LAYERS')
      elseif(incr.eq.3) then
        call cp7pt(5.,390.,16,
     +'VIEW ON MID-PLANE BETWEEN THIRD AND FOURTH
LAYERS')
      elseif(incr.eq.4) then
        call cp7pt(5.,390.,16,
     +'VIEW ON MID-PLANE BETWEEN FOURTH AND FIFTH
LAYERS')
      elseif(incr.eq.5) then
        call cp7pt(5.,390.,16,
     +'VIEW ON MID-PLANE BETWEEN FIFTH AND SIXTH LAYERS')
      elseif(incr.eq.6) then
        call cp7pt 5.,390.,16,
     +'VIEW ON MID-PLANE BETWEEN SIXTH LAYER AND APEX')
      endif

      return
      end


      subroutine wlayer(incr)
c
c     It is called by subroutine cxview
c
c     It writes the title for the plan views of the
ventricle
c
c     It calls the simpleplot routine cp7pt
c
      integer incr

      if(incr.eq.1  then
        call cp7pt(5.,390.,16,'PLAN VIEW ON FIRST PLANE')
      elseif(incr.eq.2) then
        call cp7pt 5.,390.,16,'PLAN VIEW ON SECOND
PLANE')
      elseif(incr.eq.3  then
        call cp7pt 5.,390.,16,'PLAN VIEW ON THIRD PLANE')
      elseif(incr.eq.4  then
        call cp7pt 5.,390.,16,'PLAN VIEW ON FOURTH
PLANE')
      elseif(incr.eq.5) then
        call cp7pt(5.,390.,16,'PLAN VIEW ON FIFTH PLANE')
      elseif(incr.eq.6) then
        call cp7pt 5.,390.,16,'PLAN VIEW ON SIXTH PLANE')
      endif

      return
      end


      subroutine plote
c
c     It is called by subroutine cxview
c
c     It initialises the plotting parameters of
simpleplot
c
c     It calls the simpleplot routines initsp page picsiz
textmg
c
      call initsp
      call page(29.7,21.)
      call picsiz(29.,20.3)
      call textmg(.9)
      return
      end


      subroutine calmyovol(coord,mvol,wvol)
c
c     It is called by subroutine afevolume
c
c     It calculates the volume of the endocardium form
the
c     FE mesh.
c
c     It calls subroutines vol20b forma vol15w
c
      double precision
coord(401,3),elco(20,3),elco15(15,3),evol,
     +volume,wvol
      real mvol
```

```
      integer elm,elmb(20,60),elmw(15,12)
      data
elmb/1,2,3,6,7,9,10,11,49,50,52,53,67,68,69,72,73,75,76,7
7,

+3,4,5,7,8,11,12,13,50,51,53,54,69,70,71,73,74,77,78,79,9
,10,11,

+14,15,17,18,19,52,53,55,56,75,76,77,80,81,83,84,85,11,12
,13,15,

+16,19,20,21,53,54,56,57,77,78,79,81,82,85,86,87,17,18,19
,22,23,

+25,26,27,55,56,58,59,83,84,85,88,89,91,92,93,19,20,21,23
,24,27,

+28,29,56,57,59,60,85,86,87,89,90,93,94,95,25,26,27,30,31
,33,34,

+35,58,59,61,62,91,92,93,96,97,99,100,101,27,28,29,31,32,
35,36,

+37,59,60,62,63,93,94,95,97,98,101,102,103,33,34,35,38,39
,41,42,

+43,61,62,64,65,99,100,101,104,105,107,108,109,35,36,37,3
9,40,43,

+44,45,62,63,65,66,101,102,103,105,106,109,110,111,41,42,
43,46,

+47,1,2,3,64,65,49,50,107,108,109,112,113,67,68,69,43,44,
45,47,

+48,3,4,5,65,66,50,51,109,110,111,113,114,69,70,71,67,68,
69,72,

+73,75,76,77,115,116,118,119,133,134,135,138,139,141,142,
143,69,

+70,71,73,74,77,78,79,116,117,119,120,135,136,137,139,140
,143,

+144,145,75,76,77,80,81,83,84,85,118,119,121,122,141,142,
143,146,

+147,149,150,151,77,78,79,81,82,85,86,87,119,120,122,123,
143,144,

+145,147,148,151,152,153,83,84,85,88,89,91,92,93,121,122,
124,125,

+149,150,151,154,155,157,158,159,85,86,87,89,90,93,94,95,
122,123,

+125,126,151,152,153,155,156,159,160,161,91,92,93,96,97,9
9,100,

+101,124,125,127,128,157,158,159,162,163,165,166,167,93,9
4,95,97,

+98,101,102,103,125,126,128,129,159,160,161,163,164,167,1
68,169,

+99,100,101,104,105,107,108,109,127,128,130,131,165,166,1
67,170,

+171,173,174,175,101,102,103,105,106,109,110,111,128,129,
131,132,

+167,168,169,171,172,175,176,177,107,108,109,112,113,67,6
8,69,130,

+131,115,116,173,174,175,178,179,133,134,135,109,110,111,
113,114,

+69,70,71,131,132,116,117,175,176,177,179,180,135,136,137
,133,134,

+135,138,139,141,142,143,181,182,184,185,199,200,201,204,
205,207,

+208,209,135,136,137,139,140,143,144,145,182,183,185,186,
201,202,

+203,205,206,209,210,211,141,142,143,146,147,149,150,151,
184,185,

+187,188,207,208,209,212,213,215,216,217,143,144,145,147,
148,151,

+152,153,185,186,188,189,209,210,211,213,214,217,218,219,
149,150,

+151,154,155,157,158,159,187,188,190,191,215,216,217,220,
221,223,

+224,225,151,152,153,155,156,159,160,161,188,189,191,192,
217,218,

+219,221,222,225,226,227,157,158,159,162,163,165,166,167,
190,191,

+193,194,223,224,225,228,229,231,232,233,159,160,161,163,
164,167,
+168,169,191,192,194,195,225,226,227,229,230,233,234,235,
165,166,

+167,170,171,173,174,175,193,194,196,197,231,232,233,236,
237,239,

+240,241,167,168,169,171,172,175,176,177,194,195,197,198,
233,234,

+235,237,238,241,242,243,173,174,175,178,179,133,134,135,
196,197,

+181,182,239,240,241,244,245,199,200,201,175,176,177,179,
180,135,

+136,137,197,198,182,183,241,242,243,245,246,201,202,203,
199,200,

+201,204,205,207,208,209,247,248,250,251,265,266,267,270,
271,273,

+274,275,201,202,203,205,206,209,210,211,248,249,251,252,
267,268,

+269,271,272,275,276,277,207,208,209,212,213,215,216,217,
250,251,

+253,254,273,274,275,278,279,281,282,283,209,210,211,213,
214,217,

+218,219,251,252,254,255,275,276,277,279,280,283,284,285,
215,216,

+217,220,221,223,224,225,253,254,256,257,281,282,283,286,
287,289,

+290,291,217,218,219,221,222,225,226,227,254,255,257,258,
283,284,

+285,287,288,291,292,293,223,224,225,228,229,231,232,233,
256,257,

+259,260,289,290,291,294,295,297,298,299,225,226,227,229,
230,233,

+234,235,257,258,260,261,291,292,293,295,296,299,300,301,
231,232,

+233,236,237,239,240,241,259,260,262,263,297,298,299,302,
303,305,

+306,307,230,234,235,237,238,241,242,243,262,261,263,264,
299,300,

+301,303,304,307,308,309,239,240,241,244,245,199,200,201,
262,263,

+247,248,305,306,307,310,311,265,266,267,241,242,243,245,
246,201,

+202,203,263,264,248,249,307,308,309,311,312,267,268,269,
265,266,

+267,270,271,273,274,275,313,314,316,317,331,332,333,336,
337,339,

+340,341,267,268,269,271,272,275,276,277,314,315,317,318,
333,334,

+335,337,338,341,342,343,273,274,275,278,279,281,282,283,
316,317,

+319,320,339,340,341,344,345,347,348,349,275,276,277,279,
280,283,

+284,285,317,318,320,321,341,342,343,345,346,349,350,351,
281,282,

+283,286,287,289,290,291,319,320,322,323,347,348,349,352,
353,355,

+356,357,283,284,285,287,288,291,292,293,320,321,323,324,
349,350,

+351,353,354,357,358,359,289,290,291,294,295,297,298,299,
322,323,

+325,326,355,356,359,360,361,363,364,365,291,292,293,295,
296,299,

+300,301,323,324,326,327,357,358,359,361,362,365,366,367,
297,298,

+299,302,303,305,306,307,325,326,328,329,363,364,365,368,
369,371,

+372,373,299,300,301,303,304,307,308,309,326,327,329,330,
365,366,

+367,369,370,373,374,375,305,306,307,310,311,265,266,267,
328,329,

+313,314,371,372,373,376,377,381,332,333,307,308,309,311,
312,267,

+368,369,329,330,314,315,373,374,375,377,378,333,334,335
```

```
      data elmw
     +
397,339,331,399,341,333,382,336,379,398,340,332,383,337,3
80,
     +
399,341,333,401,343,335,383,337,380,400,342,334,384,338,3
81,
     +
397,347,339,399,349,341,385,344,382,398,348,340,386,345,3
83,
     +
399,349,341,401,351,343,386,345,383,400,350,342,387,346,3
84,
     +
397,355,347,399,357,349,388,352,385,398,356,348,389,353,3
86,
     +
399,357,349,401,359,351,389,353,386,400,358,350,390,354,3
87,
     +
397,363,355,399,365,357,391,360,388,398,364,356,392,361,3
89,
     +
399,365,357,401,367,359,392,361,389,400,366,358,393,362,3
90,
     +
397,371,363,399,373,365,394,368,391,398,372,364,395,369,3
92,
     +
399,373,365,401,375,367,395,369,392,400,374,366,396,370,3
93,
     +
397,331,371,399,333,373,379,376,394,398,332,372,380,377,3
95,
     +
399,333,373,401,335,375,380,377,395,400,334,374,381,378,3
96

      volume=0.d0
      do 5 k=1,60
        do 10 i=1,20
             idum=elmb i,k)
             do 10 j=1,3
             elco i,j)=coord(idum,j
 10        continue
        call vol20b evol,elco)
        if evol.le.0.d0  then
          print*,' elem,evol ',k,evol
          evol=0.d0
        endif
        volume=volume+evol
 5    continue

      call forma
      wvol=0.d0
      do 25 k=61,72
        elm=k-60
        do 30 i=1,15
             idum=elmw(i,elm
             do 30 j=1,3
             elco15 i,j =coord(idum,j
 30     continue
        call vol15w evol,elco15,elm
        if evol.le.0.d0  then
          print*,' elem,evol ',k,evol
          evol=0.d0
        endif
        wvol=wvol+evol
 25   continue

      mvol=sngl(volume+wvol *1000000.0

      return
      end


      subroutine calsa coord,sara
c
c     It is called by subroutines arevolume and fevolume
c
c     It calculates the surface area of the endocardium
form the
c     FE mesh.
c
c     It calls subroutines sa20b forma sa15w
c
      double precision
coord 401,3),elco(20,3 ,elco15 15,3 ,esara,
     +sara
      integer elm,elmb 20,3  ,elmw 15,6

      data
elmb/3,4,5,7,8,11,12,13,50,51,53,54,69,70,71,73,74,77,78,
79,
     +
11,12,13,15,16,19,20,21,53,54,56,57,77,78,79,81,82,85,86,
87,19,
     +
20,21,23,24,27,28,29,56,57,59,60,85,86,87,89,90,93,94,95,
27,28,
     +
29,31,32,35,36,37,59,60,62,63,93,94,96,97,98,101,102,103,
35,36,
     +
37,39,40,43,44,45,62,63,65,66,101,102,103,105,106,109,11
,111,
      +
43,44,45,47,48,3,4,5,65,66,50,51,109,110,111,113,114,69,7
0,71,69,
      +
70,71,73,74,77,78,79,116,117,119,120,135,136,137,139,140,
143,144,
      +
145,77,78,79,81,82,85,86,87,119,120,122,123,143,144,145,1
47,148,
      +
151,152,153,85,86,87,89,90,93,94,95,122,123,125,126,151,1
52,153,
      +
155,156,159,160,161,93,94,95,97,98,101,102,103,125,126,12
8,129,
      +
159,160,161,163,164,167,168,169,101,102,103,105,106,109,1
10,111,
      +
128,129,131,132,167,168,169,171,172,175,176,177,109,110,1
11,113,
      +
114,69,70,71,131,132,116,117,175,176,177,179,180,135,136,
137,
      +
135,136,137,139,140,143,144,145,182,183,185,186,201,202,2
03,205,
      +
206,209,210,211,143,144,145,147,148,151,152,153,185,186,1
88,189,
      +
209,210,211,213,214,217,218,219,151,152,153,155,156,159,1
60,161,
      +
188,189,191,192,217,218,219,221,222,225,226,227,159,160,1
61,163,
      +
164,167,168,169,191,192,194,195,225,226,227,229,230,233,2
34,235,
      +
167,168,169,171,172,175,176,177,194,195,197,198,233,234,2
35,237,
      +
238,241,242,243,175,176,177,179,180,135,136,137,197,198,1
82,183,
      +
241,242,243,245,246,201,202,203,201,202,203,205,206,209,2
10,211,
      +
248,249,251,252,267,268,269,271,272,275,276,277,209,210,2
11,213,
      +
214,217,218,219,251,252,254,255,275,276,277,279,280,283,2
84,285,
      +
217,218,219,221,222,225,226,227,254,255,257,258,283,284,2
85,287,
      +
288,291,292,293,225,226,227,229,230,233,234,235,257,258,2
60,261,
      +
291,292,293,295,296,299,300,301,233,234,235,237,238,241,2
42,243,
      +
260,261,263,264,299,300,301,303,304,307,308,309,241,242,2
43,245,
      +
246,201,202,203,263,264,248,249,307,308,309,311,312,267,2
68,269,
      +
267,268,269,271,272,275,276,277,314,315,317,318,333,334,3
35,337,
      +
338,341,342,343,275,276,277,279,280,283,284,285,317,318,3
20,321,
      +
341,342,343,345,346,349,350,351,283,284,285,287,288,291,2
92,293,
      +
320,321,323,324,349,350,351,353,354,357,358,359,291,292,2
93,295,
      +
296,299,300,301,323,324,326,327,357,358,359,361,362,365,3
66,367,
      +
299,300,301,303,304,307,308,309,326,327,329,330,365,366,3
67,369,
      +
370,373,374,375,307,308,309,311,312,267,268,269,329,330,3
14,315,
      + 373,374,375,377,378,333,334,335/
       data
elmw/399,341,333,401,343,335,383,337,380,400,342,334,384,
      +
338,381,399,349,341,401,351,343,386,345,383,400,350,342,3
87,346,
      +
384,399,357,349,401,359,351,389,353,386,400,358,350,390,3
54,387,
      +
399,365,357,401,367,359,392,361,389,400,366,358,393,362,3
90,
      +
399,373,365,401,375,367,395,369,392,400,374,366,396,370,3
93,
      +
399,333,373,401,335,375,380,377,395,400,334,374,381,378,3
96/

        sara=0.d0
        do 5 k=2,60,2
          elm=k/2
          do 10 i=1,20
              idum=elmb(i,elm)
              do 10 j=1,3
              elco(i,j)=coord(idum,j)
 10         continue
          call sa20b(esara,elco)
          sara=sara+esara
 5      continue

        call forma
        do 25 k=62,72,2
          elm=k/2-30
          do 30 i=1,15
              idum=elmw(i,elm)
              do 30 j=1,3
              elco15(i,j)=coord(idum,j)
30        continue
          call sa15w(esara,elco15,elm)
          sara=sara+esara
 25     continue

        return
        end


        subroutine sa20b(esara,elco)
c
c       It is called by subroutine calsa
c
c       Calculates the area of the internal face of a
finite
c       element brick
c
c       It calls subroutines lam20b, mxm
c
        double precision hh(14),xil(15),etal(15)
        double precision deter,elco(20,3)
        double precision jay(3,3),esara
        double precision lam(3,20),xi,eta,zeta
        double precision dp(3,15),va(15,15),vinva(15,15)
        integer i
        common/lamcm/lam,xi,eta,zeta,vinva,dp,va

        data hh/8*0.335180055d0,6*0.886426593d0/
        data xil/2*-0.758786911d0,2*0.758786911d0,2*-
0.758786911d0,
      +  2*0.758786911d0,0.d0,-
0.795822426d0,0.d0,0.d0,0.795822426d0,
      + 2*0.d0/
        data etal/4*0.758786911d0,4*-
0.758786911d0,0.795822426d0,0.d0,
      +  0.d0,0.d0,0.d0,-.795822426d0,0.d0/

        esara=0.d0
        do 5 i=1,14
             xi=xil(i)
             eta=etal(i)
             zeta=1.d0
             call lam20b
          call mxm(lam,3,elco,20,jay,3)
          deter=sqrt((jay(2,2)*jay(1,3)-
jay(2,3)*jay(1,2))**2.d0
      +   +(jay(2,3)*jay(1,1)-jay(2,1)*jay(1,3))**2.d0
      +   +(jay(2,1)*jay(1,2)-jay(2,2)*jay(1,1))**2.d0)
          esara=esara+deter*hh(i)/2.d0
 5      continue

        return
        end


        subroutine sa15w(esara,elco15,elm)
c
c       It is called by subroutine calsa
c
c       It calculates the surface area of one face of a 15
noded
c       element
c
c       It calls subroutines lam15w, mxm
c
        double precision lam(3,20),lam8(3,8),xi,eta,zeta
        double precision
dp(3,15),va(15,15),vinva(15,15),lam15(3,15)
        double precision jay(3,3)
        double precision h,h1,h2,deter,esara
        double precision elco15(15,3)
        double precision
hh1(7),hh2(4),xil(7),etal(7),zetal(4)
        integer i,j,elm

        common/lamcm/lam,xi,eta,zeta,vinva,dp,va
        equivalence (lam(1,1),lam8(1,1),lam15(1,1))

        data hh1/.225d0,3*.13239415d0,3*.12593918d0/
        data hh2/2*.347854845137454d0,2*.652145154862546d0/
        data xil/0.0d0,.41042619d0,-.41042619d0,0.0d0,-
.69614048d0,
      + .69614048d0,0.0d0/
      data etal -.3333333333333333d0,2*-.0597158~d0,-
.88056825d0,
     + 2*-.7974_699d0,.59485397d0/
      data zetal .861136311594 53d0,-.861136311594053d0,
     + .339981043584856d0,-.33998104 5856d0

      if(elm.eq.1) call setupvinva

      esara=0.d0
      do 3070 i=1,7
          xi=xil(i
          eta=etal i
          zeta=1.d0
          hl hhl i
          do 3070 j=1,4
            h2-hh2 j
            h=hl*h2
            call lam15w
          call mxm lam15,3,elco15,15,jay,3)
        deter=sqrt( jay 2,2 *jay(1,3 -
jay 2,3)*jay(1,2) **2.d0
     +   + jay 2,3 *jay 1,1)-jay 2,1 *jay 1,3  **2.d0
     +   +(jay 2,1 *jay 1,2)-jay 2,2 *jay 1,1  **2.d0
         esara=esara+deter*h
 3070 continue

      return
      end


      subroutine vol_ b evol,elco
c
c     It is called by subroutine calmyovol
c
c     Calculates the volume of a finite
c     element brick
c
c     It calls subroutines lam20b, mxm
c
      double precision hh 14),xil 15 ,etal 15 ,zetal(15)
      double precision deter,elc  20,3
      double precision jay 3,3 ,evol
      double precision lam 3,2  ,xi,eta,zeta
      double precision dp 3,15 ,va 15,15 ,vinva 15,15
      integer i
      common lamcm lam,xi,eta,zeta,vinva,dp,va

      data hh 8*0.335180055d0,6*0.886426593d0
      data xil 2*-0.758786911d0,2*0.758786911d0,2*-
0.758786911d0,
     +  2*0.758786911d ,0.d ,-
0.795822426d0,0.d , .d0, .7958224_6d0,
     + 2* .d
      data etal 4* .758786911d0,4*-
0.758786911d0, .795 _2426d0, .d ,
     +  0.d0,0.d0,0.d0,-.7958224_6d0,0.d0/
      data zetal - .758786911d0,0.758786911d0,-
0.758786911d ,
     +  .758786911d0,-0.758786911d0,0.758786911d0,
     + -0.758786911d0,0.758786911d ,2*0.d0,-
0.795822426d0,
     + 0.795822426d0,3*0.d0

      evol=0.d0
      do 5 i=1,14
          xi=xil i
          eta=etal i
          zeta=zetal i)
          call lam2 b
        call mxm lam,3,elco,20,jay,3
        deter=jay 1,1 * jay 2,2)*jay 3,3 -
jay 3,2 *jay 2,3
     +-jay 2,1 * jay 1,2 *jay 3,3 -jay 3,2 *jay 1,3)
     ++jay 3,1 * jay 1,2 *jay 2,3 -jay 2,2 *jay(1,3))
        evol=evol+deter*hh i)
 5    continue

      return
      end


      subroutine vol15w evol,elco15,elm
c
c     It is called by subroutine calmyovol
c
c     It calculates the volume of a finite element.
c
c     It calls subroutines setupvinva, lam15w, mxm
      double preci ion lam(3,20),lam8 3,8),xi,eta,zeta
      double pre _sion
dp(3,15),va 15,15 ,vinva(15,15),lam15 3,15)
      double precision jay 3,3
      double precision h,hl,h2,deter,evol
      double precision elco15 15,3
      double precision
hhl 7),hh2 4),xil 7 ,etal(7 ,zetal 4)
      integer i,j,elm

      common/lamcm/lam,xi,eta,zeta,vinva,dp,va
      equivalence (lam(1,1),lam8 1,1),lam15 1,1)

      data hhl .225d0,3*.13239415d0,3*.12593918d0
      data hh2 2*.3478548451374 54d0,2*.6521451548625 46d
      data xil 0. d0,.41042619d0,-.41042619d0,0.0d0,-
.69614 48d0,
     + .6961404 8d ,).0d0
```

```
      data etal -.3333333333333333d0,2*-. 59~158~d0 -
.88056825d0,
     + 2*-.79742699d0,.59485397d0
      data zetal .8611~631159405?d0,-.8611363115?4 53d
     + .339981043584856d ,-.339981~4358 56d

      if elm.eq.1  call setupvinva

      evol=0.d0
      do 30~0 i=1,7
          xi=xil i)
          eta=etal i
          h1=hnl i
          do 3070 j-1,4
            zeta-zetal j
            h2-hh2 j
            h=h1*h2*2.d
            call lam15w
          call mxm(lam15,3,elco15,15,jay,3
          deter=jay(1,1 * jay(2,2 *jay 3,3 -
jay(3,2)*jay(2,3
     +-jay 2,1 * jay 1,2 *jay 3,3 -jay 3,2 *jay 1,3
     ++jay 3,1 * jay 1,2 *jay 2,3 -jay 2,_ *jay 1 3
         evol=evol+deter*h
 3070 continue

      return
      end


      subroutine setupvinva
c
c     It is called by subroutines vol15w and sal5w
c
c     It sets up the element coordinates for the 15 noded
c     wedge element and also constructs the derivitives
c
c     it calls subroutine fminv
c
      double precision
lc15w 3,15 ,deter,xo,eo,zo,va 15,15),
     + vinva 15,15 ,xi,eta,zeta,lam 3,20 ,dp 3,15
      integer j,i,vmwork 15 ,vlwork 15

      common/lamcm/lam,xi,eta,zeta,vinva,dp,va

      do 100 j=1,15
          lc15w 1,j =-1.
          lc15w 2,j =-1.
100     lc15w 3,j =-1.

       lc15w 1,2 =1.
       lc15w 1,3 =0.
       lc15w 1,5 =1.
       lc15w(1,6 =0.
       lc15w 1,7 =0.
       lc15w 1,8)=.5
       lc15w(1,9)=-.5
       lc15w(1,11 =1.
       lc15w(i,12.=0.
       lc15w 1,13)=0.
       lc15w(1,14)=.5
       lc15w 1,15 =-.5
       lc15w(2,3)=1.
       lc15w(2,6)=1.
       lc15w 2,8)=0.
       lc15w(2,9)=0.
       lc15w(2,12 =1.
       lc15w 2,14)=0.
       lc15w(2,15)=0.
       lc15w 3,4 =1.
       lc15w 3,5 =1.
       lc15w(3,6 =1.
       lc15w 3,10 =0.
       lc15w 3,11 =0.
       lc15w(3,12 =0.
       lc15w(3,13 =1.
       lc15w 3,14 =1.
       lc15w 3,15 =1.

       do 10 i=1,15
           xo=lc15w 1,i
           eo=lc15w 2,i)
           zo=lc15w 3,i
           va(i,1 =1.
           va(i,2 =xo
           va(i,3)=eo
           va(i,4)=zo
           va(i,5)=xo*xo
           va(i,6)=eo*eo
           va i,~ =zo*zo
           va i,8 =xo*eo
           va i,9 =xo*zo
           va(i,10 =eo*zo
           va i,11 =xo*eo*zo
           va i,12 =zo*xo*xo
           va i,13 =zo*eo*eo
           va(i,14 =xo*zo*zo
 10       va i,15 =eo*zo*zo

        call fminv va,vinva,15,deter,vlwork,vmwork,15

        return
        end
c*****************************************************
c
c     part 4 the finite element control bit
c
c*****************************************************

      subroutine part4
c
c     called from the main program
c
c     Controls the automated application of the finite
Element
c     process for diastolic and systolic investigations
c
c     It calls subroutines smoolen, afemvol, rdcoords,
femvol,
c     knumb, newfil, finelm, matchops
c
      double precision wnew(401,3),tol
      real
volum(151),preses(151),step,mat(10,9),modrat,stlen,
     +
smolens(151),bdx(4),bdy(4),imod,irat,iact,tec(3,10),sara,
sidx,
     +
tsidx,tsidx,grat,mvol,act2,iart,myovol(151),sidxs(151)
      integer
ifrmno,start,finish,iresp,first,match,nf,typ,nlm,i,
     + nfrmno,last
      character*80
filenam,cframe*2,filnam,title,answer*2,cfrm*2
      logical tof
      common/person/ filenam,cframe

      iresp=2
      first=0
      typ=1
      tol=0.1d0
      imod=5.
      irat=2.
      iact=0.2
      iart=0.15
      last=0
      grat=1.0

      write(6,39)
      read(5,'(a80)') title
      write(6,9)
      read(*,'(a80)') filenam
      write(6,19)
      read(*,*) start
      write(6,29)
      read(*,*) finish

1     write(6,49) typ,imod,iact,iart,grat,irat,tol
      read(5,'(a2)') answer
      if(answer.eq.'1') then
        typ=1
      elseif answer.eq.'2') then
        typ=2
      elseif(answer.eq.'3') then
        typ=3
      elseif(answer.eq.'4') then
        typ=4
      elseif(answer.eq.'5') then
        typ=5
      elseif(answer.eq.'6') then
        typ=6
      elseif(answer.eq.'7') then
        typ=7
      elseif(answer.eq.'8') then
        typ=8
      elseif(answer.eq.'9') then
        typ=9
      elseif(answer.eq.'10') then
        typ=10
      elseif(answer.eq.'m') then
        write(6,109)
          read(5,*) imod
      elseif(answer.eq.'r') then
        write(6,119)
          read(5,*) irat
      elseif(answer.eq.'a') then
        write(6,139)
          read(5,*) iact
      elseif(answer.eq.'x') then
        write(6,169)
          read(5,*) iart
      elseif(answer.eq.'g') then
        write(6,139)
          read(5,*) grat
      elseif(answer.eq.'t') then
        write(6,69)
          read(5,*) tol
      endif
      if((answer.ne.'Q').and.(answer.ne.'q')) goto 1

      write(6,89)
      read(5,'(a1)') answer
      if((answer.eq.'y').or.(answer.eq.'Y')) iresp=1

      write(6,79)
      read(5,'(a1)') answer
      if((answer.eq.'y').or.(answer.eq.'Y')) call
femvol(volum,
     +ifrmno,nlm,iresp)
```

