APPLICATION OF FINITE ELEMENT ANALYSIS

TO ROTATING FAN IMPELLERS

BY

MUHSIN JABER JWEEG

B.Sc., M.Sc.

A thesis submitted for the degree of

Doctor of Philosophy

of the

University of Aston in Birmingham

Department of Mechanical Engineering

July 1983

Supervisor:

T. H. Richards

THE UNIVERSITY OF ASTON IN BIRMINGHAM

APPLICATION OF FINITE ELEMENT ANALYSIS TO ROTATING FAN IMPELLERS

MUHSIN JABER JWEEG

A thesis submitted to
The University of Aston in Birmingahm
for the degree of
Doctor of Philosophy 1983

Summary

The aim of this work was to investigate the computer aided stress analysis of centrifugal fan impellers fabricated from thin plate. The finite element method has been chosen as a tool to solve this problem, since it has gained considerable attention, wide spread popularity, and proved to be a powerful tool for the designer engineer.

The semiloof shell element was adopted, in view of its demonstrated ability to deal with thin branching shells having sharp corners and multiple junction regions such as are found in fan structures.

The objectives of this research were achieved by establishing:

(i) A general purpose finite element package for plates and shells using the semiloof shell element for stresses and displacements.

(ii) A 'tailor made' finite element package for the impellers themselves where rotational stresses are significant for stresses and displacements. This package was developed further to cater for the sectorial symmetry of rotating fan impellers: by these means, data preparation, memory requirements and computing time are greatly reduced.

An experimental study was conducted, using the strain gauge technique, to evaluate the displacements and stresses predicted numerically. This was carried out for a simplified and an actual fan impeller as well as special simple test structures.

Numerical results for the thin plate and shell structures showed excellent agreement with those obtained by other methods. A satisfactory correlation was obtained between the experimental and the numerical predictions for the simplified and actual fan impellers.

A desk top computer HP9845B was used since it is a 'user friendly', low cost machine with interactive graphics and other powerful facilities.

Other topics which may be investigated as an extension for this work have been pointed out.

Key Words

ROTATING, FAN, IMPELLERS, C.A.D., F.E.M.

ACKNOWLEDGEMENTS

I am extremely grateful to my supervisor, Mr.T.H.Richards, Reader in Applied Mechanics, for advice, patient and personal guidance, inspiration and many productive discussions which covered all aspects of the work.

I would like to thank Professor K.Foster, Head of Mechanical Engineering Department, for the provision of equipment, technical staff and other extensive departmental facilities.

Thanks are also due to the Technicians in the Department of Mechanical Engineering who assisted in building the experimental rig.

Finally, I wish to express my thanks to Helen Turner for typing the thesis, whose perfectionism greatly enhanced the presentation.

LIST OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER SIX

CHAPTER ONE

INTRODUCTION

INTRODUCTION

Over recent decades, there have been considerable pressures on engineering designers to provide more accurate methods of stress analysis for engineering components subjected to specified loading. This is a significant consequence of attempting to satisfy the purchasing public's demands for better performance, economy, and a standard of safey in engineering products.

In the case of fan impellers, design stressing models have previously been extremely simple and inaccurate necessitating the use of large factors of safety which could imply excessive material and weight in the fan structure. Most methods were experimental; they were relatively time consuming and could be expensive and were reserved for background researchers. On the other hand, numerical modelling techniques have become available to designers; amongst those, the finite element method occupies an outstanding position because of its applicability to arbitrary shapes, different loading and support conditions and many aspects of practical design applications. The application of this method to the stress analysis of rotating fan impellers permits a much more sophisticated approach, but requiring the aid of a computer to provide the necessary facilities for the development of this efficient numerical method.

Since centrifugal fan impellers are fabricated from a thin sheetmetal into a backsheet, conesheet and the blades between them, the piecewise continuum model is an assemblage of plates and shells. A suitable element for a finite element discretisation is then the semiloof shell element, was designed recently by B. Irons, it was chosen for this work, because it is valuable for modelling plate and shell structures in which intersections and sharp corners exist, similar to those found in impeller structures.

The ultimate objective of this research was to develop a computer package, based on this semiloof shell element to analyse the displacements and stresses in fan impeller structure, due to rotational loads. This was achieved through the following steps:

(1) Develop a finite element computer program for general thin plate and shell structures.

(2) Test this program by correlating its predictions for selected problems with those obtained by other methods and reported in the literature.

(3) Extend the computer program of item (1) to accept a segmental solving routine in order to be able economically to tackle more complicated plate and shell structures in which the stiffness matrix cannot be held in the computer memory

for solving in one straightforward pass.

(4) Modify this program to suit fan impellers by taking a full advantage of their sectorial symmetry, and test this program against some examples of rotationally periodic structures.

(5) Build a rig for the strain gauge analysis of rotating fan impellers.

(6) Compare the experimental results using the strain gauge technique, with those predicted by the finite element.

In Chapter 2, a survey of the literature relevant to the theoretical and experimental stress analysis of the rotating fan impellers is given with an indication of their deficiencies in relation to the current problem. Chapter 3 is concerned with the chosen numerical modelling technique (finite element method) in the analysis of the rotating fan impellers. A major part of this chapter was given to a survey of shell elements, so that a suitable shell element could be chosen as a discretisation element.

Chapter 4 is a description of the semiloof shell element and the computer program built-up for general thin plate and shell structures: details of input data, generation of the system stiffness matrix, solution of

equilibrium equations and the output (displacements, strains and stresses) are given.

Chapter 5 describes a number of numerical examples used to assess the performance of the semiloof shell element in examining several plate bending and stretching as well as shell examples.

Chapter 6 deals with the modifications of the computer program, described in Chapter 4, to make it possible to analyse one sector of structures such as discs, cones and impellers in which the loading pattern, geometry and material properties; display secotrial symmetry.

In Chapter 7, a description of the test rig, strain gauge circuits and experimental procedure are given for a simplified and an actual fan impellers. The experimental results are compared with the numerical predictions for these impellers.

Chapter 8 presents a summary, conclusions and suggestions for further work.

CHAPTER TWO

REVIEW OF THE LITERATURE RELATED TO THE MECHANICAL DESIGN OF THE ROTATING FAN IMPELLERS

2.1 GENERAL INTRODUCTION

The determination of stress and deformation levels is an important element in the design of mechanical equipment. In the past, the theoretical methods used to estimate the stresses in rotating impellers proved to be unsatisfactory due to a general lack of knowledge regarding their effectiveness with respect to calculation accuracy: this problem will be discussed in the literature review later. An efficient tool is required which applies a careful and accurate stress analysis method in order to obtain the overall stress pattern that exists in thin walled rotating fan impellers under the action of rotational loads. On the one hand, experimental methods are available to provide analysis of stress in rotating impellers. Thus, strain gauge measurements are suitable for complicated structures, such as the rotating fan impellers of interest here, but this will consume time and can be expensive, especially in comparison with modern methods of analysis based on digital computers which have become more common: Most technical methods are implemented on relatively accessible computing systems.

In this chapter some of the previous theoretical and experimental investigations related to thin rotating impellers will be reviewed.

2.2 THIN WALLED CENTRIFUGAL IMPELLERS

It is probable that the majority of all fans are thin centrifugal or radial flow type. Such fans consist of an impeller running in a casing and having a spirally shaped contour. The air enters the impeller in an axial direction and is discharged at the periphery, the impeller rotation being towards the casing outlet. The amount of work done on the air, evident in the pressure developed by the fan, depends on the angle of fan blades with respect to the direction of rotation at the periphery of the impeller. The impeller is the most highly stressed part of the fan. The stresses in the impeller are caused by rotation, temperature and aerodynamic forces, but the most important are the stresses caused by rotation. The impeller consists of a back sheet and a front cone-sheet with radially spaced blades continuously welded between them.

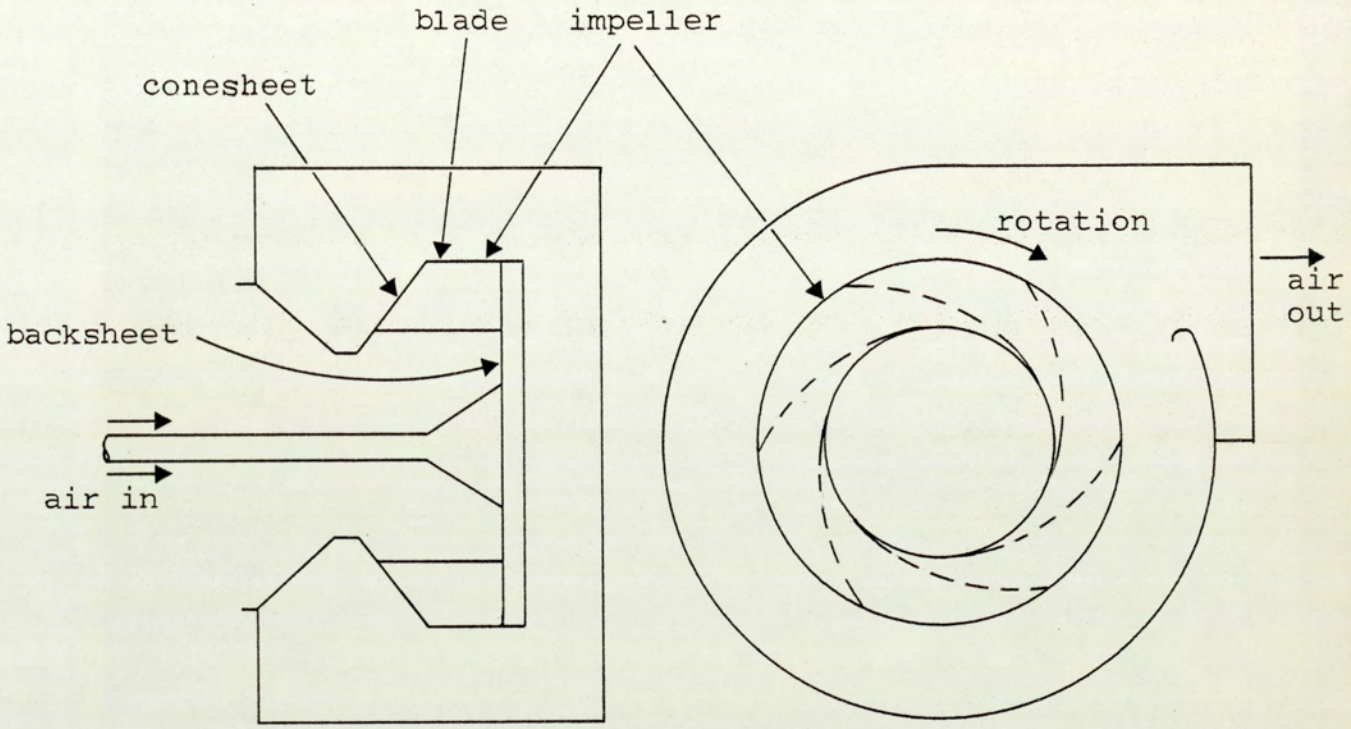
There are three main types of impellers, Fig. 2.1, depending on the blade type. These are:

(a) the backward bladed impeller, in which the blade tips incline away from the direction of rotation and the angle of the blade $\beta<90^{o}$.

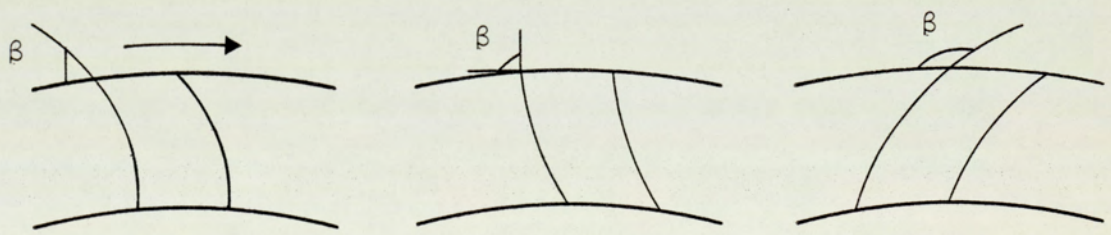
(b) the radial bladed impeller, in which the blade tips are radial, therefore, $\beta=90^{o}$.

(c) the forward bladed impeller, where the blade tips incline towards the direction of rotation and $\beta>90^{o}$.

Centrifugal fan impellers have been used in a wide range of industrial applications, such as induced draught, equipment for industrial boilers and compact units for heating and ventilating applications.



Centrifugal fan notation



Blade forms

Fig.2.1 Centrifugal fan impeller

2.3 REVIEW OF LITERATURE RELATING TO THE STRESS ANALYSIS OF FAN IMPELLERS

The relevant literature revealed few references dealing with the stress analysis of rotating impellers. The previous experimental and theoretical work related to the rotating impellers will be reviewed. In addition, some of the work carried out by previous investigators on the stress analysis of rotating discs, rotating cones and the stresses produced by rotation in the blade of centrifugal fan impellers are mentioned briefly.

2.3.1 Rotating Discs

The backsheet, which is the main part of the rotating fan impellers has previously been considered as a disc. The stress distribution in simple rotating discs with constant thickness is given by Timoshenko Ref. (1). Haerle Ref. (2) has developed a theoretical method to obtain the stress distributions in rotating discs with a hyperbolic profile. His method is based on the general formulae for the tangential and radial stresses in a rotating disc as given in Ref. (3) . These are:

$$\sigma_t = \frac{E}{1-\nu^2}\left[(1+3\nu)Kx^2+(1+\nu)b_1+(1-\nu)b_2x^{-2}\right] \qquad .....(2.1)$$

$$\sigma_r = \frac{E}{1-\nu^2}\left[(3+\nu)kx^2+(1+\nu)b_1-(1-\nu)b_2x^{-2}\right] \qquad .....(2.2)$$

$$k = \frac{-(1-\nu^2)\rho\omega^2}{8gE}$$

where σ_r, σ_t = radial and tangential stresses at any radius x respectively.

ν = Possions ratio

E = Youngs modulus of elasticity

b_1, b_2 = constants depending upon the condition (stress) prevailing at the bore and periphery of the disc

Let

$u = \omega x$ = rotational velocity of a point of a disc at radius x

Also,

$\sigma_t + \sigma_r = S$ = sum of principal stresses

$\sigma_t - \sigma_r = D$ = difference of principal stresses

Then it can be shown that

$$S = (1+\nu)\frac{\rho}{2g}\left[-u^2+k_1\right] \qquad .....(2.3)$$

$$D = (1-\nu)\frac{\rho}{4g}\left[u^2+k_2u^{-2}\right] \qquad .....(2.4)$$

where $k_1 = \frac{4gEb_1}{(1-\nu^2)\rho}$, $k_2 = \frac{8gE\omega^2 b_2}{(1-\nu^2)\rho}$

If S and D are known for any point on the disc, the stresses at that point are given by:

$$\sigma_t = \frac{S+D}{2} \qquad .....(2.5)$$

$$\sigma_r = \frac{S-D}{2} \qquad .....(2.6)$$

Equations (2.3) and (2.4) are plotted in Ref. (2) for varying speeds of rotation. In these two equations, the only variables are the constants k_1 and k_2. Therefore, Haerle Curves, as represented by equations (2.3) and (2.4), are based on different values of k_1 and k_2 respectively. For a disc of constant thickness, the stresses can be found by selecting the appropriate sum and difference curves, which conform to the applied boundary conditions.

S and D charts can be applied to the discs or hubs of a hyperbolic profile by replacing the actual hyperbolic profile by stepped discs consisting of a number of concentric rings, each with a constant thickness (as shown in Fig. 2.2). The assumption is made that, providing the steps are comparatively small the stresses in adjacent concentric rings, on either side of the step, are inversely proportional to the thickness of the ring.

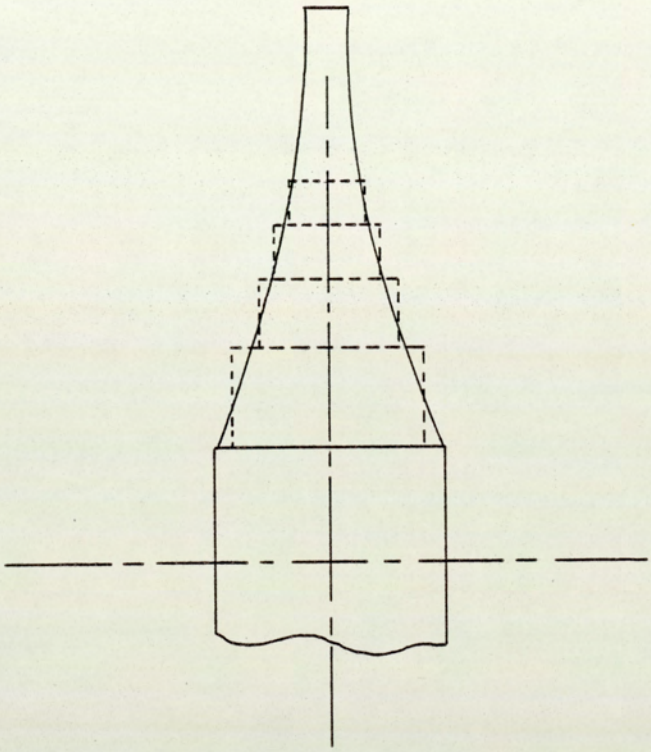


Fig.2.2 Simplified disc : Haerle

The sum and difference chart is applied to each ring by assuming an initial tangential stress at the rim. The analysis is then carried to the next ring. The procedure is repeated with a new assumed tangential stress at the rim until the stresses obtained at the bore of the disc match the desired boundary conditions.

The main disadvantage of the Haerle method is that the stresses are obtained by making initial assumptions and by working up to the final results by trial and error.

2.3.2 Rotating Cones

Another main component of the rotating fan impeller has been treated essentially as a rotating cone. The most suitable solution to predict the stress distributions in the rotating cones has been given by Meriam, Ref.(4) . Meriam has extended the theory of axially symmetric shells by including the body forces of rotation about the axis and applying the results to the rotating conical shells. The Meriam analysis is based on the linear theory of elasticity. The equations of equilibrium and compatibility are manipulated into two linear fourth-order differential equations by using the Meissner operator. A method of analysing the fan impeller, which has been used by Bell, Ref.(5) , is to treat the backsheet, cone-sheet and blades separately and examine the interaction

between them. Bell has used Meriam solution to predict the stresses in the conesheet. Another suitable approach for the rotating cone is given by Timoshenko and Woinosky-Krieger, Ref. (1).

Few references dealing with conical shells are availalbe in Bell, Ref.(5) , who concluded that they were too general for the conesheet, such as Refs.(6, 7).

2.3.3 Blade Analysis

Few references are available to estimate the blade stresses. Among these are:

Osborne, Ref.(8) has considered the blade as a beam built in at both ends. Osborne estimated the maximum bending stresses produced in a blade on the basis of the centrifugal force acting at the centroid of the blade.

Patton, Ref. (9) has attempted to use plate theory Ref. (1) to predict the stresses in the plate panels. However, this proved to be unsuccessful due to the complexity of the geometries and the inability of the plate theory to treat the interaction of backsheet and conesheet with blades.

2.3.4 Stress Analysis of Rotating Impellers

As mentioned before, the number of references dealing with the rotating impellers is limited. The important literature in this respect includes:

(a) Glessner, Ref. (10) has presented a method of analysing the stresses in a rotating impeller. His method is characterised by obtaining an elastic balance between the blades, disc and cover, when these parts are subjected to inertia forces arising from rotation. Fig. 2.3 shows a front view and section of the Glessner impeller. He approached the problem by analysing a simplified structure, the disc, blades and the cover separately. The disc carries its own centrifugal load, and some portion of the blade and cover loads. The blades are approximated by built in beams of span equal to the blade width. They are subjected to the centrifugal force due to their own mass. The cover is considered as a rotating disc with a conical profile and supports itself.

The Glessner Method depends upon:

(i) the calculation of the blade load caused by centripetal acceleration.

(ii) the calculation of the cover and disc loads by use of the Haerle sum and difference curves (discussed in Section 2.3.1).

Once the centrifugal loads for the three components have been calculated, the distribution of the blade load on the cover and disc is found by balancing forces and deformations. This can be done by assuming a spring constant for the three components which are to be calculated from their deflections under no blade load and full blade load conditions. By using these constants the proportion of the blade load on the disc and cover can be calculated. The stresses in the disc and cover can then be calculated as the sum of stresses due to the body forces and the blade bending forces.