```
      filnam="pressure."//filenam

open(7,file=filnam,access='sequential',form='formatted')
      read(7,*) nf
      read(7,*) (preses(i),i=1,nf)
      close(7)

      if((typ.eq.3).or.(typ.eq.10)) then
        call
smoolen(filenam,smolens,typ,sidxs,myovol,last,start,act2)
      elseif((typ.eq.4).or.((typ.ge.7).and.(typ.le.9)))
then
        last=finish
        finish=start+1
        call
smoolen(filenam,smolens,typ,sidxs,myovol,last,start,act2)
      endif
c
c     setup datafile or update with coordinates for the
next frame
c
      write(6,159) filenam
      do 5 ifrmno=start,finish-1
        print*,' Frame ',ifrmno
        if((typ.ge.3).and.(typ.le.10)) then
          bdx(1)=0.
          bdx(2)=0.
          bdx(3)=0.
          bdx(4)=0.
          bdy(1)=0.
          bdy(2)=0.
          bdy(3)=0.
          bdy(4)=0.
          if((typ.eq.3).or.(typ.eq.10))
stlen=smolens(ifrmno+1)
          if((typ.eq.4).or.((typ.ge.7).and.(typ.le.9)))
     +      stlen=smolens(last)
        endif
        tmvol=myovol(ifrmno)
        if(last.eq.0) then
          tsidx=sidxs(ifrmno+1)
        else
          tsidx=sidxs(last)
        endif

        nlm=0
        call knumb(ifrmno,cframe)
        if((typ.eq.1).or.(typ.eq.6).or.(typ.eq.10)) then
          call
afemvol(wnew,volum,ifrmno,nlm,iresp,filenam,1)
        else
c
c         Check for old displacement file
c
          if(ifrmno.eq.start) then
            nfrmno=ifrmno-1
            call knumb(nfrmno,cfrm)
            filnam="dcoords"//cfrm//"."//filenam
            inquire(FILE=filnam,EXIST=tof)
            if(tof) then
              call rdcoords(wnew,filnam)
            else
              call
afemvol(wnew,volum,ifrmno,nlm,iresp,filenam,1)
            endif
          endif
        endif

        modrat=irat
        call
newfil(2,typ,ifrmno,preses,step,wnew,mat,modrat,imod,
     + last,iact,tec,grat,act2,iart)

        match=0
        stepl=0
        do 10 i=1,50
          write(6,99) i
c         open output files and add headers
          filnam="disp"//cframe//"."//filenam
          open(2,file=filnam,access='sequential',
     + form='formatted')
          write(2,'(a11,i2,1x,a80)') ' Iteration
',i,title
          filnam="output"//cframe//"."//filenam
          open(3,file=filnam,access='sequential',
     + form='formatted')
          write(3,'(1x,a80,/1x,a2,a80)')
title,cframe,filenam
          call finelm
          close(1)
          close(2)
          close(3)
          call
matchops(ifrmno,match,preses,typ,tol,iresp,
     +
step,mat,wnew,stlen,modrat,bdx,bdy,last,tec,sara,sidx,mvo
l,
     + tsidx,tmvol,grat,iart)
c
c       abort conditions for non-convegence
c          if((typ.le.9).and.(grat.gt.50.)) return

if((typ.le.8).and.(typ.ge.5).and.(tec(2,1)*step.gt.2.))
     + return
          if((typ.le.5).and.(mat(1,1)/step.gt.100000.))
return
```

```
         if((typ.le.5).and.(mat(1,1)/step.lt.0.0001))
return
           if(match.eq.1) then
           write(6,15) sara*1000000.
           write(6,16) sidx
           write(6,17) mvol
           write(6,18) tec(2,1)
           write(6,219) iart
           goto 5
         elseif(match.eq.0) then
             print*,' Step is ',step
c
c     update datafile for new modulus
c
             call
newfil(1,typ,ifrmno,preses,step,wnew,mat,modrat,imod,
     + last,iact,tec,grat,act2,iart)
          elseif(match.ne.0) then

if((type.eq.1).or.(type.eq.6).or.(type.eq.10)) then
             print*,' Aborting This frame'
             goto 5
           else
              return
           endif
         endif
         if(i.ge.50) return
10      continue

5     continue

9     format(////
     +' Which patient datafile is to be analysed? ',$
19    format(////
     +' Which is the first frame for analysis? ',$)
15    format(' Surface Area=',f8.2,'mm2')
16    format(' Shape Index=',f8.6)
17    format(' Volume of Myocardium =',f8.3)
18    format(' Contraction Coefficient =',f8.5)
219   format(' Contraction Coefficient Ratio =',f8.5)
29    format(////
     +' Which is the last frame of analysis? ',$)
39    format(////' What title do you wish for this run?
',$)
49    format(////'          Volume Matching Options'/
     +'            ------------------------'//
     +'                Option ',I2,' Active'//
     +' 1  Match TARGET Volumes and use new'/
     +'    frame coords for each new frame'/
     +' 2  Match smoothed FE volumes and use same
coords'/
     +'    for each frame'/
     +' 3  Match smoothed FE volumes and ventricle
length'/
     +' 4  Match first and last diastolic frame volume
and'/
     +'    ventricle length'/
     +' 5  Match smoothed systolic volumes'/
     +' 6  Match target systolic volumes'/
     +' 7  Match two systolic volumes'/
     +' 8  Match two systolic volumes and shape'//
     +' 9  Match two systolic cavity and myocardium
volumes'//
     +' 10 Match target systolic cavity and myocardium
volumes'//
     +' m  Alter initial guess for the modulus',f8.3//
     +' a  Alter initial guess for the Contraction
Coefficient',f8.3//
     +' x  Alter initial guess for the Contraction
Coefficient ratio',f8.3//
     +' g  Alter initial guess for the shear modulus
ratio',f8.3//
     +' r  Alter initial guess for the modulus
ratio',f8.3//
     +' t  Alter the tolerance, presently ',f7.4,'% (max
error)'//
     +' Q  Quit volume matching options'///
     +'    ',$)
69    format(////' What is the new value for the
tolerance? ',$)
109   format(////' What is the new value of the modulus?
',$)
119   format(////' What is the new value of the modulus
ratio? ',$)
139   format(////' What is the new value of the
Contraction Coefficient? ',$)
169   format(////' What is the new value of the
Contraction Coefficient ratio? ',$)
149   fo mat(////' What is the new value of the shear
moulus  atio? ',$)
79    format(////
     +' Do you wish to generate a new template file? ',$)
89    format(////
     +' Do you wish to skip interpolating alternate
nodes'/
     +' when calculating the FE volume? ',$)
99    format(' Iteration',i3)
159   format(//' Patient ',a15/)

      return
      end
c****************************************************
c
c     automated routines
c
c****************************************************
```

```
      subroutine
smoolen(filenam,sm lens,typ,sidxs,myovol,last,
     +start,act2
c
c     It is called by subrotine part4
c
c     It calculates ventricle length, shapindex,
myocardial volume
c     and also fits a straight line through the length
pressure data
c     to give a best fit.
c
c     It makes no subroutine calls
c
      double precision wnod(401,3 ,vo,len,wvol
      real
lens(151),volum 151 ,vols(151),frames 151),sara,sidx,
     +
preses 151 ,press 151),smolens 151 ,ssx,sxy,grad,yint,
     + mvol,myovol 151 ,sidxs 151
      integer i,iresp,nlm,frame,n fras,typ,last,start,j
      character filenam*80,filnam*80

      filnam='pressure.'  filenam

open(7,file=filnam,access='sequential',form 'formatted'
      read 7,*) nofras
      read(7,*)  preses i),i=1,n fras
      close(7)

      filnam='voludata.'//filenam

open(1,file=filnam,access='sequential',form='formatted'
      read(1,9) nof

      do 5 i=1,nof
        read(1,19) frame,vols(i)
        call
afemvol(wnod,volum,frame,nlm,iresp,filenam,0)
        call
afevolume(wnod,vo,len,iresp,sara,sidx,mv 1,wv 1)
        frames(i)=float(frame)
        lens(i)=len
        press(i)=preses(frame)
        myovol(frame)=mvol
        sidxs(frame)=sidx
5     continue
      close(1)

      if typ.eq.3) then
c
c     linear regression
c
        ssx=0.
        sxy=0.
        do 10 i=2,nof
          ssx=ssx+ press(i)-press(1))*(press(i)-press(1)
          sxy=sxy+(press(i)-press(1))* lens(i -lens 1 )
10      continue
        grad=sxy/ssx
        yint=lens(1)-grad*press(1)
        do 15 i=frames(1),frames(nof)
          smolens(i)=yint+grad*preses(i)
15      continue

        print*,' % len inc over selected region ',
     +(lens(nof)-lens(1))/lens(1)*100.
        print*,' grad ',grad
        print*,' yint ',yint

      elseif (typ.eq.8).or.(typ.eq.10)) then
c
c     linear regression
c
        ssx=0.
        sxy=0.
        do 40 i=2,nof
          ssx=ssx+ vols(i)-vols(1 )*(vols(i -v ls(1))
          sxy=sxy+(vols i -vols(1) *(lens i)-lens 1 )
40      continue
        grad=sxy/ssx
        yint=lens 1 -grad*vols 1)
        print*,' Initi     ngths '
        do 45 i=frames(1  frames n f)
          smolens i)=yint+grad*vols i-frames(1)+1)
45      continue

        do 35 i=frames(nof -1,frames 1 ,-1
          j=i-frames 1 +1
          smolens i+1)=sm lens(i+1 +(sngl lens j  -
smolens i )
35      continue
      elseif(typ.eq.4  then
        smolens frames nof))=lens n f
      elseif((typ.eq.7 .or.(typ.eq.9 ) then
        do 20 i=1,nof
          if last.eq.frames i)) then
            smolens last  lens(i
          endir
  20    c ntinue
      endif

9     f rmat  33x,i6
19    f rmat I5,  x,f6._)
```

```
      return
      end


      subroutine rdcoords(wnew,filnam)
c
c     It is called by subroutine part4
c
c     it reads the displaced patient coordinates from a
file.
c
c     It makes no subroutine calls
c
      double precision wnew(401,3)
      integer i
      character*80 filnam,head1,head2


open(7,file=filnam,access='sequential',form='formatted')
      read(7,19) head1,head2
      read(7,20) (wnew(i,1),wnew(i,2),wnew(i,3),i=1,401)
      close(7)

19    format(a80/a80)
20    format(3f18.12)

      return
      end


      subroutine
asplit(wnod,onod,volum,ijkl,ifrmno,patr,namer,
     +comtr,iresp,nlm,wfil)
c
c     It is called by subroutine afemvol
c
c     It turns the 31 element model into a 72 element
model
c
c     It makes no subroutine calls
c
      real volum(151)
      double precision onod(255,3),wnod(401,3)
      integer ijkl,ii,jj,i,j,k,iresp,nlm,ifrmno,wfil
      character filnam*80,filenam*80,patr*6,namer*20,
     +comtr*20,cframe*2
      common/person/ filenam,cframe

C-----Subroutine splits into 2 layer model
C     Program to Change 36 element data to 72
      ii=0
      jj=0

      do 10 i=1,6
        do 20 j=1,6
          do 30 k=1,3
            wnod(ii+1,k)=onod(jj+1,k)
            wnod(ii+3,k)=onod(jj+2,k)
            wnod(ii+5,k)=onod(jj+3,k)
            wnod(ii+2,k)=(wnod(ii+1,k)+wnod(ii+3,k))/2.
            wnod(ii+4,k)=(wnod(ii+3,k)+wnod(ii+5,k))/2.
            wnod(ii+6,k)=onod(jj+4,k)
            wnod(ii+8,k)=onod(jj+5,k)
30          wnod(ii+7,k)=(wnod(ii+6,k)+wnod(ii+8,k))/2.
          jj=jj+5
20        ii=ii+8
        do 40 j=1,6
          do 50 k=1,3
            wnod(ii+1,k)=onod(jj+1,k)
            wnod(ii+3,k)=onod(jj+2,k)
50          wnod(ii+2,k)=(wnod(ii+1,k)+wnod(ii+3,k))/2.
          jj=jj+2
40        ii=ii+3
10    continue

      do 80 k=1,3
        wnod(397,k)=onod(253,k)
        wnod(399,k)=onod(254,k)
        wnod(401,k)=onod(255,k)
        wnod(398,k)=(wnod(397,k)+wnod(399,k))/2.
80      wnod(400,k)=(wnod(399,k)+wnod(401,k))/2.

      if(wfil.eq.1) then
        filnam='coords'//cframe//'.'//filenam

open(9,file=filnam,access='sequential',form='formatted')
        write(9,3) patr,namer,comtr,ifrmno
        write(9,4)
(wnod(i,1),wnod(i,2),wnod(i,3),i,i=1,401)
        close(9)
      endif

3     format(a6,1x,2(a20,1x)/' CO-ORDINATES FOR FRAME NO.
',i2)
4     format(3f18.12,i5)

      return
      end


      subroutine
newfil(type,typ,ifrmno,preses,step,wnew,mat,modrat,
     +imod,last,iact,tec,grat,act2,iart)
c
c     It is called by subroutine part4
c
c     Datafile for finite element analysis is updated
here (XL1-data)
c
c     It makes no subroutine calls
c
      double precision wnod(401,3),wnew(401,3),temp(401)
      real
modrat,mat(10,9),prelem(72),aaa(72),bbb(72),imod,

+ccc(72),presve,step,preses(151),tec(3,10),iact,grat,act2
,iart
      integer
ifrmno,nelem,nonop,mannd,lo,nplc,co,nomat,matno(72),

+elemty(72),codeno(72),angle1(72),angle2(72),angle3(72),t
ype,

+angle4(72),angle5(72),angle6(72),angle7(72),angle8(72),i
frmno,

+angle9(72),nonode(72),seqnon(72),A(72),B(72),C(72),D(72)
,

+E(72),F(72),G(72),H(72),U(72),V(72),W(72),X(72),Y(72),Z(
72),

+O(72),P(60),Q(60),R(60),S(60),T(60),aa(72),bb(72),cc(72)
,dd(72),
     +isw1,isw2,isw3,i,ik,last,isw4,typ
      character title*80,filenam*80,cframe*2

      common/person/ filenam,cframe
c
c     read data from file XL1-data
c
      open(77,file='XL1-
data',access='sequential',form='formatted')
      read(77,41)
title,nelem,nonop,lo,nplc,co,mannd,nomat
      read(77,42) isw1,isw2,isw3,isw4
      read(77,43)
(mat(i,1),mat(i,2),mat(i,3),mat(i,4),mat(i,5),

+mat(i,6),mat(i,7),mat(i,8),mat(i,9),tec(1,i),tec(2,i),te
c(3,i)
     +,i=1,nomat)
      read(77,44)
(wnod(i,1),wnod(i,2),wnod(i,3),temp(i),i=1,401)
      ik=(5*nelem)/6
      do 45 i=1,nelem
        read(77,46)
i,matno(i),elemty(i),codeno(i),angle1(i),

+angle2(i),angle3(i),angle4(i),angle5(i),angle6(i),angle7
(i),
     +angle8(i),angle9(i),nonode(i),seqnon(i)
        if(i.le.ik) read(77,47)
A(i),B(i),C(i),D(i),E(i),F(i),

+G(i),H(i),U(i),V(i),W(i),X(i),Y(i),Z(i),O(i),P(i),Q(i),R
(i),
     +S(i),T(i)
        if(i.gt.ik) read(77,46)
A(i),B(i),C(i),D(i),E(i),F(i),
     +G(i),H(i),U(i),V(i),W(i),X(i),Y(i),Z(i),O(i)
45    continue
      read(77,52) (aa(i),bb(i),cc(i),dd(i),i=1,co)
      read(77,53)
(prelem(i),aaa(i),bbb(i),ccc(i),presve,i=
     +2,72,2)
      close(77)

      act2=0.0

      if(type.eq.1) then
        if((typ.ge.5).and.(typ.le.10)) then
          tec(2,1)=tec(2,1)*step
          tec(1,1)=-tec(2,1)*iart
          tec(3,1)=tec(1,1)

          tec(2,2)=act2
          tec(1,2)=act2
          tec(3,2)=act2

          mat(1,9)=mat(1,1)/(2.0*(1.0+mat(1,4)))
          mat(1,7)=mat(1,9)*modrat*grat
          mat(1,8)=mat(1,7)
        else
          mat(1,1)=mat(1,1)/step
          mat(1,2)=mat(1,1)*modrat
          mat(1,3)=mat(1,1)
          if(modrat.gt.1.) then
            mat(1,4)=0.47
            mat(1,5)=mat(1,4)
            mat(1,6)=mat(1,4)/modrat
          else
            mat(1,6)=0.47
              mat(1,4)=mat(1,6)*modrat
            mat(1,5)=mat 1,6 *modrat
          endif
          mat(1,9)=mat(1,1)/(2.0*(1.0+mat(1,4)))
          mat(1,7)=mat(1,9)*modrat*grat
          mat(1,8)=mat(1,7)
        endif
      elseif(type.eq.2) then
        tec 2,1)=iact
        tec 1,1)=-tec(2,1)*iart
```

```
       tec(3,1)=tec(1,1)

         tec 2,2)=act2
         tec(1,2)=act2
         tec(3,2)=act2

       mat(1,1)=imod
       mat(1,2)=mat 1,1)*modrat
       mat(1,3)=mat(1,1)
       mat(1,5)=mat(1,4)
       mat(1,6)=mat(1,4)/modrat
       if(modrat.gt.1.) then
         mat(1,4)=0.47
         mat(1,5)=mat(1,4)
         mat(1,6)=mat(1,4)/modrat
       else
         mat(1,6)=0.47
           mat(1,4)=mat(1,6)*modrat
         mat(1,5)=mat(1,6)*modrat
       endif
       mat(1,9)=mat(1,1)/(2.0*(1.0+mat(1,4)))
       mat(1,7)=mat(1,9)*modrat*grat
       mat(1,8)=mat(1,7)
       do 5 i=1,401
           wnod(i,1)=wnew(i,1)
           wnod(i,2)=wnew(i,2
           wnod(i,3)=wnew(i,3)
5      continue
       if(last.eq.0) then
         presve=(preses(ifrmno+1)-
preses(ifrmno))*0.133416
       else
           presve=(preses(last)-preses(ifrmno))*0.133416
       endif
      endif

c
c     write revised data back to file XL1-data
c
      title=cframe//' '//filenam

      open(77,file='XL1-
data',access='sequential',form='formatted')
      write(77,41)
title,nelem,nonop,lo,nplc,co,mannd,nomat
      write(77,42) isw1,isw2,isw3,isw4
      write(77,43)
(mat(i,1),mat(i,2),mat(i,3),mat(i,4),mat(i,5),

+mat(i,6),mat(i,7),mat(i,8),mat(i,9),tec(1,i),tec(2,i),te
c(3,i),
     +i=1,nomat)
      write(77,44)
(wnod(i,1),wnod(i,2),wnod(i,3),temp(i),i=1,401)
      do 55 i=1,nelem
        write(77,46)
i,matno(i),elemty(i),codeno(i),angle1(i),

+angle2(i),angle3(i),angle4(i),angle5(i),angle6(i),angle7
(i),
     +angle8(i),angle9(i),nonode(i),seqnon(i)
        if(i.le.ik) write(77,47)
A(i),B(i),C(i),D(i),E(i),F(i),

+G(i),H(i),U(i),V(i),W(i),X(i),Y(i),Z(i),O(i),P(i),Q(i),R
(i),
     +S(i),T(i)
        if(i.gt.ik) write(77,46)
A(i),B(i),C(i),D(i),E(i),F(i),
     +G(i),H(i),U(i),V(i),W(i),X(i),Y(i),Z(i),O(i)
55    continue
      write(77,52) (aa(i),bb(i),cc(i),dd(i),i=1,co)
      write(77,53)
(prelem(i),aaa(i),bbb(i),ccc(i),presve,i=
     +2,72,2)
      close(77)

41
format(a80/i2,',',i3,',',i1,',',i2,',',i2,',',i3,',',i1)
42    format(i1,',',i1,',',i1,',',i1)
43
format(f11.4,',',f11.4,',',f11.4/f6.3,',',f6.3,',',f6.3/

+f11.4,',',f11.4,',',f11.4/f12.5,',',f11.5,',',f11.5)
44    format(f18.12,',',f17.12,',',f17.12,',',f17.12)
46    format(15i4)
47    format(20i4)
52    format(i3,',',i1,',',i1,',',i1)
53    format(f3.0,',',f2.0,',',f2.0,',',f2.0,',',f7.5)

      ret rn
      end


      subroutine
afevolume(array,vo,len iresp,sara,sidx,mvol,wvol)
c
c     It is called by subrotines smoolen and newvol
c
c     cavity volume calculated here
c
c     It calls subroutines rearrange,subdiv,calsa,
c     calmyovol
c
      integer j,idiv,idivi,incr,itmp,iresp
      double precision
array 401,3),CX(109 ,CY 109),CZ 109),volume,
     +
vo,CCX(109),CCY 109),CCZ 109 ,a 3 ,b 3),c 3 ,a1,a2,a3,len
,
     + surfacearea,wvol
      real sara,sidx,mvol

      call rearrange 5,1,11,1,CX,CY,CZ,array)
      call rearrange 8,2,12,1,CX,CY,CZ,array
      call rearrange(51,13,18,2,CX,CY,CZ,array)
      call rearrange 71,19,29,1,CX,CY,CZ,array)
      call rearrange 74,20,30,1,CX,CY,CZ,array)
      call rearrange 117,31,36,2,CX,CY,CZ,array)
      call rearrange 137,37,47,1,CX,CY,CZ,array)
      call rearrange(140,38,48,1,CX,CY,CZ,array)
      call rearrange(183,49,54,2,CX,CY,CZ,array)
      call rearrange 203,55,65,1,CX,CY,CZ,array)
      call rearrange(206,56,66,1,CX,CY,CZ,array)
      call rearrange(249,67,72,2,CX,CY,CZ,array)
      call rearrange(269,73,83,1,CX,CY,CZ,array)
      call rearrange 272,74,84,1,CX,CY,CZ,array)
      call rearrange 315,85,90,2,CX,CY,CZ,array
      call rearrange 335,91,101,1,CX,CY,CZ,array
      call rearrange 338,92,102,1,CX,CY,CZ,array
      call rearrange(381,103,108,2,CX,CY,CZ,array
      CX(109)=array 401,1
      CY(109)=array 401,2)
      CZ(109)=array 401,3)
      call rearrange 1,1,11,1,CCX,CCY,CCZ,array
      call rearrange(6,2,12,1,CCX,CCY,CCZ,array
      call rearrange(49,13,18,2,CCX,CCY,CCZ,array
      call rearrange 67,19,29,1,CCX,CCY,CCZ,array)
      call rearrange(72,20,30,1,CCX,CCY,CCZ,array)
      call rearrange(115,31,36,2,CCX,CCY,CCZ,array
      call rearrange 133,37,47,1,CCX,CCY,CCZ,array
      call rearrange(138,38,48,1,CCX,CCY,CCZ,array
      call rearrange(181,49,54,2,CCX,CCY,CCZ,array)
      call rearrange(199,55,65,1,CCX,CCY,CCZ,array)
      call rearrange 204,56,66,1,CCX,CCY,CCZ,array)
      call rearrange(247,67,72,2,CCX,CCY,CCZ,array)
      call rearrange(265,73,83,1,CCX,CCY,CCZ,array)
      call rearrange(270,74,84,1,CCX,CCY,CCZ,array)
      call rearrange(313,85,90,2,CCX,CCY,CCZ,array)
      call rearrange(331,91,101,1,CCX,CCY,CCZ,array)
      call rearrange(336,92,102,1,CCX,CCY,CCZ,array)
      call rearrange(379,103,108,2,CCX,CCY,CCZ,array)
      CCX(109)=array(401,1)
      CCY(109)=array(401,2)
      CCZ(109)=array(401,3)

C-----Splitting each layer into sub-divisions

C-----Top layer
          volume=0
        idiv=100
        idivi=idiv
        itmp=0*idiv
        call
subdiv(1,13,3,14,5,15,7,16,9,17,11,18,2,20,4,22,

+6,24,8,26,10,28,12,30,CX,CY,CZ,idivi,0,itmp,2,volume,ire
sp,
     +CCX,CCY,CCZ)

C-----Intermediate layers
        incr=idiv
        itmp=0*idiv+1
        call
subdiv(1,13,3,14,5,15,7,16,9,17,11,18,2,20,4,22,

+6,24,8,26,10,28,12,30,CX,CY,CZ,idiv,0,itmp,1,volume,ires
p,
     +CCX,CCY,CCZ)
        itmp=1*idiv+1
        call
subdiv(13,19,14,21,15,23,16,25,17,27,18,29,2,20,4,22,6,

+24,8,26,10,28,12,30,CX,CY,CZ,idiv,incr,itmp,1,volume,ire
sp,
     +CCX,CCY,CCZ)
        itmp=2*idiv+1
        call
subdiv(19,31,21,32,23,33,25,34,27,35,29,36,20,38,22,40,

+24,42,26,44,28,46,30,4 ,CX  Y,CZ,idiv,0,itmp,1,v lume,ir
esp,
     +CCX,CCY,CCZ
        itmp=3*idiv+1
        call
subdiv(31,37,32,39,33,41,34,43,35,45,36,47,20,38,2 ,40,

+24,42,26,44,28,46,30,48,CX,CY,CZ,idiv,incr,itmp,1,v lume
,
     +iresp,CCX,CCY,CCZ)
        itmp=4*idiv+1
        call
subdiv(37,49,39,50,41,51,43,52,45,53,47,54,38,56,40,58,

+42,60,44,62,46,64,48,66,CX,CY,CZ,idiv,0,itmp,1,v l ume,ir
esp,
     +CCX,CCY,CCZ
        itmp-5*idiv+1
        call
subdiv(49,55,50,5 ,51,59,52,61,5 ,63,54,65, 8,56,40, 8,

+42,60,44,6 ,46,64,48,66,CX,CY,CZ,idiv,incr,itmp,1,v lume
,
     +iresp,CCX,C Y,C Z
```

```
        itmp=6*idiv+1
        call
subdiv(55,67,57,68,59,69,61,70,63,71,65,72,56,74,58,76,

+60,78,62,80,64,82,66,84,CX,CY,CZ,idiv,0,itmp,1,volume,ir
esp,
     +CCX,CCY,CCZ)
        itmp=7*idiv+1
        call
subdiv(67,73,68,75,69,77,70,79,71,81,72,83,56,74,58,76,

+60,78,62,80,64,82,66,84,CX,CY,CZ,idiv,incr,itmp,1,volume
,
     +iresp,CCX,CCY,CCZ)
        itmp=8*idiv+1
        call
subdiv(73,85,75,86,77,87,79,88,81,89,83,90,74,92,76,94,

+78,96,80,98,82,100,84,102,CX,CY,CZ,idiv,0,itmp,1,volume,
     +iresp,CCX,CCY,CCZ)
        itmp=9*idiv+1
        call
subdiv(85,91,86,93,87,95,88,97,89,99,90,101,74,92,76,94,

+78,96,80,98,82,100,84,102,CX,CY,CZ,idiv,incr,itmp,1,volu
me,
     +iresp,CCX,CCY,CCZ)
        itmp=10*idiv+1
        call
subdiv(91,103,93,104,95,105,97,106,99,107,101,108,92,109,

+94,109,96,109,98,109,100,109,102,109,CX,CY,CZ,idiv,0,itm
p,
     +1,volume,iresp,CCX,CCY,CCZ)
        itmp=11*idiv+1
        call
subdiv(103,109,104,109,105,109,106,109,107,109,108,109,

+92,109,94,109,96,109,98,109,100,109,102,109,CX,CY,CZ,idi
v,incr,
     +itmp,1,volume,iresp,CCX,CCY,CCZ)

        vo=volume*1000000.D0

C-----S.Area & S.Index calculation
      do 1 j=1,3
        a(j)=array(5,j)
        b(j)=array(21,j)
1       c(j)=array(37,j)

C-----Base to apex length

a1=(array(5,1)+array(8,1)+array(13,1)+array(16,1)+array(2
1,1)+

+array(24,1)+array(29,1)+array(32,1)+array(37,1)+array(40
,1)+
     +array(45,1)+array 48,1))/12.d0

a2=(array(5,2)+array(8,2)+array(13,2)+array(16,2)+array(2
1,2)+

+array(24,2)+array(29,2)+array(32,2)+array(37,2)+array(40
,2)+
     +array(45,2)+array(48,2))/12.d0

a3=(array(5,3)+array(8,3)+array(13,3)+array(16,3)+array(2
1,3)+

+array(24,3)+array(29,3)+array(32,3)+array(37,3)+array(40
,3)+
     +array(45,3)+array(48,3))/12.d0
      len=((abs(a1-array(401,1)))**3.+(abs(a2-
array(401,2)))**3.
     ++(abs(a3-array(401,3)))**3.)**(1./3.)