(b) Patton, Ref. (9) has described an experimental programme of stress analysis of rotating impellers using brittle lacquer and strain gauge techniques for three impellers of different sizes running at different speeds. Patton has compared his results with the method suggested by Hearle, Ref. (2), which involves the treatment of the backsheet and the conesheet separately, allowing for the blade loading to add radial stresses at various radii. It was found that there was poor correlation between the experimental and analytical results, since the analytical method used proves to be unsuitable in that no account is taken of any symmetry or conical nature of the disc profile or of the discontinuous nature of the blade loading. These effects create bending and local stresses, which increase the safety factors required in fan design to cover these

effects, especially in the design of high pressure fans. Patton observed this poor correlation at critical points of high stress on the conesheet and backsheet.

(c) Thurgood, Ref. (11) has investigated the surface stresses in rotating discs of an asymmetric profile. The results are compared with the measurements made on photoelastic models by the 'frozen-stress' technique. His analysis is based on the 'step-by-step' method of Ref. (12). but is extended to include the effects of blade loading, asymmetry, and blade stiffening when necessary. Thurgood assumes that the stresses vary linearly across the thickness of the disc, the disc only carrying loads and moments in the tangential direction, but both discs and blades carrying loads and moments radially, as shown in Fig. 2.4. Reasonably good correlation has been obtained between the predicted stress distributions and those obtained from the photoelastic models, except where few blades give rise to interblade bending and occasionally in the bore region.

(d) Bell and Benham, Ref. (13) have discussed the theoretical and experimental stress analysis of centrifugal fan impellers. Brittle lacquer and strain gauge methods used in the experimental programme to determine stress distributions in a simplified model and an actual centrifugal fan impeller. Bell and Benham have applied the finite element method using a flat triangular

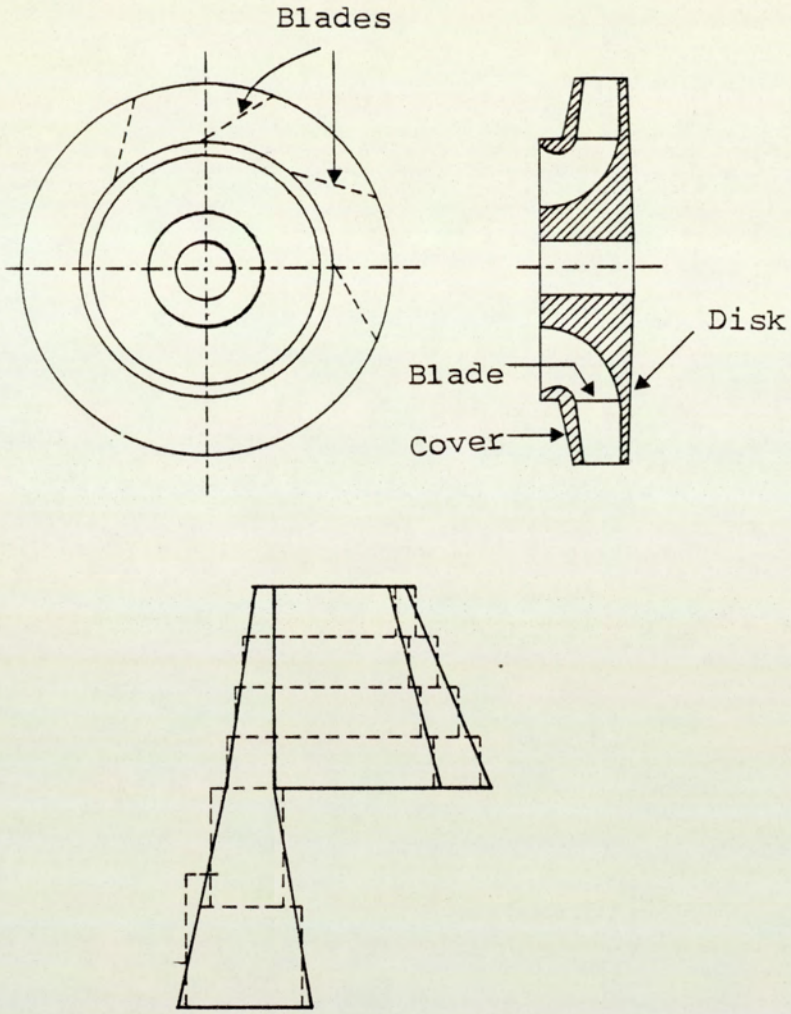


Fig.2.3 Glessner impeller

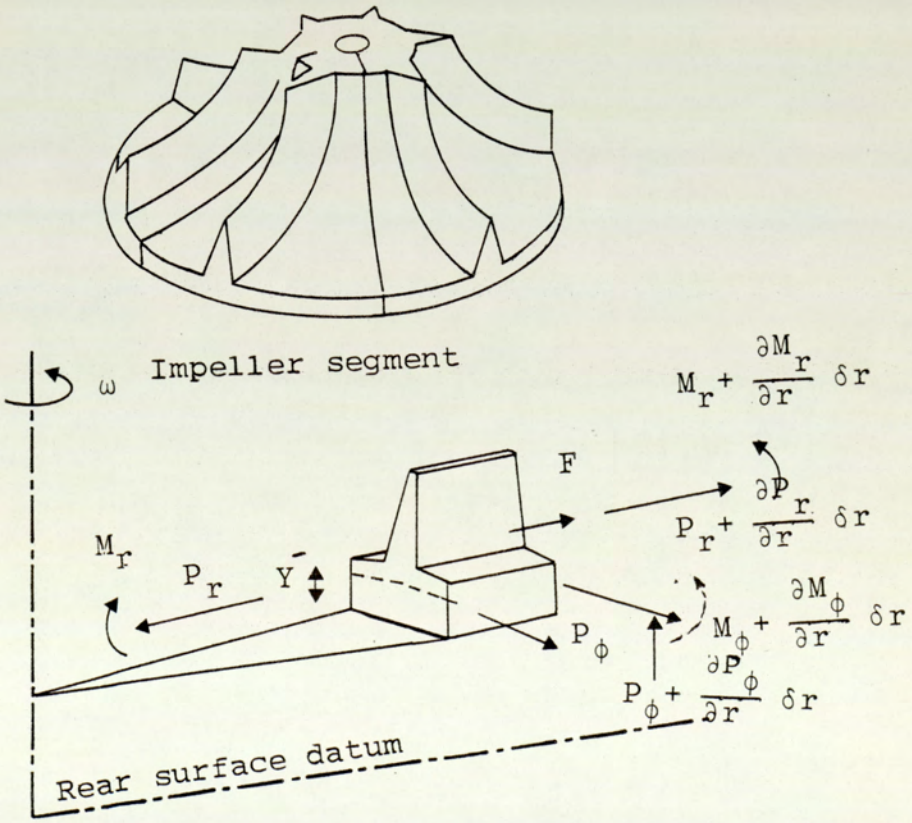


Fig.2.4 Dimensions and loading of a basic element of an impeller

element in which the bending and membrane behaviour are separately represented. (This type of element will be discussed in Chapter Three).

The centrifugal forces are calculated at the centre of each element and one third of the force is applied at each node of the element. Good agreement has not been obtained between the experimental and the finite element method for the bending moments in the region of fixed edges.

Bell's work Ref. (5) on the rotating fan impellers is the first which attempts to apply a modern general method (finite elements) for obtaining the stress levels in these structures. He has used a simple element for discretisation , which needs a very fine mesh in order to converge to the exact solution. Bell assumed that the fan structure is quarterly symmetric. The research project described in the present thesis is aimed to develop a method which overcomes the disadvantages mentioned above. The method is structured to suit the requirements of industry and widely available but powerful user friendly desk-top computers.

2.4 CLOSING REMARKS

The literature review, discussed in the previous sections, related to analytical and experimental methods of stress analysis for thin rotating impellers. It has shown relatively little work for dealing with such complex structures. Consequently, a general lack of knowledge regarding detailed investigations of the intersection areas (blade-backsheet and blade-conesheet) can be seen. It has been pointed out that the most of the methods used were experimental. The analysis provided in the literature deals with the classical approach, except the work presented in Ref. (5).

The new analysis reported here deals with the stress analysis of thin rotating fan impellers by the finite element method. A suitable shell element is chosen for discretisating the impeller structure. The element should provide reasonable modelling of the interactions between the blades and backsheet and conesheet.

In the next chapter, the finite element method is briefly explained, and afterwards the existing shell elements are presented. Suitable shell element can then be selected for the discretisation of thin rotating fan impellers.

CHAPTER THREE

MODELLING FOR STRESS ANALYSIS IN DESIGN: APPLICATION TO FAN IMPELLERS

3.1 INTRODUCTION

In the stress analysis of complex engineering structures, numerical modelling techniques are now widely used (see Refs (14,15)). The quality of the models depend substantially on the experience and judgement of the analyst (defining the detailed geometry and loading and interpreting the results). At present numerical modelling techniques are well-known and the following factors have contributed to their development.

(i) the rapid decline in digital computing costs.

(ii) the firm establishment of rational numerical modelling techniques (particularly the finite element method of linear elastic and non-linear elastic problems).

The traditional approach to structural design modelling was simple and this was successful when the loading regimes and structural forces were essentially simple. Analogue models, particularly photoelastic ones, could be used to get detailed localised stress data. Such experimental techniques yield accurate data if there is an adequate investment of effort, but the cost is high. These experiments were usually reserved for background research and development studies. Widely available cheap computing power has given the stress analysis engineer a new capability of better modelling of complex

engineering structures. However, the approach to the problem is still based on understanding and modelling the problem. Furthermore, the familiarity of the analyst engineer with the limitations of the description of the model under investigation, as well as his ability to sense or judge the solutions play a major role in obtaining a reasonable solution to the problem in hand.

In this chapter, the appproximate techniques to form a stress analysis model will be discussed. Depending on the requirements of the problem concerned, the approximate method is chosen, which in turn requires experience in order to make adequate judgement. The finite element method will be explained, including its mathematical basis, accuracy and advantages. This is the technique used in developing a stress analysis computer package for general thin plate and shell structures which is then tailored for application to fan impellers (to be presented later).

As mentioned in Chapter 2, fan impellers are fabricated from a thin backsheet, a thin conesheet and the blades between them. Thus, it is necessary to have a general knowledge of the shell elements, so that a suitable element for the mesh discretisation can be chosen. The blades may be treated as a special case of the shell, perhaps with zero curvature. The shell elements will be presented in Section 3.4.

3.2 METHODS FOR SOLVING ELASTICITY PROBLEMS

3.2.1 Exact Method

The exact solution for any elasticity problem defined by the differential equations, for the equilibrium and the compatibility requirements, needs a great effort due to the difficulty of satisfying the boundary conditions. An approach to this is to get the stresses and strains using trial and error (for example Ref. (2)). This implies that if the trial functions satisfy the equilibrium, the compatibility and the boundary conditions, then they are the correct solution to the problem, otherwise, another trial function should be used until the boundary conditions are satisfied. For some types of problems, the classical analytical method is either impractical, or virtually impossible. As an example, Ref. (16) , in the plain plate problem the general governing equation for the bending of thin plates is a fourth order partial differential one; for circular plate defined in terms of cylindrical co-ordinates r, θ and z in which the governing equation is of the form:

$$\nabla^4 w = \frac{P}{D}$$

where ∇^2 is the operator $\left(\frac{\partial^2}{\partial r^2} + \frac{1}{r}\frac{\partial}{\partial r} + \frac{1}{r^2}\frac{\partial^2}{\partial \theta^2}\right)$

P is the intensity of loading on the plate

w is the transverse deflection of the plate

D is the flexural rigidity of the plate.

A direct analytical solution of this equation is rarely possible; except in the case of axisymmetric loading and it implies that the plate is homogenous and of constant thickness. In this case the equation is reduced to a relatively simple ordinary differential equation where the loading and deflection are functions of the radius only. In almost all other cases, approximate methods of solutions are necessary. The basic outlines of the approximate methods will be discussed in the following section.

3.2.2 Approximation Techniques

As mentioned before, the approximate techniques are necessary to deal with stress analysis problems when considering complex structures, in which a complete theoretical solution may be impractical with respect to time, cost and degree of difficulty. As pointed out, the component under investigation here is a complex structure; therefore an approximation technique needs to be used in order to obtain the stress levels in these types of structures. Methods of approximations may be classified into either,

(i) approximate treatment of an exact differential equations, such as finite difference method.

or,

(ii) exact treatment of an approximate system, such as Rayleigh-Ritz and finite element methods (Energy methods).

3.2.2.1 The Finite Difference Technique

In this technique, the continuous functions, such as the displacements or stresses, are represented by values at a finite number of points within the solution domain. This, then reduces the problem to a set of simultaneous algebraic equations. The accuracy of the method depends upon the size of the intervals, or the mesh size. The finer mesh increases the accuracy. On the other hand, the resulting equations are increased and hence the amount of work required for a solution increases with higher round-off errors, in addition to large computer storage requirements and running times. An important advantage of the finite difference methods is the ease with which they may be formulated and programmed for a solution by a digital computer. In case of irregular boundary slopes and in concentrating points in regions of the solution domain, the lack of the geometric flexibility in fitting makes the finite

difference method to be unsatisfactory for solving these cases. This is considered as a main disadvantage of this method. However, the technique becomes very hard to use with a component containing assemblages of plates or shells, such as those found in fan impellers due to difficulties of writing the mathematical description of the structure geometry. Another approach should be used when dealing with these types of structures.

3.2.2.2 Energy Methods (Variational Methods)

In these methods, a functional which describes a physical property of the system, such as the potential energy, is extremised with respect to independent parameters (displacements and/or stresses), thus resulting in the governing differential equations of the problem. These methods can also be used as a means for solving problems by an approximate technique. The principle of minimum potential energy, for instance, has been widely used in connection with Rayleigh-Ritz and finite element techniques to solve complex problems in the context of solid mechanics. In this principle, an approximate displacement field which satisfies the compatibility condition, is used to obtain an expression for the total potential energy of the system. By minimising this expression, one obtains the approximate

equilibrium equations of the system.

In the Rayleigh-Ritz technique (see Ref. (17), the solution is assumed in the form of a series of function which satisfies the boundary conditions, but with undetermined parameters. These functions are inserted into the expression of the potential energy and the required integration is carried out. The minimisation of the potential energy results in equation from which the undetermined parameters can be found. The accuracy of the solution improves if more undetermined parameters are taken. Even in the days of the finite element method, Shebak Ref. (18) has recently used this method successfully to predict the mechanical vibration characteristics of Bell-type pump mountings..

The finite element method may be viewed as piecewise a Ritz method. According to this method the continuum is imagined as divided into a finite number of elements, connected with each other through nodal points. The displacement function within each element is assumed in a suitable form, then the strain energy is computed. Summation over all the elements yields the total energy of the whole structure. As in the Ritz method, minimisation of the total energy generates the static equilibrium equation in the form:

$$[K]\{q\} = \{Q\} \qquad \ldots\ldots(3.1)$$

The detailed contents of $[K]$ depend upon the technique used and displacement model assumed. A discussion of the finite element method, adequate for the purposes of this report, will be given in Section 3.3.

3.3 THE FINITE ELEMENT METHOD IN STRESS ANALYSIS

3.3.1 Introduction

As mentioned in the previous section, the finite element method is an approximation technique in which a continuum, with infinite degrees of freedom, is approximated by an assemblage of elements, each with a finite number of degrees of freedom. The element behaviour is described by a set of assumed functions representing the stresses or displacements. The assumed functions are usually of a polynomial form which is chosen because of the ease with which the mathematics can be manipulated when formulating the stress or strain equations. By using a sufficient number of elements, an acceptable representation of the overall real situation of the structure under consideration is obtained. The advances made in the design of digital computers in the 1950s have given a new impetus to the development of this technique, as it enables structural engineers using the technique of matrix algebra, to deal with problems which would otherwise be too large to tackle. This represents the beginning of the finite element method as a significant tool for engineering analysis. Initially, structural mechanics reasoning was used to extend the matrix methods to continuum problems. It was not until the early 1960s that stiffness finite

element analysis was formulated in terms of the principle of stationary total potential energy. Today the finite element method has already become a very powerful tool for analysing complex problems in structural and continuum mechanics. A vast number of papers and books have been written on this subject (see Refs.(19,20)). Now this method is firmly established in civil engineering and widely used by mechanical and aeronautical engineers for the analysis of stress in solid components. In general, the finite element method has the following advantages:

(i) the ability to model structures of arbitrary geometry.

(ii) the ability to deal with structures with arbitrary loading, including thermal loading and arbitrary support conditions.

(iii) the ability to model composite structures involving different structural components, such as combination of plates, bars and solids.

(iv) the ease with which the actual structure may be visualised because the finite element structure closely resembles the actual structure.

Inspite of all the advantages, there are disadvantages which are not unique to the finite element method. These include:

(i) a specific numerical result is obtained for a specific problem.

(ii) to construct a good and efficient finite element model, experience and judgement are needed.

(iii) a large computer is essential for large structures.

(iv) input and output may be large and tedious.

In the following section a brief outline of the process will be presented.

3.2.2 Outline of the Finite Elements Process

When applying the finite element method, two phases are to be considered. These phases consist of studying the individual elements and assemblages of the elements. In the following section the various steps involved when the finite element method is applied to stress analysis problems. The method can be conveniently summarised into five essential steps; for the so-called displacement or stiffness formulation, these are as follows (see Ref. (19)).

(i) Definition of the Finite Element Mesh

At this step, the continuum is subdivided into subregions or finite elements. Each element is connected

to the next through node points at its boundary; the nodes are numbered and referenced to a co-ordinate origin. Depending on the nature of the problem, such elements can be one, two or three dimensional.

Bar type elements (Fig. 3.1) are appropriate when the geometry and dependent variables, such as, displacement can all be expressed in terms of one independent co-ordinate. This co-ordinate is measured along the axis of the element. In case of two-dimensional problems (Fig. 3.2) the element usually takes the form of a triangle or quadrilateral. The general three-dimensional solid (Fig. 3.3) may be divided-up into tetrahedron or more general shapes, while if the solid enjoys axial symmetry in its geometry, the ring type elements of Fig. 3.4 may be used. As a general guide, the element density is increased in areas of rapid stress changes and, conversely, reduced in areas of nearly uniform stress.

(ii) Selection of the Displacement Model

In the finite element method the assumed element behaviour is governed by a displacement function. These functions are approximate and chosen to define uniquely the displacement field within the element in terms of its nodal displacements. The simple functions representing the approximate displacements are variously

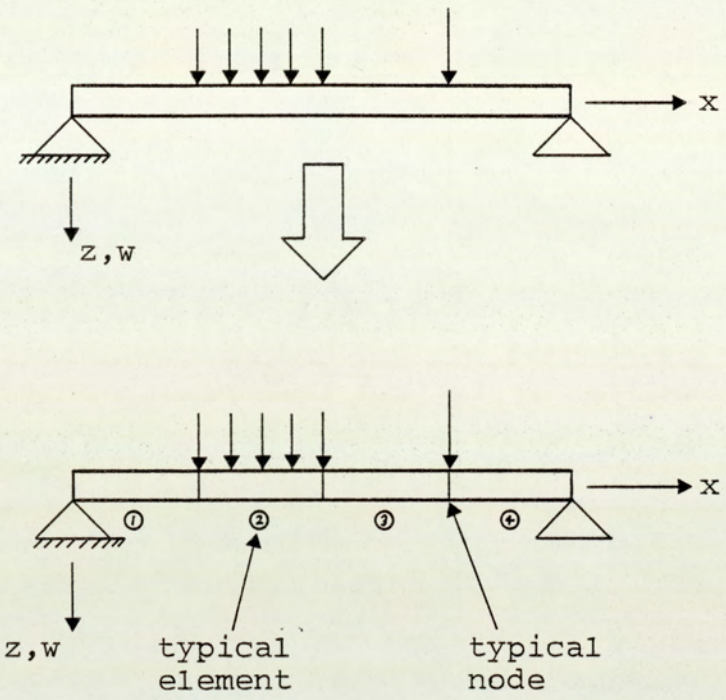


Fig.3.1 One dimensional structure idealisation

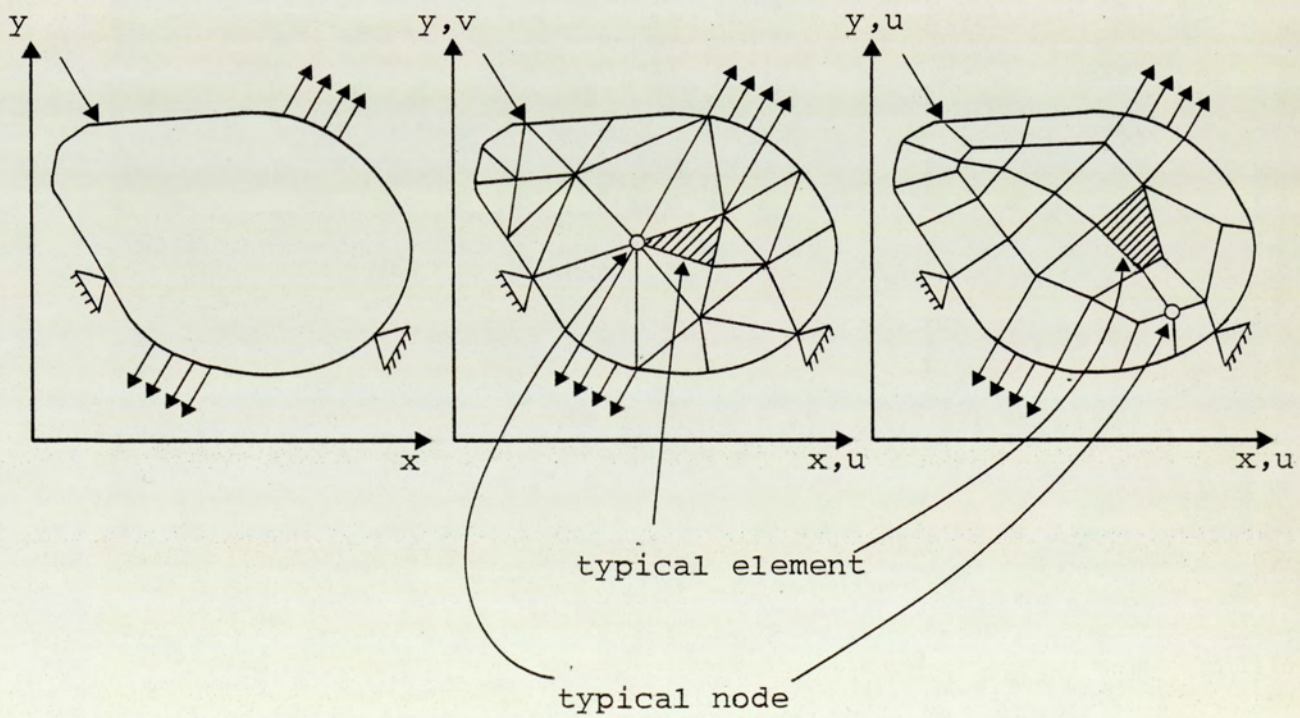


Fig.3.2 Idealisation of two-dimensional structure using triangular or quadrilateral elements

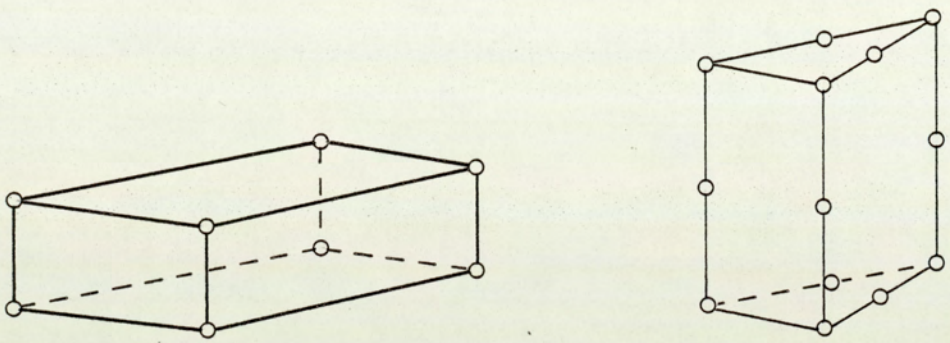


Fig.3.3 Three dimensional finite elements

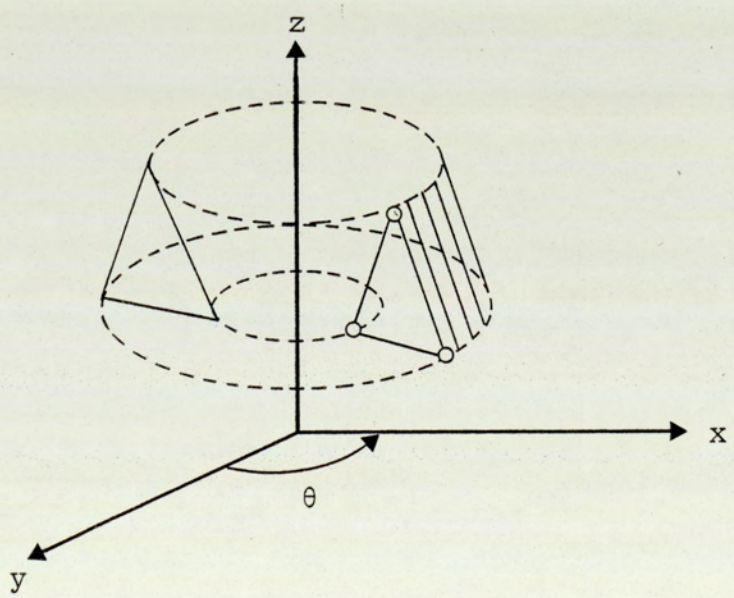


Fig.3.4 An axisymmetric finite element

called displacement models, functions, fields, patterns, or shape functions. The displacement functions are commonly assumed in a polynomial form. The degree of the polynomial governs the ability of the element to approximate the true displacement field. The simplest form for the displacement of an element is a linear function, and higher order elements may be used for improved accuracy in any particular problem (see Ref.(20)). Usually the same element formulation is used throughout the discretisation process, but it is possible to mix element types in order to gain a more efficient approximation to the real structure. The choice of a suitable displacement function, which limits the infinite degree of freedom of the system can be very difficult. As a rule, the solution converges to the true one for the structure as the number of elements is successfully increased but a true minimisation of the potential energy never be reached, irrespective of the fineness of the subdivision. To ensure this convergence, the displacement function must satisfy the following convergence requirements (See Ref. (19)):

(i) The displacement function must be continuous within the element, and between the adjacent elements, i.e. no opening or overlaps must be implied.

(ii) The displacement function must include the rigid body displacement of the element. This condition implies that all the points in the element should experience the same displacement.

(iii) The displacement function should allow for all the states of uniform strain within the element. This is a necessary condition, because if the body is imagined to be divided into smaller and smaller elements, the strain state of each of the infinitesimal elements approaches a constant value.

The starting point for the process is to assume a displacement pattern for an arbitrary element of the solid:

$$\{q\}^e = [N] \{q\}_o^e \qquad \text{.....(3.2)}$$

where $\{q\}^e$ is the displacement field within the elements whereas $\{q\}_o^e$ is a vector of element nodal displacements; and $[N]$ is the shape function.

(iii) Formulating the Discrete Stiffness Equations

The strains at any point within the element may be expressed in terms of the element nodal displacements. This can be done by a suitable differentiation of the displacements defined by equation (3.2), so that:

$$\{\varepsilon\}^e = [\partial]\ [N]\ \{q\}_o^e$$

or

$$\{\varepsilon\}^e = [B]\ \{q\}_o^e \qquad .....(3.3)$$

where $\{\varepsilon\}^e$ is the element strain vector;

$[\partial]$ is a matrix of differential operators; and

$[B]$ contains the appropriate derivatives of $[N]$.

The strain energy stored in a typical element is (see Ref.(19)):

$$U^e = \frac{1}{2} \int_{vol} \{\varepsilon\}^{et} [D]\{\varepsilon\}^e \, dvol \qquad .....(3.4)$$

substituting from equation (3.2),

$$U^e = \frac{1}{2}\{q\}_o^{et} \left(\int_e [B]^t [D] [B] dvol\right)\{q\}_o^e$$

or

$$U^e = \frac{1}{2}\{q\}_o^{et} [k]^e \{q\}_o^e \qquad .....(3.5)$$

The matrix

$$[k]^e = \int_e [B]^t [D] [B] \, dvol \qquad .....(3.6)$$

is the element stiffness matrix.

Now, the total strain energy stored in the system is:

$$U = \sum_{e=1}^{n} U^e \qquad \text{.....(3.7)}$$

where n is the total number of elements in the assemblage.

If all the element nodal displacements are denoted by:

$$\{\tilde{q}\} = \{q\}_o^e = \begin{Bmatrix} \{q\}_o^1 \\ \{q\}_o^2 \\ \{q\}_o^3 \end{Bmatrix} \qquad \text{.....(3.8)}$$

and the element stiffness matrices are displayed as:

$$[\tilde{k}] = \begin{bmatrix} [k]^1 & & \\ & [k]^2 & \\ & & [k]^n \end{bmatrix} \qquad \text{.....(3.9)}$$

The equation (3.5) can be written as:

$$U = \frac{1}{2} \{\tilde{q}\}^t [\tilde{k}] \{\tilde{q}\} \qquad \text{.....(3.10)}$$

Because the displacements are matched at the nodes; a compatible or connection matrix $[C]$ is formed which expresses the necessary compatibility between the locally and globally measured displacements, therefore:

$$\{\tilde{q}\} = [C]\{q\} \qquad \text{.....(3.11)}$$

where $\{q\}$ is the vector of system generalised co-ordinates.

Equation (3.10) can be expressed in terms of $\{q\}$. That is,

$$U = \frac{1}{2}\{q\}^t[C]^t[\tilde{k}][C]\{q\}$$

or

$$U = \frac{1}{2}\{q\}^t[K]\{q\} \qquad \text{.....(3.12)}$$

where $[K]$ is the assemblage or system stiffness matrix.

In order to establish the stiffness equilibrium equations, the total potential of the applied loads must be determined. If the external applied loads are denoted by $(Q^1, Q^2, \ldots, Q^N)$ corresponding to $(q^1, q^2, \ldots, q^N)$, then

$$\Omega = -\{q\}^t\{Q\} \qquad \text{.....(3.13)}$$

where N is the total number of nodal degress of freedom that is, the total number of equations to be solved for the assemblage. The total potential energy in the discrete system is then.

$$V = \frac{1}{2}\{q\}^t[K]\{q\} - \{q\}^t\{Q\} \qquad \text{.....(3.14)}$$

Now, the principle of minimum potential energy to the assemblage is applied, i.e. for equilibrium $\delta V=0$, so that:

$$\{\delta q\}^t([K]\{q\} - \{Q\}) = 0$$

But δq are arbitrary so that:

$$[K]\{q\} = \{Q\} \qquad .....(3.15)$$

These are the required equilibrium equations for the 'assembled' approximate system.

(iv) Solution of the Stiffness Equations

The solution of matrix equation (3.15) is a standard procedure in matrix algebra. A number of equation solving routines, usually based on the Gaussian elimination or Cholesky decomposition processes, are available. In order to obtain an acceptable representation of the continuum system, the number of unknowns resulting from the discretisation needs to be very large.

As a consequence, the solution cannot be contemplated without the aid of a high speed digital computer. The efficient solving routines take into account the symmetric, banded nature of the stiffness matrix in order to reduce the storage requirements demanded of the computer.

(v) Determining Element Strains and Stresses

Once the nodal displacements have been determined the element strains can be calculated from the displacement model using the strain-displacement relations. The stresses are then obtained by means of Hooke's law.

If the strains are calculated from equation (3.3) and the material obeys Hooke's law, the stress vector $\{\sigma\}$ may then be written as follows:

$$\{\sigma\} = [D]\{\varepsilon\} \qquad \text{.....(3.16)}$$

Since the equilibrium conditions are only satisfied in some average overall manner and not point by point: at particular stations within an element and on inter-element boundaries. Consequently, the accuracy of the stresses at a particular node may be improved by taking the average of the individual element stresses at that node.

3.3.3 Convergence and the Patch Test

As mentioned in Section 3.2.2, for convergence to exact solution, the assumed displacements for an element should be continuous within the element and across the element boundaries. In addition, the rigid body motion needs to be included and the displacement field must be able to accommodate the state of constant strain. The

reason for this is that any real problems, when idealised by a very fine mesh of elements over any assemblage of few neighbouring elements, experiences sensibly uniform stress in the correct solution. However, if the necessary condition of inter-element continuity is not met fully, these elements may not necessarily reproduce uniform strain conditions, even though the individual elements can give constant strains. If these classical convergence requirements are fully met, the element is known as a conforming one. If displacement discontinuity exists along the element boundaries, the element is known as a non-conforming element. An example is the non-conforming bending triangle Ref. (21) in which an arbitrary assemblage of elements fail to reproduce constant strain conditions while the individual element can satisfy the constant strain criterion. In practice, some types of non-conforming finite elements have been found acceptable in engineering applications. Among these are:

(i) Elements that do not meet the classical requirements, i.e. non-conforming elements.

(ii) Elements that contain signularities.

(iii) Elements in which the energy is approximately integrated.

(iv) Elements with no clear physical basis.

Irons Ref. (22) has recently established a test to guarantee the convergence for any type of element. This is called Patch Test. A necessary and sufficient condition for convergence is that the element is required to pass the patch test. This test involves prescribing displacements to the external nodes of a small patch of elements which corresponds to a known, but arbitrary state of constant strains. If the free nodes voluntarily take-up the assumed displacements giving constant strains at every point within the patch, the element converges in the limit.

In conclusion, if a non-conforming element passes the patch test, it can be used with confidence and convergence can be achieved. A comprehensive study of this test is given in Refs. (22, 23, 24) .

3.4 THIN SHELL ELEMENTS

3.4.1 Introduction

Thin shells provide some of the more difficult problems that have been attempted by finite element analysis. A number of shell elements have been found in the literature, as will be illustrated in this section. In the selection of an element to solve shell problems, it is necessary to consider the geometry and symmetry of the structures, the type of behaviour to be studied and the computing facilities which are available. There are many different elements dependent on shell theory, which can lead to very complex formulations. Each element has its own advantages and disadvantages. The following section deals with the literature on thin shell elements, later the selection of one of these elements, suitable for discretisation of thin plate and shell structures having intersections similar to those junctions found between blades, backsheet and conesheet of thin rotating impellers will be described. There are two main approaches to thin shell elements:

(a) Flat shell elements.

(b) Curved shell elements.

3.4.2 Flat Shell Elements

This type of formulation has been used by various researchers to model shells due to its simplicity. In this approach Ref. (20) the curved surface is approximated by an assemblage of flat elements, such as plane stress, to represent the membrane stiffness of the shell and plate bending elements to represent flexure.

Fig. 3.5(c) shows a triangular element formulated by superposition of stretching behaviour (Fig. 3.5(a)) and bending behaviour (Fig. 3.5(b)).

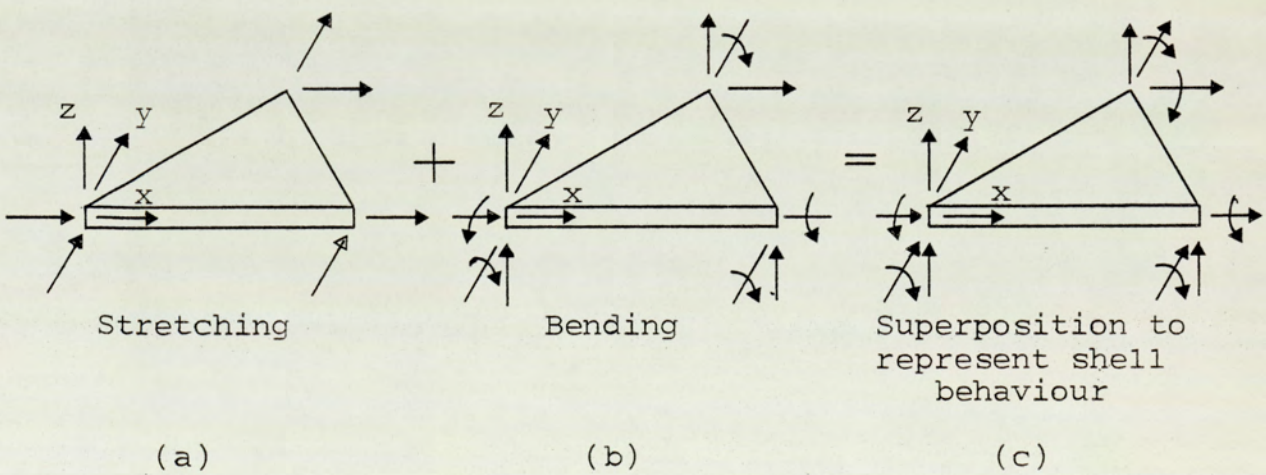


Fig.3.5 Flat shell element

Flat rectangular and quadrilateral elements are restricted in use, but they were the earliest used for analysing cylindrical shell roofs, Ref. (25); cylindrical shapes can be represented well by these elements. In the idealisation of an arbitrary shell into flat elements only triangular elements can be used. Flat elements have the capability of rigid body motion without strain. Their mathematical formulation is found in Ref. (20). These elements have in common the disadvantage of uncoupled representation of membrane and bending behaviour. For a curved geometry idealisation it is necessary to construct a fine mesh in order to achieve a reasonable degree of accuracy. Among these, flat triangular elements are:

(i) The Carr triangle Ref.(26) with nine degrees of freedom; cubic interpolation has been used for both membrane and bending behaviours.

(ii) The Chu and Schnobrich triangle Ref. (27). This uses quadratic in-plane displacement functions, together with quartic bending displacement functions Ref. (21). The element has three corner nodes and three midside nodes. The degrees of freedom at corner nodes are u, v, w, $\frac{\partial w}{\partial x}$, $\frac{\partial w}{\partial y}$ and the midside nodes are u, v, w, $\frac{\partial w}{\partial n}$.

(iii) Dawe Ref. (28) used a facet element in the analysis of shells. The element is of triangular plane form and it's assumed displacement field is such that the membrane stresses and the bending moments are uniform within the elements. The twelve degrees of freedom per element are three displacement components u, v and w at each corner node, plus the value of the slope normal to the edge, $\frac{\partial w}{\partial n}$ at the midside of each edge.

(iv) Mervyn, et al. Ref. (29) has formulated the 18 degrees of freedom flat triangular shell element. A three node stress triangle with three degrees of freedom per node is used (in-plane displacements and the in-plane rotation). The plane stress triangle is combined with the well-known 9 parameter bending triangle. The final element stiffness matrix is (18×18).

(v) Klaus and Lee-Wing Ref (30) have developed a simple flat three-node triangular shell element for linear and non-linear analysis of general shell structures. The element stiffness matrix with six degrees of freedom per node is obtained by superimposing its bending and membrane stiffness matrices.

In conclusion, if these elements are used on shells of high curvature, the number of elements required for discretisation should be high.

3.4.3 Curved Shell Elements

These types of elements generally have a complex formulation, but this means that fewer elements are needed accurately to model a shell than in the case of flat type elements. Curved shell elements, based on the assumption of different thin-shell theories, have been developed as an attempt to overcome the disadvantages of flat elements. The curved shell elements are either:

(a) cylindrical shell elements

or

(b) doubly curved shells and doubly curved shallow elements.

(a) Cylindrical Shell Elements

The curved cylindrical shell elements have been investigated by many researchers, which has resulted in a great many papers. They have been written, to take into account the type of geometric symmetry existing in cylindrical shells. Among these are:

(i) Connor and Brebbia Ref. (31) have formulated a simple cylindrical shell element. The four term polynomial (1 x y xy) is chosen to describe the membrane displacement, while the radial displacement is described by a twenty-term polynomial ($1\ x\ y\ x^2\ xy\ y^2\ x^3\ x^2y\ xy^2\ y^3\ x^3y\ xy^3$). The nodal degrees of freedom are five at each node, namely,

three displacements u, v, w and two rotations w_x and $(w_y - \frac{v}{R})$ where R is the shell radius. The element has 20 d.o.f. and is a non-conforming element. It has proved to be the simplest cylindrical shell curved element.

(ii) Gallagher Ref.(32) gave a 24 d.o.f. conforming element similar to the one suggested by Ref.(31), but has the following term added to w (x^2y^2 x^3y^2 x^2y^3 x^3y^3) and the additional nodal d.o.f. w_{xy}.