      call calsa(array,surfacearea)
      call calmyovol(array,mvol,wvol)
      sara=sngl(surfacearea)
      sidx=6.*dsqrt(3.141592654D0)*vo*1.E-
6/(surfacearea**1.5)
c      sidx2=0.75*vo*1.E-
6/(((len/2.)**3.0)*3.141592654D0)

      return
      end


      subroutine
matchops(itrame,match,preses,typ,tol,iresp,step

+,mat,wnod,stlen,modrat,bdx,bdy,last,tec,sara,sidx,mvol,
     + tsidx,tmvol,grat,iart)
c
c     called from subroutine part4
c
c     calls the subroutines volmatch, lenmatch and
dispmat
c
c     controls the application of the different matching
options
c     deffined by the variable typ and when a match is
found saves
c     the displacements by calling dispmat.
c
      double precision
t 1,wnod(401,3),disp(401,3),vo,tv 1
```

```
      real
preses 151),step,mat 10,9 ,stlen,modrat,delvol 151 ,len,

+bdx(4),bdy(4 ,tec(3,10 ,sara,sidx,tsidx,tmvol,mvol,grat,
iart
      integer typ,iresp,iframe,match,i,last
      character*80
filnam,filenam,patr*6,namer*20,comtr*20,cframe*2
      common/person  filenam,cframe

      if((typ.eq.1).or. typ.eq.6)) then
c
c     Inital frame co-ordinates read in here
c
        filnam='coords'  cframe  '.'  filenam

open(9,file=filnam,access='sequential',form-'formatted'
        read(9,3  patr,namer,comtr,iframe
        read(9,5   wnod i,1),wnod ı,2 ,wnod(i,3 ,ı 1,4 1
        close 9)
        call
volmatch iframe,match,preses,typ,tol,iresp,step,mat
     +
,wnod,len,delvol,disp,patr,namer,comtr,last,bdy,bdx,tec,
     + sara,sidx,tsidx,mvol,tmvol,vo,tvol)
      elseif((typ.eq.2).or.(typ.eq.5   then
        call
volmatch(iframe,match,preses,typ,tol,iresp,step,mat
     +
,wnod,len,delvol,disp,patr,namer,comtr,last,bdy,bdx,tec,
     + sara,sidx,tsidx,mvol,tmv l,vo,tvol
      elseif((typ.eq.3).or.(typ.eq.4)) then
        call
volmatch(iframe,match,preses,typ,tol,iresp,step,mat
     +
,wnod,len,delvol,disp,patr,namer,comtr,last,bdy,bdx,tec,
     + sara,sidx,tsidx,mvol,tmvol,vo,tvol)
        write(6,19) stlen*1000.
        call lenmatch(stlen,len,modrat,match,bdx,bdy,t l)
      elseif(typ.eq.7) then
        write(6,19) stlen*1000.
        call
volmatch(iframe,match,preses,typ,tol,iresp,step,mat
     +
,wnod,len,delvol,disp,patr,namer,comtr,last,bdy,bdx,tec,
     + sara,sidx,tsidx,mvol,tmvol,vo,tvol)
      elseif(typ.eq.8) then
        write(6,19) stlen*1000.
        call
volmatch(iframe,match,preses,typ,tol,iresp,step,mat
     +
,wnod,len,delvol,disp,patr,namer,comtr,last,bdy,bdx,tec,
     + sara,sidx,tsidx,mvol,tmvol,vo,tv l)
       if(match.eq.1) then
c         call
matshape(tsidx,sidx,grat,match,bdx,bdy,tol
          call matshape  tlen,len,grat,match,bdx,bdy,t l)
       endif
      elseif(typ.eq.9) then
        write(6,19) stlen*1000.
        call
volmatch(iframe,match,preses,typ,tol,iresp,step,mat
     +
,wnod,len,delvol,disp,patr,namer,comtr,last,bdy,bdx,tec,
     + sara,sidx,tsidx,mvol,tmvol,v ,tvol)
        call
myovol(tmvol,mvol,iart,bdy,bdx,match,tol,vo,tvol)
      elseif(typ.eq.10) then
        filnam='coords'//cframe//'.'//filenam

open(9,file=filnam,access='sequential',form 'f rmatted')
        read(9,3) patr,namer,c mtr,iframe
        read(9,5) (wnod(i,1),wnod(i,2),wnod(i,3),i 1,401)
        close(9)
        write(6,19) stlen*1000.
        call
volmatch(iframe,match,preses,typ,t l,iresp,step,mat
     +
,wnod,len,delvol,disp,patr,namer,comtr,last,bdy,bdx,tec,
     + sara,sidx,tsidx,mvol,tmvol,vo,tvol)
        call
myovol(tmvol,mvol,iart,bdy,bdx,match,tol,vo,tv l
      endif

      if(match.eq.1) then
        call
dispmat(wnod,mat,preses,delvol,iframe,typ,disp,patr,
     + namer,comtr,tec,last)
      endif

3     format(a6,1x,2(a20,1x /28x,i2)
5     format(3f18.12
 19   format(' Target Base to Apex Length ',f7.3,'mms'

      return
      end


      subroutine
matshape tsidx,sidx,grat,match,bdx,bdy,t l
c
c     called by subr utine match ps
c
c     calls n  subr tine
c
c     matched th- sidx u ing a linear  tep meth d and
c     linear interp lati n.
c
      double precision tol
      real tsidx,sidx,grat,bdx(4 ,bdy(4)
      integer match

      if(((abs(tsidx-sidx)/tsidx*100.).gt.tol).and.
     +((abs(bdx(3)-
bdx(4)).ge.tol).or.(bdx(3)*bdx(4).eq.0.))) then
          if(sidx.gt.tsidx) then
            bdx(4)=grat
            bdy(4)=sidx
            if(bdx(3).eq.0.) then
              grat=grat*0.5
            else
              grat=bdx(3)+(bdx(4)-bdx(3))*(tsidx-bdy(3))/
     +(bdy(4)-bdy(3))
            endif
          elseif(sidx.lt.tsidx) then
            bdx(3)=grat
            bdy(3)=sidx
            if(bdx(4).eq.0.) then
              grat=grat*1.5
            else
              grat=bdx(3)+(bdx(4)-bdx(3))*(tsidx-bdy(3))/
     +(bdy(4)-bdy(3))
            endif
          else
            print*,' exact value found'
            return
          endif

      print*,' bounds upper sidx ',bdx(4),bdy(4)
      print*,' grat         sidx ',grat,sidx
      print*,' bounds lower sidx ',bdx(3),bdy(3)

        if(grat.le.0.) then
          match=3
        else
          match=0
        endif
      endif

      bdx(1)=0.
      bdx(2)=0.
      bdy(1)=0.
      bdy(2)=0.

      return
      end


      subroutine
lenmatch(stlen,len,modrat,match,bdx,bdy,tol)
c
c     called by subroutine matchops
c
c     calls no subrotines
c
c     matched the length using a linear step method and
c     quadratic interpolation.
c
      double precision tol
      real stlen,len,modrat,bdx(4),bdy(4)
      integer match

      if((len.lt.bdy(1)).and.(bdx(2).eq.0.)) then
        print*,' '
        print*,' GONE PAST MAXIMUM '
        print*,' '
        match=3
      else
        if(((abs(stlen-len)/stlen*100.).gt.tol).and.
     +((abs(bdx(1)-
bdx(2)).ge.tol).or.(bdx(1)*bdx(2).eq.0.))) then
c        if(((abs(stlen-len)/stlen*100.).gt.tol).or.
c     +(abs(bdx(1)-
bdx(2)).ge.tol*5.).or.(bdx(1)*bdx(2).eq.0.)) then
          if((bdx(1).eq.0.).or.(bdx(2).eq.0.)) then
            if(len.gt.stlen) then
              bdx(2)=modrat
              bdy(2)=len
              if(bdx(1).eq.0.) then
                modrat=modrat-0.5
                if(modrat.le.0.) then
                  modrat=0.01
                endif
              else
                modrat=bdx(1)+(bdx(2)-bdx(1))*(stlen-
bdy(1))/
     +(bdy(2)-bdy(1))
              endif
            elseif(len.lt.stlen) then
              bdx(1)=modrat
              bdy(1 =len
              if(bdx 2).eq.0.) then
c                mod at=modrat+1.
                modrat=modrat+2.
              else
                modrat=bdx(1)+(bdx(2)-bdx(1) *(stlen-
bdy(1))/
     +(bdy(2)-bdy(1))
              endif
            endif
          else
            x1=bdy(1)-stlen
            x2-len-stlen
            x3=bdy(2)-stlen
            xx1= x1*x2*x2-x2*x1*x1
            xx3= x3*x2*x2-x2*x3*x3
            modrat=  bdx 1 *x2*x2-modrat*x1*x1 *xx3-
     + bdx(2 *x2*x2-modrat*x3*x3 *xx1    x2*x2-
x1*x1)*xx3-
     + x2*x2-x3*x3 *xx1)
            if  modrat-bdx(1   (bdx(2)-bdx(1  .gt.0.9
then
              modrat=bdx 1 +0.7*(bdx 2)-bdx 1))
            elseif  modrat-bdx 1 ) (bdx 2 -
bdx(1) .lt.0.1) then
              modrat=bdx 1)+0.3*(bdx 2)-bdx 1)
            endif

      print*,' bounds upper len ',bdx 2 ,bdy 2 *1000.
      print*,' modrat       len ',modrat,len*10 0.
      print*,' bounds lower len ',bdx 1 ,bdy 1 *1000.

            if len.gt.stlen  then
              bdx 2)=modrat
              bdy 2)=len
            elseif len.lt.stlen) then
              bdx 1)=modrat
              bdy(1 =len
            else
              print*,' exact value found'
                match=1
              return
            endif
          endif
          match=0.
        endif
      endif

      return
      end


      subroutine dispmat(wnod,mat,preses,delvol,iframe,
     +typ,disp,patr,namer,comtr,tec,last
c
c     It is called by subroutine matchops
c
c     writes to a file the material properties and the
displaced
c     coordinate data
c
c     It makes no subroutine calls
c
      double precision
wnod(401,3),wspu 401,3),disp 401,3)
      real
preses 151),mat(10,9),delvol 151),tec 3,10),deltat
      integer i,j,iframe,typ,tframe,last
      character*80
filnam,filenam,patr*6,namer*20,comtr*20,
     +cframe*2,cha
      logical torf
      common/person/ filenam,cframe

      if((typ.eq.4).or.(typ.eq.7). r. typ.eq.8)  then
        tframe-last
      else
        tframe=iframe+1
      endif
      deltat=0.02*float(tframe-iframe

c
c     Patient displaced coordinates written to file
c
      filnam='dcoords'//cframe//'.'//filenam

open(7,file=filnam,access='sequential',form='formatted'
      write(7,19) patr,namer,comtr,tframe
      if((typ.eq.1 .or.(typ.eq.6).or.(typ.eq.10 ) then
        do 35 i=1,401
          do 30 j=1,3
            wspu(i,j)-wnod(i,j)+disp(i,j)
30        continue
35      continue
        write 7,20)
 (wspu(i,1),wspu(i,2),wspu i,3),i,i=1,401)
      else
        do 5 i-1,401
          do 10 j=1,3
            wnod(i,j =wn d i,j)+disp i,j
10        continue
5       continue
        write(7,20)
 (wnod(i,1),wnod i,2),wnod(i,3),i,i 1,401
      endif
      close(7
c
c     Patient Material Results File created here
c
      filnam='matprop' /'.'  filenam
      inquire file=filnam,exist t rf)
      if(torf  then

open 7,file filnam,access='sequential',f rm 'f rmatted'
444     read 7,' a1 ',end 911  cha
        got  444
911     write(7,22
iframe,tframe,mat 1,1 ,mat 1,  ,mat 1,  ,

+tec 2,1),mat 1,4 ,mat 1,5 ,mat(1,6 ,mat 1,7 ,mat 1,  ,ma
t 1,9),
     +(preses(tframe)-
preses(iframe))/deltat, delvol tframe)-
     +delvol(iframe))/deltat,(delvol(tframe)-
delvol(iframe))/
     +preses(tframe)-preses(iframe))
        close(7)
      else

open(7,file=filnam,access='sequential',form='formatted')
        write(7,14)
patr,iframe,tframe,mat(1,1),mat(1,2),mat(1,3),

+tec(2,1),mat(1,4),mat(1,5),mat(1,6),mat(1,7),mat 1,8),ma
t(1,9),
     +(preses(tframe)-
preses(iframe))/deltat,(delvol(tframe)
     +-delvol(iframe))/deltat,(delvol(tframe)-
delvol(iframe))/
     +(preses(tframe)-preses(iframe))
        close(7)
      endif

14    format(
     +' MATERIAL PROPERTIES FOR ',a6/' -----------------
------------'/
     +' Frame
Nos.',3x,'Ex',5x,'Ey',5x,'Ez',5x,'Ay',4x,'Muyx',4x,
     +'Muzx',4x,'Muzy',4x,'Gxy',4x,'Gyz',4x,
     +'Gzx',5x,'Dp/Dt',4x,'Dv/Dt',4x,
     +'Dv/Dp'/' ----------   --     --     --     --    -
---    ----'
     +,4x,'----    ---    ---    ---    -----    -----
-----'/

+13x,'KN/m2',2x,'KN/m2',2x,'KN/m2',32x,'KN/m2',2x,'KN m2'
,2x,
     +'KN/m2',3x,'mmHg/s',4x,'cm3/s',3x,'cm3/mmHg'/
     +'              -----  -----  ----- ------',25x,
     +'-----  -----  -----  -------  -------   --------'
     +/i5,'-
',i2,f9.2,f8.2,f7.2,f7.4,f7.3,f8.3,f8.3,f7.2,f7.2,f7.2,f9
.3,
     +f9.3,f9.3)
19    format(a6,1x,2(a20,1x)/' CO-ORDINATES FOR FRAME NO.
',i2)
20    format(3f18.12,i5)
22    format(i5,'-
',i2,f9.2,f8.2,f7.2,f7.4,f7.3,f8.3,f8.3,f7.2,f7.2,f7.2,
     +f9.3,f9.3,f9.3)

      return
      end


      subroutine
volmatch(iframe,match,preses,typ,tol,iresp,step,

+mat,wnod,len,delvol,disp,patr,namer,comtr,last,bdy,bdx,t
ec,
     +sara,sidx,tsidx,mvol,tmvol,vol,smovol)
c
c     It is called by subroutine matchops
c
c     Results from FE part analysed here for ongoing
analysis;
c     displacements are added and scaled accordingly and
a new
c     guess for contration coefficient or elastic modulus
formed
c
c     It calls subroutines thrmvol and goldsec
c
      double precision wnod(401,3),disp(401,3),vo,vol,
     + smovol,tol,dlen,ddlen
      real
volone(151),voltwo(151),volthr(151),preses(151),tsidx,
     +
delvol(151),step,mat(10,9),len,bdy(4),bdx(4),tec(3,10),sa
ra,
     + sidx, mvol,tmvol
      integer iframe,nof,int(151),typ,iresp,i,match,last
      character*80
filnam,filenam,title,head1,head2,patr*6,namer*20
     + ,comtr*20,cframe*2
      common/person/ filenam,cframe

      step=1.0
c
c     displacements read in
c
      filnam='disp'//cframe//'.'//filenam

open(9,file=filnam,access='sequential',form='formatted')
      read(9,7) title,head1,head2
      read(9,8) (disp(i,1),disp(i,2),disp i,3),i=1,401)
      close(9)

C-----Matching volume result with targeted ones tabulated
      filnam='voludata.'//filenam

open(9,file=filnam,access='sequential',form='formatted')
      read(9,11) patr,namer,comtr,nof,title

      do 12 i-1,nof
        read(9,*) int(i ,volone(i),voltwo i ,volthr i
        if (typ.eq.1).or.(typ.eq.6).or. typ.eq.1    then
          delvol(int i))=volthr(i)
        elseif  typ.eq.2 .or. typ.eq.3 . r. typ.eq.5 )
then
          delvol ınt i )=voltwo ı
        elseif  typ.eq.4 .or. typ.ge.7 . r. typ.le.9
then
          delvol int i )=volone i
        end if
12    continue
      close 9

      do 13 i=1,n f
        if( int i .eq.iframe+1 .and. typ.eq.1 )
smovol-volthr ı
        if( int i .eq.iframe+1 .and.(typ.eq.2 )
smovol=voltwo i
        if( int i).eq.iframe+1 .and.(typ.eq.3))
smovol=voltwo i)
        if( int i .eq.last).and. typ.eq.4 )
smovol=volone i
        if (int i .eq.iframe+1).and. typ.eq.5)
smovol=voltwo i
        if( int i .eq.iframe+1 .and. typ.eq.6 )
smovol=volthr i)
        if  int i .eq.last .and.(typ.eq.7))
smovol=volone i
        if((int i .eq.last).and. typ.eq.8
smovol=volone i)
        if((int(i .eq.last .and. typ.eq.9))
smovol=volone(i)
        if( int(i).eq.iframe+1 .and. typ.eq.10))
smovol=volone(i)
13    continue

      call
newvol(wnod,disp,vo,iframe,0.0,iresp,dlen,sara,sidx,
     + mvol)
      ddlen=dlen
c      print*,' base vol   ',vo
c      print*,' base len   ',dlen
c      print*,' base sara  ',sara
c      print*,' base sidx  ',sidx
c      print*,' base mvol ',mvol

      call
newvol(wnod,disp,vo,iframe,1.0,iresp,dlen,sara,sidx,
     + mvol)

      print*,' % L.R.  ',(ddlen-dlen) ddlen*100.

      vol=vo
      write(6,29) smovol
      write 6,39  vo
      write 6,49  dlen*1000.

      if(vo/10..gt.smovol) then
        match=3
        return
      endif
c      if(typ.eq.7) then
        write(6,59  tsidx
        write(6,69) sidx
        write(6,79) tmvol
        write(6,89) mvol
c      endif

      print*,' vol error    ',dabs(vo-sm vol) sm vol*100.
      if(typ.eq.7) print*,' shindx error ',abs(sidx-
tsidx)/tsidx*100.
      if(typ.eq.9) print*,' shindx error ',abs(sidx-
tsidx)/tsidx*100.

      if(tol.le.dabs(vo-smovol)/smovol*100.) then
        if((typ.ge.5).and.(typ.le.10)) then
          call thmvol smovol,vo,step,bdy,bdx,tec)
        else
          call
goldsec(smovol,wnod,disp,vo,tol,iresp,step,dlen)
        endif
      else
        match=1
      endif

      len=dlen

7     format(a80/a80/a80)
8     format(11x,3e14.7
9     format(///' Warning!! Occilitary behavıour'  /
11    format(a6,1x,2(a20,1x  33x,i6/a80)
19    format(a6,1x,2(a20,1x) 'CO-ORDINATES FOP FRAME
NO.',i2)
29    format(' Target volume is     ',f8.3)
39    format(' Calculated v lume is ',f8.3)
49    format(' Calculated Base to Apex
Length=',f7.3,'mms')
59    format(' Target Shape Index     ',f8.3)
69    format ' Calculated Shape Index ',f8.3)
79    format(' Target My cardial Volume     ',f8.3
89    format(' Calculated My cardial Volume ',f8.

      return
      end


      subr utine
g ldsec sm v l,wn d,disp,v ,r l,iresp, tep,dlen
c
c     this subr utine i  called by v lmat h
```

```
c
c     this routine uses a linear step search to find an
upper and
c     lower bound on the correct scalling factor then
uses the
c     golden section method to find the correct value
c
c     It calls only subroutine newvol
c
      double precision smovol,wnod(401,3),disp(401,3),
     +vo,tol,toll,vol(4),dlen
      real step,stepx(4),fact,sara,sidx
      integer iframe,iresp,i

      fact=(dsqrt(5.d0)-1.d0)/2.d0
      toll=tol/100.d0

      if(vo.lt.smovol) then
        stepx(4)=1.0
        vol(1)=vo
1       step=step+1.0
        call
newvol(wnod,disp,vo,iframe,step,iresp,dlen,sara,
     +sidx,mvol)
        stepx(1)=stepx(4)
        stepx(4)=step
        if(vo.lt.smovol) goto 1
        vol(4)=vo
      else
        stepx(4)=1.0
        stepx(1)=0.0
        vol(4)=vo
        call
newvol(wnod,disp,vo,iframe,0.0,iresp,dlen,sara,
     +sidx,mvol)
        vol(1)=vo
      endif
c
c     do while tolerance condition not met
c
      stepx(3)=stepx(1)+(stepx(4)-stepx(1))*fact
      call
newvol(wnod,disp,vo,iframe,stepx(3),iresp,dlen,sara,
     +sidx,mvol)
      vol(3)=vo
      stepx(2)=stepx(4)-(stepx(4)-stepx(1))*fact
      call
newvol(wnod,disp,vo,iframe,stepx(2),iresp,dlen,sara,
     +sidx,mvol)
      vol(2)=vo
      do 10 i=1,300
        if(((vol(2)-smovol)**2.0).le. (vol(3)-
smovol)**2.0)) then
          vol(4)=vol(3)
          vol(3)=vol(2)
          stepx(4)=stepx(3)
          stepx(3)=stepx(2)
          stepx(2)=stepx(4)-(stepx(4)-stepx(1))*fact
          call
newvol(wnod,disp,vo,iframe,stepx(2),iresp,dlen,sara,
     +sidx,mvol)
          vol(2)=vo
        else
          vol(1)=vol(2)
          vol(2)=vol(3)
          stepx(1)=stepx(2)
          stepx(2)=stepx(3)
          stepx(3)=stepx(1)+(stepx(4)-stepx(1))*fact
          call
newvol(wnod,disp,vo,iframe,stepx(3),iresp,dlen,sara,
     +sidx,mvol)
          vol(3)=vo
        endif
        if(toll.ge.dabs(vol(4)-vol(1))/vol(1)*100.) goto
2
10    continue

2     vo=(vol(4)+vol(1))/2.
      step=(stepx(4)+stepx(1))/2.0
      call
newvol(wnod,disp,vo,iframe,step,iresp,dlen,sara,
     + sidx,mvol)

      write 6,9) vo
      write(6,19) dlen*1000.

      if(i.gt.299) then
        print*,' ERROR!! no convergence on step size'
        if tol*10..lt.dabs(vol 4 -vol(1))/vol(1)*100.)
stop
      en _f

9     format(' Matched volume is     ',f8.3)
19    format(' Matched Base to Apex Length=',f7.3,'mms')

      return
      end


      subroutine thmvol(smovol,vo,step,bdy,bdx,tec
c
c     this subr utine is called by subroutine volmatch
c
c     It calculates the next guess for the Contracti n
C efficient required for
c     a volume match. It uses a geometric process
followed by
c     quadratic _nterpolation.
c
c     It calls no subroutines
c
      d uble precisi n smovol,vo
      real bdy 4 ,bdx 4 ,tec 3,10 ,step

      if vo.gt.smovol  then
        bdx 1 =tec 2,1
        bdy 1 -vo
      elseif vo.lt.smovol  then
        bdx 2 =tec 2,1
        bdy 2 =v
      endif

c      print*,' bounds lower ',bdx(1),bdy(1)
c      print*,' bounds upper ',bdx 2),bdy(2)

      if((bdx 3 .eq.0 .and.(bdx 4).eq.0 ) then
        if  bdx(1).eq.0. .or. bdx 2 .eq.0.   then
          if vo.gt.smovol  then
c          step= tec 2,1)+0.05) tec 2,1
c             step=2.0
            step= tec 2,1 +0.1) tec 2,1
          elseif vo.lt.smovol  then
            step=0.5
          endif
        else
          step=(bdx 1)+ bdy 1 -sngl sm vol)   bdy 1 -
bdy 2))*
     +  bdx(2 -bdx 1)   tec 2,1
        endif
      else
        if((bdx(1).eq.0.).or. bdx 2 .eq.0.   then
          if vo.gt.smovol  then
            step=1.1
          elseif(vo.lt.smovol) then
            step=0.9
          endif
        else
          step=(bdx(1 + bdy 1)-sngl smovol    bdy 1 -
bdy(2) *
     + (bdx 2)-bdx 1) ) tec 2,1)
        endif
      endif

      print*,' bounds lower ',bdx(1),bdy(1
      print*,' new act ',tec 2,1)*step
      print*,' bounds upper ',bdx 2),bdy(2

      return
      end


      subroutine
myovol(tmvol,mvol,iart,bdy,bdx,match,tol,v ,tv 1
c
c     this subroutine is called by subroutine match ps
c
c     It calculates the next guess for the C ntracti n
Coefficient ratio to give
c     constant myocardial volume. It uses a geometric
step process
c     followed by quadratic interpolati n once b unds
have been
c     established
c
c     It calls no subroutines
c
      double precision tol,vo,tvol
      real bdy(4),bdx 4),iart,tmv 1,mvol
      integer match

      if match.eq.1) then
        if(abs(tmv 1-mvol  tmv 1*100..gt.sngl tol ) then
          if(mvol.gt.tmvol) then
            bdx(4)=iart
            bdy(4)=mvol
          elseif(mvol.lt.tmvol) then
            bdx(3 -iart
            bdy 3)-mvol
          endif

          if((bdx(3 .eq. .  or. bdx 4 .eq.0.)  then
            if mvol.gt.tm      then
              if 4.0* tmv   mv 1) tmv 1.lt.-iart  then
                iart-0.75*iart
              else
                iart-iart* 1.0+3.0* tmvol-mv 1) tmv 1
              endif
            elseit mvol.lt.tmvol) then
              iart=iart* 1. +3.0* tmv 1-mv 1 /tmv 1
            endif
          else
            iart odx 3 +  bdy 3 -tmv 1   bdy 3 -bdy 4  *
     +   bdx 4 -bdx 3 ) * .5
          endif

          match

          bdx 1 -0.
          bdy 1    .
          bdx      .
          bdy _    .

          print*,' o unds lower ',odx    ,bdy 3
          print*,' new ra*    ',iart
          print*,' bounds upper ',bdx 4),bdy(4

        endif
      else
        if((mvol.gt.tmvol*1.1).and.(vo.gt.tvol*1.1 ) then
          iart=iart*.95
          bdx(1)=0.0
          bdy(1)=0.0
          bdx(2)=0.0
          bdy(2)=0.0
        endif
c        if((mvol*1.1.lt.tmvol).and.(vo*1.1.lt.tvol))
then
c          iart=iart*1.01
c          bdx(1)=0.0
c          bdy(1)=0.0
c          bdx(2)=0.0
c          bdy(2)=0.0
c        endif
      endif

      return
      end


      subroutine
newvol(wnod,disp,vo,iframe,step,iresp,len,sara,
     + sidx,mvol)
c
c     This subroutine is called by subroutine volmatch
and goldsec
c
c     this subroutine adds the scaled displacements to
the original
c     nodal coordinates and returns the resulting cavity
volume.
c
c     It makes one subroutine call to afevolume
c
      double precision wnod(401,3),disp(401,3),vo,
     +xtdisp(401,3),len,wvol
      real step,sara,sidx,mvol
      integer iframe,j,k,iresp

      do 2 j=1,401
         do 2 k=1,3
2           xtdisp(j,k)=disp(j,k)*dble(step)+wnod(j,k)

      call
afevolume(xtdisp,vo,len,iresp,sara,sidx,mvol,wvol)

      return
      end


      subroutine
afemvol(wnod,volum,ifrmno,nlm,iresp,filenam,wfil)
c
c     this subroutine is called by subroutines part4 and
smoolen
c
c     It calculates the FE mesh for a specified patient
frame
c
c     It makes calls to
readfil,chkpnt,moveshp,rotclk,noloop,sketch,
c     tidyup,rotatenv,arcbas,slicenv,fillup,mmlt,asplit.
c
      integer
jxrn(151),jyrn(151),jxln(151),jyln(151),jxrp(151),
     +jyrp(151),jxlp(151),jylp(151),ixrn(151,151),

+iyrn(151,151),ixrp(151,151),iyrp(151,151),ixln(151,151),
     +iyln(151,151),ixlp(151,151),iylp(151,151),
     +nrnr(151),nrpr(151),nlnl(151),nlpl(151),lowy,lowz,

+lowye,lowze,n1,n2,n3,n4,n1p,n2p,n3p,n4p,nlm,nf,ifrmno,ij
kl,ij,
     +i,nrn,nrp,nln,nlp,j,k,iresp,jj,wfil
      double precision
onod(255,3),x2,plow,xaom,yaom,hyp,rsinl,rcosl,

+rsinr,rcosr,val,temx(200),temz(200),rxrn(151),ryrn(151),

+rxrp(151),ryrp(151),rxln(151),ryln(151),rxlp(151),rylp(1
51),

+ylow,zlow,width,widthe,high,slice,env(50,4,3),envp(50,4,
3),

+geom(3,255),yelow,zelow,x,y,z,geoma(3,255),t3(3,3),ht,
     +plowl,th1,th2,wnod(401,3)
      real preses(151),volum(151)
      character patr*6,namer*20,comtr*20,filenam*80

      call readfil(fi enam,patr,namer,comtr,nf,ixrn,iyrn,

+nrnr,ixln,iyln,nlnl,ixrp,iyrp,nrpr,ixlp,iylp,nlpl,preses
)

      nrn=nrnr(ifrmno)
      nrp=nrpr(ifrmno)
      nln=nlnl(ifrmno)
      nlp=nlpl(ifrmno)
      do 86 i=1,151
        jxrn(i)=ixrn(ifrmno,i)
86      jyrn(i)=iyrn(ifrmno,i)
      do 87 i=1,151
        jxrp i =ixrp(ifrmno,i)
87      jyrp i =iyrp(ifrmno,i)
      do 88 i=1,151
        jxln i =ixln ifrmno,i)
88      jyln i)=iyln(ifrmno,i)
      do 89 i=1,151
        jxlp i)=ixlp(ifrmno,i)
89      jylp i =iylp ifrmno,i)

        call chkpnt jxrn,jyrn,nrn
        call chkpnt(jxln,jyln,nln)
        call chkpnt(jxrp,jyrp,nrp
        call chkpnt(jxlp,jylp,nlp

C-----Float integer values to metres
        do 7000 i=1,151
          rxrn i)=dble(jxrn(i )*(.0001)
          ryrn i)=dble(jyrn i))* .0001)
          rxrp(i =dble(jxrp(i))*(. 001)
          ryrp(i)=dble(jyrp i))*(. 001)
          rxln(i)-dble(jxln(i))*(.0001)
          ryln(i)=dble(jyln i))*(.0001)
          rxlp(i)=dble(jxlp(i))*(.0001)
7000      rylp i =dble(jylp(i  *(.0001

c         call
tao(nln,nlp,nrn,nrp,rxrn,ryrn,rxrp,ryrp,rxln,
c     +ryln,rxlp,rylp)

C-----Move shapes to convenient origin at apex  defined
as being
C     farthest from mid-aortic root) for x and then
lowest point on
C     endocardium for y
        call moveshp(nrn,rxrn,ryrn,nrp,rxrp,ryrp)
        call moveshp(nln,rxln,ryln,nlp,rxlp,rylp)

C-----Moves shape up for sketch
        xaom=(rxln(1)+rxln(nln)) 2.
        yaom=(ryln(1)+ryln(nln))/2.
        hyp=dsqrt(xaom*xaom+yaom*yaom)
        rcosl=yaom/hyp
        rsinl=xaom/hyp
        call rotclk(nln,rxln,ryln,rcosl,rsinl)
        call rotclk(nlp,rxlp,rylp,rcosl,rsinl)
        xaom=(rxrn(1)+rxrn(nrn )/2.
        yaom=(ryrn(1)+ryrn(nrn )/2.
        hyp=dsqrt(xaom*xaom+yaom*yaom)
        rcosr=yaom/hyp
        rsinr=xaom/hyp
        call rotclk(nrn,rxrn,ryrn,rcosr,rsinr)
        call rotclk(nrp,rxrp,ryrp,rcosr,r ınr)
        call noloop(nln,rxln,ryln)
        call noloop(nrn,rxrn,ryrn)

C-----Move Rao to ensure we can always fit a box. this
has a
c     negligable effect upon the volume
      do 1001 i=1,nrn
        if(ryrn(i).lt.0.0001d0) ryrn i) 0.0001d0
1001  continue

C-----Sketch in pericardium
        ij=0
        call
sketch(nrp,nrn,rxrp,ryrp,rxrn,ryrn,ij,0.,th1,th2)
          call
tidyup(rxrn,ryrn,rxrp,ryrp,nrn,nrp,th1,th2)
        rsinr=(-1.)*rsinr
        call rotclk(nrp,rxrp,ryrp,rcosr,rsinr)
        call rotclk(nrn,rxrn,ryrn,rcosr,rsinr)

        val=ryrp 1
        do 1012 i=2,nrp
1012      if(ryrp(i).lt.val) val ryrp(i)

        jj=1
        call
sketch(nlp,nln,rxlp,rylp,rxln,ryln,jj,val,th1,th2)
        rsinl=(-1.)*rsinl
        call rotclk(nln,rxln,ryln,rcosl,rsinl)
        call rotclk(nlp,r  p,rylp,rcosl,rsinl)
          call
tidyup(rxln,ryln,rxlp,ryl  nln,nlp,th1,th2)

c         call
tao(nln,nlp,nrn,nrp,rxrn,ryrn,rxrp,ryrp,rxln,
c     +ryln,rxlp,rylp)

C-----Move Rao to ensure we can always fit a box. this
has a
c     negligable effect upon the volume
c          do 1001 i=1,nrn
c            ryrn(i) ryrn i)+0.001
c1001      continue
c
c     Reposition points on Rao Pericardium ıf they are
below
c     or cl se t  the lowest  n the La
c
c          do 1 0_ i 1,nrp
c            ryrp i =ryrp(i)+ . 01
c1002      continue

          val rylp 1
          do 518 ı 2,nıp
          if(rylp(i).lt.val) val=rylp(i)
518     continue
          val=val+0.0001D0
          do 519 i=1,nrp
          if(ryrp(i).lt.val) ryrp(i)=val
519     continue

c         call
tao(nln,nlp,nrn,nrp,rxrn,ryrn,rxrp,ryrp,rxln,
c      +ryln,rxlp,rylp)

c
c      check for over flow in the arrays
c

if((nrn.gt.150).or.(nln.gt.150).or.(nrp.gt.150).or.
     +(nlp.gt.150)) then
          print*,' ERROR .! array is not large enough '
          stop
        endif

C-----Draw aorta on lao if low compared with rao
        ht=ryrn(1)
        do 515 i=2,nrn
          if(ryrn(i).gt.ht) ht=ryrn(i)
515     continue

        if(ryln(nln).lt.ht) then
          nln=nln+1
          ryln(nln)=ht
          rxln(nln)=rxln(nln-1)
        endif

        if(ryln(1).lt.ht) then
          do 1006 i=1,nln
            j=nln+1-i
            rxln(j+1)=rxln(j)
1006        ryln(j+1)=ryln(j)
          rxln(1)=rxln(1)
          ryln(1)=ht
          nln=nln+1
        endif

        ht=ryrp(1)
        do 525 i=2,nrp
          if(ryrp(i).gt.ht) ht=ryrp(i)
525     continue

        if(rylp(nlp).lt.ht) then
          nlp=nlp+1
          rylp(nlp)=ht
          rxlp(nlp)=rxlp(nlp-1)
        endif

        if(rylp(1).lt.ht) then
          do 2006 i=1,nlp
            j=nlp+1-i
            rxlp(j+1)=rxlp(j)
2006        rylp(j+1)=rylp(j)
          rxlp(1)=rxlp(1)
          rylp(1)=ht
          nlp=nlp+1
        endif

        rsinr=(-1.)*rsinr
        call rotclk(nrp,rxrp,ryrp,rcosr,rsinr)
        call rotclk(nrn,rxrn,ryrn,rcosr,rsinr)

C-----Reverse sign x rao
        do 1011 i=1,100
          rxrn(i)=rxrn(i)*(-1.)
1011      rxrp(i)=rxrp(i)*(-1.)

C-----Find highest end point on endocardium
        high=ryrn(1)
        if(high.lt.ryrn(nrn)) high=ryrn(nrn)
        if(high.lt.ryln(1)) high=ryln(1)
        if(high.lt.ryln(nln)) high=ryln(nln)

C-----Find lowest point on the endocardium
        yelow=0.
        plowl=12345678.
        lowye=0
        do 1010 i=1,nln
          if(ryln(i).le.plowl) then
            yelow=rxln(i)
            plowl=ryln(i)
            lowye=i
          endif
1010    c ntinue
        .elow=0.
        plow=12345678.
        lowze=0
        do 1020 i=1,nrn
          if(ryrn(i).le.plow) then
            zelow=rx n i)
            plow=ryrn(i)
            lowze=i
          endif
1020    continue

        if(plowl.gt.plow) plow=plowl

C-----Slice model horizontally at fifty levels to produce
envelope .
C-----where nothing to slice place 0.
C-----First endocardium
```

```
c
        widthe= high-plow  50.
        slice=plow

        do 1040 i=1,50

          slice=slice+widthe
          do 1090 j-1,lowze
            if  slice.gt.ryrn 1 ) then
              env(i,1,3 -0.
              goto 1112
            endif
            if  slice.gt.ryrn j)) then
              env(i,1,3  rxrn j)+(slice-ryrn(j))
     +(ryrn(j-1 -ryrn j) * rxrn j-1)-rxrn(j))
              goto 1112
            endif
1090      continue
1112      env i,2,3)=env(i,1,3

          do 1120 j=lowze,nrn
            k=nrn-j+lowze
            if(slice.gt.ryrn nrn)) then
              env i,3,3)-0.
              goto 1132
            endif
            if slice.gt.ryrn(k ) then
              env i,3,3 =rxrn k +(slice-ryrn k )
     +(ryrn(k+1)-ryrn(k))* rxrn(k+1)-rxrn k))
              goto 1132
            endif
1120      continue
1132      env i,4,3)=env i,3,3

          do 1040 j=1,4
            ij=((i-1 *4 +j
            temx(ij)=slice
            temz(ij)=env(i,j,3)
1040        env(i,j,1)=slice

C-----Now rotate rao points back and use x to slice lao
        call rotclk(200,temz,temx,rcosr,rsinr)

        do 1042 i=1,50
          do 1042 j=1,4
            ij=((i-1)*4)+j
            if(env(i,j,3).eq.0.) temz(ij)=0.
            env(i,j,3)=temz(ij)
1042        env(i,j,1)=temx(ij)

        do 1041 i=1,50

          do 1050 j=1,lowye
            if(env(i,2,1).gt.ryln(1)) then
              env(i,2,2)=0.
              goto 1062
            endif
            if(env(i,2,1 .gt.ryln(j)) then
              env(i,2,2)-rxln(j)+ env(i,2,1)-ryln(j))/
     +(ryln(j-1)-ryln(j))*(rxln(j-1)-rxln(j))
              goto 1062
            endif
1050      continue
1062      do 1051 j=1,lowye
            if(env(i,4,1).gt.ryln(1)) then
              env(i,4,2)=0.
              goto 1065
            endif
            if(env(i,4,1).gt.ryln(j)) then
              env(i,4,2)=rxln(j)+(env(i,4,1)-ryln(j  /
     +(ryln(j-1)-ryln(j) *(rxln(j-1)-rxln(j))
              goto 1065
            endif
1051      continue

1065      do 1070 j=lowye,nln
            k=nln-j+lowye
            if(env(i,1,1).gt.ryln nln)) then
              env(i,1,2)=0.
              goto 1082
            endif
            if (env i,1,1).gt.ryln(k ) then
              env(i,1,2 =rxln(k)+(env(i,1,1)-ryln k )/
     +(ryln(k+1)-ryln(k)   rxln(k+1)-rxln(k))
              goto 1082
            endif
1070      continue
1082      do 1071 j=lowye,nln
            k nln-j+lowye
            if(env(i,3,1).gt.ryln(nln)) then
              env(i,3,2 =0.
              goto 1 41
            endif
            if(env i,3,1).gt.ryln k)  then
              env(i,3,2 =rxln(k)+(env i,3,1 -ryln k  /
     + ryln(k+1 -ryln k  *(rxln(k+1)-rxln(k
              g to 1 41
            endif
1071      continue

1041    continue

C-----Repeat f r pericardium
C-----Find highest end p int  n peri ardium
        nigh ryrp 1
        if(high.lt.ryrp nrp   high ryrp nrp
        if(high.lt.rylp 1   high rylp(1
       if(high.lt.rylp(nlp)) high=rylp(nlp)

C-----Find lowest point on the pericardium
       ylow=0.
       plowl=12345678.
       lowy=0

       do 8010 i=1,nlp
         if(rylp(i).lt.plowl) then
           ylow=rxlp(i)
           plowl=rylp(i)
           lowy=i
         endif
8010   continue

       zlow=0.
       plow=12345678.
       lowz=0
       do 8020 i=1,nrp
         if(ryrp(i).lt.plow) then
           zlow=rxrp(i)
           plow=ryrp(i)
           lowz=i
         endif
8020   continue

       if(plowl.gt.plow) plow=plowl

       width=(high-plow)/50.
       slice=plow

       do 2040 i=1,50
         slice=slice+width
         do 2090 j=1,lowz
           if (slice.gt.ryrp(1)) then
             envp(i,1,3)=0.
             goto 2112
           endif
           if (slice.gt.ryrp(j)) then
             envp(i,1,3)=rxrp(j)+(slice-ryrp(j )/
    +(ryrp(j-1)-ryrp(j))*(rxrp j-1)-rxrp(j)
             goto 2112
           endif
2090     continue
2112     envp(i,2,3)=envp(i,1,3)
         do 2120 j=lowz,nrp
           k=nrp-j+lowz
           if (slice.gt.ryrp(nrp)) then
             envp(i,3,3)=0.
             goto 2132
           endif
           if slice.gt.ryrp(k)) then
             envp i,3,3)=rxrp(k)+(slice-ryrp(k))/
    +(ryrp(k+1)-ryrp(k))*(rxrp(k+1)-rxrp(k))
             goto 2132
           endif
2120     continue
2132     envp(i,4,3)=envp(i,3,3)

         do 2040 j=1,4
           ij=((i-1)*4)+j
           temx(ij =slice
           temz(ij)=envp(i,j,3)
2040       envp(i,j,1)=slice

C-----Now rotate rao points back and use x to slice lao
        call rotclk(200,temz,temx,rcosr,rsinr)

        do 2042 i=1,50
          do 2042 j=1,4
            ij=((i-1)*4)+j
            if(envp(i,j,3).eq.0.) temz(ij)=0.
            envp(i,j,3)=temz(ij)
2042        envp(i,j,1)=temx(ij)

        do 2041 i=1,50

          do 2050 j=1,lowy
            if (envp(i,2,1).gt.rylp(1)) then
              envp(i,2,2)=0.
              goto 2062
            endif
            if (envp(i,2,1).gt.rylp(j)) then
              envp(i,2,2)=rxlp(j)+(envp(i,2,1)-rylp(j))/
    +(rylp(j-1)-rylp(j))*(rxlp(j-1)-rxlp(j)
              goto 2062
            endif
2050      continue
2062      do 2051 j=1,lowy
            if(envp(i,4,1).gt.rylp(1)) then
              envp(i,4,2)=0.
              goto 2065
            endif
            if(envp(i,4,1).gt.rylp(j)) then
              envp(i,4,2)=rxlp(j)+(envp(i,4,1 -rylp(j))/
    +(rylp(j-1)-rylp j))*(rxlp(j-1)-rxlp(j )
              goto 2065
            endif
2051      continue

2065      do .070 j=lowy,nlp
            k=nlp-j+lowy
            if envp(i,1,1).gt.rylp nlp)) then
              envp(i,1,2 =0.
              goto 2082
            endif
            if envp(i,1,1 .gt.rylp(k   then
              envp i,1,2 =rxlp k + envp i,1,1 -rylp k
    +(rylp k+1)-rylp k  * rxlp k+1 -rxlp k
              goto 2082
            endif
2070      continue
2082      do 2071 j=lowy,nlp
            k=nlp-j+lowy
            if(envp i,3,1).gt.rylp nlp   then
              envp(i,3,2)=0.
              goto 2041
            endif
            if(envp(i,3,1 .gt.rylp k   then
              envp(i,3,2)-rxlp(k + envp i,3,1 -rylp k
    +(rylp(k+1)-rylp(k))*(rxlp(k+1 -rxlp k
              goto 2041
            endif
2071      continue

2041    continue

C-----Rotate actual rao back
        call rotclk(nrn,rxrn,ryrn,rcosr,rsinr
        call rotclk nrp,rxrp,ryrp,rc sr,rsinr)

C-----Find lowest xyz for end points
        do 3060 j=50,1,-1
3060
if(dabs(env(j,1,3) .gt.0..and.dabs env(j,1,2 ).gt.0.
     +goto 3070
        goto 2999
3070    n1=j
        do 3080 j=50,1,-1
3080
if(dabs(env(j,2,3)).gt.0..and.dabs env j,2,2 ).gt.0.)
     +goto 3090
        goto 2999
3090    n2=j
        do 4000 j=50,1,-1
4000
if(dabs(env(j,3,3)).gt.0..and.dabs(env(j,3,2)).gt.0.)
     +goto 4010
        goto 2999
4010    n3=j
        do 4020 j=50,1,-1
4020
if(dabs(env(j,4,3)).gt.0..and.dabs(env j,4,2)).gt.0.
     +goto 4030
        goto 2999
4030    n4=j

c------epi
        do 4060 j=50,1,-1
4060
if(dabs(envp(j,1,3)).gt.0..and.dabs(envp(j,1,2)).gt.0.)
     +goto 4070
        goto 2999
4070    nlp=j
        do 4080 j=50,1,-1
4080
if(dabs(envp(j,2,3)).gt.0..and.dabs(envp(j,2,2)).gt.0.)
     +goto 4090
        goto 2999
4090    n2p=j
        do 5000 j=50,1,-1
5000
if(dabs(envp(j,3,3)).gt.0..and.dabs(envp(j,3,2)).gt.0.)
     +goto 5010
        goto 2999
5010    n3p=j
        do 5020 j=50,1,-1
5020
if(dabs(envp(j,4,3)).gt.0..and.dabs(envp(j,4,2)).gt.0.)
     +goto 5030
        goto 2999
5030    n4p=j

C----- Rotate envelope in three dimensi ns so that mid-
aortic line
C-----is vertical
        x=(ryln(1 +ryln(nln))/2.
        y=(rxln(1 +rxln nln))/2.
        z=(rxrn(1)+rxrn(nrn))/2.
        x2=(ryrn(1)+ryrn nrn))/2.
        z=z* x/x2

c         call
tpenv(env,envp,n1,n2,n3,n4,nlp,n2p,n3p,n4p

        call rotatenv(x,y,z,env,envp,t3

        call
arcbas(env,envp,geom,n1,n2,n3,n4,nlp,n2p,n3p,n4p

C----- Slice remaining ventricle h riz ntally
        call slicenv env,envp,geom,n1,n2,n ,n4,nlp,
     +n2p,n3p,n4p
        call fillup(ge m)

        call mmlt(ge ma,t3,geom,3,3,255

        do 6000 i=1,255
           nod i,1 =ge ma 1,i
          onod i,2  geoma 2,.
600       onod i,3  ge ma 3,i


c        d  60 5 i 1,.
c6005      onod i,1)=0.0705

        goto 9999
2999    print*,'Run stopped-end of envelope not found'
        stop
9999    continue

      call
asplit(wnod,onod,volum,ijkl,ifrmno,patr,namer,comtr,
     +iresp,nlm,wfil)

85    return
      end


c*****************************************************
c
c     end of automated routines
c
c*****************************************************

c*****************************************************
c
c     Finite Element part (formaly Program XL1)
c
c*****************************************************


      subroutine finelm
c
c     this subroutine is called only from subroutine
part4
c     it was formally the main routine of program XL1
c     It makes numerious subrotine calls:
storemat,lcoord,
c
forma,stf20b,stf15w,pr20b,pr15w,tl20b,tl15w,dblokg,smhstr
,
c     strain.
c
c     This subroutine controls the setup and solution
caculation
c     of the finite element problem specified by the file
XL1-data
c
      double precision wrmbk(3,339),stiff(3,339,401)
      common/wmbk/wmbk,prmbk
      double precision wmbk(3,339),prmbk(3,339)
      equivalence(wmbk(1,1),wrmbk(1,1))
      double precision lam(3,20),lam8(3,8),xi,eta,zeta
      double precision
dp(3,15),va(15,15),vinva(15,15),lam15(3,15)
      common/lamcm/lam,xi,eta,zeta,vinva,dp,va
      equivalence (lam(1,1),lam8(1,1),lam15(1,1))
      integer lwork(6),mwork(6),isw1,isw2,isw3,isw4
      integer vlwork(15),vmwork(15)
      integer
nrmbk,nrb,nelem,nonop,lo,co,mannd,nomat,nplc,ncmbk
      integer
ncb,ze,et,xii,prevty,p,p1,p2,counte,prevma,post
      integer nonp1(72,15),nonp2(72,20),rest(99,4),eno
      double precision coord(3,401),dis(1203),f(1203)
      double precision work(360),temp(401),tec(3,10)
      double precision mat(10,9)
      double precision
fx(401),fy(401),fz(401),dpx(401),dpy(401)
      double precision dpz(401)
      double precision wdx,wddx,wddy,wddz
      double precision tr(6,6),d2(6,6),m(6,6),a(3,3)
      double precision xi2(6),eta2(6),zeta2(6)
      double precision
elco(20,3),elco8(8,3),k(60,60),jay(3,3)
      double precision
invjla(3,20),invjay(3,3),b(6,60),d(6,6),d1(6,6)
      double precision
eldis(60),stn(6),invjl8(3,8),b8(6,24)
      double precision
ap(3),sts(6),lc20b(3,20),lc08b(3,8),lc15w(3,15)
      double precision
n(20),fn(3),pforc(36,5),kb(24,24),detj,tmp
      double precision
k15(45,45),elco15(15,3),invjl5(3,15),b15(6,45)
      double precision vp(15),n15(15)
      character hed*80

common/femacm/mat,work,tr,d2,m,a,coord,xi2,eta2,zeta2,
     +
elco,k,jay,invjay,invjla,b,d,d1,dis,f,eldis,stn,sts,
     + ap,fn,n,vmwork,vlwork,lc20b,lc08b,lc15w,
     +
vp,detj,nrmbk,nrb,eno,nelem,nonop,lo,co,mannd,nomat,nplc,
     + ncmbk,icom,ncb,ne,loop,ioop,nonp1,nonp2,pforc,
     + ze,et,xii,prevty,p,p1,p2,counte,prevma,post,rest
      common/smatrix/stiff
      equivalence(k(1,1),kb(1,1),k15(1,1))
      equivalence(invjla(1,1),invjl8(1,1),invjl5(1,1))
      equivalence(n(1),n15(1))
      equivalence(elco(1,1),elco8(1,1),elco15(1,1))
      equivalence(b(1,1),b8(1,1),b15(1,1))
      equivalence(vlwork(1),lwork(1))
      equivalence(vmwork(1),mwork(1))

      data lin/1/,lout/3/,ldsp/2/

      open(1,file='XL1-
data',access='sequential',form-'formatted')
```

```
      read(lin,1000  hed
1000  format(a80
1001  format 1x,a80
      read(lin,*  nelem,nonop,lo,nplc,co,mannd,nomat
c     ---------------------------------------------------
c     ---------------------------------------------------
      read lin,*) isw1,isw2,isw3,isw4
c     ---------------------------------------------------
c     ---------------------------------------------------
      nrb=nonop
      ncb=mannd+1
      nrmbk=nonop*3
      ncmbk=3*ncb
      iscr=3*ncmbk

      do 20 i=1,3
          do 20 j=1,ncmbk
20        wmbk(i,j =0.

      do 22 i=1,nrb
22      call storemat(i,1,wmbk,stiff)

      write(lout,111)
nelem,nonop,lo,nplc,co,mannd,nomat,iscr
111   format(1x,'number of elements = ',i6,/,1x,
     +'number of nodes = ',i6, ,1x,'number of point 1 ads
= ',i6,/
     +,1x,'number of pressure loads = ',i6, ,1x,
     +'number of constraints = ',i6, ,1x,
     +'max node number difference',i6, ,1x,
     +'number of materials = ',i6,/,1x,
     +'elements on backing file = ',i6)
c     ---------------------------------------------------
c     ---------------------------------------------------
      write(lout,115) isw1,isw2,isw3,isw4
115   format(1x,/' stress plotting option isw1
',i2,//,1x,
     +'strain output isw2 = ',i2,//,1x,
     +'data reflection only isw3 = ',i2,////,1x,
     +'thermal loading required isw4 = ',i2,////,1x,
     +'if option = 0  option inactive',/,1x,
     +'if option = 1  option active',//,1x,
     +'stress plot option relevant to 72 element model
only',/,1x,
     +'ensure plot option 5 off otherwise')
c     ---------------------------------------------------
c     ---------------------------------------------------

101   format(10i6)

1102  format(9d9.0)

      do 199 i=1,nomat
        read(lin,*)
(mat(i,j),j=1,9),tec 1,`),tec(2,i),tec(3,i)
199     write(lout,112)
i,mat(i,1),mat(i,2),mat(i,3),mat(i,4),

+mat(i,5),mat(i,6),mat(i,7),mat(i,8),mat(i,9),tec(1,i),te
c(2,i),
     +tec(3,i)
112   format(1x,'material no ',i6,' material
props',/,1x,'ex=',e14.4,
     +5x,'ey=',e14.4,5x,'ez=',e14.4,/,1x,'mu-
yx=',f10.5,5x,'mu-zx=',
     +f10.5,5x,'mu-zy=',f10.5,/,1x,'g-xy ',e14.4,5x,'g-
yz=',e14.4,5x,
     +'g-zx=',e14.4,/' Themal Expansion Coefficients',/,
     +' X =',E14.6,' Y =',E14.6,' Z =',E14.6)

      do 5189 i=1,nonop
5189    read(lin,*) (coord(j,i),j=1,3),temp(i)

      do 1234 i=1,nelem
          read(lin,*) (nonp1(i,j),j 1,15)
          j=nonp1(i,14)
1234    read(lin,*) (nonp2(i,jj),jj 1,j)

105   format(36i2)
      read(lin,*) ((rest(i,j),j=1,4),i=1,co)
      read(lin,*)  (pforc(i,j),j=1,5),i-1,nplc)
977   format(5e12.5)
978   format(5d12.5)

c      --------------------------------------------------
c      --------------------------------------------------
      if(isw3.eq.1) goto 9999
c      --------------------------------------------------
c      --------------------------------------------------
      call lcoord

      do 1989 ne=1,nelem
1989    if(nonp1 ne,14).eq.15  call forma

      prevty-0

      do 2000 ne-1,nelem
          if (nonp1(ne,14).eq.20) call stf20b
          if (nonp1 ne,14).eq.15) call stf15w
2000  continue

      do 2010 i 1,nrmbk
          dis i  0.
2010    f i) 0.