(iii) Cantin and Clough Ref.(33) have modified the polynomial displacement used by Gallagher to reproduce the rigid-body modes of the element. The element has been modified so that all rigid-body displacements are explicitly represented. Trigonometric terms are introduced into the shape function, as well as the coupling between expressions for u, v and w. The element is a conforming one with 24 d.o.f. having the same six nodal d.o.f. as the Gallagher element.

(iv) Olsen and Lindberg Ref. (34) have formulated the stiffness matrix for a four-sided cylindrical shell element using Love's strain-displacement equations Ref.(35). The twelve-term polynomial (1 x y x^2 xy y^2 x^3 x^2y xy^2 y^3 x^3y xy^3) has been chosen to describe the radial displacement w, while the membrane displacements are described by u or v= 1, x, y, xy, y^2, xy^2, y^3, xy^3. The displacement

parameters at each node are: u,v,w, $\frac{\partial u}{\partial x}$, $\frac{\partial v}{\partial y}$, $\frac{\partial w}{\partial x}$ and $\frac{\partial w}{\partial y}$. The rigid body mode requirements are satisfied, but full compatibility is not.

(v) Giannini and Miles, Ref. (36) have derived the stiffness equations for curved elements of orthotropic axi-symmetric thin shells. In their analysis a stiffness matrix was derived for an element defined by arbitrary parametric equations. Polynomials of any degree are included in the displacement field approximation.

(vi) Henshell et al. Ref. (37) have formulated a cylindrical hybrid shell element by assuming a fourteen-term opitmum stress function within the element. Bilinear polynomials are used for u and v and the Birkhoff, et al Ref. (38) polynomial (modified so as to include rigid body modes) has been used for w.

(v) Fonder Ref. (39) has developed a 48 d.o.f. element using the higher-order polynomials to represent rigid-body displacements. This gives a satisfactory approximation to rigid body displacements with the modification carried out by Ref. (33). Fonder has taken bicubic interpolation polynomials for all the curvilinear displacements u, v and w, which leads to 48 d.o.f. per element with twelve unknown displacements at four corners. This element has proved able to ensure continuity of displacement, their first derivatives and some of their second derivatives.

(b) Doubly Curved Shell Elements and Doubly Curved Shallow Elements

In the following section, a review of the elements published to deal with types of thin doubly curved shell elements and doubly curved shallow elements is introduced. Extensive literature on this subject is available. It includes

(i) Key et al. Ref. (40) have described a curved quadrilateral shell element with transverse shear. The displacement field is represented by Hermition polynomials (values and their first derivatives) coupled with the isoparametric concept Ref. (41). Rigid body mode requirements are not satisfied by the shell theory used.

(ii) Greene, et al. Ref. (42) have described an approach to shell analysis based on the generalised variational principle Ref. (43). The displacements are represented by cubic polynomials.

(iii) Wempner, et al. Ref. (44) have derived a linear theory for thin shells, which includes transverse shear deformation in terms of the middle surface displacements and the rotation of the normal to the middle surface. These displacement parameters, for a curvilinear quadrilateral element, are then represented by simple

polynomials in the curvilinear surface co-ordinates. Continuity of displacement is imposed at the four nodes only.

(iv) Argyris and Scharpf Ref. (45) have formulated the SHEBA element. The formulation is based on a complete quintic polynomial (21 terms each) for all three displacement components. Inter-element continuity is assured by the use of midside nodes having derivatives at three nodes as degrees of freedom. The corner nodes have 18 degrees of freedom each: three translational displacements, all six of the first derivatives and all nine of the second derivatives. Results in Ref. (46) showed that this element is reliable to produce accurate solutions.

(v) The super-parametric quadratic shell elements of Ahmed Ref. (47) as modified by Too Ref. (48), are efficient in representing both thin and thick plates and shells.

(vi) Cowper, et al. Ref. (49) have developed a conforming shallow shell finite element of arbitrary triangular shape. The element incorporates 36 generalised co-ordinates, namely, the displacement, w, and its first and second derivatives as well as the tangential displacements u, v and their first derivatives at each vertex. The

displacement function for the normal deflection, w, of the shell is taken as a polynomial (21 terms) while the tangential displacements, u and v for the shell are each expressed as cubic polynomials (20 terms each). The final element has 36 d.o.f., with 12 at each of the corner nodes; these consist of u, u_x, u_y, v, v_x, v_y, w, w_x, w_y, w_{xy}, w_{xx} and w_{yy}).

(vii) Dupis and Goel Ref. (84) have used a triangular thin shell finite element. The formulation satisfies all necessary requirements for convergence of the potential energy. Rational functions are used to define the displacement functions (see Ref. (20)). The rigid body motion is satisfied.

(viii) Ernest Ref. (50) has examined some aspects of the developments of a curved triangular shell element, which is capable of describing the general non-linear theory of thin elastic shells. A correct description of the rigid body displacements is given.

(ix) Mohr Ref. (51) has developed a doubly curved isoparametric triangular shell element with six nodes. Quadratic interpolation is applied to the freedoms u, v, w , $\frac{\partial w}{\partial x}$ and $\frac{\partial w}{\partial y}$ at each node. Consequently, the element has 30 freedoms.

(x) Sander Ref. (52) has developed a family of quadrilateral finite elements allowing for variable curvature. It is reported that in deep shell elements a complex formulation of the shell theory exists: In particular it is not easy to preserve the continuity of both the displacements and the rotation of the middle surface. Sander's elements are conforming elements and each element contains three displacement components and two rotation components in each corner. Along the interface, a tangential rotation is always present. It is recommended that these elements are used for the study of shells with small curvature radii.

(xi) Recently, Irons Ref. (53) has developed the semiloof shell element formulated on the basis of curved shell theory. The semiloof shell element has nodal parameters selected in such away as to simplify the treatment of problems with multiple junctions. It is essentially a non-conforming element incorporating discrete Kirchoff assumptions. Since this element is a non-conforming one, it owes its usefulness to the Patch Test. This element passes the Patch Test Ref. (53) therefore, the convergence is achieved. The rigid body motions are satisfied exactly.

3.5 CLOSING REMARKS

In the present work, the finite element method was chosen as a tool to predict the stress levels in rotating fan impellers, since it is the most powerful method to deal with such complicated structures. The element chosen for discretisation is the semiloof shell element, because (as explained in Chapter Two) the centrifugal fan impellers are essentially thin shells with sharp corners having junctions between the blades and sheets (Backsheet and Conesheet). All these problems can be overcome by using this element as recommended by Irons.

In the next chapter, the mathematical formulation of the semiloof shell element will be summarised in order to aid an account of the development of a computer program implemented in desk top computer HP9845B. This program is designed to solve for displacements, strains and stresses for general thin plate and shell structures using the semiloof shell element.

CHAPTER FOUR

SEMILOOF SHELL ELEMENT IN THE ANALYSIS OF THIN PLATE AND SHELL STRUCTURES

4.1 INTRODUCTION

The finite element method described in Chapter Three has become a standard powerful tool in the analysis of thin plate and shell structures, to find their stresses and displacements. This is because the available analytical solutions to such structural problems are limited and do not apply to arbitrary shapes, load conditions, irregular stiffening, support conditions and many other practical problems. In contrast the finite element technique can deal with the above requirements efficiently.

This technique has been used in the analysis of thin plate and shell structures such as the work involving aerospace structural design and naval architecture. The stress analysis of such structures has been subjected to large research activities, since the early days of the finite element method, by virtue of the specific needs of aerospace structural design. Different discretisation approaches were presented (see Refs.(53,54)) such as the method involving flat, isoparametric and curved thin shell elements. In this project the semiloof shell element has been chosen, since this element has nodal parameters which are selected to simplify the treatment of the problem with multiple junctions and intersection situations.

A description of the semiloof shell element is given here. The computer programs were developed using this element to solve the stresses and displacements in thin plate and shell structures. The plate is considered as a shell with zero curvature. Therefore, all the explanations are related to shell elements.

4.2 THEORY OF THE SEMILOOF SHELL ELEMENT

4.2.1 Introduction

A large proportion of plate and shell structures are thin with sharp corners and multiple junction regions. The semiloof shell element was developed by B. Irons(53) to meet the practical needs of analysing such structures. This element is a non-conforming one which has passed the Patch Test. Kirchhoff assumptions of shell theory, that normals to the mid-surface plane remain normal in deformation, are made at the integration points. The versatility of the most familiar isoparametric elements have been passed on to semiloof one. The element was developed as a direct stiffness displacement element. In fact, it is the result of a gradual process beginning at the isoparametric elements developed by Ergatoudis(55) and a membrane and membrane stack elements, developed by Ahmad (47) which in turn was derived from the solid elements. An improvement to this model was achieved by

the use of reduced integration see Ref.(56) which made the element more economical, since it could be used to model thin shells. It has been used by various researchers due to its generality, type of formulation and excellent performance,it is now regarded as one of the most efficient elements for engineering shell and plate applications. Amongst those investigators, Martins Ref (57) showed that the semiloof shell element is extremely competitive for the solution of structural stability and natural vibration problems. Javaherian et al.(58) have extended the use of the element to include large deflection and plastic deformation effects, while Martins Ref (59) has extended the use of this element to nonlinear situations in which elastic-plastic or large deflection effects are present. Denis et al. Ref (60) used the element for the solution of elastic-viscoplastic and elasto-plastic large deformation thin plate and shell problems. It can be concluded that this versatile element is one of the most suitable elements to deal with thin engineering structures. Its application to the solution of structural problems follows the same general procedure outlined in Chapter 3. In the following section, the semiloof shell element theory will be discussed while the relevant matrix equations are described in a form suitable to be implemented in the computer programs the development of which is described in Section 4.3.

The discussion presented here covers the quadrilateral semiloof shell only, but both quadrilateral and triangular versions have been implemented in the computer programs.

4.2.2 Nodal Configuration

Two versions of the semiloof shell element are available: triangular with three corner nodes and three midside nodes and quadrilateral with four corner nodes and four midside nodes. The midside node lies on the perpendicular bisector plane of the two end nodes allowing parabolically curved edges. Topology is ordered by starting at any corner node and progressing round the element, as shown in Figs. 4.1(a) and 4.1(b). This order defines the direction of the local Z-coordinates using a right hand screw rule. The element degrees of freedom are: three displacements at each corner and midside node; two rotations normal to the side at two points along each side, positioned at a distant $\frac{1}{2\sqrt{3}} \times$ side length from the middle of the side, these nodes are called loof nodes. This gives a total of 32 d.o.f. for a quadrilateral and 24 for a triangular element.

4.2.3 Shape Function Polynomials

The semiloof shell element adopts the well-known isoparametric 8-noded parabolic model. The introduction

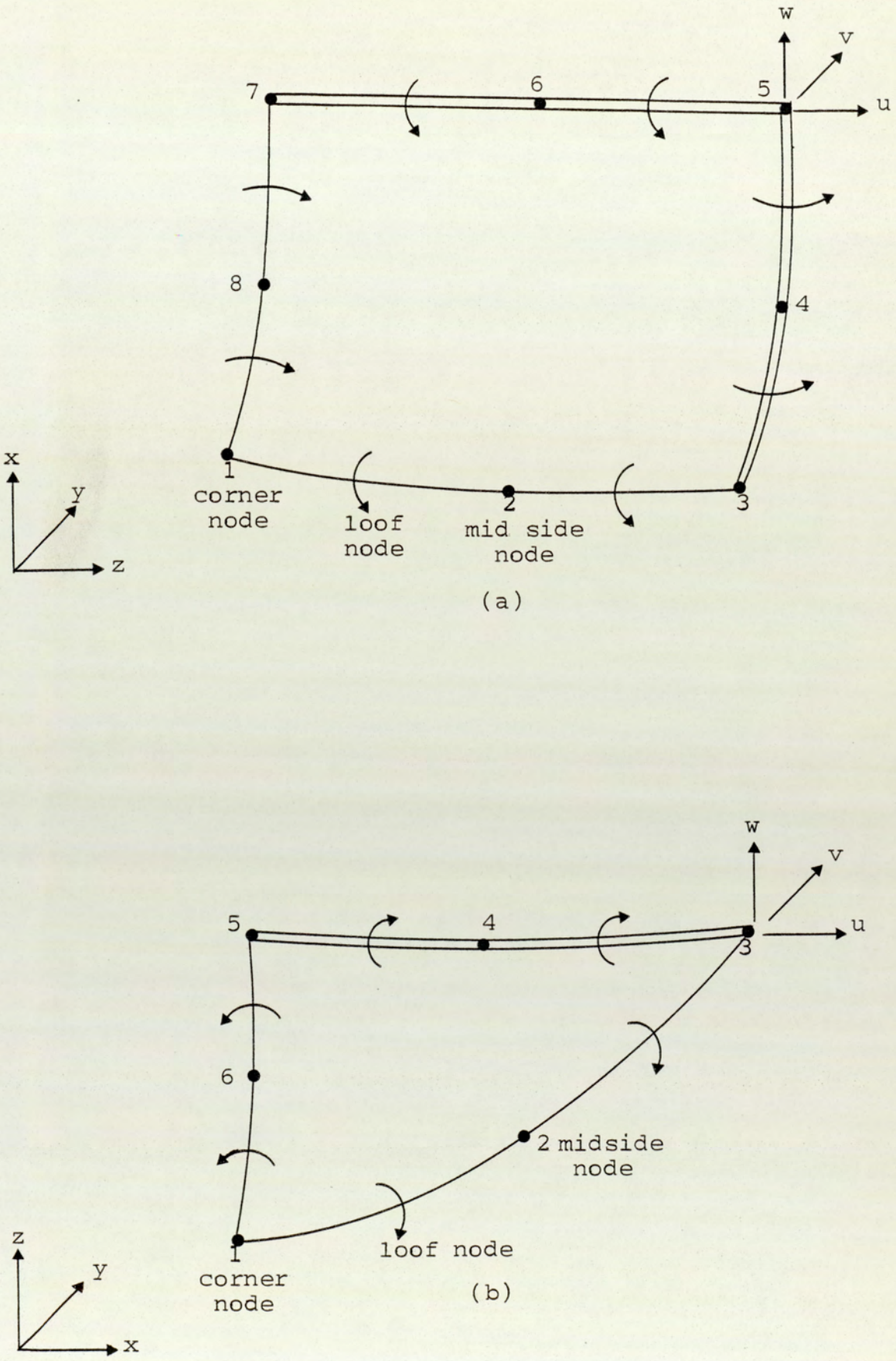


Fig.4.1 Semiloof shell element. (a) Quadrilateral type (32 d.o.f). (b) Triangular type (24 d.o.f)

of the normal rotation variables at the loof nodes on the element periphery is to maintain the C^1-continuity. The following shape functions in terms of curvilinear coordinate system (ξ,η) are used for quadrilateral elements ($\xi_o = \xi_i\ \xi$, $\eta_o = \eta_i\ \eta$) (See Fig. 4.2)

(a) For corner nodes

$$N^i = \frac{1}{4}(1 + \xi_o)(1 + \eta_o)(\xi_o + \eta_o - 1) \qquad \ldots \quad (4.1)$$

(b) For midside nodes

$$N^i = \frac{1}{2}(1 - \xi^2)(1 + \eta_o), \qquad \xi_i = 0 \qquad \ldots \quad (4.2)$$

$$N^i = \frac{1}{2}(1 - \eta^2)(1 + \xi), \qquad \eta_i = 0 \qquad \ldots \quad (4.3)$$

(c) For loof nodes:

The shape function of the loof nodes $L(\xi,\eta)$ is a polynomial with 8 terms, which is given in Ref. (58)

$$L^j = \frac{3}{32}(3\xi^2-\eta^2)+ \frac{1}{8}\left[3\xi_o(1-\eta^2)+3\eta_o\{3\xi^2+\eta_o-1+ \frac{3}{32}.\xi(\xi^2-\eta^2)\}\right]$$

$$\xi_i = \pm 1 \qquad \ldots \quad (4.4)$$

$$L^j = \frac{3}{32}(3\eta^2-\xi^2)+ \frac{1}{8}\left[3\eta^2(1-\xi^2)+3\xi_o\{3\eta^2+\eta_o-1+ \frac{3}{32}.\eta_o(\eta_o-\xi^2)\}\right]$$

$$\eta_i = \pm 1 \qquad \ldots \quad (4.5)$$

(d) Central node: the bubble function (Fig. 4.3)

$$N^9 = (1 - \xi^2)(1 - \eta^2) \quad \ldots \quad (4.6)$$

The consideration of this shape function has been given in Ref.(61).

4.2.4 In Plane Behaviour

Fig. 4.4 shows a quadrilateral semiloof shell element, relative to global axes (O,x,y,z), and a general point p with local coordinates X, Y, Z. Describing the in-plane behaviour means determination of the displacement of a particular point contained in a plane tangent to the element. A set of local axes comprises one axis normal to the surface, while the other two are contained in a plane tangent to the element at a point of interest. The displacement with respect to local axes may be found by determinating the displacements in the global axes and projecting them on the directions of the local axes.

Define the displacements in global axes as follows:

$$\{d\} = \begin{Bmatrix} u \\ v \\ w \end{Bmatrix} = \begin{Bmatrix} \text{displacement in x direction} \\ \text{displacement in y direction} \\ \text{displacement in z direction} \end{Bmatrix} \quad \ldots \quad (4.7)$$

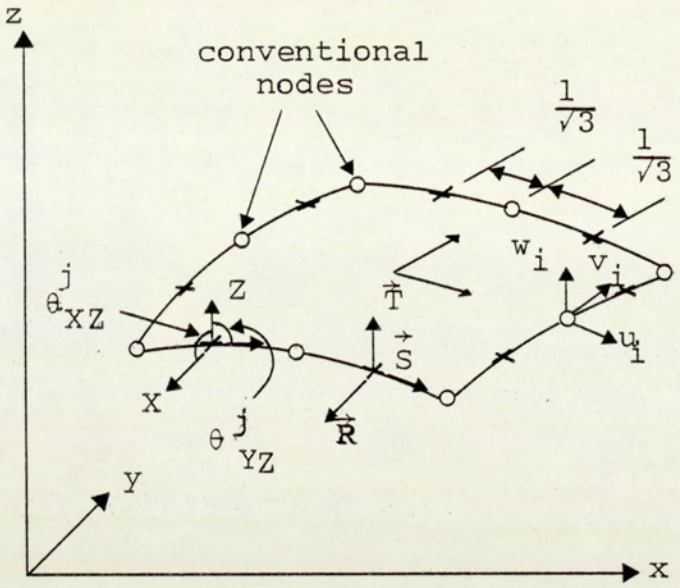


Fig.4.2 Nodal configuration of the semiloof shell element (quadrilateral type)

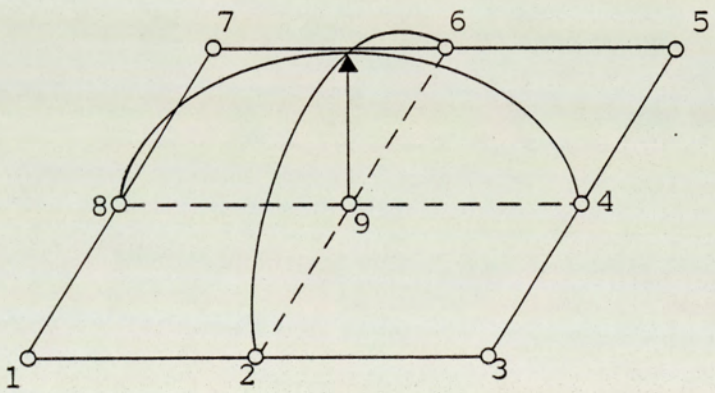


Fig.4.3 The bubble function

N.B The bubble function N^9 had to be added in order that the patch test be satisfied. See Ref.(61) for a discussion relating to the quadrilateral semiloof shell element.