9 _   f rmat 4i1
```

```
      mxrest=0

      do 2030 i=1,co
          j=rest(i,1)
          if (j.gt.mxrest) mxrest=j
          jj=3*(j-1)
        do 2040 ii=1,3
            tmp=rest(i,ii+1)
            if(tmp.eq.0.) goto 2040
            tmp=0.000001
            idum=jj+ii
            dis(idum)=tmp
            goto 2040
2040    continue
2030  continue

      if (lo.eq.0) goto 2060
2060  if (nplc.eq.0) goto 3000

      do 3040 icom=1,nplc
          j=idint(pforc(icom,1))
          if(nonpl(j,14).eq.20) call pr20b
          if(nonpl(j,14).eq.15) call pr15w
3040  continue

      if(isw4.eq.1) then
        do 3050 ne=1,nelem
           if(nonpl(ne,14).eq.20) call tl20b(temp,tec)
           if(nonpl(ne,14).eq.15) call tl15w(temp,tec)
3050    continue
      endif

3000  continue

      do 4010 i=1,co
          do 4010 j=2,4
            if (rest(i,j).eq.0) go to 4020
            idum=3*rest(i,1)+j-4
            if(f(idum).eq.0.d0) go to 4020
            f(idum)=0.
4020      continue
4010  continue

      write(lout,305)
305   format(//1x,'final nodal
forces',/,1x,'node',13x,1hx,10x,
     +1hy,10x,1hz)
      icount=0

      do 4030 i=1,nrmbk,3
          icount=icount+1
          jdum=(i+2)/3
          fx(icount)=f(i)
          fy(icount)=f(i+1)
          fz(icount)=f(i+2)
4030  write(lout,306) jdum,f(i),f(i+1),f(i+2)

306   format(1x,i3,7x,3f12.6)
410   format(3f10.4)

      do 4040 i=1,mxrest
          call storemat(i,0,wmbk,stiff)
          do 4050 j=1,3
            idum=3*(i-1)+j
            if(dis(idum).eq.0.) goto 4050
              do 4070 ii=1,ncmbk
4070          wmbk(j,ii)=0.
            do 4080 ii=1,3
4080          wmbk(ii,j)=0.
              wmbk(j,j)=1.
4050    continue
          idum=ncb-1
          do 4090 j=1,idum
            if (i+j.gt.nrb) go to 4090
            do 5010 ii=1,3
              kdum=3*(i+j-1)+ii
              if (dis(kdum).eq.0.) go to 5010
              do 5020 jj=1,3
                jdum=3*j+ii
                wmbk(jj,jdum)=0.
5020          continue
5010        continue
4090      continue
            call storemat(i,1,wmbk,stiff)
4040  continue

      call dblokg(f,ncb,nrb)
      write(ldsp,190)
190   format(1x,'nodal
displacements',/,1x,'node',10x,'dx',
     +11x, dy',11x,'dz')
      icount=0

      do 6000 i=1,nrmbk,3
          icount=icount+1
          j=(i+3)/3
          dpx(icount =f(i)
          dpy icount)=f(i+1)
          dpz(icount)=f(i+2)
6000  write(ldsp,400) j,f(i),f(i+1),f(i+2)

400   format(1x,i3,7x,3e14.7)
6001  format(1e20.0

      wdx=0.d0
      do 1 i=1,401
          wddx=fx i)*dpx(i)
          wddy=fy i *dpy i
          wddz-fz i)*dpz i
          wdx=wddx+wddy+wddz+wdx
1     continue

      wdx=wdx/2.d0
      write(lout,3  wdx
      write(6,3) wdx
3     format ' Work done = ',f20.14

      call smhstr isw1,hed)

      if(isw2.eq.1  call strain

9999  return
      end


      subroutine tl20b temp,tec)
c
c     called from the main program.
c
c     this subroutine calculates the thermal loading  n a
node
c     for a 20 noded iso parametric brick shaped element.
c
      double precision temp(401),thta,ethr 6),ethrl(6 ,
     + sthg(6),ff 60),tec 3,10)

      integer lwork(6),mwork(6 ,nd,ln
      integer vlwork 15),vmwork(15)
      integer
nrmbk,nrb,nelem,nonop,lo,co,mannd,nomat,nplc,ncmbk
      integer
ncb,ze,et,xii,prevty,p,pl,p2,counte,prevma,post
      integer nonpl(72,15 ,nonp2(72,20 ,rest 99,4),eno
      double precision
coord(3,401),dis(1203),f(1203),deter
      double precision work(360),xi,eta,zeta,lam 3,20)
      double precision mat(10,9 ,dp 3,15),va(15,15)
      double precision tr(6,6),d2(6,6),m(6,6),a(3,3)
      double precision
xi2(6),eta2(6),zeta2(6),vinva 15,15)
      double precision elco(20,3),k 60,60),jay(3,3)
      double precision
invjla(3,20 ,invjay(3,3),b(6,60),d(6,6),dl 6,6
      double precision eldis(60),stn(6)
      double precision
ap(3),sts 6),lc20b(3,20),lc08b(3,8),lc15w(3,15)
      double precision n 20),fn 3),pf rc(36,5),detj
      double precision vp(15)
      double precision hh(14),xil(15),etal 15),zetal(15

      common/lamcm/lam,xi,eta,zeta,vinva,dp,va

common/femacm/mat,work,tr,d2,m,a,coord,xi2,eta2,zeta2,
     +
elco,k,jay,invjay,invjla,b,d,dl,dis,f,eldis,stn,sts,
     + ap,fn,n,vmwork,vlwork,lc20b,lc08b,lc15w,
     +
vp,detj,nrmbk,nrb,eno,nelem,nonop,lo,co,mannd,nomat,nplc,
     + ncmbk,icom,ncb,ne,loop,ioop,nonpl,nonp2,pf rc,
     + ze,et,xii,prevty,p,pl,p2,counte,prevma,post,rest

      data hh/8*0.335180055d0,6*0.886426593d0/
      data xil/2*-0.758786911d0,2*0.758786911d0,2*-
0.758786911d0,
     +  2*0.758786911d0,0.d0,-
0.795822426d0,0.d0,0.d0,0.795822426d0,
     + 2*0.d0/
      data etal/4*0.758786911d0,4*-
0.758786911d0,0.795822426d0,0.d0,
     +  0.d0,0.d0,0.d0,-.795822426d0,0.d0/
      data zetal/-0.758786911d0,0.758786911d0,-
0.758786911d0,
     + 0.758786911d0,-0.758786911d0,0.758786911d0,
     + -0.758786911d0,0.758786911d0,2*0.d0,-
0.795822426d0,
     + 0.795822426d0,3*0.d0/

      do 70 i=1,60
        ff(i)=0.d0
70    continue

      if(nonpl ne,4 .ge.2     l transf 1, .d0)

      do 1111 i=1,20
          do 1111 j=1,3
1111    elco i,j =coord(j,nonp2(ne,i))


      do 5 i=1,14
          xi=xil i
          eta=etal(i
          zeta=zetal i)
          h=nh i

c     calculate temp at node fr m shape fun t  n
contributions
        call nodetemp20b temp,lc2 b,thta,xi,eta,zeta,
     + ne,nonp2

c     calculate thermal l ads in l  al   rd nate
        ethr 1  tec 1,n npl ne,   *thta
        ethr 2 -tec 2,n npl ne,2  *thta
        ethr    -t-   ,n npl ne,2) *thta
        ethr 4  ).d
      ethr(5)=0.d0
      ethr(6)=0.d0

        call lam20b
      call mxm(lam,3,elco,20,jay,3)
      detj=jay(1,1)*(jay(2,2)*jay(3,3)-
jay(3,2)*jay(2,3))
     +-jay(2,1)*(jay(1,2)*jay(3,3)-jay(3,2)*jay(1,3))
     ++jay(3,1)*(jay(1,2)*jay(2,3)-jay(2,2)*jay(1,3))
      if(detj.le.0.) then
c         print*,' node connection Error Element ',ne
        detj=0.d0
c         stop
      endif
      call fminv(jay,invjay,3,deter,lwork,mwork,6)
      call mxm(invjay,3,lam,3,invjla,20)

        if(nonpl(ne,4).le.1) then
          call transf(1,zeta)
          do 60 j=1,6
            ethrl(j)=0.d0
            do 65 jj=1,6
              ethrl(j)=ethrl(j)+tr(j,jj)*ethr(jj)
 65         continue
 60       continue
        else
          do 152 j=1,6
            ethrl(j)=ethr(j)
 152      continue
       endif

       if(nonpl(ne,4).le.1) call transf(2,0.d0)

         do 10 j=1,6
           ethr(j)=0.d0
           do 15 jj=1,6
             ethr(j)=ethr(j)+tr(j,jj)*ethrl(jj)
 15        continue
 10      continue

         do 20 j=1,6
           sthg(j)=0.d0
           do 25 jj=1,6
             sthg(j)=sthg(j)-d1(j,jj)*ethr(jj)
 25        continue
 20      continue

c     calculate coefficient strain matrix
        do 1460 jj=1,6
           do 1460 j=1,60
1460       b(jj,j)=0.

        do 1470 j=1,20
           b(1,3*j-2)=invjla(1,j)
           b(4,3*j-1)=invjla(1,j)
1470     b(6,3*j)=invjla(1,j)

        do 1471 j=1,20
           b(2,3*j-1)=invjla(2,j)
           b(4,3*j-2)=invjla(2,j)
1471     b(5,3*j)=invjla(2,j)

        do 1472 j=1,20
           b(3,3*j)=invjla(3,j)
           b(5,3*j-1)=invjla(3,j)
1472     b(6,3*j-2)=invjla(3,j)

        dum=detj*h

c     calculate thermal stresses * shape fn derivatives

c     calculate thermal loads in global coordinates

         do 30 j=1,60
           sum=0.d0
           do 35 jj=1,6
             sum=sum+b(jj,j)*sthg(jj)
 35        continue
           ff(j)=ff(j)-sum*dum
 30      continue

 5    continue

c     global node and direction 'nd' from local 'j'
        do 55 j=1,60
          ln=((j-1)/3+1)
          nd=(nonp2(ne,ln)-1)*3+(j-(ln-1)*3)
          f(nd)=f(nd)+ff(j)
 55     continue

      return
      end


      subroutine tl15w(temp,tec)
c
c     called from the subroutine finelm.
c
c     this subroutine calculates the thermal loading on a
node
c     for a 15 noded wedge shaped element.
c
      double precision temp(401),thta,ethr(6),
     + sthg(6),tec(3,10),ethrl(6),ff(60)

      integer lwork(6),mwork(6),nd,ln
      integer vlwork(15),vmwork(15)
```

```
      integer
nrmbk,nrb,nelem,nonop,lo,co,mannd,nomat,nplc,ncmbk
      integer
ncb,ze,et,xii,prevty,p,p1,p2,counte,prevma,post
      integer nonp1(72,15),nonp2(72,20),rest(99,4),eno
      double precision
coord(3,401),dis(1203),f(1203),deter,h1,h2
      double precision
work(360),xi,eta,zeta,lam(3,20),lam15(3,15)
      double precision
mat(10,9),dp(3,15),va(15,15),lam8(3,8)
      double precision tr(6,6),d2(6,6),m(6,6),a(3,3)
      double precision
xi2(6),eta2(6),zeta2(6),vinva(15,15)
      double precision elco(20,3),k(60,60),jay(3,3)
      double precision
invjla(3,20),invjay(3,3),b(6,60),d(6,6),d1(6,6)
      double precision eldis(60),stn(6)
      double precision
ap(3),sts(6),lc20b(3,20),lc08b(3,8),lc15w(3,15)
      double precision n(20),fn(3),pforc(36,5),detj
      double precision
elco15(15,3),invj15(3,15),b15(6,45)
      double precision vp(15)
      double precision
hh1(7),hh2(4),xi1(7),eta1(7),zeta1(4)

      equivalence (lam(1,1),lam8(1,1),lam15(1,1))

      common/lamcm/lam,xi,eta,zeta,vinva,dp,va

common/femacm/mat,work,tr,d2,m,a,coord,xi2,eta2,zeta2,
     +
elco,k,jay,invjay,invjla,b,d,d1,dis,f,eldis,stn,sts,
     + ap,fn,n,vmwork,vlwork,lc20b,lc08b,lc15w,
     +
vp,detj,nrmbk,nrb,eno,nelem,nonop,lo,co,mannd,nomat,nplc,
     + ncmbk,icom,ncb,ne,loop,ioop,nonp1,nonp2,pforc,
     + ze,et,xii,prevty,p,p1,p2,counte,prevma,post,rest

      data hh1/.225d0,3*.13239415d0,3*.12593918d0/
      data hh2/2*.347854845137454d0,2*.652145154862546d0/
      data xi1/0.0d0,.41042619d0,-.41042619d0,0.0d0,-
.69614048d0,
     + .69614048d0,0.0d0/
      data eta1/-.333333333333333d0,2*-.05971587d0,-
.88056825d0,
     + 2*-.79742699d0,.59485397d0/
      data zeta1/.861136311594053d0,-.861136311594053d0,
     + .339981043584856d0,-.3399810435856d0/

      do 1111 i=1,15
          idum=nonp2(ne,i)
          do 1111 j=1,3
1111      elco15(i,j)=coord(j,idum)

      do 95 j=1,45
        ff(j)=0.d0
 95   continue

      if(nonp1(ne,4).ge.2) then
        call transf(1,0.d0)
      else
        do 100 i=1,6
          do 105 j=1,6
            m(i,j)=0.d0
105       continue
100     continue
        m(1,1)=1./mat(nonp1(ne,2),1)
        m(2,2)=1./mat(nonp1(ne,2),2)
        m(3,3)=1./mat(nonp1(ne,2),3)
        m(4,4)=1./mat(nonp1(ne,2),7)
        m(5,5)=1./mat(nonp1(ne,2),8)
        m(6,6)=1./mat(nonp1(ne,2),9)
        m(1,2)=-mat(nonp1(ne,2),4)/mat(nonp1(ne,2),2)
        m(2,1)=m(1,2)
        m(1,3)=-mat(nonp1(ne,2),5)/mat(nonp1(ne,2),3)
        m(3,1)=m(1,3)
        m(2,3)=-mat(nonp1(ne,2),6)/mat(nonp1(ne,2),3)
        m(3,2)=m(2,3)
      endif
      call fminv(m,d1,6,deter,lwork,mwork,6)

      do 5 i=1,7
          xi=xi1(i)
          eta=eta1(i)
        h1=hh1(i)
        do 55 ii=1,4
            zeta=zeta1(ii)
            h2=hh2(ii)
            h=h1*h2*2.
c     calculate temp at node from shape function
contributions
          call nodetemp15w(temp,lc15w,thta,xi,eta,zeta,
     + ne,nonp2)
c     calculate thermal loads in local coordinates
          ethr(1)=tec(1,nonp1(ne,2))*thta
          ethr(2)=tec(2,nonp1(ne,2))*thta
          ethr(3)=tec(3,nonp1(ne,2))*thta
          ethr(4)=0.d0
          ethr(5)=0.d0
          ethr(6)=0.d0

            call lam15w
          call mxm(lam15,3,elco15,15,jay,3)
          detj=jay(1,1)*(jay(2,2)*jay(3,3)-
jay(3,2)*jay(2,3))
```

```
     +-jay(2,1)*(jay(1,2)*jay(3,3)-jay(3,2)*jay(1,3))
     ++jay(3,1)*(jay(1,2)*jay(2,3)-jay(2,2)*jay(1,3))
          if(detj.le.0.) then
c             print*,' node connection Error Element ',ne
            detj=0.d0
c             stop
          endif
          call fminv(jay,invjay,3,deter,lwork,mwork,6)
          call mxm(invjay,3,lam15,3,invj15,15)

          if(nonpl(ne,4).le.1) then
            call transf(1,zeta)
            do 60 j=1,6
              ethrl(j)=0.d0
              do 65 jj=1,6
                ethrl(j)=ethrl(j)+tr(j,jj)*ethr(jj)
 65           continue
 60         continue
          else
            do 152 j=1,6
              ethrl(j)=ethr(j)
 152        continue
        endif

        if(nonpl(ne,4).le.1) call transf(2,0.d0)

          do 10 j=1,6
            ethr(j)=0.d0
            do 15 jj=1,6
              ethr(j)=ethr(j)+tr(j,jj)*ethrl(jj)
 15         continue
 10       continue

          do 20 j=1,6
            sthg(j)=0.d0
            do 25 jj=1,6
              sthg(j)=sthg(j)-dl(j,jj)*ethr(jj)
 25         continue
 20       continue

c      calculate coefficient strain matrix
          do 1460 jj=1,6
              do 1460 j=1,45
1460          b15(jj,j)=0.

          do 1470 j=1,15
              b15(1,3*j-2)=invj15(1,j)
              b15(4,3*j-1)=invj15(1,j)
1470        b15(6,3*j)=invj15(1,j)

          do 1471 j=1,15
              b15(2,3*j-1)=invj15(2,j)
              b15(4,3*j-2)=invj15(2,j)
1471        b15(5,3*j)=invj15(2,j)

          do 1472 j=1,15
              b15(3,3*j)=invj15(3,j)
              b15(5,3*j-1)=invj15(3,j)
1472        b15(6,3*j-2)=invj15(3,j)

          dum=detj*h

c      calculate thermal stresses * shape fn derivatives
          do 30 j=1,45
            sum=0.d0
            do 35 jj=1,6
              sum=sum+b15(jj,j)*sthg(jj)
 35         continue
            ff(j)=ff(j)-sum*dum
 30       continue

 55     continue

 5    continue

      do 85 j=1,45
        ln=((j-1)/3+1)
        nd=(nonp2(ne,ln)-1)*3+(j-(ln-1)*3)
        f(nd)=f(nd)+ff(j)
 85   continue

      return
      end


      subroutine
nodetemp20b(temp,lc20b,thta,r,s,t,ne,nonp2)
c
c     Called by subroutine tl20b
c
c     Calculates the temperature a a given point
(integration point)
c     from the contributions from the shape functions.
c
      double precision temp(401),thta,r,s,t,lc20b(3,20),
     + n(20)
      integer i,ne,nonp2(72,20)
c
c      corner nodes
c
      n(1)=0.125*(1.+r*lc20b(1,1))*(1.+s*lc20b(2,1))*
     +
(1.+t*lc20b(3,1))*(r*lc20b(1,1)+s*lc20b(2,1)+t*lc20b(3,1)
-2.)
      n(3)=0.125*(1.+r*lc20b(1,3))*(1.+s*lc20b(2,3))*
     +
(1.+t*lc20b(3,3))*(r*lc20b(1,3)+s*lc20b(2,3)+t*lc20b(3,3)
-2.)
      n(6)=0.125*(1.+r*lc20b(1,6))*(1.+s*lc20b(2,6))*
     +
(1.+t*lc20b(3,6))*(r*lc20b(1,6)+s*lc20b(2,6)+t*lc20b(3,6)
-2.)
      n(8)=0.125*(1.+r*lc20b(1,8))*(1.+s*lc20b(2,8))*
     +
(1.+t*lc20b(3,8))*(r*lc20b(1,8)+s*lc20b(2,8)+t*lc20b(3,8)
-2.)
      n(13)=0.125*(1.+r*lc20b(1,13))*(1.+s*lc20b(2,13))*
     +
(1.+t*lc20b(3,13))*(r*lc20b(1,13)+s*lc20b(2,13)+t*lc20b(3
,13)-2.)
      n(15)=0.125*(1.+r*lc20b(1,15))*(1.+s*lc20b(2,15))*
     +
(1.+t*lc20b(3,15))*(r*lc20b(1,15)+s*lc20b(2,15)+t*lc20b(3
,15)-2.)
      n(18)=0.125*(1.+r*lc20b(1,18))*(1.+s*lc20b(2,18))*
     +
(1.+t*lc20b(3,18))*(r*lc20b(1,18)+s*lc20b(2,18)+t*lc20b(3
,18)-2.)
      n(20)=0.125*(1.+r*lc20b(1,20))*(1.+s*lc20b(2,20))*
     +
(1.+t*lc20b(3,20))*(r*lc20b(1,20)+s*lc20b(2,20)+t*lc20b(3
,20)-2.)
c
c      mid nodes
c
      n(9)=0.25*(1.-r*r)*(1.+s*lc20b(2,9))*
     + (1.+t*lc20b(3,9))
      n(10)=0.25*(1.-r*r)*(1.+s*lc20b(2,10))*
     + (1.+t*lc20b(3,10))
      n(11)=0.25*(1.-r*r)*(1.+s*lc20b(2,11))*
     + (1.+t*lc20b(3,11))
      n(12)=0.25*(1.-r*r)*(1.+s*lc20b(2,12))*
     + (1.+t*lc20b(3,12))

      n(4)=0.25*(1.-s*s)*(1.+r*lc20b(1,4))*
     + (1.+t*lc20b(3,4))
      n(5)=0.25*(1.-s*s)*(1.+r*lc20b(1,5))*
     + (1.+t*lc20b(3,5))
      n(16)=0.25*(1.-s*s)*(1.+r*lc20b(1,16))*
     + (1.+t*lc20b(3,16))
      n(17)=0.25*(1.-s*s)*(1.+r*lc20b(1,17))*
     + (1.+t*lc20b(3,17))

      n(2)=0.25*(1.-t*t)*(1.+r*lc20b(1,2))*
     + (1.+s*lc20b(2,2))
      n(7)=0.25*(1.-t*t)*(1.+r*lc20b(1,7))*
     + (1.+s*lc20b(2,7))
      n(14)=0.25*(1.-t*t)*(1.+r*lc20b(1,14))*
     + (1.+s*lc20b(2,14))
      n(19)=0.25*(1.-t*t)*(1.+r*lc20b(1,19))*
     + (1.+s*lc20b(2,19))
c
c     calculate the thermal load (thta) at a point from
the individual
c     shape function contributions
c
      thta=0.d0
      do 5 i=1,20
        thta=thta+n(i)*temp(nonp2(ne,i))
 5    continue

      return
      end


      subroutine
nodetemp15w(temp,lc15w,thta,r,s,t,ne,nonp2)
c
c     Called by subroutine tl20b
c
c     Calculates the temperature a a given point
(integration point)
c     from the contributions from the shape functions.
c
      double precision temp(401),thta,r,s,t,lc15w(3,15),
     + n(15),l(6),a(3),b(3),c(3),dlt
      integer i,ne,nonp2(72,20)

      dlt=0.5*(lc15w(1,2)*lc15w(2,3)-
lc15w(2,2)*lc15w(1,3)
     + -lc15w(1,1)*(lc15w(2,3)-lc15w(2,2))+
     + lc15w(2,1)*(lc15w(1,3)-lc15w(1,2)))

      a(1)=lc15w(1,2)*lc15w(2,3)-lc15w(2,2)*lc15w(1,3)
      b(1)=lc15w(2,2)-lc15w(2,3)
      c(1)=lc15w(1,3)-lc15w(1,2)

      a(2)=lc15w(1,3)*lc15w(2,1)-lc15w(2,3)*lc15w(1,1)
      b(2)=lc15w(2,3)-lc15w(2,1)
      c(2)=lc15w(1,1)-lc15w(1,3)

      a(3)=lc15w(1,1)*lc15w(2,2)-lc15w(2,1)*lc15w(1,2)
      b(3)=lc15w(2,1)-lc15w(2,2)
      c(3)=lc15w(1,2)-lc15w(1,1)


l(1)=1/(4.*dlt)*(a(1)+b(1)*r+c(1)*s)*(1+t*lc15w(3,1))

l(2)=1/(4.*dlt)*(a(2)+b(2)*r+c(2)*s)*(1+t*lc15w(3,2))

l(3)=1/(4.*dlt)*(a(3)+b(3)*r+c(3)*s)*(1+t*lc15w(3,3))
```

```
l(4 =1/(4.*dlt * a 1)+b 1)*r+c(1 *s * 1+t*lc15w(3,4

l 5 =1/(4.*dlt)*(a(2)+b(2)*r+c(2)*s)* 1+t*lc15w(3,5)

l(6)=1/(4.*dlt)*(a 3)+b 3)*r+c(3)*s)*(1+t*lc15w(3,6))

      l(1)=1/(2.*dlt)*(a(1)+b(1 *r+c 1)*s)
      l(2 =1 (2.*dlt)* a 2 +b(2)*r+c 2)*s
      l(3)=1/(2.*dlt)*(a 3)+b(3)*r+c 3)*s
      l 4)=1/(2.*dlt)*(a 1)+b 1)*r+c(1)*s)
      l(5)=1/(2.*dlt)*(a 2)+b 2)*r+c 2)*s
      l(6)=1/(2.*dlt)*(a 3)+b(3)*r+c 3)*s)
c
c     corner nodes
c
      n(1)=0.5*l(1)* 2.*l(1)-1.)*(1.+t*lc15w(3,1))-
0.5*l(1)*(1.-t*t)
      n(2)=0.5*l(2)*(2.*l 2)-1.)*(1.+t*lc15w(3,2))-
0.5*l(2)*(1.-t*t
      n(3)=0.5*l(3)* 2.*l 3)-1.)*(1.+t*lc15w(3,3))-
0.5*l(3)*(1.-t*t
      n(4)=0.5*l(4 *(2.*l(4)-1.)* 1.+t*lc15w 3,4))-
0.5*l(4)*(1.-t*t)
      n(5)=0.5*l 5 * 2.*l(5)-1. * 1.+t*lc15w(3,5))-
0.5*l(5)*(1.-t*t)
      n(6)=0.5*l 6)*(2.*l(6)-1.)*(1.+t*lc15w(3,6) -
0.5*l(6)*(1.-t*t)
c
c     mid-nodes on the triangle
c
      n(7)=2.*l 1)*l 2)*(1.+t*lc15w(3,7))
      n(8 =2.*l 2)*l 3)* 1.+t*lc15w 3,8))
      n(9)=2.*l 3)*l 1)*(1.+t*lc15w(3,9))
      n(13)=2.*l 1)*l 2)* 1.+t*lc15w(3,13)
      n(14)=2.*l 2)*l 3)*(1.+t*lc15w 3,14))
      n(15)=2.*l(3)*l(1)*(1.+t*lc15w(3,15))
c
c     mid-nodes on the rectangle
c
      n(10)=l 1 *(1-t*t)
      n(11)=l 2 * 1-t*t)
      n(12 =l 3)*(1-t*t)
c
c     calculate the thermal load (thta) at a point from
the individual
c     shape function contributions
c
      thta=0.d0
      do 5 i=1,15
        thta=thta+n i *temp(nonp2 ne,i))
 5    continue

      return
      end


      subroutine storemat(rec,rw,part,stiff)
c
c     saves and retrives a local stiffness matrix to and
from
c     the global stiffness matrix
c
      double precision part(3,339 ,stiff(3,339,401)
      integer rec,rw,i,j

      if(rw.eq.0  then
        do 5 i=1,3
          do 5 j=1,339
            part(i,j)=stiff(i,j,rec
5       continue
      else
        do 10 i=1,3
          do 10 j=1,339
            stiff i,j,rec =part i,j)
10      continue
      endif

      return
      end


      subroutine stf20b
      double precision wrmbk 3,339 ,stiff(3,339,401
      common/wmbk/wmbk,prmbk
      double precision wmbk 3,339),prmbk(3,339)
      equivalence(wmbk 1,1 ,wrmbk(1,1))
      double precision lam(3,20 ,lam8(3,8),xi,eta,zeta
      double precision
dp(3,15   a(15,15),vinva 15,15 ,lam15 3,15
      c mmon/lamcm/lam,xi,eta,zeta,vinva,dp,va
      equivalence  lam 1,1 ,lam8 1,1 ,lam15 1,1 )
      integer lwork 6),mwork(6
      integer vlwork(15 ,vmwork 15
      integer
nrmbk,nrb,nelem,nono  lo,co,mannd,nomat,nplc,ncmbk
      integer
ncb,ze,et,xii,prevty,p,p1,p2,counte,prevma,post
      integer nonp1(72,15),nonp2(72,20),rest(99,4 ,eno
      double precision coord(3,401),dis 1203),f 1203
      double precision work(36 )
      double precision mat 10,9
      double precision tr 6,6 ,d  6,6 ,m 6,6 ,a 3,3
      double precision xi2(6 ,eta2 6 ,zeta2 6
      double precision
elco  ,  ,elc 8(3,3 ,k 60,60 ,jay 3,3
```

```
      double precision
invjla 3,20 ,invjay 3,3 ,b 6,6  ,d 6,6 ,dl 6,6
      double precision
eldis 6  ,stn 6),invj18 3,8 ,b8 6,24
      double precision
ap(3),sts 6),lc20b 3,20 ,lc08b(3,8 ,lc15w 3,1
      double precision n 20 ,fn(3),pf rc  6,5  kb 24,24
      double precision dum,h,detj,deter
      double precision
k15(45,45 ,elco15 15,3 ,invj15 3,15 ,b15 6,45
      double precision vp 15 ,n15 15

common femacm mat,work,tr,d2,m,a,coord,xi2,eta ,zeta ,
     +
elco,k,jay,invjay,invjla,b,d,dl,dis,f,eldis,stn,sts,
     + ap,fn,n,vmwork,vlwork,lc20b,lc 8b,lc15w,
     +
vp,detj,nrmbk,nrb,eno,nelem,n nop,lo,c ,mannd,n mat,npl ,
     + ncmbk,icom,ncb,ne,loop,ioop,nonp1,nonp2,pr rc,
     + ze,et,xii,prevty,p,p1,p2,counte,prevma,post,rest
      common smatrix stiff

      equivalence k 1,1 ,kb 1,1 ,k15 1,1
      equivalence invjla 1,1 ,invj18 1,1 ,invj 5 1,1
      equivalence n 1 ,n15(1
      equivalence elco(1,1),elco8 1,1 ,elco15 1,1
      equivalence b 1,1 ,b8 1,1),b15 1,1
      equivalence(vlwork 1 ,lwork 1
      equivalence vmwork 1 ,mwork 1 )
      double precision hh 14 ,xi1 15 ,eta1 15 ,zeta1 15

      data hh 8*0.33518 055d ,6* .886426593d
      data xi1/2*-0.758786911d0,2* .75 786911d , *-
0.758786911d0,
     +2*0.758786911d0,0.d0,-
0.795822426d0,0.d0,0.d0,0.795822426d0,
     +2*0.d0/
      data eta1/4*0.758786911d ,4*-
0.758786911d0,0.795822426d0,
     +0.d0,0.d0,0.d0,0.d0,-.795822426d , .d
      data zeta1 -0.758786911d , .758786911d ,-
0.758786911d0,
     +0.758786911d0,-0.758786911d0, .758786911d ,
     +-0.758786911d0,0.758786911d0,2* .d0,- .79    4 6d ,
     +0.795822426d0,3*0.d0/
      data lin/1/,lout/3/,ldsp 2/

6789  format(i4)
      if nonp1(ne,3).eq.prevty  g to 1700

      do 1100 i=1,60
          do 1100 j=1,60
1100      k(i,j)=0.

      if(nonp1(ne,4).ge.2) call transf 1, .

      do 1600 loop-1, 4
          xi=xi1(loop)
          eta-eta1 loop
          zeta-zeta1 loop
          h-hh(loop
          if nonp1 ne,4 .le.1  call transf 1,zeta
          call lam2 b
        call mxm(lam,3,elco,2 ,jay,3
        detj=jay(1,1 *(jay 2,2)*jay 3,3 -
jay(3,2 *jay(2,3 )
     +-jay(2,1)*(jay(1,2)*jay(3,3)-jay 3,2)*jay 1,3)
     ++jay(3,1)*(jay(1,2)*jay(2,3)-jay 2,2)*jay 1,3
        if(detj.le.0.) then
          print*,' node connecti n Error Element
',ne,detj
          detj=0.d0
c          stop
        endif
        call fminv jay,invjay,3,deter,lw rk,mw rk,6
        if n np1 ne,4).le.1) call transf(2, .d0
        call mxm invjay,3,lam,3,invjla,2

        do 1460 i 1,6
            do 1460 j-1,6
1460        b i,j   .

        do 147  i=1,20
            b 1,3*i-2 -in    a 1,i
            b 4,3*i-1  i     a 1,i
1470      b(6,3*i -invjl  ,i)

        do 1471 i 1,20
            b 2,3*i-1 -inv la  ,i
            b 4,3*i-  -invjla  ,i
1471      b 5, *i  nvjla(2,i

        do 1472 i=1,2
            b 3,3*i =invjla 3,
            b 5,3*i-1 =inv  a 3,
1472      b 6,3*i-2  invj a 3,i

        dum-d tj*h

        do 1 20 i 1,6
            do 15     1,6
1520        d i,j  dl i,  *dum

         all btdb(b,1,k,6,6 , r

16      ntin e
```

```
      j=1
      do 1620 i=1,20
          k(j+1,j)=k(j,j+1)
          k(j+2,j)=k(j,j+2)
          k(j+2,j+1 =k j+1,j+2)
1620    j=j+3

1700  continue

      if(nonpl(ne,15).ne.1) goto 1800

      do 1780 i=1,20
          ipi=nonp2(ne,i)
          call storemat ipi,0,wmbk,stiff)
          do 1770 j=i,20
            ipj=nonp2(ne,j)
            do 1760 ii=1,3
              jdum=3*(i-1)+ii
              do 1760 jj=1,3
                idum=3*(ipj-ipi)+jj
                kdum=3*(j-1)+jj
1760          wmbk(ii,idum)=wmbk(ii,idum +k(jdum,kdum)
1770    continue
          call storemat(ipi,1,wmbk,stiff)
1780  continue

      goto 1900
1800  continue

      do 1880 i=1,20
          ipi=nonp2(ne,i)
          call storemat(ipi,0,prmbk,stiff)
          do 1879 j=i,20
            ipj=nonp2(ne,j)
            if(ipi.gt.ipj) goto 1840
            do 1830 ii=1,3
              jdum=3*(i-1)+ii
              do 1830 jj=1,3
                idum=3*(ipj-ipi)+jj
                kdum=3*(j-1)+jj
1830          prmbk(ii,idum)=prmbk(ii,idum)+k(jdum,kdum)

      go to 1879
1840  call storemat(ipj,0,wmbk,stiff)
      idum=3*(ipi-ipj)
      jdum=3*(i-1)
      kdum=3*(j-1)
      wmbk(1,idum+1)=wmbk(1,idum+1)+k(jdum+1,kdum+1)
      wmbk(1,idum+2)=wmbk(1,idum+2)+k(jdum+2,kdum+1
      wmbk(1,idum+3)=wmbk(1,idum+3)+k(jdum+3,kdum+1)
      wmbk(2,idum+1)=wmbk(2,idum+1)+k(jdum+1,kdum+2)
      wmbk(2,idum+2)=wmbk(2,idum+2)+k(jdum+2,kdum+2)
      wmbk(2,idum+3)=wmbk(2,idum+3)+k(jdum+3,kdum+2)
      wmbk(3,idum+1)=wmbk(3,idum+1)+k(jdum+1,kdum+3)
      wmbk(3,idum+2)=wmbk(3,idum+2)+k(jdum+2,kdum+3)
      wmbk(3,idum+3)=wmbk(3,idum+3)+k(jdum+3,kdum+3)
      call storemat(ipj,1,wmbk,stiff)
1879  continue
      call storemat(ipi,1,prmbk,stiff)
1880  continue
1900  continue
      prevty=nonp2(ne,3)
2000  continue

      return
      end


      subroutine stfl5w
      double precision wrmbk(3,339),stiff(3,339,401)
      common/wmbk/wmbk,prmbk
      double precision wmbk(3,339),prmbk(3,339)
      equivalence(wmbk(1,1),wrmbk(1,1))
      double precision lam(3,20),lam8(3,8),xi,eta,zeta
      double precision
dp(3,15),va(15,15),vinva(15,15),lam15(3,15)
      common/lamcm/lam,xi,eta,zeta,vinva,dp,va
      equivalence (lam(1,1),lam8(1,1),lam15(1,1))
      integer lwork(6),mwork(6)
      integer vlwork(15),vmwork(15)
      integer
nrmbk,nrb,nelem,nonop,lo,co,mannd,nomat,nplc,ncmbk
      integer
ncb,ze,et,xii,prevty,p,p1,p2,counte,prevma,post
      integer nonpl(72,15),nonp2(72,20),rest(99,4),eno
      double precision coord(3,401),dis(1203),f 1203
      double precision work(360)
      double precision mat(10,9)
      double precision tr(6,6),d2(6,6),m 6,6),a(3,3)
      double precision xi2(6),eta2(6),zeta2(6
      double precision
elco(20,3),elco8(8,3),k(60,60),jay(3,3)
      double precision
invjla(3,20),invjay(3,3 ,b(6,60),d 6,6 ,dl(6,6
      double precision
eldis(60),stn(6),invjl8 3,8 ,b8 6,24
      double precision
ap(3),sts(6),lc20b(3,20),lc08b(3,8),lcl5w 3,15)
      double precision n 20 ,fn 3 ,pforc(36,5 ,kb _4,_4
      double precision dum,h,h1,h2,detj,deter
      double precision
k15(45,45),elco15 15,3 ,invj15 3,15 ,b15(6,45
      double precision vp 15),n15 15

common remacm mat,work,tr,d2,m,a,coord,xi2,eta_,ze_a_,
     +
elco,k,jay,invjay,invjla,b,d,dl,dis,f,eldis,stn,sts,
     + ap,fn,n,vmwork,vlwork,lc20b,lc08b,lc15w,
     +
vp,detj,nrmbk,nrb,eno,nelem,nonop,lo,co,mannd,nomat,nplc,
     + ncmbk,icom,ncb,ne,lo p,io p,n npl,n np_,pforc,
     + ze,et,xii,prevty,p,pl,p2,counte,prevma,p st,rest
      equivalence k 1,1),kb(1,1 ,k15 1,1
      equivalence invjla 1,1),invjl8 1,1 ,invj15 1,1
      equivalence n 1),n15(1))
      equivalence elc (1,1),elco8 1,1),elco15(1,1
      equivalence b 1,1),b8(1,1 ,b15 1,1 )
      equivalence vlwork(1),lwork 1 )
      equivalence vmwork(1),mwork 1 )
      double precision
hh1(7),hh2 4),xil 7 ,etal(7),zetal 4
      common smatrix stiff

      data hh1 .225d0,3*.13239415d0,3*.1259 918d0
      data hh2 2*.347854845137454d0,2*.652145154862546d0
      data xil/0.0d0,.41042619d0,-.41042619d0,0.0d0,-
.69614048d0,
     + .69614048d0,0.0d0
      data etal -.333333333333333d0,2*-.05971587d0,-
.88056825d0,
     + 2*-.79742699d0,.59485397d0
      data zetal .861136311594053d0,-.861136311594053d0,
     + .339981043584856d0,-.3399810435856d0
      data lin/1 ,lout/3 ,ldsp 2/

6789  format(i4)
      if(nonpl(ne,3 .eq.prevty) goto 1700

      do 1100 i=1,45
          do 1100 j=1,45
1100  k15 i,j -0.

      call transf 1,0.d0)

      do 1111 i=1,15
          idum=nonp2(ne,i)
          do 1111 j=1,3
1111    elco15(i,j)=coord(j,idum)

      do 1600 loop=1,7
          xi=xil loop)
          eta=etal(loop)
          h1=hh1 loop)
          do 1600 ioop=1,4
            zeta=zetal(io p)
            h2=hh2 ioop)
            h=h1*h2*2.
            call lam15w
          call mxm(lam15,3,elco15,15,jay,3
            detj=jay(1,1)*(jay 2,2 *jay 3,3 -
jay 3,2)*jay(2,3))
     +  -jay(2,1 *(jay 1,2)*jay 3, )-jay  ,  *jay(1,3 )
     +  +jay(3,1 * jay 1,2)*jay(_,3)-jay 2,  *jay(1,3))
          if detj.le.0.) then
            print*,' node connecti n Err r Element
',ne,detj
            detj=0.d0
c            stop
          endif
            call fminv jay,invjay,3,deter,lw rk,mw rk,6
          call mxm(invjay,3,lam15,3,invj15,15
              do 1460 i=1,6
                do 1460 j=1,45
1460          b15(i,j)=0.
              do 1470 i=1,15
                b15(1,3*i-2)=invj15(1,i)
                b15(4,3*i-1)=invj15 1,i)
1470          b15(6,3*i) invj15(1,i)
              do 1471 i 1,15
                b15(2,3*i-1)=invj15(2,i)
                b15 4,3*i-2)=invj15 2,i
1471          b15 5,3*i)-invj15 2,i)
              do 1472 i 1,15
                b15 3,3*i) invj15(3,i
                b15(5,3*i-1)=invj15 3,i
1472          b15 6,3*i-2  invj15(3,i)
              dum=detj*h
              do 1520 i=1,6
                do 1520 j=1,6
1520          d 1,j  dl i,j *dum
              call bt    15,d,k15,6,45,w rk
1600  continue

      j=1

      do 1620 i=1,15
          k15(j+1,j =k15(j,j+1)
          k15(j+2,j  k15 j,j+2)
          k15 j+_,j+1  k15  +1,j+2
1620    j j+3

1700  continue
      if n npl ne,15 .ne.1  g t  1

      do 178  i 1,15
          ipi n np2 ne,1)
          call  toremat ipi,0,wmbk, tift
          do 1      1,15
            ip  n np2 ne,
            d  176  11 1,3
             dum  * i-1 + 1
            d  1 6     1,3
              idum 3* ip -ipi +
              kdum  *  -1 +
 1760          wmbk(ii,idum)=wmbk(ii,idum)+k15(jdum,kdum)
 1770     continue
           call storemat ipi,1,wmbk,stiff)
 1780   continue
        goto 1900
 1800   continue

        do 1880 i=1,15
            ipi=nonp2(ne,i)
            call storemat(ipi,0,prmbk,stiff)
            do 1879 j=i,15
              ipj=nonp2(ne,j)
              if(ipi.gt.ipj) goto 1840
              do 1830 ii=1,3
                jdum=3*(i-1)+ii
                do 1830 jj=1,3
                  idum=3*(ipj-ipi)+jj
                  kdum=3*(j-1)+jj
1830
prmbk(ii,idum)=prmbk(ii,idum)+k15(jdum,kdum)
                go to 1879
 1840         call storemat(ipj,0,wmbk,stiff)
                idum=3*(ipi-ipj)
                jdum=3*(i-1)
                kdum=3*(j-1)

wmbk(1,idum+1)=wmbk(1,idum+1)+k15(jdum+1,kdum+1)

wmbk(1,idum+2)=wmbk(1,idum+2)+k15(jdum+2,kdum+1)

wmbk(1,idum+3)=wmbk(1,idum+3)+k15(jdum+3,kdum+1)

wmbk(2,idum+1)=wmbk(2,idum+1)+k15(jdum+1,kdum+2)

wmbk(2,idum+2)=wmbk(2,idum+2)+k15(jdum+2,kdum+2)

wmbk(2,idum+3)=wmbk(2,idum+3)+k15(jdum+3,kdum+2)

wmbk(3,idum+1)=wmbk(3,idum+1)+k15(jdum+1,kdum+3)

wmbk(3,idum+2)=wmbk(3,idum+2)+k15(jdum+2,kdum+3)

wmbk(3,idum+3)=wmbk(3,idum+3)+k15(jdum+3,kdum+3)
               call storemat(ipj,1,wmbk,stiff)
 1879     continue
            call storemat(ipi,1,prmbk,stiff)
 1880   continue

 1900   continue
        prevty=nonp1(ne,3)
        return
        end


        subroutine pr20b
        double precision lam(3,20),lam8(3,8),xi,eta,zeta
        double precision
 dp(3,15),va(15,15),vinva(15,15),lam15(3,15)
        common/lamcm/lam,xi,eta,zeta,vinva,dp,va
        equivalence (lam(1,1),lam8 1,1),lam15(1,1))
        integer lwork 6),mwork(6)
        integer vlwork(15),vmwork(15)
        integer
 nrmbk,nrb,nelem,nonop,lo,co,mannd,nomat,nplc,ncmbk
        integer
 ncb,ze,et,xii,prevty,p,p1,p2,counte,prevma,post
        integer nonp1(72,15),nonp2(72,20),rest(99,4),eno
        double precision coord(3,401),dis(1203),f(1203)
        double precision work(360)
        double precision mat(10,9)
        double precision tr(6,6),d2(6,6),m(6,6),a(3,3)
        double precision xi2(6),eta2(6),zeta2(6)
        double precision
 elco(20,3),elco8(8,3),k(60,60),jay(3,3)
        double precision
 invjla(3,20),invjay(3,3),b(6,60),d(6,6),d1(6,6)
        double precision
 eldis(60),stn(6),invj18(3,8),b8(6,24)
        double precision
 ap(3),sts(6),lc20b(3,20),lc08b(3,8),lc15w(3,15)
        double precision n(20),fn(3 ,pforc(36,5),kb(24,24)
        double precision dum,xo,zo,eo,h,detj
        double precision deter,pp
        double precision
 k15(45,45),elco15(15,3),invj15(3,15),b15(6,45)
        double precision vp(15),n15 15)

 common/femacm/mat,work,tr,d2,m,a,coord,xi2,eta2,zeta2,
      +
 elco,k,jay invjay,invjla,b,d,d1,dis,f,eldis,stn,sts,
      + ap fn,n,vmwork,vlwork,lc20b,lc08b,lc15w,
      +
 vp,detj,nrmbk,nrb,eno,nelem,nonop,lo,co,mannd,nomat,nplc,
      + ncmbk,icom,ncb,ne,loop,ioop,nonp1,nonp2,pforc,
      + ze,et,xii,prevty,p,p1,p2,counte,prevma,post,rest
        equivalence(k(1,  ,kb(1,1),k15(1,1))
        equivalence(invjla(1,1),invj18(1,1),invj15(1,1))
        equivalence(n(1),n15(1))
        equivalence(elco(1,1),elco8(1,1),elco15(1,1))
        equivalence(b(1,1),b8(1,1 ,b15(1,1))
        equivalence(vlwork 1),lwork 1))
        equivalence(vmwork 1),mwork 1))
        double precision hh(14),xil 15),etal 15 ,zetal 15

        data hh 8*0.335180055d0,6*0.886426593d0
        data xil 2*-0.758786911d0,2*0.758786911d0,2*-
 0.758786911d0,
```

```
      +  2*0.758786911d0,0.d0,-
 0.795822426d0,0.d0,0.d0,0.795822426d0,
      + 2*0.d0
       data etal 4*0.758786911d0,4*-
 0.758786911d0,0.7958_2426d0,0.d0,
      +  0.d0,0.d0,0.d0,-.7958_24_6d0,0.d0
       data zetal -0.758786911d0,0.758786911d0,-
 0.758786911d0,
      + 0.758786911d0,-0.758786911d0,0.758786911d0,
      + -0.758786911d0,0.758786911d0,_*0.d ,-
 0.795822426d0,
      + 0.795822426d0,3*0.d0
       data lin 1 ,lout 3 ,ldsp 2

       eno=idint pforc icom,1 )
       pp=pforc(icom,5
       pp=-pp

       do 3111 kk=2,4
3111     fn(kk-1 =pforc ic m,kk

       do 3060 i=1,20
           idum=nonp2 eno,i
           do 3060 j=1,3
           elco(i,j =coord j,idum)
3060   continue

       do 3070 loop=1,14
           xi=xil loop
           eta=etal loop)
           zeta=zetal loop
           h=hh loop
           h=h*.5
           if (fn 1 .ne.0.d0) xi fn(1
           if (fn 2 .ne.0.d0) eta fn 2
           if (fn 3 .ne.0.d0) zeta fn(3
           call lam20b
         call mxm(lam,3,elco,20,jay,3)
         detj=jay 1,1)*(jay(2,2)*jay 3,3 -
jay(3,2)*jay(2,3))
      +-jay(2,1)*(jay(1,2)*jay(3,3)-jay(3,2 *jay 1,3))
      ++jay(3,1)*(jay 1,2)*jay(2,3)-jay(2,2 *jay 1,3)
         call fminv jay,invjay,3,deter,lwork,mw rk,6)

         if (fn(1 .eq.0.d0) goto 3080

         do 3090 i=1,3
3090       ap(i)=pp*invjay(i,1 *detj*fn(1)

3080     if  fn(2).eq.0.d0) g to 3100

         do 3110 i=1,3
3110       ap(i)=pp*invjay i,2)*detj*fn 2

3100     if (fn(3).eq.0.d0) goto 3120

         do 3130 i=1,3
3130       ap(i =pp*invjay(i,3)*detj*fn 3)

3120     do 3140 i=1,20
             xo=lc20b(1,i *xi
             eo=lc20b(2,i *eta
             zo=lc20b 3,i *zeta
             dum=0.125*(1.+xo)*(1.+zo)*(1.+e )* xo+eo+z -
2.)
             n(1)=dum
             n(3)=dum
             n(6 =dum
             n(8)=dum
             n(13)=dum
             n(15)=dum
             n(18)=dum
             n(20 =dum
             dum=0.25*(1.-xi*xi)*(1.+eo)* 1.+zo
             n(9)=dum
             n(10 =dum
             n(11)=dum
             n(12)=dum
             dum=0.25*(1.+xo)*(1.-eta*eta)*(1.+zo)
             n(4)=dum
             n 5)=dum
             n(16 =dum
             n(17)=dum
             dum 0.2 * 1.+      1.+eo)*(1.-zeta*zeta
             n(2)=dum
             n(7)=dum
             n(14)=dum
             n(19)=dum
             do 3140 j=1,3
               idum=3*nonp2(eno,i -3+j
 3140        f idum =f(idum)+n(i *ap j *h

 3070 continue

6789  format i4)

      return
      end


      subr utine pr15w
      d uble precisi n lam  ,_0 ,lam   ,   ,xi,eta,zeta
      d uble precisi n
dp 3,15 ,va(15,15 ,vinva 15,15),lam15 _,15
      c mm n lam m lam,xi,eta,zeta,vinva,dp,va
      equivalen e  lam 1,1 ,lam  1,1 ,laml  1,
      integer lw rk 6 ,mw rk 6
      integer vlwork(15),vmwork(15)
      integer
nrmbk,nrb,nelem,nonop,lo,co,mannd,nomat,nplc,ncmbk
      integer
ncb,ze,et,xii,prevty,p,pl,p2,counte,prevma,post
      integer nonpl(72,15),nonp2(72,20),rest(99,4),eno
      double precision coord(3,401),dis(1203),f(1203)
      double precision work(360)
      double precision mat(10,9)
      double precision tr(6,6),d2(6,6),m(6,6),a(3,3)
      double precision xi2(6),eta2(6),zeta2(6)
      double precision
elco(20,3),elco8(8,3),k(60,60),jay(3,3)
      double precision
invjla(3,20),invjay(3,3),b(6,60),d(6,6),d1(6,6)
      double precision
eldis(60),stn(6),invjl8(3,8),b8(6,24)
      double precision
ap(3),sts(6),lc20b(3,20),lc08b(3,8),lc15w(3,15)
      double precision n(20),fn(3),pforc(36,5),kb(24,24)
      double precision h,hl,h2,detj,deter,pp
      double precision
k15(45,45),elco15(15,3),invjl5(3,15),b15(6,45)
      double precision vp(15),n15(15)

common/femacm/mat,work,tr,d2,m,a,coord,xi2,eta2,zeta2,
     +
elco,k,jay,invjay,invjla,b,d,dl,dis,f,eldis,stn,sts,
     + ap,fn,n,vmwork,vlwork,lc20b,lc08b,lc15w,
     +
vp,detj,nrmbk,nrb,eno,nelem,nonop,lo,co,mannd,nomat,nplc,
     + ncmbk,icom,ncb,ne,loop,ioop,nonpl,nonp2,pforc,
     + ze,et,xii,prevty,p,pl,p2,counte,prevma,post,rest
      equivalence(k(1,1),kb(1,1),k15(1,1))
      equivalence(invjla(1,1),invjl8(1,1),invjl5(1,1))
      equivalence(n(1),n15(1))
      equivalence(elco(1,1),elco8(1,1),elco15(1,1))
      equivalence(b(1,1),b8(1,1),b15(1,1))
      equivalence(vlwork(1),lwork(1))
      equivalence(vmwork(1),mwork(1))
      double precision
hhl(7),hh2(4),xil(7),etal(7),zetal(4)
      data hhl/.225d0,3*.13239415d0,3*.12593918d0/
      data hh2/2*.347854845137454d0,2*.652145154862546d0/
      data xil/0.0d0,.41042619d0,-.41042619d0,0.0d0,-
.69614048d0,
     + .69614048d0,0.0d0/
      data etal/-.333333333333333d0,2*-.05971587d0,-
.88056825d0,
     + 2*-.79742699d0,.59485397d0/
      data zetal/.861136311594053d0,-.861136311594053d0,
     + .339981043584856d0,-.3399810435856d0/
      data lin/1/,lout/3 ,ldsp/2

      eno=idint(pforc(icom,1 )
6789  format(i4)
      pp=pforc(icom,5)
      pp=-pp

      do 3111 kk=2,4
3111    fn(kk-1)=pforc icom,kk

      if (fn(3).eq.0.) goto 9998

      do 3060 i=1,15
          idum=nonp2(eno,i)
          do 3060 j=1,3
3060      elco15(i,j)=coord(j,idum)

      do 3070 loop=1,7
          xi=xil(loop)
          eta=etal(loop)
          hl=hhl(loop)
          do 3070 ioop=1,4
          zeta=zetal(ioop)
          h2=hh2(ioop)
          h=hl*h2
          if (fn(1).ne.0.d0) xi=fn(1)
          if (fn(2).ne.0.d0) eta=fn(2)
          if (fn(3).ne.0.d0) zeta=fn(3)
          call lam15w
        call mxm(lam15,3,elco15,15,jay,3)

          detj=jay(1,1)*(jay(2,2)*jay(3,3)-
jay(3,2)*jay(2,3))
     +-jay(2,1)*(jay(1,2)*jay(3,3)-jay(3,2)*jay(1,3))
     ++jay(3,1)*(jay(1,2)*jay(2,3)-jay(2,2)*jay(1,3))
          call fminv(jay,invjay,3,deter,lwork,mwork,6)
          if (fn(1).eq.0.d0) goto 3080

          do 3090 i=1,3
3090      ap(i)=pp*invjay(i,1)*detj*fn(1)
3080    if (fn(2).eq.0.d0) goto 3100
          do 3110 i=1,3
3110      ap(i)=pp*invjay(i,2)*detj*fn(2)
3100    if (fn(3).eq.0.d0) goto 3120
          do 3130 i=1,3
3130     ap(i)=pp*invjay(i,3)*detj*fn(3)

3120    vp(1)=1.
          vp(2)=xi
          vp(3)=eta
          vp(4)=zeta
          vp(5)=xi*xi
          vp(6)=eta*eta
          vp(7)=zeta*zeta
          vp(8)=xi*eta
```

```
          vp(9)=xi*zeta
          vp(10)=eta*zeta
          vp(11)=xi*eta*zeta
          vp(12)=zeta*xi*xi
          vp(13)=zeta*eta*eta
          vp(14)=xi*zeta*zeta
          vp(15)=eta*zeta*zeta
        call mxm(vp,1,vinva,15,n15,15)
          do 3140 i=1,15
            do 3140 j=1,3
              idum=3*nonp2(eno,i)-3+j
3140        f(idum)=f(idum)+n15(i)*ap(j)*h
 3070 continue

3040  continue
      goto 9999
9998  write(lout,7771)
7771  format(1x,'  loaded on wrong face')
9999  continue

      return
      end


      subroutine forma
      double precision lam(3,20),lam8(3,8),xi,eta,zeta
      double precision
dp(3,15),va(15,15),vinva(15,15),lam15(3,15)
      common/lamcm/lam,xi,eta,zeta,vinva,dp,va
      equivalence (lam(1,1),lam8(1,1),lam15(1,1))
      integer lwork(6),mwork(6)
      integer vlwork(15),vmwork(15)
      integer
nrmbk,nrb,nelem,nonop,lo,co,mannd,nomat,nplc,ncmbk
      integer
ncb,ze,et,xii,prevty,p,pl,p2,counte,prevma,post
      integer nonpl(72,15),nonp2(72,20),rest(99,4),eno
      double precision coord(3,401),dis(1203),f(1203)
      double precision work(360)
      double precision mat(10,9)
      double precision tr(6,6),d2(6,6),m(6,6),a(3,3)
      double precision xi2(6),eta2(6),zeta2(6)
      double precision
elco(20,3),elco8(8,3),k(60,60),jay(3,3)
      double precision
invjla(3,20),invjay(3,3),b(6,60),d(6,6),dl(6,6)
      double precision
eldis(60),stn(6),invjl8(3,8),b8(6,24)
      double precision
ap(3),sts(6),lc20b(3,20),lc08b(3,8),lc15w(3,15)
      double precision n(20),fn(3),pforc(36,5),kb(24,24)
      double precision xo,zo,eo,detj,deter
      double precision
k15(45,45),elco15(15,3),invjl5(3,15),b15(6,45)
      double precision vp(15),n15(15)

common/femacm/mat,work,tr,d2,m,a,coord,xi2,eta2,zeta2,
     +
elco,k,jay,invjay,invjla,b,d,dl,dis,f,eldis,stn,sts,
     + ap,fn,n,vmwork,vlwork,lc20b,lc08b,lc15w,
     +
vp,detj,nrmbk,nrb,eno,nelem,nonop,lo,co,mannd,nomat,nplc,
     + ncmbk,icom,ncb,ne,loop,ioop,nonpl,nonp2,pforc,
     + ze,et,xii,prevty,p,pl,p2,counte,prevma,post,rest
      equivalence(k(1,1),kb(1,1),k15(1,1))
      equivalence(invjla(1,1),invjl8(1,1),invjl5(1,1))
      equivalence(n(1),n15(1))
      equivalence(elco(1,1),elco8(1,1),elco15(1,1))
      equivalence(b(1,1),b8(1,1),b15(1,1))
      equivalence(vlwork(1),lwork(1))
      equivalence(vmwork(1),mwork(1))

      do 10 i=1,15
          xo=lc15w(1,i)
          eo=lc15w(2,i)
          zo=lc15w(3,i)
          va(i,1)=1.
          va(i,2)=xo
          va(i,3)=eo
          va(i,4)=zo
          va(i,5)=xo*xo
          va(i,6)=eo*eo
          va(i,7)=zo*zo
          va(i,8)=xo*eo
          va(i,9)=xo*zo
          va(i,10)=eo*zo
          va(i,11)=xo*eo*zo
          va(i,12)=zo*xo*xo
          va(i,13)=zo*eo*eo
          va(i,14)=xo*zo*zo
10      va(i,15)=eo*zo*zo

      call fminv(va,vinva,15,deter,vlwork,vmwork,15)

      return
      end


      subroutine lcoord
      integer lwork(6),mwork(6)
      integer vlwork(15),vmwork(15)
      integer
nrmbk,nrb,nelem,nonop,lo,co,mannd,nomat,nplc,ncmbk
      integer
ncb,ze,et,xii,prevty,p,pl,p2,counte,prevma,post
      integer nonpl(72,15),nonp2(72,20),rest(99,4),eno
      double precision coord(3,401),dis 1203),f(1203 .
      double precision work(360)
      double precision mat(10,9)
      double precision tr(6,6 ,d2(6,6),m(6,6),a(3,3)
      double precision xi2(6 ,eta2(6),zeta2 6)
      double precision
elco(20,3),elco8(8,3),k(60,60),jay(3,3)
      double precision
invjla(3,20),invjay(3,3),b(6,60),d(6,6),d1(6,6
      double precision
eldis(60),stn(6),invjl8(3,8),b8(6,24)
      double precision
ap(3),sts(6),lc20b 3,20),lc08b(3,8),lc15w(3,15)
      double precision
n(20),fn(3),pforc(36,5),kb(24,24),detj
      double precision
k15(45,45),elco15(15,3),invjl5(3,15),b15(6,45)
      double precision vp(15),n15(15)

common/femacm/mat,work,tr,d2,m,a,coord,xi2,eta2,zeta2,
     +
elco,k,jay,invjay,invjla,b,d,d1,dis,f,eldis,stn,sts,
     + ap,fn,n,vmwork,vlwork,lc20b,lc08b,lc15w,
     +
vp,detj,nrmbk,nrb,eno,nelem,nonop,lo,co,mannd,nomat,nplc,
     + ncmbk,icom,ncb,ne,loop,ioop,nonp1,nonp2,pforc,
     + ze,et,xii,prevty,p,p1,p2,counte,prevma,post,rest
      equivalence(k(1,1),kb(1,1),k15(1,1))
      equivalence(invjla(1,1),invjl8(1,1),invjl5(1,1))
      equivalence(n(1),n15(1))
      equivalence(elco(1,1),elco8(1,1),elco15(1,1))
      equivalence(b(1,1),b8(1,1),b15(1,1))
      equivalence(vlwork(1),lwork(1))
      equivalence(vmwork(1),mwork(1))

      do 10 i=1,nelem
          if (nonp1(i,14).eq.20) goto 20
10    continue

      goto 15

20    do 3010 j=1,20
          lc20b(1,j)=-1.
          lc20b(2,j)=-1.
3010    lc20b(3,j)=-1.

      do 3020 j=9,12
3020    lc20b(1,j)=0.

      do 3030 j=13,20
3030    lc20b(1,j)=1.

      lc20b(2,4)=0.
      lc20b(2,5)=0.
      lc20b(2,16)=0.
      lc20b(2,17)=0.
      lc20b(3,2)=0.
      lc20b(3,7)=0.
      lc20b(3,14)=0.
      lc20b(3,19)=0.
      lc20b(2,6)=1.
      lc20b(2,8)=1.
      lc20b(2,7)=1.
      lc20b(2,11)=1.
      lc20b(2,18)=1.
      lc20b(2,19)=1.
      lc20b(2,20)=1.
      lc20b(3,3)=1.
      lc20b(3,5)=1.
      lc20b(3,8)=1.
      lc20b(3,10)=1.
      lc20b(3,12)=1.
      lc20b(3,15)=1.
      lc20b(3,17)=1.
      lc20b(3,20)=1.
      lc20b(2,12)=1.