These displacements are interpolated inside the element using the shape functions defined by equations (4.1 - 4.3) and the shape function for the central node defined by equation (4.6) as follows:

$$u = u_1 N^1 + u_2 N^2 + \ldots\ldots + u_9 N^9 \qquad \ldots \quad (4.8)$$

$$v = v_1 N^1 + v_2 N^2 + \ldots\ldots + v_9 N^9 \qquad \ldots \quad (4.9)$$

$$w = w_1 N^1 + w_2 N^2 + \ldots\ldots + w_9 N^9 \qquad \ldots \quad (4.10)$$

If a matrix of shape functions $|N|$ is defined by the following expression:

$$|N| = \begin{bmatrix} N^1 & 0 & 0 & N^2 & 0 & 0 & & N^9 & 0 & 0 \\ 0 & N^1 & 0 & 0 & N^2 & 0 & \ldots\ldots & 0 & N^9 & 0 \\ 0 & 0 & N^1 & 0 & 0 & N^2 & & 0 & 0 & N^9 \end{bmatrix} \qquad \ldots \quad (4.11)$$

then, the displacements with respect to global axes may be written in a simple matrix equation:

$$\{d\} = [N]\{\delta\}^e \qquad \ldots \quad (4.12)$$

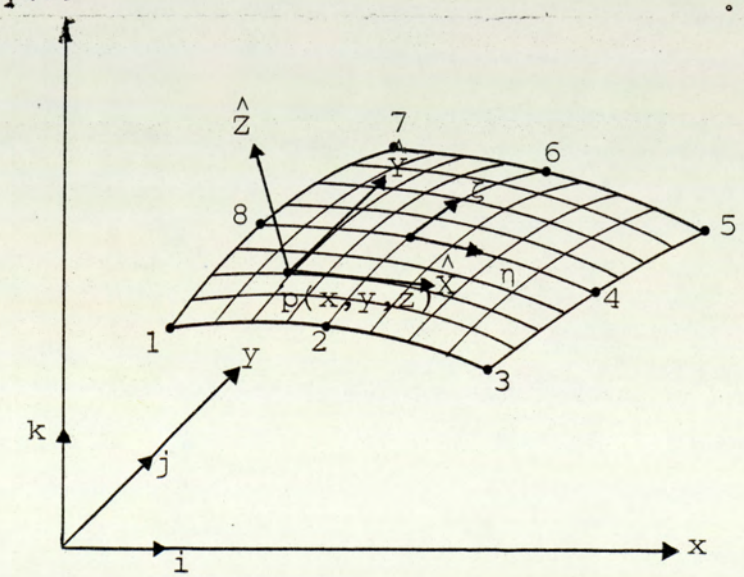


Fig 4.4

Unit local axis

$$\text{where } \{\delta\}^e = \begin{Bmatrix} u_1 \\ v_1 \\ w_1 \\ u_2 \\ v_2 \\ w_2 \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ u_9 \\ v_9 \\ w_9 \end{Bmatrix} = \begin{Bmatrix} \text{displ. along x at corner node 1} \\ \text{displ. along y at corner node 1} \\ \text{displ. along z at corner node 1} \\ \text{displ. along x at midside node 2} \\ \text{displ. along y at midside node 2} \\ \text{displ. along z at midside node 2} \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \text{displ. along x at central node 9} \\ \text{displ. along y at central node 9} \\ \text{displ. along z at central node 9} \end{Bmatrix} \quad \ldots \quad (4.13)$$

For convenience it is defined:

$$\{\theta^e_{XZ}\} = \begin{Bmatrix} \theta^1_{XZ} \\ \theta^2_{XZ} \\ \cdot \\ \cdot \\ \cdot \\ \theta^9_{XZ} \end{Bmatrix} = \begin{Bmatrix} \text{rotation normal to the side at loof node 1} \\ \text{rotation normal to the side at loof node 2} \\ \cdot \\ \cdot \\ \cdot \\ \text{rotation along } \xi \text{ at the central node 9} \end{Bmatrix} \quad \ldots \quad (4.14)$$

$$\{\theta^e_{YZ}\} = \begin{Bmatrix} \theta^1_{YX} \\ \theta^2_{YZ} \\ \cdot \\ \cdot \\ \cdot \\ \theta^9_{YZ} \end{Bmatrix} = \begin{Bmatrix} \text{rotation along the side at the loof node 1} \\ \text{rotation along the side at the loof node 2} \\ \cdot \\ \cdot \\ \cdot \\ \text{rotation along the } \eta \text{ at the central node 9} \end{Bmatrix} \quad \ldots \quad (4.15)$$

The displacements, U and V in the directions of the local axes X and Y respectively may be obtained as follows:

$$U = \{X\}^t\{d\} \qquad \ldots \quad (4.16)$$

$$V = \{Y\}^t\{d\} \qquad \ldots \quad (4.17)$$

The vectors {X}, {Y} together with {Z} Fig. (4.4) correspond a set of local axes one being normal to the surface, another which is oriented along the ξ-direction and a third defined by vector multiplication to give an orthogonal system.

If a point p(x,y,z) of the mid-surface of the element is considered, the vectors $\vec{X}$, $\vec{Y}$ and $\vec{Z}$ may be written in terms of non-dimensional parameters (ξ,η) as follows:

$$\vec{Z} = \frac{\partial\vec{P}}{\partial\xi} \times \frac{\partial\vec{P}}{\partial\eta}\ , \quad \vec{X} = \frac{\partial\vec{P}}{\partial\xi}, \quad \vec{Y} = \vec{X}\times\vec{Z}$$

These vectors are normalised to give $\hat{X}$, $\hat{Y}$ and $\hat{Z}$ (see Ref.(61)),

where $$\hat{X} = \{X\} = \begin{Bmatrix} X_x \\ X_y \\ X_z \end{Bmatrix} \quad , \quad \hat{Y} = \{Y\} = \begin{Bmatrix} Y_x \\ Y_y \\ Y_z \end{Bmatrix}$$

$\ldots$ (4.18)

and $$\hat{Z} = \{Z\} = \begin{Bmatrix} Z_x \\ Z_y \\ Z_z \end{Bmatrix}$$

To define the membrane strains, the derivatives of the membrane displacements have to be found. i.e.

$$\frac{\partial U}{\partial X} = \{X\}^t \left[\frac{\partial N}{\partial X}\right]\{\delta^e\}$$

$$\frac{\partial U}{\partial Y} = \{X\}^t \left[\frac{\partial N}{\partial Y}\right]\{\delta^e\}$$

$$\frac{\partial V}{\partial X} = \{Y\}^t \left[\frac{\partial N}{\partial X}\right]\{\delta^e\} \quad \ldots (4.19)$$

$$\frac{\partial V}{\partial Y} = \{Y\}^t \left[\frac{\partial N}{\partial Y}\right]\{\delta^e\}$$

where $$\left[\frac{\partial N}{\partial X}\right] = \begin{bmatrix} \frac{\partial N^1}{\partial X} & 0 & & \frac{\partial N^9}{\partial X} & 0 & 0 \\ 0 & \frac{\partial N^1}{\partial X} & & 0 & \frac{\partial N^9}{\partial X} & 0 \\ 0 & 0 & \frac{\partial N^1}{\partial X} & 0 & 0 & \frac{\partial N^9}{\partial X} \end{bmatrix} \quad \ldots (4.20)$$

similarly

$$\left[\frac{\partial N}{\partial Y}\right] = \begin{bmatrix} \frac{\partial N^1}{\partial Y} & 0 & 0 & \frac{\partial N^9}{\partial Y} & 0 & 0 \\ 0 & \frac{\partial N^1}{\partial Y} & 0 & 0 & \frac{\partial N^9}{\partial Y} & 0 \\ 0 & 0 & \frac{\partial N^1}{\partial Y} & 0 & 0 & \frac{\partial N^9}{\partial Y} \end{bmatrix} \quad \ldots (4.21)$$

4.2.5 Out-Of-Plane Behaviour

As far as out-of-plane behaviour is concerned, it is of interest to find the displacement, W, at each point along the normal to the element surface (i.e. the displacement along the local axis $\hat{Z}$).

As has been done for U and V in the previous section, here, the {d} is projected on to $\hat{Z}$ in order to obtain the displacement, W, as follows:

$$W = \{Z\}^t\{d\}$$

or using (4.12)

$$W = \{Z\}^t [N]\{\delta^e\} \qquad \ldots \quad (4.22)$$

In order to write the bending strain expressions for this element, it is necessary to obtain the contribution of the loof nodes and the corner midside nodes as follows:

The first derivative of the out-of-plane displacement of a particular point p(x,y,z) of the element can be written as:

$$\frac{\partial U}{\partial Z} = \left(\frac{\partial U}{\partial Z}\right)^{\ell} + \left(\frac{\partial U}{\partial Z}\right)^{n}$$

$$\frac{\partial V}{\partial Z} = \left(\frac{\partial V}{\partial Z}\right)^{\ell} + \left(\frac{\partial V}{\partial Z}\right)^{n} \qquad \ldots \quad (4.23)$$

where

ℓ represents the effect of the rotations at the loof nodes and central node, and

n represents the effect of the displacement at the corner and midside nodes.

A comprehensive discussion of the out-of-plane behaviour is available in Ref.(61). However, in this section the equations are only dealt with briefly in order to show the expressions which are sued in deriving the relation between strain and element nodal displacements. To define the bending behaviour in the semiloof shell element, a vector thickness at the loof nodes and central node, j, needs to be defined.

Vector thickness is specified as

$$\vec{T}^j = t^j \hat{Z}^j \qquad \ldots \quad (4.24)$$

where t^j is the scalar shell thickness at node j.

In addition, the rotation and the slope along the edge are to be defined (see Fig. 4.2).

$$\vec{R}^j = \vec{T}^j \times \vec{Y}^j = \begin{Bmatrix} \vec{R}_x^j \\ \vec{R}_y^j \\ \vec{R}_z^j \end{Bmatrix} \qquad \ldots \quad (4.25)$$

and $\quad \overset{*}{S}^j = t^j \hat{Y}^j = \begin{Bmatrix} S_x^j \\ S_y^j \\ S_z^j \end{Bmatrix}$ ... (4.26)

As mentioned earlier, the derivative $\frac{\partial U}{\partial Z}$ has two contributions one from the displacements $\{\delta\}^e$; the other due to the rotations $\{\theta^e_{XZ}\}$. These are respectively:

$$\left(\frac{\partial U}{\partial Z}\right)^n = \frac{1}{t}\left[-T_x\{X\}^t\left[\frac{\partial N}{\partial X}\right] - T_y\{X\}^t\left[\frac{\partial N}{\partial X}\right]\right]\{\delta\}^e \quad \ldots \quad (4.27)$$

$$\left(\frac{\partial U}{\partial Z}\right)^\ell = \frac{1}{t}\{X\}^t[R][L]\{\theta^e_{XZ}\} + \frac{1}{t}[S][L]\{\theta^e_{YZ}\} \quad \ldots \quad (4.28)$$

where $[L]$ is a (9,9) diagonal matrix with shape functions for the 8 loof nodes on the element boundary and the central loof node:

$$\text{i.e. } [L] = \begin{bmatrix} L^1 & 0 & & & 0 \\ 0 & L^2 & & & 0 \\ . & & & & 0 \\ . & & & & . \\ . & & & & . \\ . & & & & . \\ . & & & & L^9 \end{bmatrix} \quad \ldots \quad (4.29)$$

and where $[R]$ and $[S]$ represent two (3×9) matrice, with (x,y,z) components:

$$\{R\} = \begin{bmatrix} R_x^1 & R_x^2 & \cdots\cdots\cdots & R_x^9 \\ R_y^1 & R_y^2 & \cdots\cdots\cdots & R_y^9 \\ R_z^1 & R_z^2 & \cdots\cdots\cdots & R_z^9 \end{bmatrix} \quad \ldots \quad (4.30)$$

$$|S| = \begin{bmatrix} S_x^1 & S_x^2 & \cdots\cdots\cdots & S_x^9 \\ S_y^1 & S_y^2 & \cdots\cdots\cdots & S_y^9 \\ S_z^1 & S_z^2 & \cdots\cdots\cdots & S_z^9 \end{bmatrix} \quad \ldots \quad (4.31)$$

$\{\theta^e_{XZ}\}$ and $\{\theta^e_{YZ}\}$ were defined in equations (4.14) and (4.15).

The derivatives required to define the bending behaviour are given by:

$$\frac{\partial^2 U}{\partial X \partial Z} = \frac{1}{t}\{- \frac{\partial T_x}{\partial Y}\{X\}^t[\frac{\partial N}{\partial X}] - \frac{\partial T_y}{\partial X}\{X\}^t[\frac{\partial N}{\partial Y}]$$

$$+ \frac{\partial T_z}{\partial X}\{Z\}^t[\frac{\partial N}{\partial X}]\}\{\delta^e\} + \frac{1}{t}\{X\}^t[R][\frac{\partial L}{\partial X}]\{\theta^e_{XZ}\}$$

$$+ \frac{1}{t}\{X\}^t[S][\frac{\partial L}{\partial X}]\{\theta^e_{YZ}\} \quad \ldots \quad (4.32)$$

$$\frac{\partial^2 U}{\partial Y \partial Z} = \frac{1}{t}\{- \frac{\partial T_x}{\partial Y}\{X\}^t[\frac{\partial N}{\partial X}] - \frac{\partial T_y}{\partial Y}\{X\}^t[\frac{\partial N}{\partial Y}] + \frac{\partial T_z}{\partial X}$$

$$\{Z\}^t[\frac{\partial N}{\partial Y}]\}\{\delta^e\} + \frac{1}{t}\{X\}^t[R][\frac{\partial L}{\partial Y}]\{\theta^e_{XZ}\} +$$

$$\frac{1}{t}\{X\}^t[S][\frac{\partial L}{\partial Y}]\{\theta^e_{YX}\} \quad \ldots \quad (4.33)$$

$$\frac{\partial^2 V}{\partial X \partial Z} = \frac{1}{t}\{- \frac{\partial T_x}{\partial X} \{Y\}^t \left[\frac{\partial N}{\partial X}\right] - \frac{\partial T_y}{\partial X} \{Y\}^t \left[\frac{\partial N}{\partial Y}\right] +$$

$$\frac{\partial T_z}{\partial Y} \{Z\}^t \left[\frac{\partial N}{\partial X}\right]\}\{\delta^e\} + \frac{1}{t}\{Y\}^t [R] \left[\frac{\partial L}{\partial X}\right] \{\theta^e_{XZ}\}$$

$$+ \frac{1}{t}\{Y\}^t [S] \left[\frac{\partial L}{\partial X}\right] \{\theta^e_{YZ}\} \quad \ldots \quad (4.34)$$

$$\frac{\partial^2 V}{\partial Y \partial Z} = \frac{1}{t}\{- \frac{\partial T_x}{\partial Y} \{Y\}^t \left[\frac{\partial N}{\partial X}\right] - \frac{\partial T_y}{\partial Y} \{Y\}^t \left[\frac{\partial N}{\partial Y}\right] +$$

$$\frac{\partial T_z}{\partial Y} \{Z\}^t \left[\frac{\partial N}{\partial Y}\right]\}\{\delta^e\} + \frac{1}{t} \{Y\}^t [R] \left[\frac{\partial L}{\partial Y}\right] \{\theta^e_{XZ}\}$$

$$+ \frac{1}{t} \{Y\}^t [S] \left[\frac{\partial L}{\partial Y}\right] \{\theta^e_{YZ}\} \quad \ldots \quad (4.35)$$

Each of the equations (4.32 - 4.35) contains the initial 45 d.o.f. of the element. The displacements at the centre are combined to create a displacement normal to the element giving the unconstrained element 43 d.o.f.

4.2.6 Application of Constraints

As mentioned in the previous section, the number of degrees of freedom for the unconstrained semiloof shell element (quadrilateral type) is 43 d.o.f. At this stage 11 variables are eliminated to give a final total of 32 d.o.f. for the element. In fact, the most

critical stage in the preparation of the semiloof shell element is the definition and application of shear constraints. The application of these constraints is treated briefly in order to obtain the unconstrained semiloof shell element. These constraints are as follows: (for details see Ref.(62)).

(i) Shear strain γ_{YZ} caused by the rotation θ_{YZ} to be zero at the loof nodes, this provides 8 constraints.

(ii) The area integrals

$$\int_A \hat{X}_c . \gamma . dA = \int_A \hat{Y}_c . \gamma . dA = 0$$

where $\hat{X}_c$ and $\hat{Y}_c$ represent unit vectors at the centre ($\xi=\eta=0$) and $\gamma = \hat{X} . \gamma_{XZ} + \hat{Y} . \gamma_{YZ}$ represent the vector of lateral shears.

(iii) $\int$(thickness) γ_{XZ}. d(boundary) = 0

The integral which is carried over the element boundary provides another constraint.

After the application of the above 11 constraints the final nodal variables have 24 displacement components with respect to the global axes at the corner and midside nodes, as well as 8 rotations , θ_{XZ}, normal to the edge element at each loof node. The application of the shear constraints to the equations (4.19) ($\frac{\partial U}{\partial X}$, $\frac{\partial U}{\partial Y}$, ... $\frac{\partial V}{\partial Y}$)

and the equations (4.30) ($\frac{\partial^2 U}{\partial X \partial Z}$, ..., $\frac{\partial^2 U}{\partial Y \partial Z}$) given them in terms of non-constrained degrees of freedom for the element. These degrees of freedom can be arranged in vector $\{\delta\}$ as follows:

$$\{\delta\} = \begin{Bmatrix} u_1 \\ v_1 \\ w_1 \\ u_2 \\ v_2 \\ w_2 \\ \theta_{XZ} \\ \theta_{XZ} \\ \vdots \\ u_8 \\ v_8 \\ w_8 \\ \theta_{XZ} \\ \theta_{XZ} \end{Bmatrix} \qquad \ldots \quad (4.36)$$

The sequence of the degrees of freedom collected in vector $\{\delta\}$is shown in Fig. 4.5.

4.2.7 Strain-Displacement Relations

The strain-displacement relationships for thin shells of general geometry using the semiloof shell element may be

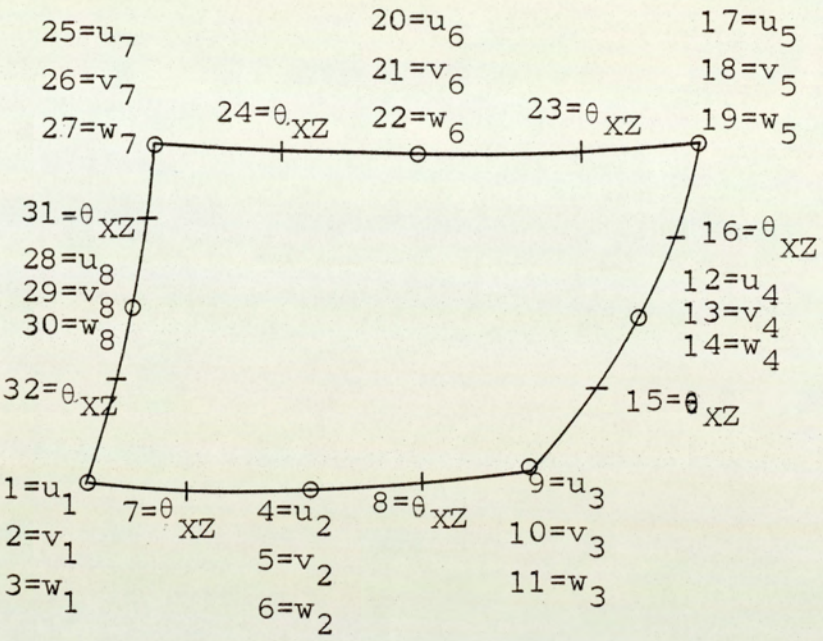


Fig.4.5 Evolution of semiloof shell element

formed by considering the corresponding relations for infinitesimal displacements. The strain components can be separated into the in-plane components $\{\varepsilon^m\}$ and the bending terms $\{\varepsilon^b\}$. They are given by:

$$\{\varepsilon\} = \begin{Bmatrix} \varepsilon^m \\ \varepsilon^b \end{Bmatrix} = \begin{Bmatrix} \varepsilon_X^m \\ \varepsilon_Y^m \\ \varepsilon_{XY}^m \\ \varepsilon_X^b \\ \varepsilon_Y^b \\ \varepsilon_{XY}^b \end{Bmatrix} = \begin{Bmatrix} \frac{\partial U}{\partial X} \\ \frac{\partial V}{\partial Y} \\ \frac{\partial U}{\partial Y} + \frac{\partial V}{\partial X} \\ - \frac{\partial^2 U}{\partial X \partial Z} \\ - \frac{\partial^2 V}{\partial Y \partial Z} \\ - \left(\frac{\partial^2 U}{\partial Y \partial Z} + \frac{\partial^2 V}{\partial X \partial Z}\right) \end{Bmatrix} \quad \ldots \quad (4.37)$$

The displacements U and V, with respect to the local coordinates system X, Y and Z are set up at every point of the shell. For a point at a distance Z from the shell midsurface, the contributions due to both membrane and bending strains are:

$$\varepsilon_X = \varepsilon_X^m + Z\varepsilon_X^b$$

$$\varepsilon_Y = \varepsilon_Y^m + Z\varepsilon_Y^b \quad \ldots \quad (4.38)$$

$$\varepsilon_{XY} = \varepsilon_{XY}^m + Z\varepsilon_{XY}^b$$

Equation (4.37) may be written as follows:

$$\begin{Bmatrix} \varepsilon^m \\ - - \\ \varepsilon^b \end{Bmatrix} = \begin{bmatrix} B^m \\ - - - - \\ B^b \end{bmatrix} \{\delta\} \qquad \ldots \quad (4.39)$$

where $\{\delta\}$ is given by equation (4.36) $[B^m]$ and $[B^b]$ represent the strain-displacement matrices for the membrane and bending effects respectively.