15    do 30 i=1,nelem
          if (nonp1(i,14).eq.15) goto 40
30    continue

      goto 8

40    do 100 j=1,15
          lc15w(1,j)=-1.
          lc15w(2,j)=-1.
100     lc15w(3,j)=-1.

      lc15w(1,2)=1.
      lc15w(1,3)=0.
      lc15w(1,5)=1.
      l 15w(1,6)=0.
      lc15w(1,7)=0.
      lc15w(1,8)=.5
      lc15w(1,9)=-.5
      lc15w(1,11)=1.
      lc15w(1,12)=0.
      lc15w(1,13)=0.
      lc15w(1,14)=.5
      lc15w 1,15)=-.5
      lc15w(2,3)=1.
      lc15w(2,6)=1.
      lc15w 2,8)=0.
      lc15w 2,9)=0.
      lc15w  ,12)=1.
      lc15w 2,14)=0.
      lc15w 2,15)-0.
      lc15w 3,4)=1.
      lc15w 3,5 =1.
      lc15w 3,6 =1.
      lc15w 3,10 =0.
      lc15w 3,11 =0.
      lc15w 3,12 -0.
      lc15w 3,13 =1.
      lc15w 3,14)=1.
      lc15w 3,15 =1.

8     do 50 i=1,nelem
          if  nonp1 i,14 .eq.8) g to 60
50    continue

      goto 1111

60    do 1000 j=1,8
          lc08b 1,j -1.
          lc08b 2,j)=1.
1000    lc08b 3,j =1.

      do 1010 j=1,4
1010    lc08b 1,j =-1.
      lc08b 2,1)=-1.
      lc08b(2,2 =-1.
      lc08b 2,5 --1.
      lc08b 2,6)=-1.

      do 1020 j=1,7,2
1020    lc08b(3,i)=-1.

1111  continue

      return
      end


      subroutine lam20b
      double precision
x1,x2,x3,x4,x5,x6,x7,x8,x9,x10,x11,x12,x13,x14
      double precision lam(3,20),lam8(3,8),xi,eta,zeta
      double precision
dp 3,15),va(15,15),vinva(15,15 ,lam15 3,15)
      common/lamcm/lam,xi,eta,zeta,vinva,dp,va
      equivalence (lam(1,1),lam8(1,1),lam15(1,1)
      x1=1.+xi
      x2=1.-xi
      x3=1.+eta
      x4=1.-eta
      x5=1.+zeta
      x6=1.-zeta
      x7=1.-xi*xi
      x8=1.-eta*eta
      x9=1.-zeta*zeta
      x10=xi+xi
      x11=eta+eta
      x12-zeta+zeta
      x13-.125
      x14=.25
      lam(1,1)=x13*x4*x6*(x10+eta+zeta+1.
      lam(1,2)=-x14*x9*x4
      lam(1,3)=x13*x4*x5*(x10+eta-zeta+1.)
      lam(1,4)--x14*x8*x6
      lam(1,5)=-x14*x8*x5
      lam(1,6 =x13*x3*x6*(x10-eta+zeta+1.)
      lam(1,7)=-x14*x9*x3
      lam(1,8)=x13*x3*x5* x10-eta-zeta+1.
      lam(1,9)=-.5*xi*x4*x6
      lam(1,10)=-.5*xi*x4*x5
      lam(1,11)=-.5*xi*x3*x6
      lam 1,12)=-.5*xi*x3*x5
      lam(1,13 =x13*x4*x6*(x10-eta-zeta-1.
      lam(1,14 =x14*x9*x4
      lam(1,15 =x13*x4*x5*(x10-eta+zeta-1.)
      lam(1,16 =x14*x8*x6
      lam(1,17) x14*x8*x5
      lam(1,18) x13*x3*x6*(x10+eta-zeta-1.)
      lam(1,19)=x14*x9*x3
      lam(1,20) x13*x3*x5*(x10+eta+zeta-1.)
      lam(2,1)=x13*x2*x6*(x11+xi+zeta+1.)
      lam(2,2)=-x14*x9*x2
      lam(2,3)=x13*x2*x5*(x11+xi-zeta+1.)
      lam(2,4)=-.5*eta*x2*x6
      lam(2,5)=-.5*eta*x2*x5
      lam(2,6 =x13*x2*x6  x11-xi-zeta-1.)
      lam(2,7)=x14*x9*x
      lam(2,8)=x13*x2*x   x11-xi+zeta-1.
      lam(2,9)=-x14*x7*x6
      lam(2,10 =-x14*x7*x5
      lam 2,11) x14*x7*x6
      lam(2,12 =x14*x7*x5
      lam(2,13 =x13*x1*x6*(x11-xi+zeta+1.
      lam(2,14)=-x14*x9*x1
      lam(2,15 -x13*x1*x5*(x11-xi-zeta+1.
      lam 2,16 =-.5*eta*x1*x6
      lam(2,17)=-.5*eta*x1*x5
      lam 2,18 -x13*x1*x6*(x11+xi-zeta-1.)
      lam(2,19)-x14*x9*x1
      lam 2,20 =x13*x1*x5*(x11+xi+zeta-1.
      lam 3,1) x13*x2*x4* x12+xi+eta+1.)
      lam 3,2  -.5*zeta*x2*x4
      lam 3,3  x13*x2*x4* x12-xi-eta-1.)
      lam 3,4 =-x14*x8*x2
      lam 3,5  x14*x8*x
      lam  ,6  x1 *x *x * x1 +xi-eta+1.
      lam  ,7  -.5*zeta*x *x3
      lam(3,8  x1 *x2*x3* x12-xi+eta-1.
      lam 3,9  -x14*x7*x4
      lam(3,10)=x14*x7*x4
      lam(3,11)=-x14*x7*x3
      lam(3,12)=x14*x7*x3
      lam(3,13)=x13*x1*x4*(x12-xi+eta+1.)
      lam(3,14)=-.5*zeta*x1*x4
      lam(3,15)=x13*x1*x4* x12+xi-eta-1.)
      lam(3,16)=-x14*x8*x1
      lam(3,17)=x14*x8*x1
      lam(3,18)=x13*x1*x3*(x12-xi-eta+1.)
      lam(3,19)=-.5*zeta*x1*x3
      lam(3,20)=x13*x1*x3*(x12+xi+eta-1.)
      return
      end


      subroutine lam15w
      double precision lam(3,20),lam8(3,8),xi,eta,zeta
      double precision
dp(3,15),va(15,15),vinva(15,15),lam15(3,15)
      common/lamcm/lam,xi,eta,zeta,vinva,dp,va
      equivalence (lam(1,1),lam8(1,1),lam15 1,1 )

      do 10 i=1,15
        dp(1,i)=0.
        dp(2,i)=0.
10      dp(3,i)=0.

      dp(1,2)=1.
      dp(1,5)=2.*xi
      dp(1,8)=eta
      dp(1,9)=zeta
      dp(1,11)=eta*zeta
      dp(1,12)=2.*xi*zeta
      dp(1,14)=zeta*zeta
      dp(2,3)=1.
      dp(2,6)=2.*eta
      dp(2,8)=xi
      dp(2,10)=zeta
      dp(2,11)=xi*zeta
      dp(2,13)=2.*eta*zeta
      dp(2,15)=zeta*zeta
      dp(3,4)=1.
      dp(3,7)=2.*zeta
      dp(3,9)=xi
      dp(3,10)=eta
      dp(3,11)=xi*eta
      dp(3,12)=xi*xi
      dp(3,13)=eta*eta
      dp(3,14)=2.*xi*zeta
      dp(3,15)=2.*eta*zeta
      call mxm(dp,3,vinva,15,lam15,15)

      return
      end


      subroutine transf(ium,zeta)
      integer lwork 6),mwork(6)
      integer vlwork(15),vmwork(15)
      integer
nrmbk,nrb,nelem,nonop,lo,co,mannd,nomat,nplc,ncmbk
      integer
ncb,ze,et,xii,prevty,p,p1,p2,counte,prevma,post
      integer nonp1 72,15),nonp2(72,20),rest 99,4),eno
      double precision coord 3,401),dis(1203 ,f(1203)
      double precision work 360),ffact
      double precision mat(10,9)
      double precision tr(6,6),d2(6,6),m(6,6),a(3,3)
      double precision xi2(6),eta2(6),zeta2(6
      double precision
elco(20,3),elco8(8,3),k(60,60),jay(3,3)
      double precision
invjla(3,20),invjay(3,3),b(6,60),d(6,6),d1(6,6)
      double precision
eldis(60),stn(6),invjl8(3,8),b8(6,24)
      double precision
ap(3),sts(6),lc20b 3,20),lc08b(3,8),lc15w(3,15)
      double precision n(20 ,fn(3),pforc(36,5),kb(24,24)
      double precision dum,detj
      double precision degrad,pi,deter
      double precision
k15(45,45),elco15(15,3),invj15(3,15),b15(6,45)
      double precision vp(15),n15(15)

common femacm/mat,work,tr,d2,m,a,coord,xi2,eta2,zeta2,
     +
elco,k,jay,invjay,invjla,b,d,d1,dis,f,eldis,stn,sts,
     + ap,fn,n,vmwork,vlwork,lc20b,lc08b,lc15w,
     +
vp,detj nrmbk,nrb,eno,nelem,nonop,lo,co,mannd,nomat,nplc,
     + ncmbk,icom,ncb,ne,loop,ioop,nonp1,nonp2,pforc,
     + ze,et,xii,prevty,p,p1,p2,counte,prevma,post,rest
      equivalence(k(1,1),kb 1,1),k15(1,1 )
      equivalence(invjla(1,1 ,invjl8(1,1 ,invj15(1,1 )
      equivalence n 1),n15 1))
      equivalence e co 1,1 ,elco8(1,1 ,elco15 1,1 )
      equivalence b 1,1),b8 1,1),b15(1,1))
      equivalence(vlwork(1),lwork(1 )
      equivalence vmwork(1),mwork(1))
      double precision zeta
      integer ium

      data pi/3.14159265359d0/

      degrad=pi 18 .d0

      if ium.eq.. goto 1209
```

```
      kkk=nonp1 ne,14

      do 1120 i=1,kkk
         nonnei=n np2 ne,i
         do 1120 j 1,3
1120     elco(i,j -coord j,nonnei

      do 1140 i=1,6
         do 114  j=1,6
1140   m i,j)=0.

      j=nonp1(ne,2)
      m 1,1 =1. mat j,1
      m 2,2)=1. mat(j,2
      m(3,3 =1. mat(j,3)
      m(4,4)=1. mat(j,7)
      m(5,5)=1. mat(j,8)
      m(6,6)=1. mat(j,9)
      dum=-mat j,4) mat(j,2)
      m(1,2)=dum
      m(2,1)=dum
      dum=-mat j,5  mat(j,3)
      m 1,3)=dum
      m 3,1)=dum
      dum=-mat j,6  mat j,3)
      m 2,3)=dum
      m(3,2)=dum
      call fminv m,d1,6,deter,lwork,mw rk,6
      if(nonp1 ne,4 .eq.0.or.nonp1 ne,4 .eq.2)  g t  14 0
      if(nonp1 ne,4).eq.1) g to 1206
      a(1,1)=dble float(nonp1 ne,5  )
      a(2,1)=dble float n np1(ne,6  )
      a(3,1)-dble float(nonp1 ne,7
      a(1,2)=dble float(nonp1(ne,8))
      a(2,2)=dble float nonp1(ne,9
      a(3,2)=dble(float(nonp1(ne,10) )
      a(1,3)=dble(float(nonp1(ne,11  )
      a(2,3)=dble(float(nonp1(ne,12)))
      a(3,3)=dble(float(nonp1(ne,13)))

      goto 1207

1206  ffact=( float(nonp1 ne,6)-nonp1 ne,5  )*
     + ((zeta+1.d0)/2.d0))+dble(float nonp1 ne,5 ))
      a(1,1)=ffact
      a(2,1)=ffact+270.d0
      a(3,1)=90.d0
      a(1,2)=ffact+90.d0
      a(2,2)=ffact
c     end of edits to generalise to all

      a(3,2)=90.d0
      a 1,3 =90.d0
      a 2,3 =90.d0
      a(3,3)=0.d0

1207  do 1210 i=1,3
         do 1210 j=1,3
1210     a(i,j)-dcos(a i,j)*degrad)

      goto 1208
c
c     setup local orthogonal (to the fibre  material
c     directions
c

1209  call elemdir(a,invjay)

1208  continue

      tr(1,1)=a(1,1)*a(1,1)
      tr(1,2)=a(2,1)*a(2,1)
      tr(1,3)=a(3,1)*a(3,1)
      tr(1,4)=a 1,1 *a 2,1)
      tr(1,5)=a 2,1)*a(3,1)
      tr(1,6)=a(1,1)*a(3,1)
      tr(2,1)=a(1,2)*a 1,2)
      tr(2,2)=a 2,2)*a 2,2)
      tr(2,3)=a 3,2)*a(3,2)
      tr(2,4)=a 1,2)*a(2,2)
      tr(2,5)=a(2,2 *a(3,2
      tr(2,6)=a 1,2 *a(3,2)
      tr(3,1)=a(1,3 *a     )
      tr(3,2)=a 2,3 *a
      tr(3,3)=a 3,3 *a  ,3)
      tr(3,4)-a(1,3 *a 2,3)
      tr(3,5 =a(2,3)*a(3,3)
      tr(3,6)-a(1,3)*a(3,3)
      tr(4,1 =2.d0*a 1,1 *a(1,2
      tr(4,2)=2.d0*a 2,1)*a(2,2
      tr(4,3)=2.d0*a 3,1)*a(3,2
      tr(4,4 =a 1,1 *a(2,2 +a 2,1 *a 1,
      tr(4,5 -a 2,1 *a 3,2 +a 3,1 *a 2,2
      tr 4,6  a 1,1 *a 3,2 +a  ,1 *a 1,
      tr 5,1 =2.d0*a 1,2 *a 1,
      tr 5,2  2.d0*a 2,2 *a  ,3
      tr 5, )=2.d0*a 3,2 *a 3,3
      tr 5,4) a(1,2 *a 2,3 +a 2,2 *a 1,
      tr 5,5)-a  ,2)*a 3,3 +a(3,  *a  ,3
      tr 5,6 =a 1,2 *a 3, )+a 3,  *a(1,
      tr 6,1   .d *a 1,  *a 1,1
      tr 6,2  2.d *a 2,  *a 2,1
      tr 6,3  2.d *a  ,   *a 3,1
      tr 6,4  a 1,  *a  ,1 +a  ,  *a  ,
      tr 6,5  a  ,  *a 3,1 +a  ,  *a  ,
      tr 6,6  a 1,  *a  ,1 +a  ,  *a  ,
      do 1220 i=1,6
          do 1220 j=1,6
1220      d2(i,j)=0.d0

      call btdb(tr,d1,d2,6,6,work)

      do 1230 i=1,6
          do 1230 j=i,6
1230      d1(i,j)=d2(i,j)

      d1(2,1)=d1(1,2)
      d1(3,1)=d1(1,3)
      d1(4,1)=d1(1,4)
      d1(5,1)=d1(1,5)
      d1(6,1)=d1(1,6)
      d1(3,2)=d1(2,3)
      d1(4,2)=d1(2,4)
      d1(5,2)=d1(2,5)
      d1(6,2)=d1(2,6)
      d1(4,3)=d1(3,4)
      d1(5,3)=d1(3,5)
      d1(6,3)=d1(3,6)
      d1(5,4)=d1(4,5)
      d1(6,4)=d1(4,6)
      d1(6,5)=d1(5,6)

1400  continue

      return
      end


      subroutine elemdir(a,invjay)
c
c     This subroutine is called only by subroutine transf
c
c     This subroutine makes calls to the following
c     subroutines: vecang, mortho and munit
c     it also uses the function rotang
c
c     this routine calculates the orthogonal coordinate
axes
c     for the fibres within an ansiotropic material
c
      double precision a(3,3),v0(3),v1(3),v2(3),v3(3),
     + ang,invjay(3,3)

c
c     set new y-axis which is also the fibre direction
and
c     also set initial aproximations to the local
orthogonal
c     x and z-axes
c
      v1(1)=invjay(1,1)
      v1(2)=invjay(2,1)
      v1(3)=invjay(3,1)
      v2(1)=invjay(1,2)
      v2(2)=invjay(2,2)
      v2(3)=invjay(3,2)
      v3(1)=invjay(1,3)
      v3(2)=invjay(2,3)
      v3(3)=invjay(3,3)
      call munit(v2)
c
c     make local direction vector v1 orthogonal to vector
v2
c
      call vecang(v2,v1,ang)
      call mortho(v2,v1,ang)
      call munit(v1)
c
c     make local direction vector v3 orthogonal to vector
v2
c     and also make it a unit vector
c
      call vecang(v2,v3,ang)
      call mortho(v2,v3,ang)
      call munit(v3)
c
c     make local direction vector v3 and v1 orthogonal
c     and also make it a unit vector
c
      call vecang(v1,v3,ang)
      v0(1)=v1(1)
      v0(2)=v1(2)
      v0(3)=v1(3)
      ang=ang/2.d0+3.14159265359d0/4.d0
       all mortho(v3,v1,ang)
      call mortho(v0,v3,ang
      call munit(v1)
      call munit(v3)

      a(1,1)=v1(1)
      a(2,1)=v1(2
      a(3,1)=v1(3
      a(1,2)=v2(1)
      a(2,2)=v2(2)
      a(3,2)=v2(3)
      a(1,3)=v3 1)
      a(2,3)=v3(2)
      a(3,3)=v3 3)

      return
      end
```

```
      subroutine vecang rfv,vec,ang
c
c     This subroutine is called only by the subroutine
elemdir
c
c     This subroutine calculates the angle between vect r
c     "vec" and the unit vector "rfv"

      double precision rfv(3),vec(3 ,ang,cosang

      cosang=(rfv 1)*vec(1)+rfv 2)*vec 2 +rtv 3)*vec 3)
     + dsqrt(vec 1)*vec(1)+vec(2)*vec 2)+vec 3 *vec 3 )

      if((cosang.ge.1.d0).or.(cosang.le.-1.d0)) then
        ang=0.d0
      else
        ang=dacos(cosang)
      endif

      return
      end


      subroutine mortho(rfv,vec,ang
c
c     This subroutine is called only by the subr utine
elemdir
c
c     This subroutine makes vector "vec" orthogonal to
vector
c     "rfv" if ang is the angle between the two vect rs
c
      double precision rfv(3),vec 3),ang,fact


fact=dsqrt(vec(1)*vec(1)+vec(2)*vec 2)+vec(3)*vec 3))*dc
s(ang)

      vec(1)=vec(1)-rfv(1)*fact
      vec(2)=vec(2)-rfv(2)*fact
      vec(3)=vec(3)-rfv(3)*fact

      return
      end


      subroutine munit(vec)
c
c     This subroutine is called only by the subr utine
elemdir
c
c     This subroutine makes vector "vec" into a unit
vector
c
      double precision vec(3),fact


fact=dsqrt(vec(1)*vec 1)+vec(2)*vec(2 +vec(3)*vec 3))

      vec(1)=vec(1)/fact
      vec(2)=vec(2)/fact
      vec(3)=vec(3)/fact

      return
      end


      subroutine fminv b,a,n,d,l,m,nn)
      double precision a n*n),b n*n),d,biga,h ld
      integer j,n,nn,l nn ,m(nn)

      j=n*n
      do 4 i=1,j
4       a(i)=b i

      d=1.
      nk=-n
      do 80 k=1,n
      nk=nk+n
      l(k)-k
      m(k)-k
      kk=nk+k
      biga a kk)
      do 20 j=k,n
          iz=n* j-1
          do 20 i k,n
            ij=iz+i
10        if  dabs biga -dabs a(ij  ) 15,2 ,2
15        biga-a ij
            l k =i
            m k -j
20    continue

      j l(k
      if(j-k) 35,35,25

25    ki k-n
      do 3  i 1,n
          ki=ki+n
          h ld=-a ki
          ji ki-k+j
          a ki  a  i
30      a 'i  h ld

35    i-m k
      if i-r  45,45,
38     jp=n*(i-1)
       do 40 j=1,n
          jk=nk+j
            ji=jp+j
            hold=-a(jk)
            a(jk)=a(ji)
40        a(ji)=hold

45       if(dabs(biga)-1.d-20) 46,46,48

46       d=0.
        return
48      do 55 i=1,n
              if(i-k) 50,55,50
50         ik=nk+i
              a(ik)=a(ik)/(-biga)
55      continue

        do 65 i=1,n
             ik=nk+i
             hold=a(ik)
             ij=i-n
             do 65 j=1,n
                ij=ij+n
                if(i-k) 60,65,60
60           if(j-k) 62,65,62
62           kj=ij-i+k
                a(ij)=hold*a(kj)+a(ij)
65          continue

         kj=k-n

         do 75 j=1,n
              kj=kj+n
              if(j-k) 70,75,70
70         a(kj)=a(kj)/biga
75       continue

         d=d*biga
         a(kk)=1./biga
80       continue
         k=n
100      k=(k-1)
         if(k) 150,150,105
105      i=l(k)
         if(i-k) 120,120,108
108      jq=n*(k-1)
         jr=n*(i-1)

         do 110 j=1,n
              jk=jq+j
              hold=a(jk)
              ji=jr+j
              a(jk)=-a(ji)
110      a(ji)=hold
120      j=m(k)
         if(j-k) 100,100,125
125      ki=k-n

         do 130 i=1,n
              ki=ki+n
              hold=a(ki)
              ji=ki-k+j
              a(ki)=-a(ji)
130         a(ji)=hold

          go to 100
150       return
          end


          subroutine btdb(b,d,ak,ix,iy,btd)
          double precision
 btd(iy,ix),b(ix,iy),d(ix,ix),ak(iy,iy)

          do 1000 j=1,ix
               do 500 i=1,iy
500            btd(i,j)=0.
                  do 3000 l=1,ix
                     do 3000 i=1,iy
3000                 btd(i,j)=btd(i,j)+b(l,i)*d(l,j)
1000      continue

          do 4000 im=1,iy
               i=iy+1-im
               do 5000 l=1,ix
                  do 5000 j=1,i
5000              ak(j,i)=ak(j,i)+btd(j,l)*b(l,i)
4000      continue

          return
          end


          subroutine dblok (f,ncb,nrb)
          double precision wrmbk(3,339),stiff(3,339,401)
          common/wmbk/wmbk,prmbk
          double precision wmbk(3,339),prmbk(3,339)
          equivalence(wmbk(1,1),wrmbk(1,1 )
          double precision  invsea(3,3)
          double precision deter,bug
          double precision
  f(1203),ff1(3),ff2(3),a(3,3),a2(3,3)
          double precision dd(113,3,3),a3(3,3)
          integer map(113),lwork(3),mwork 3 ,ncb,nrb
          common smatrix/stiff
```