The element stiffness matrix [Ke] is then

$$[Ke] = \int_V [B]^t [D][B] \, dvol \qquad \ldots \quad (4.40)$$

The elasticity matrix [D], which includes both a membrane and bending part, is defined by:

$$|D| = \left[\begin{array}{c|c} D^m & 0 \\ \hline 0 & D^b \end{array}\right] \qquad \ldots \quad (4.41)$$

where

$$[D^m] = \frac{E}{1-\nu^2} \begin{bmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & \frac{1-\nu}{2} \end{bmatrix} \qquad \ldots \quad (4.42)$$

and

$$[D^b] = \frac{Et^3}{12(1-\nu^2)} \begin{bmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & \frac{1-\nu}{2} \end{bmatrix}$$

4.2.8 Numerical Integration

Equation (4.40) should be integrated numerically, using a five-point rule for the quadrilateral element, as suggested by Irons.

This rule is as follows:

$$\int_{-1}^{1}\int_{-1}^{1} \phi(\xi,\eta)d\xi d\eta = 0.2\phi(0,0) + 0.95 \sum_{1}^{4} \phi(\pm 0.592348878 \pm 0.592348878) \quad \ldots \quad (4.43)$$

4.2.9 Element Strains and Stresses

As will be seen later, the element strains and stresses are calculated at the integrating points. Smoothed nodal stresses may also be obtained for the quadrilateral element. Using equation (4.39) to obtain the nodal strains, the nodal stresses can then be calculated as follows:-

$$\{\sigma\} = [D]\{\varepsilon\} \quad \ldots \quad (4.44)$$

where $[D]$ is given by equation (4.41) and $\{\delta\}$ is the vector of nodal stresses.

4.3 DEVELOPMENT OF THE FINITE ELEMENT PROGRAMS

4.3.1 General Introduction

The work implemented in the programs has been developed employing the application of the finite element method to analyse the thin shell structures. In thin shell structures the effects of bending and in-plane actions have been taken into consideration, these are achieved by using the semiloof shell element as a discretisation element in this work. Equations necessary for implementation have been described in Section 4.2. Thin plate structures can be solved as shells with zero curvature. The programs include both versions of the semiloof shell element (quadrilateral type with 32 d.o.f. and triangular type with 24 d.o.f. as shown in Fig. 4.6.

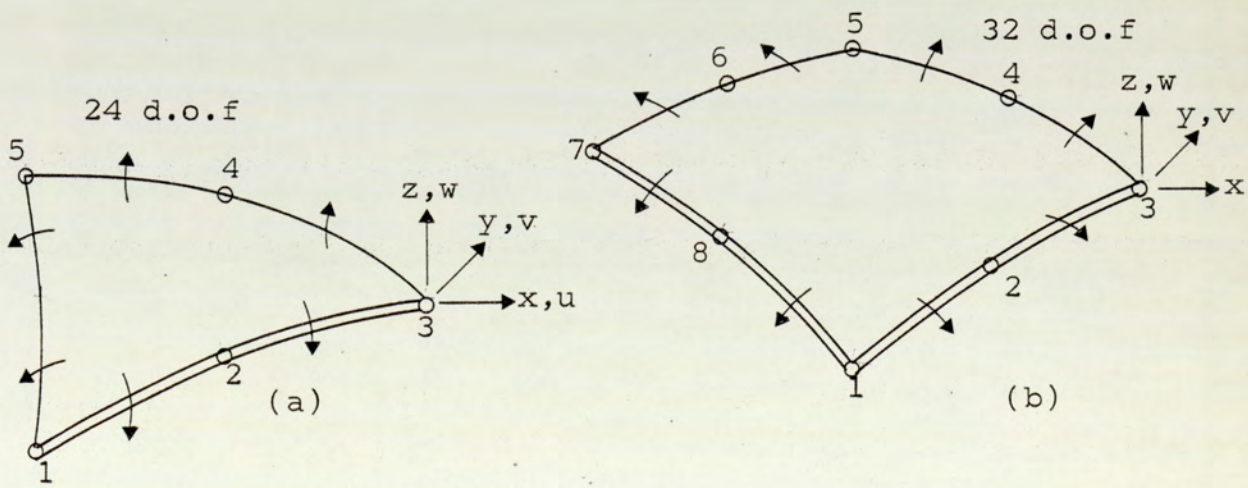


Fig 4.6 Versions of the semiloof shell element

The formation of the finite element method in general leads to a large number of simultaneous equations. For a reasonable solution accuracy, a large number of elements are required in the mesh subdivision. Consequently, the system stiffness matrix $[K]$ requires a large storage computer. The overall stiffness matrix storage can be minimised as follows:

Since $[K]$ is symmetric, only the upper or lower triangle needs to be generated and stored. In addition, since it is banded, only the coefficients of either the upper or lower band need to be retained. By using the scheme developed by Jennings and Tuff , Ref. (63), the overall stiffness may be stored as a one-dimensional array, in which only the coefficients between the first non-zero terms in any row and diagonal are retained. The coefficients of the one-dimensional array are referenced to the two-dimensional array through an address sequence. This will be explained later. The structure of the programs including the implementation of the essential steps in the finite element programs is presented in the following section.

These programs are required to have:

(i) Taking in the required input data. Such data includes the geometry of the structure, support conditions, material properties and loading to which the structure is subjected.

(ii) An analysis of element stiffness matrices.

(iii) An assembly of the overall stiffness matrix.

(iv) A solution of the force-displacement matrix equation.

(v) An evaluation of strains and stresses.

The programs were required for implementation on the Hewlett Packard HP9845B computer. The machine has a Read/Write Memory (RAM) of 187060 bytes available, with mass storage provided by two tape cartridges drives and a single flexible disc unit. Furthermore, it is possible to add another flexible disc unit, in case more mass storage is required. In fact, this desk top computer is hard-wired to interpret the BASIC language only, so that the programs need to be written in this language. However, these can be changed into other languages by simple alterations. All calculations are performed with 12-digit precision.

In the following section, the programs developed in order to prepare a stress analysis package for thin plate and shell structures using the semiloof shell element for the displacements, strains and stresses are described. The flow charts for these programs are also presented.

4.3.2 Structure of the Developed Programs

As in any systematic programming, the finite element programs are divided into various subprograms or subroutines. These are written separately and may be called either once or several times.

The programs are developed in a form suitable for a desk top computer (Hewlett Packard HP9845B) in such a way that there is a master program called SMILOF through which the other subprograms are linked whenever they are required. This is discussed in Section 4.3.2.1.

The control data, nodal coordinates, nodal connections and boundary conditions are all read by subprogram SMINPT via the "Feinput" subroutine. The address array is formed in this subprogram through the "Addarray" subroutine. The address sequence is a one-dimensional array of length equal to the structure's degrees of freedom. This is explained in Section 4.3.2.2.

In section 4.3.2.3, the elasticity matrix [D] is formed in subprogram SMNSTR. When the subroutine "Constrel" is called by the main Program. The SMASBL subprogram is presented in section 4.3.2.4 which is concerned with assembly of the overall stiffness matrix [K].

The external loading, such as concentrated loading, is transferred to the load vector {Q} in the subprogram

SMLDAP using the subroutine "Loadapp". This is discussed in Section 4.3.2.7. The subprogram SMYVBS, presented in Section 4.3.2.9, contains the following subroutines:

(a) "Geombc" : through which the boundary conditions are applied.

(b) "Symvbsol" : This solving subroutine, using cholesky decomposition, has been written by Ref.(63) to obtain the nodal displacements.

In using the computer nodal displacements, the related subprogram is linked to the main program depending on whether the nodal or element strains and stresses are required. This is explained in Section 4.3.2.10. An overall block diagram showing the broad structure of the program is shown in Fig. 4.7.

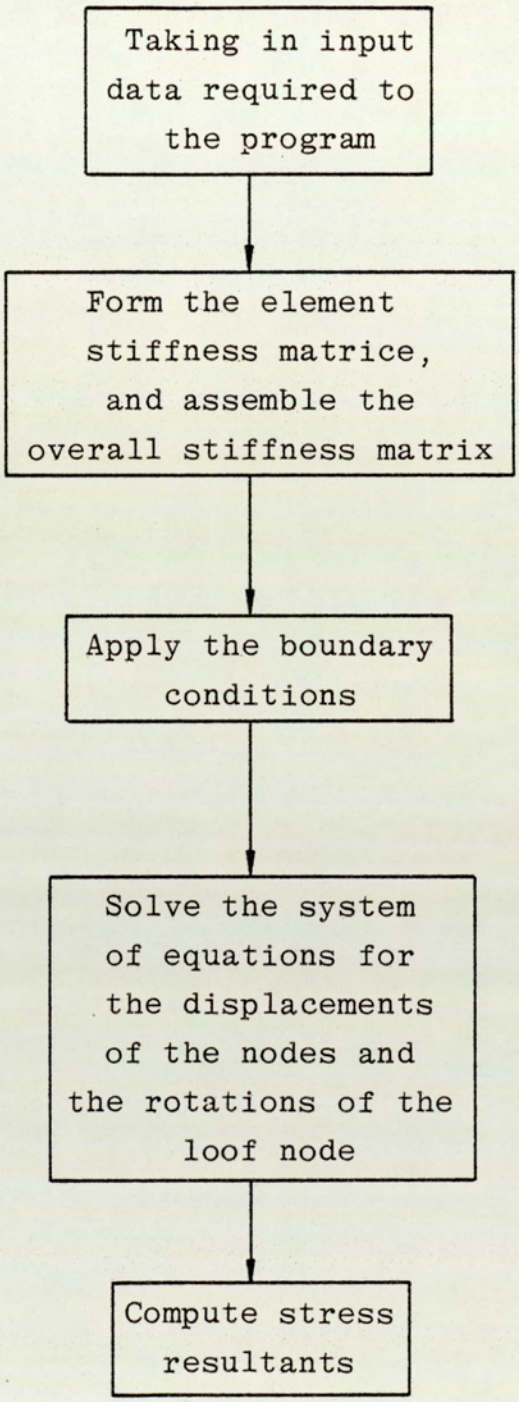


Fig. 4.7 Block diagram of the program

4.3.2.1 SMILOF

This is the master program through which the other subprograms are linked. This is done by using the LINK statement available in the HP9845B computer. The appropriate subroutine is called to perform a specific calculation or formulation, and the control variables are read via this program. With reference to the flow chart, shown in Fig. 4.8, the main steps in program SMILOF are as follows:

1. Dimension of arrays. Some arrays and variables are declared as integers in order to save some memory space. Two classes of input data are read: type of element (Qort) which is either zero for triangular or one for quadrilateral and the number of jobs (Njob) to be solved.

2. A loop is set for the number of jobs to be solved. Mesh details of each job are read, where:

Nelemt = Number of elements
Nnode = Number of nodes
Nsetfs = Number of set of forces
Princ = Input 1 if principal stresses are required; otherwise input 0
Nskw = Number of skewed nodes
Nmat = Number of materials
Prnt = 1,2,3 depending whether the element or nodal strains and stresses are required.

Ch1 = Gravity effect (weight/unit area)

Ch2 = Normal distributed load

Omg = Rotational speed (r.p.m.)

The rest of the input data are read through the subroutine "Feinput".

Two files are created at this stage (using CREATE statement - see Appendix A). N$ file is used to store the following data for each integrating point except the central point of the element;

(i) [B]matrix (strain-element nodal displacement relationship).

(ii) Thik (interpolating thickness at the current integrating point).

(iii) [Ro] array (direction cosine of the local axes).

(iv) {poin} array (x,y and z coordinate of the current integrating point).

While E$ file is used to store the data mentioned above for the element centroids. This is required in view of the complicated formulation of the [B] matrix for the semiloof shell element and takes significant computing time. These data are stored at the stage of formulating the element matrices and retreived at the stage of calculating the strains and stresses instead

of formulating them again.

3. A loop is set for the number of materials, for each material number, the elastic properties are read via the subroutine "constrel". Furthermore, the elasticity matrix [C] is formed.

4. If skewed conditions are found, the required data for each skewed node are read. This involves the reading of its number and the angle of skew.

5. Element stiffness matrices are calculated and the overall stiffness matrix is assembled. Within the subroutine "Assembly", the following subroutines are called:

(a) "Haloof"; in order to form the strain-element nodal displacements relationship for every integrating point in each element.

(b) "Skewedcon": at this subroutine, the application of skewed conditions is considered for each element before the assembly of the overall stiffness matrix.

6. The external loads are applied and transferred to the overall load vector {Q}.

7. The subroutine "skewload" is called to modify the overall load vector for any skewed displacements.

8. Boundary conditions are applied using the subroutine "Geombc", while the equilibrium equations are solved using the "Symvbsol" solving routine.

9. "Skewload" is recalled in order to convert skewed displacements back to the global coordinate system. Subsequently, the nodal displacements in global directions and the loof rotations are printed.

10. Element and/or nodal strains and stresses are calculated if required.

Steps 2-10 are repeated for the next job.

4.3.2.2 Subprogram SMINPT

This contains two subroutines, namely "Feinput" and "Addarray".

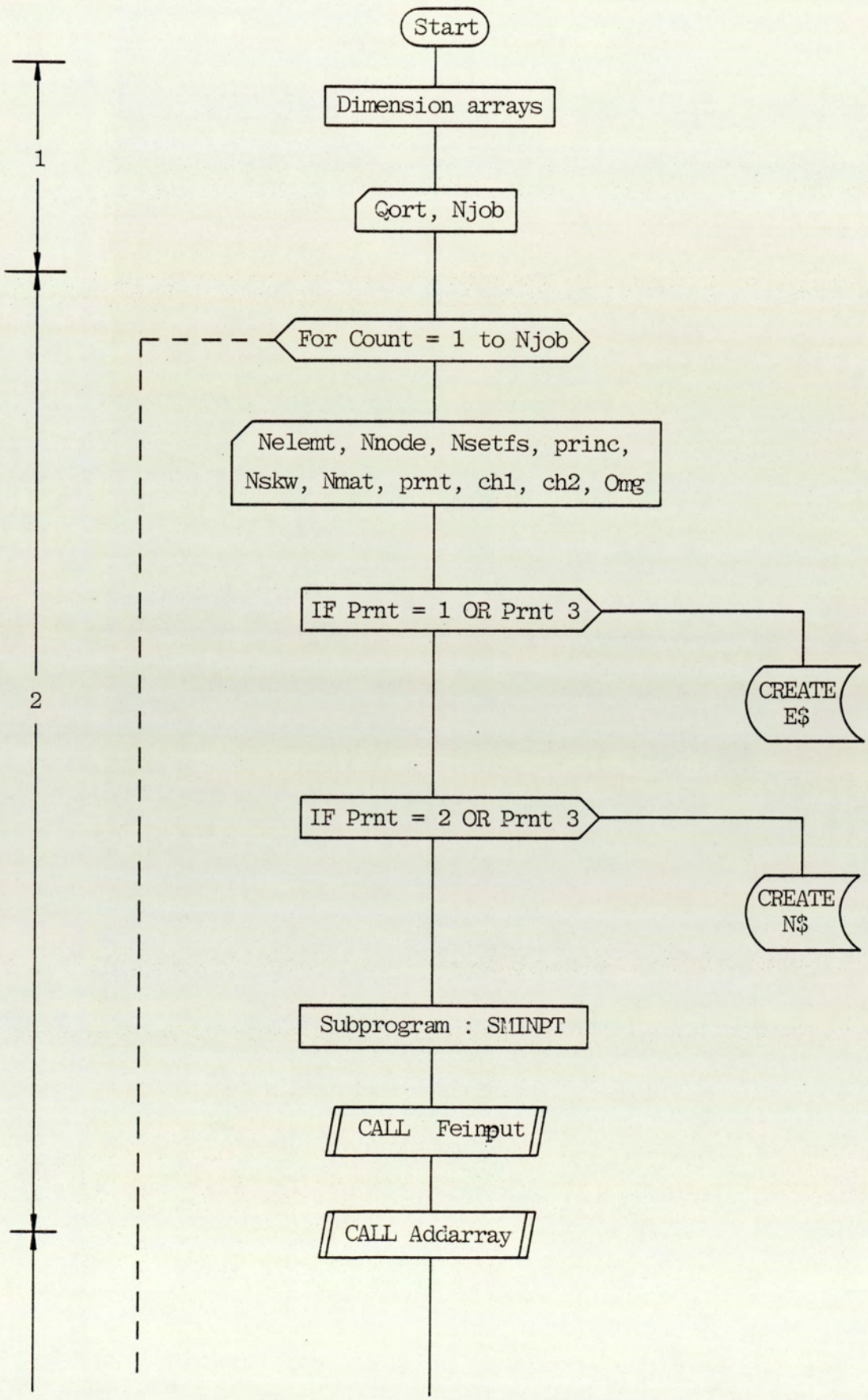
(a) "Feinput"

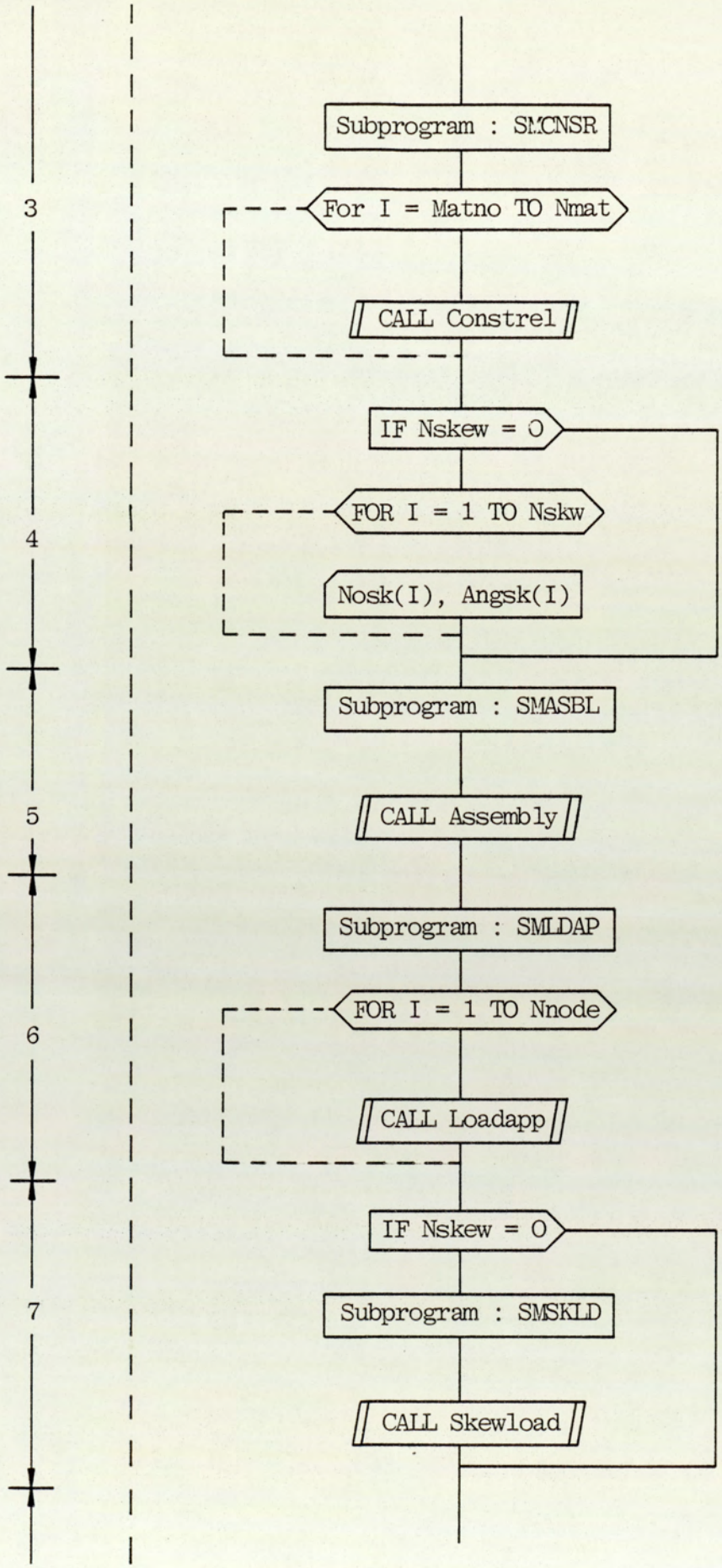
This subroutine is written in order to read and print the input data necessary to specify the mesh. Such data includes the nodal coordinates and the element nodal connections with reference to the flow chart, shown in Fig. 4.9, the various steps involved are:

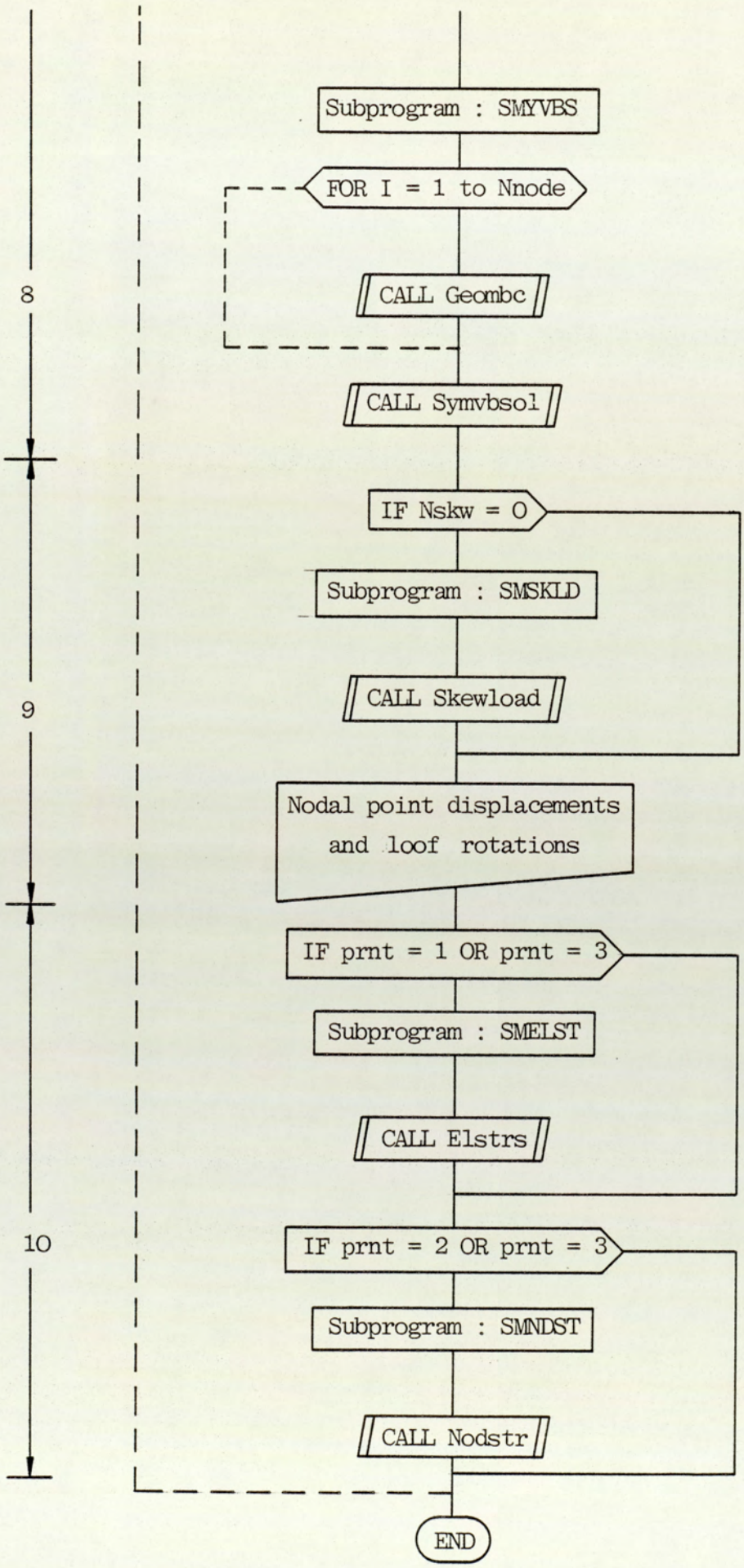
1. Initialisation of various arrays used later in the subroutine.

2. The reading of the nodal coordinates.

Fig. 4.8 Flow Chart of Program SMILOF




Subprogram : SMCNSR
For I = Matno TO Nmat
CALL Constrel
IF Nskew = 0
FOR I = 1 TO Nskw
Nosk(I), Angsk(I)
Subprogram : SMASBL
CALL Assembly
Subprogram : SMLDAP
FOR I = 1 TO Nnode
CALL Loadapp
IF Nskew = 0
Subprogram : SMSKLD
CALL Skewload
3
4
5
6
7


8
9
10
Subprogram : SMYVBS
FOR I = 1 to Nnode
CALL Geombc
CALL Symvbsol
IF Nskw = 0
Subprogram : SMSKLD
CALL Skewload
Nodal point displacements and loof rotations
IF prnt = 1 OR prnt = 3
Subprogram : SMELST
CALL Elstrs
IF prnt = 2 OR prnt = 3
Subprogram : SMNDST
CALL Nodstr
END

3. The reading of element nodal connections and material number for each element.

4. The reading of the thickness for each element.

5. Formulation of the array {Test}. This array is introduced in a size equal to the number of nodes in the discretised structure. A loop is set for the number of nodes. Each element in the {Test} array is set 1 for each corner node and 0 for each midside node.

6. Degrees of freedom are calculated on the basis that each corner node has 3 variable displacements (u,v and w) in the global directions (x,y and z respectively) and each midside node has 5 variable displacements (namely, 3 displacements u, v and w in the global directions x,y and z respectively, and 2 rotations θ^1_{XZ} and θ^2_{XZ} at the loof nodes on both sides of each midside node).

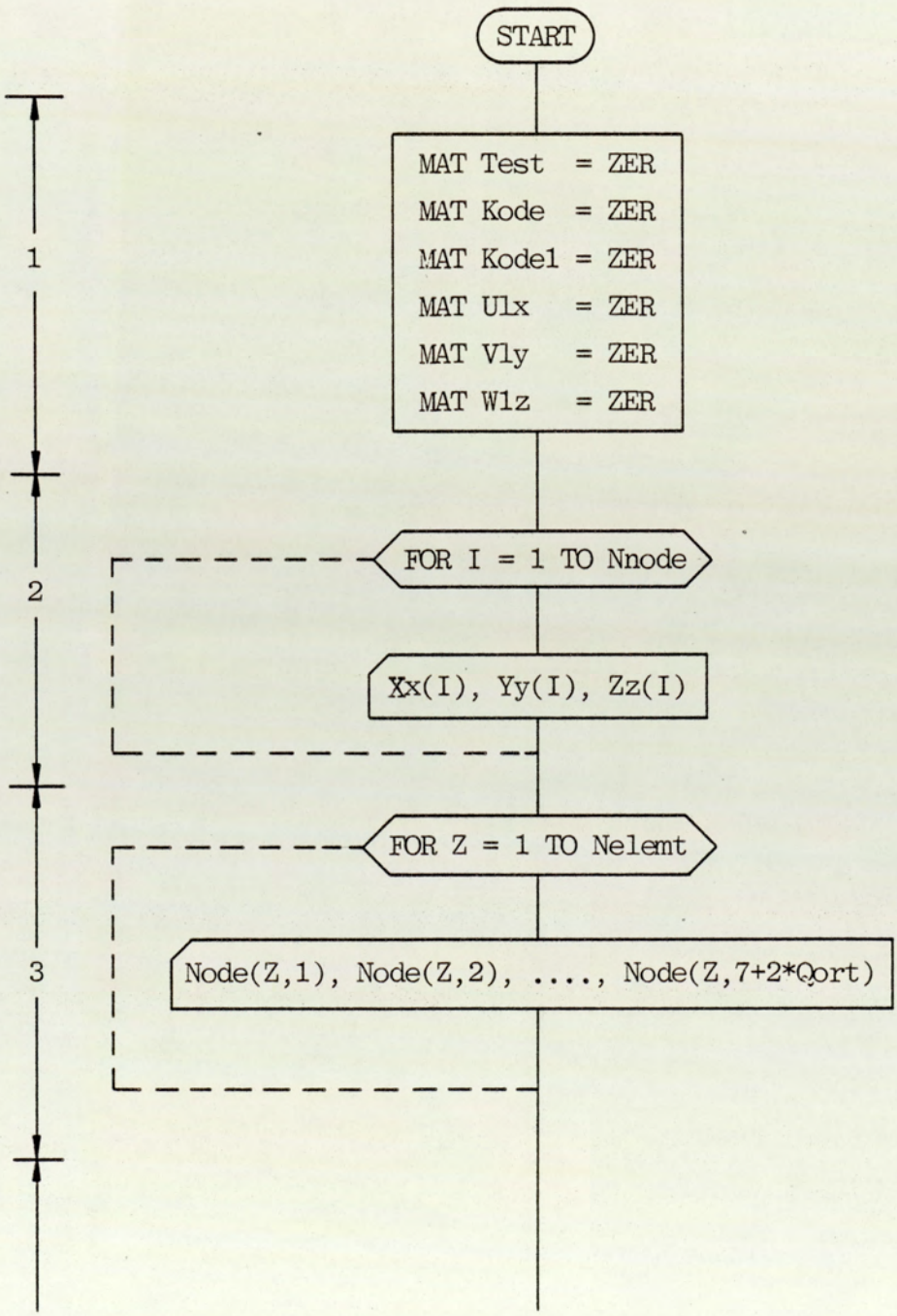
7. Specification of the boundary conditions of the corner nodes and midside nodes, i.e. specifying their code (see Appendix A). This does not include the specifications of the rotations at the loof nodes.

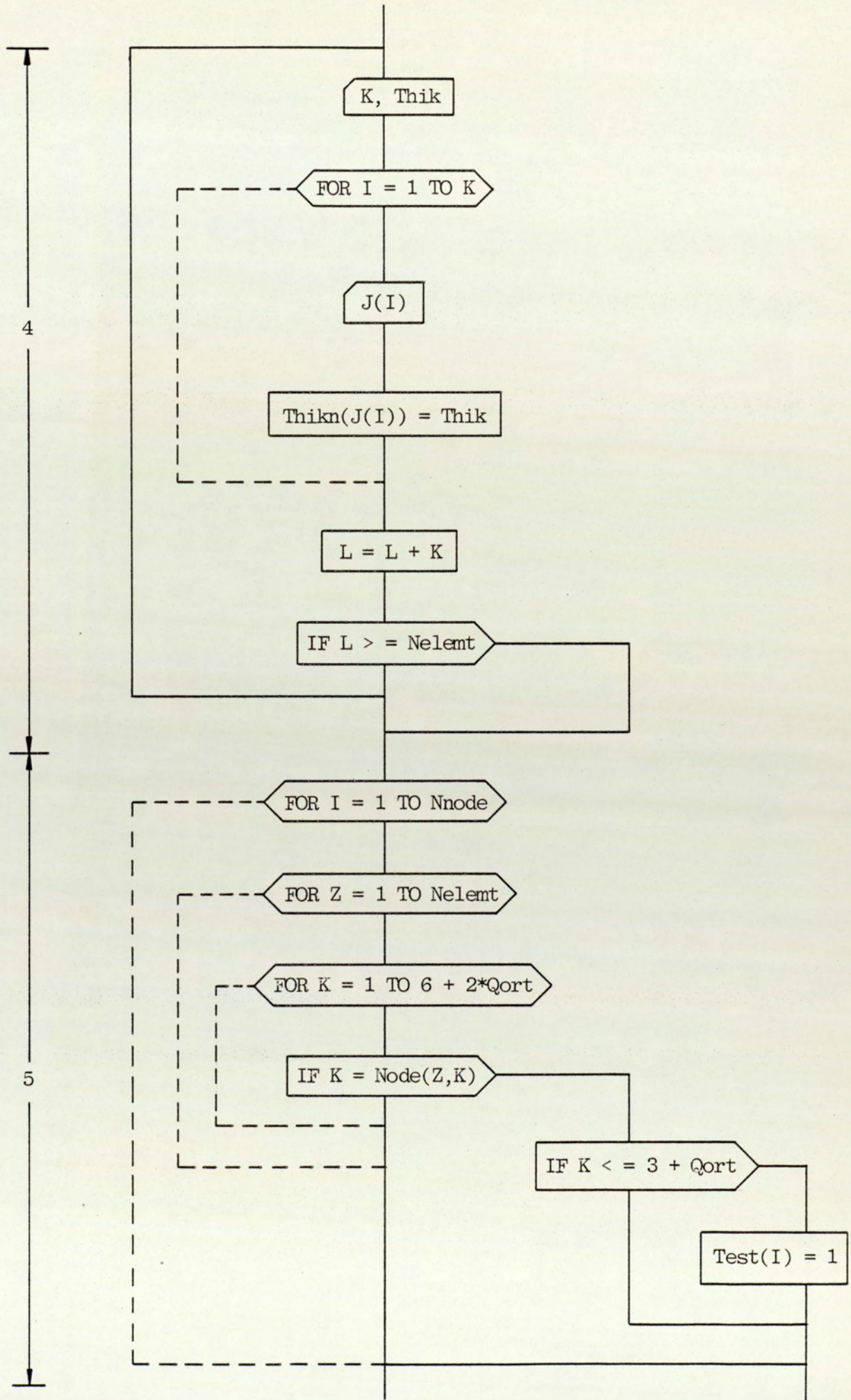
8. Description of the rotations at the loof nodes: this is to specify the rotations on both sides of each midside node (see Appendix A).

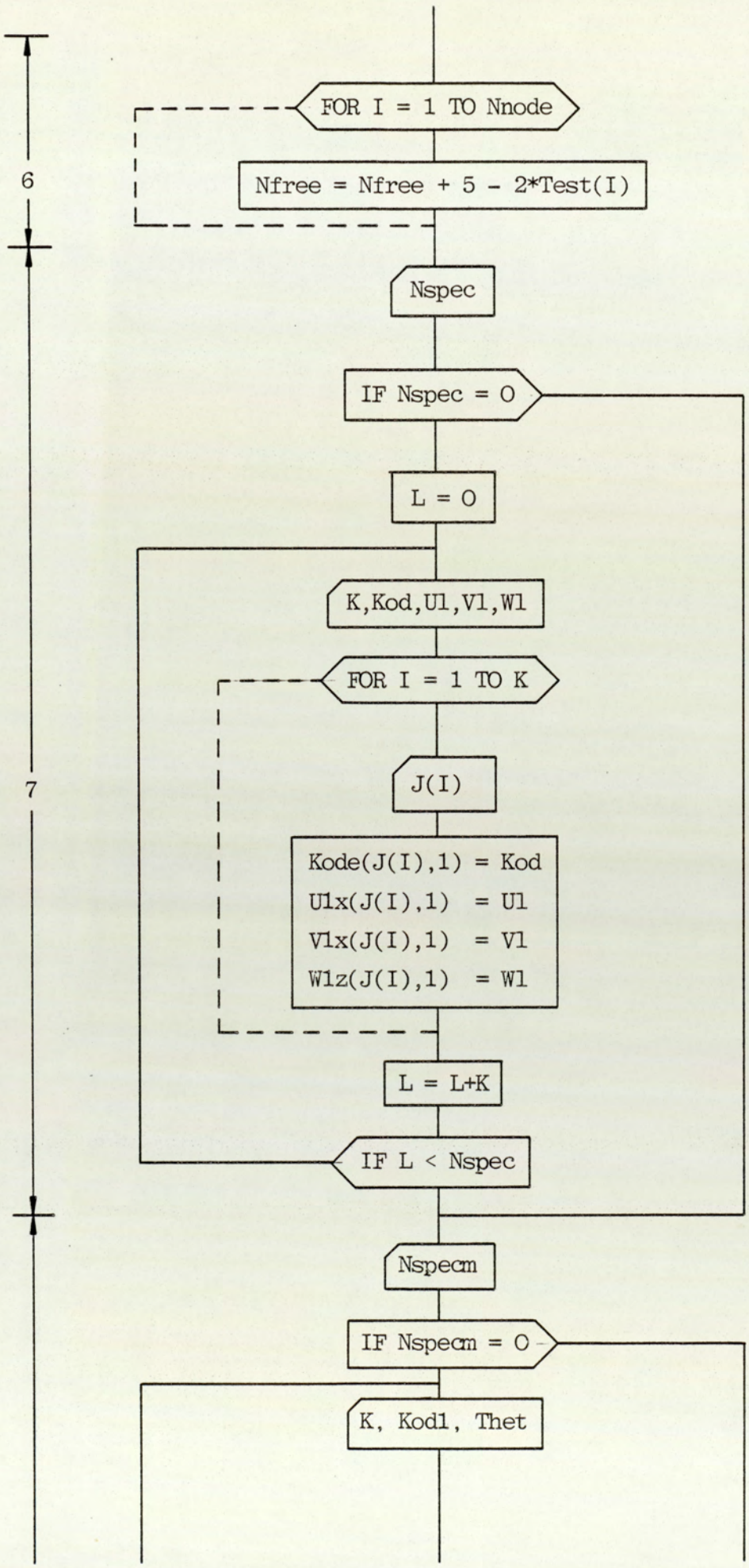
9. Nodal coordinates and element nodal connections are output.