```
       ncmbk=3*ncb
       mannd=ncb-1

       do 1000 i=1,nrb
6789     format i4)
           call storemat(i,0,prmbk,stiff)
           do 1020 ii=1,3
             do 1020 jj=1,3
1020       a ii,jj)=prmbk(ii,jj)
           call fminv(a,invsea,3,deter,lwork,mwork,3)
           jdum=3*i
           idum=3*i-2
           kdum=0
           do 1030 ii=idum,jdum
             kdum=kdum+1
1030       ff1(kdum)=f(ii)
         call mxm(invsea,3,ff1,3,ff2,1
           kdum=0
           do 1040 ii=idum,jdum
             kdum=kdum+1
1040       f ii)=ff2(kdum)

        do 1045 imk=1,ncb
            k=ncb-imk+1
            kdum=i-(nrb-mannd)+k
            if  kdum.gt.ncb) goto 1045
            bug=0.
            jdum=3*k
            idum=3*k-2
          do 1060 ii=1,3
              ldum=0
              do 1060 jj=idum,jdum
                ldum=ldum+1
                a(ii,ldum)=prmbk(ii,jj)
                dd(k,ii,ldum)=a(ii,ldum
1060          bug=bug+prmbk(ii,jj)

          if (bug.ne.0.) goto 1070
          map(k)=0
          goto 1045

1070      call mxm(invsea,3,a,3,a2,3)

          do 1080 ii=1,3
              ldum-0
              do 1080 jj=idum,jdum
              ldum=ldum+1
1080        prmbk(ii,jj)=a2(ii,ldum)

          map(k)=1
1045    continue

        do 1090 j=1,mannd
            if (i+j.gt.nrb) goto 1090
            if (map(j+1).eq.0) goto 1090
            do 1110 ii-1,3
              do 1110 jj 1 3
1110          a3(jj,ii)=dd  +1,ii,jj)
            ip=i+j
            call storemat(ip,0,wrmbk,stiff)
            jdum=ncb-j
            do 1120 k=1,jdum
              kdum=3*(j+k)-2
              ldum=3*(j+k)
              do 1130 ii=1,3
                mdum=0
                do 1130 jj=kdum,ldum
                mdum=mdum+1
1130          a(ii,mdum)=prmbk(ii,jj)
           call mxm(a3,3,a,3,a2,3)
             ldum=3*k
             kdum=ldum-2
             do 1140 ii=1,3
               mdum=0
               do 114  jj-kdum,ldum
               mdum=mdum+1
1140         wrmbk(ii,jj =wrmbk ii,jj)-a2(ii,mdum)
1120       continue
             call storemat(ip,1,wrmbk,stiff)
           call mxm a3,3,ff2,3,ff1,1)
             kdum=3*(i+j
             jdum=kdum-2
             ldum=0
             do 1150 ii-jdum,kdum
               ldum ldum+1
1150         f(ii)=f(ii)-ff1 l  m)
1090    continue

        call storemat(i,1,prmbk,stiff)
1000    continue
        idum=nrb-1

        do 1160 im=1,idum
            i-idum-im+1
            call storemat(i,0,wrmbk,stiff)
            do 1170 j-2,ncb
              jdum=i+j-1
              if  jdum.gt.nrb  goto 1170
              bug-0.
              ldum 3*j
              kdum=ldum-2
              d  1190 ii 1,3
                mdum-0
                do 1190 jj kdum,ldum
                  md m mdum+1
                  a ii,mdum -wrmbk ii,
                  bug bug+wrmbk ii,
1190        continue
            if (bug.eq.0.) goto 1170
            jd=3*(i+j-1)
            id=jd-2
            kd=0
            do 1200 ii=id,jd
              kd=kd+1
1200        ff1(kd)=f(ii)
          call mxm(a,3,ff1,3,ff2,1)
            jd=3*i
            id-jd-2
            kd=0
            do 1210 ii=id,jd
              kd=kd+1
1210        f(ii)=f(ii)-ff2(kd)
1170     continue
1160   continue

       return
       end


       subroutine smhstr(iswl,hed)
       double precision lam(3,20),lam8(3,8),xi,eta,zeta
       double precision
 dp(3,15),va(15,15),vinva(15,15),lam15(3,15)
       common/lamcm/lam,xi,eta,zeta,vinva,dp,va
       equivalence (lam(1,1),lam8(1,1),lam15(1,1))
       integer lwork(6),mwork(6)
       integer vlwork(15),vmwork(15)
       integer
 nrmbk,nrb,nelem,nonop,lo,co,mannd,nomat,nplc,ncmbk
       integer
 ncb,ze,et,xii,prevty,p,p1,p2,counte,prevma,post
       integer nonpl(72,15),nonp2(72,20),rest(99,4),eno
       double precision coord(3,401),dis(1203),f(1203)
       double precision work(360)
       double precision mat(10,9)
       double precision tr(6,6),d2(6,6),m(6,6),a(3,3)
       double precision xi2(6),eta2(6),zeta2(6)
       double precision
 elco(20,3),elco8(8,3),k(60,60),jay(3,3)
       double precision
 invjla(3,20),invjay(3,3),b(6,60),d(6,6),d1(6,6)
       double precision
 eldis(60),stn(6),invjl8(3,8),b8(6,24)
       double precision
 ap(3),sts(6),lc20b(3,20),lc08b(3,8),lc15w(3,15)
       double precision
 n(20),fn(3),pforc(36,5),kb(24,24),detj,deter
       double precision
 k15(45,45),elco15(15,3),invjl5(3,15),b15(6,45)
       double precision vp(15),n15(15)
       double precision asig(8,8),bsts(8,6),ssts(8,6)
       double precision dhp,dhm,sq3,adum,bdum,cdum,ddum
       double precision
 hh(14),xil(15),etal(15),zetal(15),gsts(60,8,6)
       integer id(8)
       character hed*80

 common/femacm/mat,work,tr,d2,m,a,coord,xi2,eta2,zeta2,
       +
 elco,k,jay,invjay,invjla,b,d,dl,dis,f,eldis,stn,sts,
       + ap,fn,n,vmwork,vlwork,lc20b,lc08b,lc15w,
       +
 vp,detj,nrmbk,nrb,eno,nelem,nonop,lo,co,mannd,nomat,nplc,
       + ncmbk,icom,ncb,ne,loop,ioop,nonpl,nonp2,pforc,
       + ze,et,xii,prevty,p,pl,p2,counte,prevma,post,rest
        equivalence(k(1,1),kb(1,1),k15(1,1))
        equivalence(invjla(1,1),invjl8(1,1),invjl5(1,1))
        equivalence(n(1),n15(1))
        equivalence(elco(1,1),elco8(1,1),elco15(1,1))
        equivalence(b(1,1),b8(1,1),b15(1,1))
        equivalence(vlwork(1),lwork(1))
        equivalence(vmwork(1),mwork(1))

        data hh/8*0.335180055d0,6*0.886426593d0/
        data xil/2*-0.758786911d0,2*0.758786911d0,2*-
 0.758786911d0,
       +  2*0.758786911d0,0.d0,-
 0.795822426d0,0.d0,0.d0,0.795822426d0,
       + 2*0.d0/
        data etal/4*0.758786911d0,4*-
 0.758786911d0,0.795822426d0,0.d0,
       +  0.d0,0.d0,0.d0,-.795822426d0,0.d0/
        data zetal/-0.758786911d0,0.758786911d0,-
 0.758786911d0,
       + 0.758786911d0,-0.758786911d0,0.758786911d0,
       + -0.758786911d0,0.758786911d0,2*0.d0,-
 0.795822426d0,
       + 0.7958 _426d0,3*0.d0/
        data id/13,18,6,1,15,20,8,3/
        data lin/1/,lout/3/,ldsp/2/

        dhp=.57735026918962
        dhm=dhp*(-1.)

        do 1010 i=1,8
            xil(i)=dhp
            etal(i)=dhp
 1010     zetal(i)=dhp

        xil(3)=dhm
        xil(4)=dhm
        xil(7)=dhm
        xil(8)=dhm
        etal(1)=dhm
        etal(4)=dhm
        etal(5)=dhm
        etal(8)=dhm
        zetal(1)=dhm
        zetal(2)=dhm
        zetal(3)=dhm
        zetal(4)=dhm
        sq3=dsqrt(3.d0)
        adum=(5.+(3.*sq3))*.25
        bdum=(sq3+1.)*(-.25)
        cdum=(sq3-1.)*.25
        ddum=(5.-(sq3*3.))*.25

        do 1020 i=1,8
            ki=i+1
            do 1020 j=ki,8
1020        asig(i,j)=bdum

        do 1030 i=1,8
1030      asig(i,i)=adum

        asig(1,3)=cdum
        asig(1,6)=cdum
        asig(1,8)=cdum
        asig(2,4)=cdum
        asig(2,5)=cdum
        asig(2,7)=cdum
        asig(3,6)=cdum
        asig(3,8)=cdum
        asig(4,5)=cdum
        asig(4,7)=cdum
        asig(5,7)=cdum
        asig(6,8)=cdum
        asig(1,7)-ddum
        asig(2,8)=ddum
        asig(3,5)=ddum
        asig(4,6)=ddum

        do 1040 i=1,8
            ki=i+1
            do 1040 j=ki,8
1040        asig(j,i)=asig(i,j)

        prevty=0

        do 1050 ne=1,nelem
6789      format(i4)
            if(nonpl(ne,14).ne.20) goto 1050
            write(lout,100)
100       format(1x,'element centroidal stress ',/,
       +1x, 'element node    stress xx',6x,
       +'stress yy',6x,'stress zz',6x,'stress
 xy',6x,'stress yz',6x,
       +'stress zx')

            do 1060 i=1,20
              ii=3*(i-1)
              jj=3*(nonp2(ne,i) 1
              do 1060 j=1,3
                iij=ii+j
                jjj=jj+j
1060          eldis(iij)=f(jjj)

            if(nonpl(ne,4).ge.2) call transf(1,0.d0)
            prevty=nonpl(ne,3)

            do 1070 post=1,8
              xi=xil(post)
              eta=etal(post)
              zeta=zetal(post)
              if(nonpl(ne,4).le.1) call transf(1,zeta)
              call lam20b
            call mxm(lam,3,elco,20,jay,3)
              call fminv(jay,invjay,3,deter,lwork,mwork,6)
              if(nonpl(ne,4).le.1) call transf(2,0.d0)
            call mxm(invjay,3,lam,3,invjla,20)
              do 1080 i=1,6
                do 1080 j=1,60
1080            b(i,j)=0.

              do 1090 i=1,20
                b(1,3*i-2)=invjla(1,i)
                b(4,3*i-1)=invjla(1,i)
1090          b(6,3*i)=invjla(i,i)

              do 1100 i=1,20
                b(2,3*i-1)=invjla  ,i)
                b(4,3*i-2)=invjla(2,i)
1100          b(5,3*i)=invjla(2,i)
              do 1110 i=1,20
                b(3,3*i)=invjla(3,i)
                b(5,3*i-1) invjla(3,i)
1110          b(6,3*i-2)=invjla(3,i)
            call mxm(b,6,eldis,60,stn,1)
              do 1120 i=1,6
1120          if(stn(i +0.9.ge.0.) stn(i)-dl g(1.+stn(i )
            call mxm(dl,6,stn,6,sts,1)
              do 1130 i=4,6
1130          stn(i)-0.5*stn i)
              do 1140 i-1,6
1140          bsts post,i =sts(i)
1070        continue
            call mxm asig,8,bsts,8,ssts,6
              do 1150 p st 1,8
                idn=id post
                id=-n npl ne,1
```

```
write(lout,110)ide,nonp2(ide,idn),(ssts(post,i),i=1,6)
110        format(1x,i3,4x,i3,1x,6e14.7)
             if(iswl.eq.0) goto 1150
             if (ne.gt.60) goto 1150
             do 2001 i=1,6
2001         gsts(ne,post,i) = ssts(post,i)
1150   continue

111    format(6f10.3)
1050   continue

       if (iswl.eq.1) call plot72(gsts,hed)

       return
       end


       subroutine plot72(gsts,tit)
       double precision
gsts(60,8,6),casts(8,28),ctsts(8,30)
       double precision xcof(4),cof(4),rootr(3),rooti(3)
       double precision
s1,s2,s3,s4,s5,s6,rootm,sigev,shrm,hsts
       double precision a,b,c,ab,bc,ca,abss,bcs,cas
       integer ier
       character tit*80
c
c        calculate centre stresses
c
       do 1000 i=1,30
           do 2000 j=1,2
             s1=0.
             s2=0.
             s3=0.
             s4=0.
             s5=0.
             s6=0.
             if(j.eq.2) ii=2*i
             if(j.eq.1) ii=(2*i)-1
             do 3000 k=1,4
c            average mid face values
               l=k+(j-1)*4
               s1=gsts(ii,l,1)+s1
               s2=gsts(ii,l,2)+s2
               s3=gsts(ii,l,3)+s3
               s4=gsts(ii,l,4)+s4
               s5=gsts(ii,l,5)+s5
               s6=gsts(ii,l,6)+s6
3000       continue
             s1=s1*0.25
             s2=s2*0.25
             s3=s3*0.25
             s4=s4*0.25
             s5=s5*0.25
             s6=s6*0.25
             xcof(3)=(s1+s2+s3)*(-1.)
             xcof(2)=s1*s2+s2*s3+s3*s1-s4*s4-s5*s5-s6*s6
             xcof(1)=(s1*s2*s3-s1*s5*s5-s2*s6*s6-
s3*s4*s4+2*s4*s5*s6)*(-1.)
             xcof(4)=1.
             call polrt(xcof,cof,3,rootr,rooti,ier)
             rootm=rootr(1)
             if(rootr(2).gt.rootm) rootm=rootr(2)
             if(rootr(3).gt.rootm) rootm=rootr(3)
             ctsts(j,i)=rootm
             a=rootr(1)
             b=rootr(2)
             c=rootr(3)
             ab=a-b
             bc=b-c
             ca=c-a
             abss=ab*ab
             bcs=bc*bc
             cas=ca*ca
             sigev=dsqrt(0.5*(abss+bcs+cas))
             ctsts(j+2,i)=sigev
             shrm=s4
             if(s5.gt.shrm) shrm=s5
             if(s6.gt.shrm) shrm=s6
             hsts=(rootr(1)+rootr(2)+rootr(3))/3.
             ctsts(j+4,i)=shrm
             ctsts(j+6,i)=hsts
2000     continue
1000   continue

c        calculate corner stresses
       do 4000 i=1,4
           do 5000 k=1,2
             if(k.eq.2) ii=1
             if(k.eq.1) ii=0
             mmo=((i-1)*6)+1
             m=((i-1)*12)+1+ii
             l=(k-1)*4
             m5=m+10
             m11=m+22
             m6=m+12
             l1=l+1
             l2=l+2
             l3=l+3
             l4=l+4

s1=gsts(m,l1,1)+gsts(m5,l2,1)+gsts(m11,l3,1)+gsts(m6,l4,1
)

s2-gsts(m,l1,2)+gsts(m5,l2,2)+gsts(m11,l3,2 +gsts(m6,l4,2
)

s3=gsts(m,l1,3)+gsts(m5,l2,3)+gsts m11,l3,3 +gsts(m6,l4,3
)

s4=gsts(m,l1,4)+gsts(m5,l2,4)+gsts(m11,l3,4)+gsts(mb,l4,4
)

s5=gsts(m,l1,5)+gsts m5,l2,5 +gsts m11,l3,5 +gsts mb,l4,5
)

s6=gsts(m,l1,6 +gsts(m5,l2,6)+gsts m11,l3,6)+gsts(mb,l4,6
)
             s1=s1*0.25
             s2=s2*0.25
             s3=s3*0.25
             s4=s4*0.25
             s5=s5*0.25
             s6=s6*0.25
             xcof(3)=(s1+s2+s3)*(-1.)
             xcof(2)=s1*s2+s2*s3+s3*s1-s4*s4-s5*s5-s6*sb
             xcof(1)=(s1*s2*s3-s1*s5*s5-s2*s6*s6-
s3*s4*s4+2*s4*s5*s6)*(-1.)
             xcof 4 =1.
             call polrt(xcof,cof,3,rootr,rooti,ier
             rootm=rootr(1)
             if(rootr(2).gt.rootm) rootm=rootr 2
             if(rootr(3).gt.rootm  rootm rootr 3
             casts(k,mmo)=rootm
             a=rootr(1)
             b=rootr 2)
             c=rootr(3)
             ab=a-b
             bc=b-c
             ca=c-a
             abss-ab*ab
             bcs=bc*bc
             cas=ca*ca
             sigev=dsgrt(0.5*(abss+bcs+cas))
             casts(k+2,mmo)=sigev
             shrm=s4
             if(s5.gt.shrm) shrm=s5
             if(s6.gt.shrm) shrm=s6
             hsts=(rootr(1)+rootr(2)+rootr(3))/3.
             casts(k+4,mmo)=shrm
             casts(k+6,mmo)=hsts
             do 6000 n=1,5
               nno=mmo+n
               nn=m+2+(n-1)*2
               nn1=nn-2
               nn5=nn+10
               nn6=nn+12

s1=gsts(nn,l1,1)+gsts(nn1,l2,1)+gsts(nn5,l3,1)+gsts(nn6,l
4,1)

s2=gsts(nn,l1,2)+gsts(nn1,l2,2)+gsts(nn5,l3,2  gsts(nn6,l
4,2)

s3=gsts(nn,l1,3)+gsts(nn1,l2,3)+gsts(nn5,l3,3)+gsts(nn6,l
4,3)

s4=gsts(nn,l1,4)+gsts(nn1,l2,4)+gsts(nn5,l3,4)+gsts(nn6,l
4,4)

s5=gsts(nn,l1,5)+gsts(nn1,l2,5)+gsts(nn5,l3,5)+gsts(nn6,l
4,5)

s6=gsts(nn,l1,6)+gsts(nn1,l2,6)+gsts(nn5,l3,6)+gsts(nn6,l
4,6)
               s1=s1*0.25
               s2=s2*0.25
               s3=s3*0.25
               s4=s4*0.25
               s5=s5*0.25
               s6=s6*0.25
               xcof(3)=(s1+s2+s3)*(-1.)
               xcof(2)=s1*s2+s2*s3+s3*s1-s4*s4-s5*s5-s6*s6
               xcof(1)=(s1*s2*s3-s1*s5*s5-s2*s6*s6-
s3*s4*s4+2*s4*s5*s6)*(-1.)
               xcof(4)=1.
               call polrt(xcof,cof,3,rootr,ro ti,ier
               rootm=rootr(1)
               if(rootr(2).gt.rootm) rootm=rootr(2)
               if(rootr(3).gt.rootm) rootm rootr(3)
               casts(k,nn     otm
               a=rootr(1)
               b=rootr(2)
               c=rootr(3)
               ab=a-b
               bc=b-c
               ca=c-a
               abss=ab*ab
               bcs=bc*bc
               cas=ca*ca
               sigev=dsqrt(0.5*(abss+bcs+cas))
               casts(k+2,nno)-sigev
               shrm=s4
               if s5.gt.shrm) shrm=s5
               if(s6.gt.shrm) shrm s6
               hsts=(rootr(1)+rootr(2)+r  tr(3 )/3.
               casts k+4,nno) shrm
               casts k+6,nno -hsts
6000         continue
5000       continue
4000   continue
1002   format 6f10.3
```

```
      call prints(casts,ctsts,tit)

      return
      end


      subroutine polrt(xcof,cof,m,rootr,rooti,ier)
      double precision xcof(4),cof(4),rootr(3),rooti(3
      double precision
xo,yo,x,y,xpr,ypr,ux,uy,v,yt,xt,u,xt2,yt2
      double precision sumsq,tmp,alpha,dx,dy,fi
      integer m,ier

      ifit=0
      n=m
      ier=0
      if(xcof(n+1)) 10,25,10
10    if(n) 15,15,32
c       set error code to 1
15    ier=1
20    return
c       set error code to 4
25    ier=4
      goto 20
c      set error code to 2
30    ier=2
      goto 20
32    if(n-36) 35,35,30
35    nx=n
      nxx=n+1
      n2=1
      kj1=n+1
      do 40 l=1,kj1
          mt=kj1-l+1
40      cof(mt)=xcof(l)

c      set initial values
45    xo=.00500101
      yo=.01000101
c      zero initial value counter
      in=0
50    x=xo
c      increment initial values and counter
      xo=-10.0*yo
      yo=-10.0*x
c      set x and y to current value
      x=xo
      y=yo
      in =in+1
      goto 59
55    ifit=1
      xpr=x
      ypr=y
c      evaluate polynomial and derivatives
59    ict=0
60    ux=0.0
      uy=0.0
      v=0.0
      yt=0.0
      xt=1.0
      u=cof(n+1)
      if(u) 65,130,65

65    do 70 i=1,n
          l=n-i+1
          xt2=x*xt-y*yt
          yt2=x*yt+y*xt
          u=u+cof(l)*xt2
          v=v+cof(l)*yt2
          fi=i
          ux=ux+fi*xt*cof(l)
          uy=uy-fi*yt*cof(l)
          xt=xt2
70      yt=yt2

      sumsq=ux*ux+uy*uy
      if(sumsq) 75,110,75
75    dx=(v*uy-u*ux)/sumsq
      x=x+dx
      dy=-(u*uy+v*ux)/sumsq
      y=y+dy
78    if(dabs(dy)+dabs(dx)-1.0d-05) 100,80,80
c      set iteration counter
80    ict=ict+1
      if(ict-500) 60,85,85
85    if(ifit) 100,90,100
90    if in-5) 50,95,95
c      set error code to 3
95    ier=3
      g to 20

100   do 105 l=1,nxx
          mt=kj1-l+1
          tmp=xcof(mt)
          xcof(mt)=cof(l)
105     cof(l)=tmp

      itmp=n
      n=nx
      nx=itmp
      if(ifit) 120,55,120
110   if(ifit) 115,50,115
115   x=xpr
      y=ypr
120   ifit=0
122   if dabs(y x)-1.0d-04) 135,125,1_5
125   alpha-x+x
```

```
      sumsq=x*x+y*y
      n-n-2
      goto 140
130   x=0.0
      nx=nx-1
      nxx=nxx-1
135   y=0.0
      sumsq=0.0
      alpha=x
      n=n-1
140   cor 2)=cof 2 +alpha*cof 1

145   do 150 l=2,n
150     cof(l+1)=cof l+1 +alpha*cof l -sumsq*cof l-1)

155   rooti(n2)=y
      rootr(n2 =x
      n2=n2+1
      if(sumsq  160,165,160
160   y=-y
      sumsq=0.0
      goto 155
165   if n) 20,20,45
      end


      subroutine prints casts,ctsts,tit
      double precision casts 8,28 ,ctsts 8,3
      integer i,k
      character tit*80,st*12
      data lin 1/,lout/3 ,ldsp 2

      st=' ********** '


      do 1000 i=1,8
          write(lout,10) tit
10      format(//' ',a80,//)
          if(i.eq.1.or.i.eq.2) write(lout,20  'max
principal stress'
20      format(' ',a20,//)
          if(i.eq.3.or.i.eq.4) write lout,21  'v n mises
equivalent stress'
21      format(' ',a27,//)
          if(i.eq.5.or.i.eq.6) write(lout,22) 'max shear
stress'
22      format(' ',a16,//)
          if(i.eq.7.or.i.eq.8) write(lout,20)
'hydrostatic stress  '
          if(i.eq.1) write lout,40) 'epicardium '
          if(i.eq.3) write(lout,40) 'epicardium '
          if(i.eq.5) write lout,40) 'epicardium '
          if(i.eq.7) write(lout,40) 'epicardium '
          if(i.eq.2) write lout,40) 'endocardium'
          if(i.eq.4) write lout,40) 'end cardium'
          if(i.eq.6  write lout,40) 'end  ardium'
          if(i.eq.8  write lout,40) 'endocardium'
        write(lout, 19)

919
format(17x,'POSTERIOR',10x,'LEFT',20x,'ANTERIOR',19x,
     +'RIGHT',8x,'BASE'/)

40      format(' ',a11,///)
          write(lout,50) st, st, st, st, st, st
          do 1100 j=1,4
1100      write(lout,80)
          write(lout,60)(ctsts(i,j),j=1,6)
          do 1200 j=1,4
1200      write(lout,80)
          do 1300 j=1,4
            write(lout,70) (casts(i,6*(j-1)+k),k 1,6
70        format(f7.3,1x,'*********',1x,
     +       f7.3,1x,'*********',1x,
     +       f7.3,1x,'*********',1x,
     +       f7.3,1x,'*********',1x,
     +       f7.3,1x,'*********',1x,
     +       f7.3,1x,'*********',1x)

          do 1301 k=1,4
1301      write(lout,80)

          write(lout,60) (ctsts(i,6*j+l),l=1,6)
60
format(4x,'*',4x,f7.3,5x '*',4x,f7.3,5x,'*',4x,f7.3,
     +
5x,'*',4x,f7.3,5x,'*',4 ,f7.3,5x,'*',4x,f7.3
          do 1302 k=1,4
1302      write(lout,80)
80
format(4x,'*',16x,'*',16x,'*',16x,'*',16x,'*',16x,'*'
1300    continue
          write(l ut,50) st, st, st, st, st, st
50      format(5x,a12,6x,a12,5x,a12,5x,a12,5x,a12,5x,a1
1000  continue

      write(lout,929)
929   format(100x,'APEX')
      return
      end


      subroutine strain
      double pre-ision lam 3,2  ,lam8 3,8),xi,eta,-eta
      double precisi n
dp(3,15),va(15,15 ,vinva 15,15 ,lam15 3,15
      common lam-n lam,xi,eta,zeta,vinva,dp,va
      equivalence  lam 1,1 ,lam8 1,1 ,lam15 1,1
      integer lwork(6),mwork(6)
      integer vlwork(15),vmwork(15)
      integer
nrmbk,nrb,nelem,nonop,lo,co,mannd,nomat,nplc,ncmbk
      integer
ncb,ze,et,xii,prevty,p,p1,p2,counte,prevma,post
      integer nonp1(72,15),nonp2(72,20),rest(99,4),eno
      double precision coord(3,401),dis(1203),f 1203)
      double precision work(360)
      double precision mat 10,9)
      double precision tr(6,6),d2(6,6),m(6,6),a 3,3
      double precision xi2(6),eta2(6),zeta2(6)
      double precision
elco(20,3),elco8(8,3),k(60,60),jay(3,3)
      double precision
invjla(3,20),invjay(3,3),b(6,60),d(6,6),d1(6,6)
      double precision
eldis(60),stn(6),invjl8(3,8),b8(6,24)
      double precision
ap(3),sts(6),lc20b(3,20),lc08b(3,8),lc15w(3,15)
      double precision n(20),fn(3),pforc(36,5),kb(24,24)
      double precision detj
      double precision deter
      double precision
k15(45,45),elco15(15,3),invjl5(3,15),b15(6,45)
      double precision vp(15),n15(15)

common/femacm/mat,work,tr,d2,m,a,coord,xi2,eta2,zeta2,
     +
elco,k,jay,invjay,invjla,b,d,d1,dis,f,eldis,stn,sts,
     + ap,fn,n,vmwork,vlwork,lc20b,lc08b,lc15w,
     +
vp,detj,nrmbk,nrb,eno,nelem,nonop,lo,co,mannd,nomat,nplc,
     + ncmbk,icom,ncb,ne,loop,ioop,nonpl,nonp2,pforc,
     + ze,et,xii,prevty,p,p1,p2,counte,prevma,post,rest
      equivalence(k(1,1),kb(1,1),k15(1,1))
      equivalence(invjla(1,1),invjl8(1,1),invjl5(1,1))
      equivalence(n(1),n15(1))
      equivalence(elco(1,1),elco8(1,1),elco15(1,1))
      equivalence(b(1,1),b8(1,1),b15(1,1))
      equivalence(vlwork(1),lwork(1))
      equivalence(vmwork(1),mwork(1))
      double precision bstn(8,6),dhp,dhm
      double precision hh(14),xil(15),etal(15),zetal(15)
      integer id(8)
      data id/13,18,6,1,15,20,8,3/
      data hh/8*0.335180055d0,6*0.886426593d0/
      data xil/2*-0.758786911d0,2*0.758786911d0,2*-
0.758786911d0,
     +  2*0.758786911d0,0.d0,-
0.795822426d0,0.d0,0.d0,0.795822426d0,
     + 2*0.d0/
      data etal/4*0.758786911d0,4*-
0.758786911d0,0.795822426d0,0.d0,
     +  0.d0,0.d0,0.d0,-.795822426d0,0.d0/
      data zetal/-0.758786911d0,0.758786911d0,-
0.758786911d0,
     + 0.758786911d0,-0.758786911d0,0.758786911d0,
     + -0.758786911d0,0.758786911d0,2*0.d0,-
0.795822426d0,
     + 0.795822426d0,3*0.d0/
      data lin/1/,lout/3/,ldsp/2/

      dhp=1.
      dhm=-1.
      do 1010 i=1,8
      xil(i)=dhp
      etal(i)=dhp
1010  zetal(i)=dhp
      xil(3)=dhm
      xil(4)=dhm
      xil(7)=dhm
      xil(8)=dhm
      etal(1)=dhm
      etal(4)=dhm
      etal(5)=dhm
      etal(8)=dhm
      zetal(1)=dhm
      zetal(2)=dhm
      zetal(3)=dhm
      zetal(4)=dhm
      prevty=0

      do 1050 ne=1,nelem
6789    format(i4)
          if(nonpl(ne,14).ne.20) goto 1050
          write(lout,100)
100     format('element centroidal  strain components'/
     + 2x,'no.    no.   strain xx',5x,'strain
yy',5x,'strain zz'
     +,5x,'strain xy',5x,'strain yz',5x,'strain zx')
          do 1060 i=1,20
            ii=3*(i-1)
            jj=3*(nonp2(ne,i)-1)
            do 1060 j=1,3
              ii =ii+j
              jj -jj+j
1060        eldis(iij)=f(jjj)

      if(nonpl(ne,4).ge.2) call transf(1,0.d0
      prevty-nonpl(ne,3)

      do 1070 post=1,8
          xi=xil(post)
          eta-etal(post)
          zeta=zetal(post)
          if(nonpl(ne,4 .le.1) call transf(1,zeta
          call lam20b
        call mxm lam,3,elco,20,jay,3
          call fminv jay,invjay,3,deter,lwork,mw rk,b
          if(nonpl ne,4 .le.1  call transf 2,0.d0
        call mxm invjay,3,lam,3,invjla,20
          do 1080 i-1,6
            do 1080 j=1,60
1080        b(i,j =0.
          do 1090 i=1,20
            b(1,3*i-2)=invjla 1,i
            b(4,3*i-1)=invjla 1,i
1090      b 6,3*i)=invjla(1,i
          do 1100 i=1,20
            b(2,3*i-1 =invjla 2,i
            b 4,3*i-2)=invjla(2,i)
1100      b(5,3*i)=invjla 2,i
          do 1110 i=1,20
            b(3,3*i)=invjla(3,i
            b(5,3*i-1)=invjla 3,i)
1110      b(6,3*i-2)=invjla(3,i)
        call mxm(b,6,eldis,60,stn,1
          do 1120 i-1,6
1120      if(stn(i)+0.9.ge.0.) stn i -dl g 1.+stn ı
            do 1130 i=4,6
1130        stn(i)=0.5*stn(i)
          do 1140 i=1,6
1140      bstn(post,i)=stn i
1070  continue

      do 1150 post=1,8
          idn=id post)
          ide=nonpl(ne,1)
          write lout,110 ide,nonp2 ide,idn), bstn p st,ı
     + ,i=1,6)
110     format(1x,i3,4x,i3,1x,6e14.7)
1150  continue

111   format(6f10.3)
1050  continue

      write(lout,200)
200   format(1h ,'internal subroutine check F r straın')
      return
      end


      subroutine mxm(a,n1,b,n2,c,n3)
c
c     this subroutine takes two matrices a and b and
c     multiplies them together to form matrix c.
c
      integer n1,n2,n3,i,j,l
      double precision a(n1,n2),b n2,n3),c(n1,n3)

      do 10 l=1,n1
          do 10 j-1,n3
            c(l,j)=0.0
            do 10 ı 1,n2
              c(l,j  c(l,j)+a(l,i)*b i,j)
 10   continue

      return
      end
```