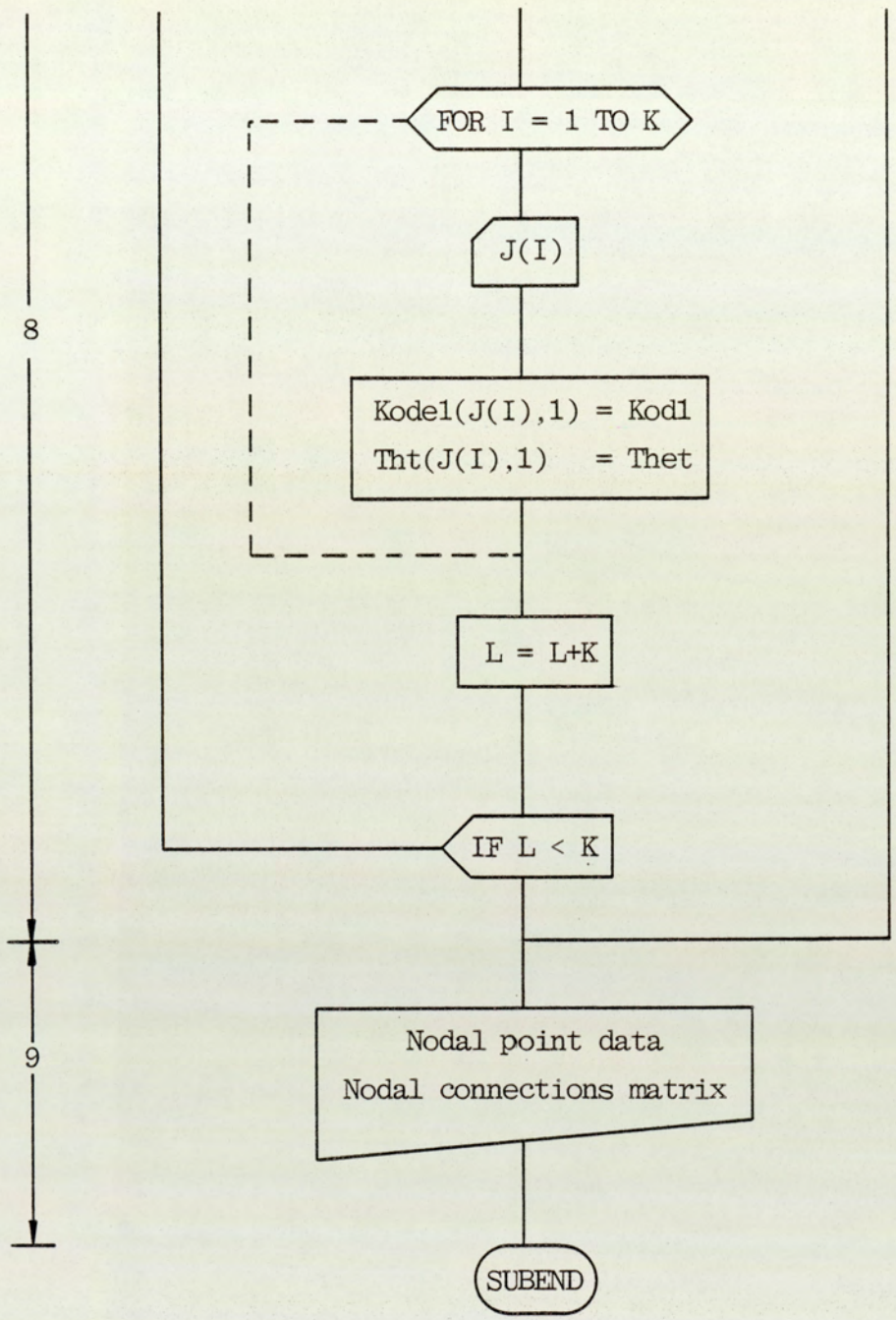
Fig. 4.9 Flowchart of subroutine "Feinput"

SUB Feinput (Xx(*), Yy(*), Zz(*), U1x(*), V1y(*), W1z(*), Tht(*), Thikn(*)
INTEGER Kode(*), Kode1(*), Test(*), Node(*), Nfree, Nnode, Nelemt, Qort)




4
5
K, Thik
FOR I = 1 TO K
J(I)
Thikn(J(I)) = Thik
L = L + K
IF L > = Nelemt
FOR I = 1 TO Nnode
FOR Z = 1 TO Nelemt
FOR K = 1 TO 6 + 2*Qort
IF K = Node(Z,K)
IF K < = 3 + Qort
Test(I) = 1


6
7
FOR I = 1 TO Nnode
Nfree = Nfree + 5 - 2*Test(I)
Nspec
IF Nspec = 0
L = 0
K,Kod,U1,V1,W1
FOR I = 1 TO K
J(I)
Kode(J(I),1) = Kod
U1x(J(I),1) = U1
V1x(J(I),1) = V1
W1z(J(I),1) = W1
L = L+K
IF L < Nspec
Nspecm
IF Nspecm = 0
K, Kod1, Thet


8
9
FOR I = 1 TO K
J(I)
Kode1(J(I),1) = Kod1
Tht(J(I),1) = Thet
L = L+K
IF L < K
Nodal point data
Nodal connections matrix
SUBEND

(b) "Addarray"

This subroutine forms the address sequence for the 1-D stiffness array. It is formed from the input data. This is the next subroutine to be accessed. Each coefficient in the sequence is equivalent to the sum of all previous coefficients in the array, plus the number of terms between the first non-zero term and the leading diagonal of the stiffness matrix. The address sequence locates the position of the diagonal elements within the one-dimensional representative of the [K] array. This scheme, as mentioned before, has been developed by Ref.(63).

If [K] is considered (in Fig. 4.10), only either the lower or the upper triangle of [K] is stored by rows in a line array. Only those elements from the first non-zero elements to the diagonal elements are stored for each row. The address sequence is used to locate the position of the diagonal elements in this array (as will be demonstrated). Moreover, the method of storing the matrix and forming the address sequence is dealt with here.

$$[K] = \begin{bmatrix} 10 & & & & \\ 3 & 6 & & & \\ 4 & 2 & 8 & & \\ 0 & 0 & 0 & 11 & \\ 0 & 0 & 5 & 2 & 7 \end{bmatrix}$$

Fig. 4.10 Address Array Sequence

According to this technique, only the enclosed area of Fig. 4.10 is stored as follows:

K[10, 3, 6, 4, 2, 8, 11, 5, 2, 7]

The address sequence locating the position of the diagonal elements in this array is as follows:

Add(1, 3, 6, 7, 10)

The coefficients of the Add array are determined before the assembly of the overall stiffness matrix starts. This is done so that the stiffness coefficients can be located directly in their proper positions in the one-dimensional stiffness array. For this reason the "Addarray" subroutine is called before the "Assembly" subroutine.

In the example of Fig. 4.11, 4 quadrilateral semiloof shell elements are used to represent a quarter of a thin rotating disc.

The form of the overall stiffness matrix is shown in Fig. 4.12 and is determined by the assembly rules, where nodes connected through a common element have a connection in the overall stiffness matrix.

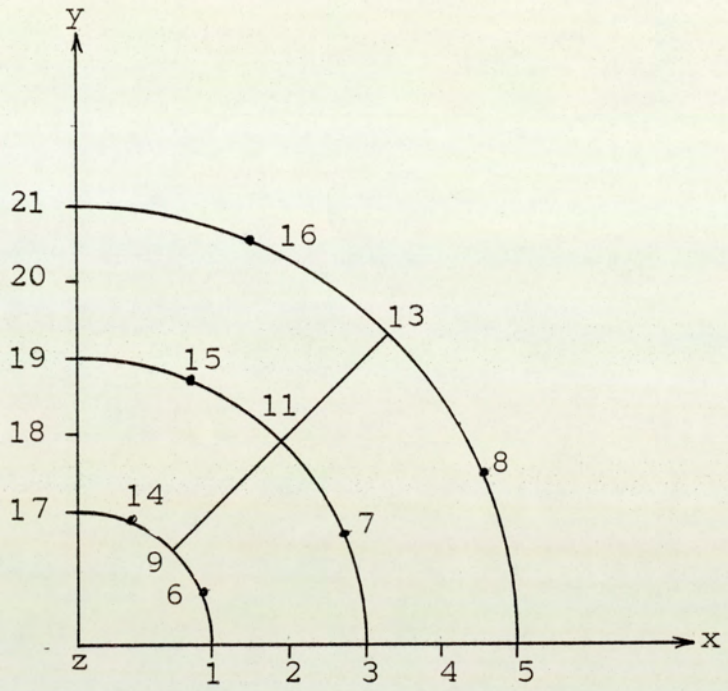


Fig. 4.11 (2×2) Mesh (quadrilateral semiloof shell element)

With reference to the flow chart shown in Fig. 4.13, the steps involved are as follows:

1. The Add array coefficients for the first three rows in the overall 2D stiffness matrix are set; first node is assumed as a corner node. Initialisation of two integers Addtemp and Itemp.

Addtemp = is the Add array coefficient for the last row number in the overall 2D stiffness matrix.

Itemp = is the last row number in the overall 2D stiffness matrix plus one.

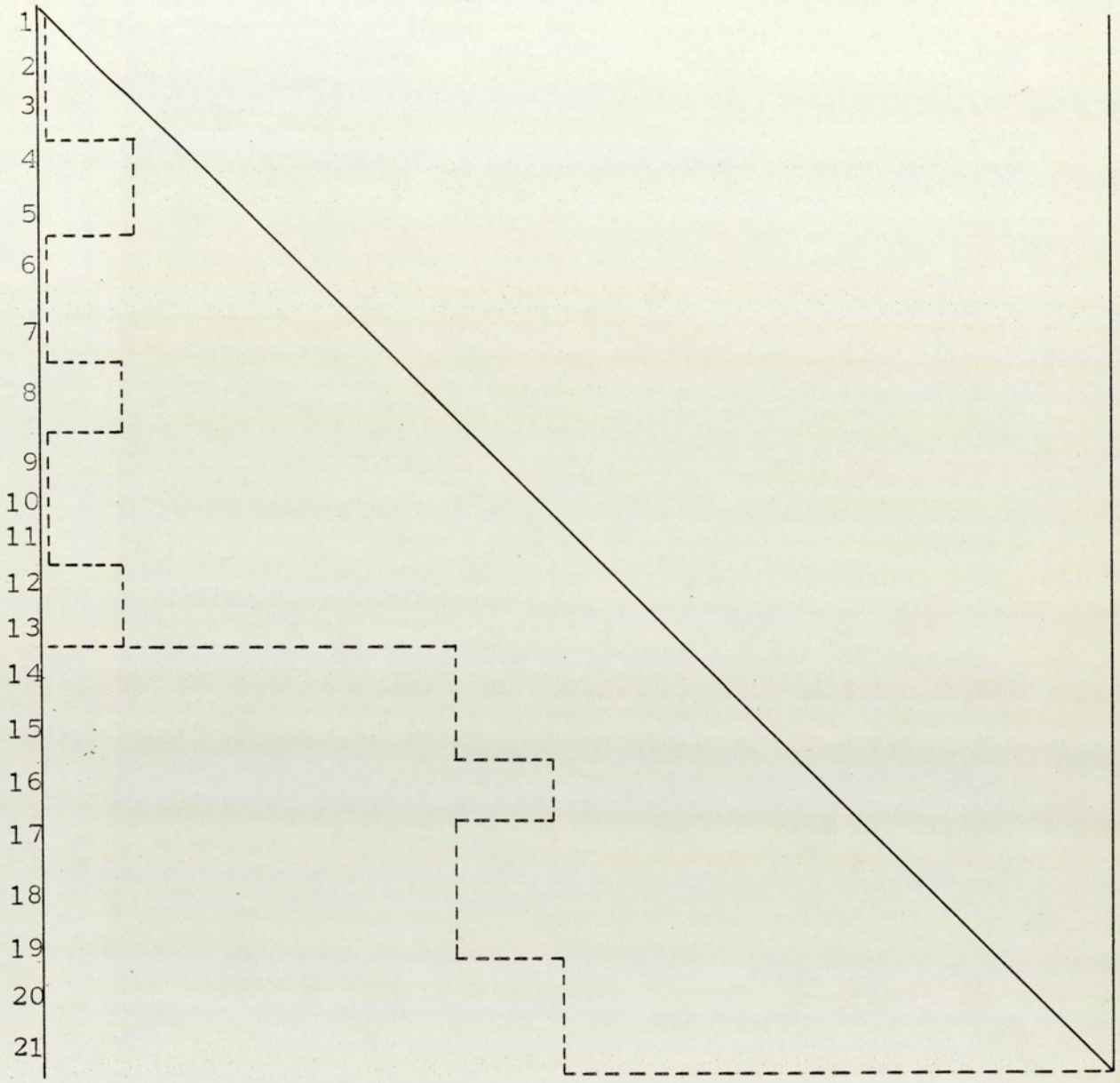


Fig.4.12 Schematic representation of overall stiffness matrix

2. A loop is set to the number of nodes. For the second node, each element is searched as to whether it contains this node. The elements sharing this node are stored in an array [A].

3. This step comprises scanning the nodal connections of the first element found in step 2 which contains this node in order to find the lowest node number connected to the current node. This is repeated for all the elements of the array [A]. Thus, the smallest node number connected to the current node can be found.

4. The number of columns present in the overall stiffness matrix for the current node is calculated.

5. The address array coefficients are determined for the number of columns present in this node.

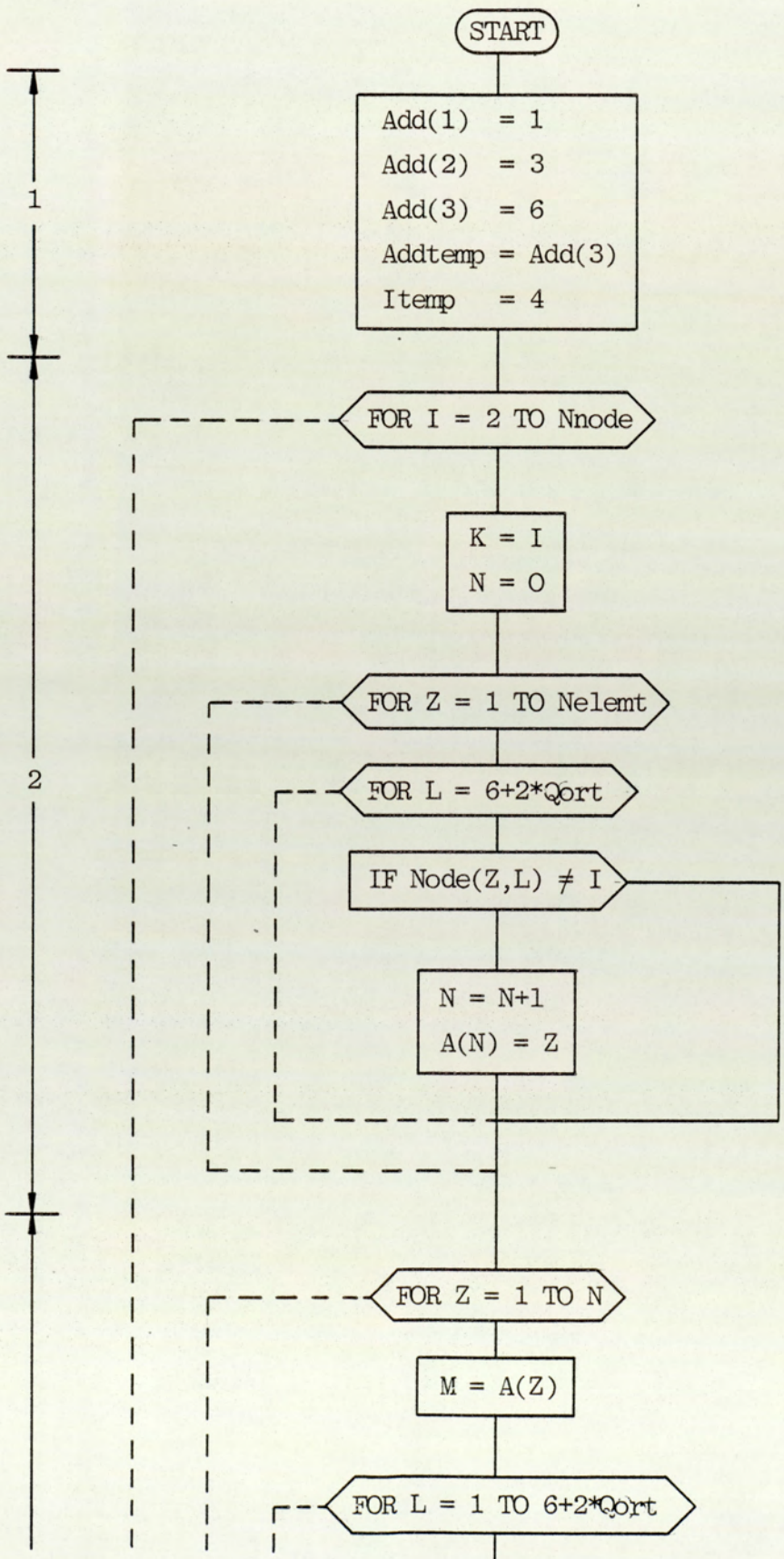
Steps 2-5 are repeated for all the nodes.

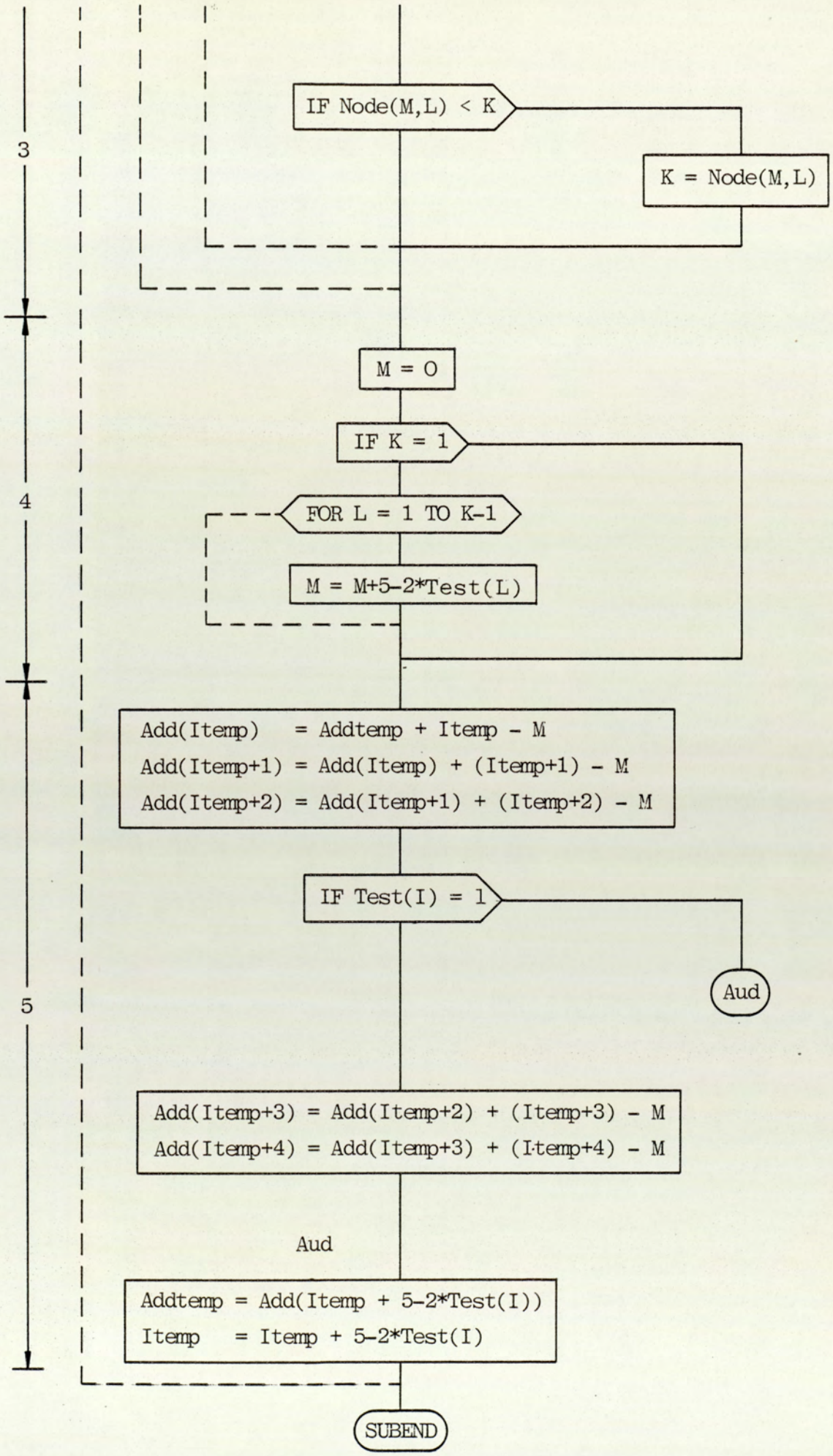
4.3.2.3 Subprogram SMCNTR

The construction of the elasticity matrix [D] is carried out in this subprogram. For this purpose the "Constrel" subroutine was written to generate this matrix , which relates the stresses to the strains. Isotropic material is used in which the two independent elastic constants are read, Modulus of elasticity E, and Poisson's ratio ν. The coefficients of the elasticity

Fig. 4.13 Flow Chart of Subroutine"Addarray"

SUB Addarray (INTEGER Test(*), Nelemt, Nnode, Add(*), Node(*), Qort, Nfree)




3
4
5
IF Node(M,L) < K
K = Node(M,L)
M = 0
IF K = 1
FOR L = 1 TO K-1
M = M+5-2*Test(L)
Add(Itemp) = Addtemp + Itemp - M
Add(Itemp+1) = Add(Itemp) + (Itemp+1) - M
Add(Itemp+2) = Add(Itemp+1) + (Itemp+2) - M
IF Test(I) = 1
Aud
Add(Itemp+3) = Add(Itemp+2) + (Itemp+3) - M
Add(Itemp+4) = Add(Itemp+3) + (Itemp+4) - M
Aud
Addtemp = Add(Itemp + 5-2*Test(I))
Itemp = Itemp + 5-2*Test(I)
SUBEND

matrix [D] according to equation (4.41) was programmed. The flow chart and the various steps involved in the subroutine are as follows: (see Fig. 4.14):

1. The material number and its elastic constants are read.

2. Print headings, data of elastic constants and the density of the material on the line printer which has been chosen.

3. Certain variables are declared and defined in terms of the elastic constants.

4. The coefficients of the elasticity matrix are stored in the one dimensional array [Z].

4.3.2.4 Subprogram SMASBL

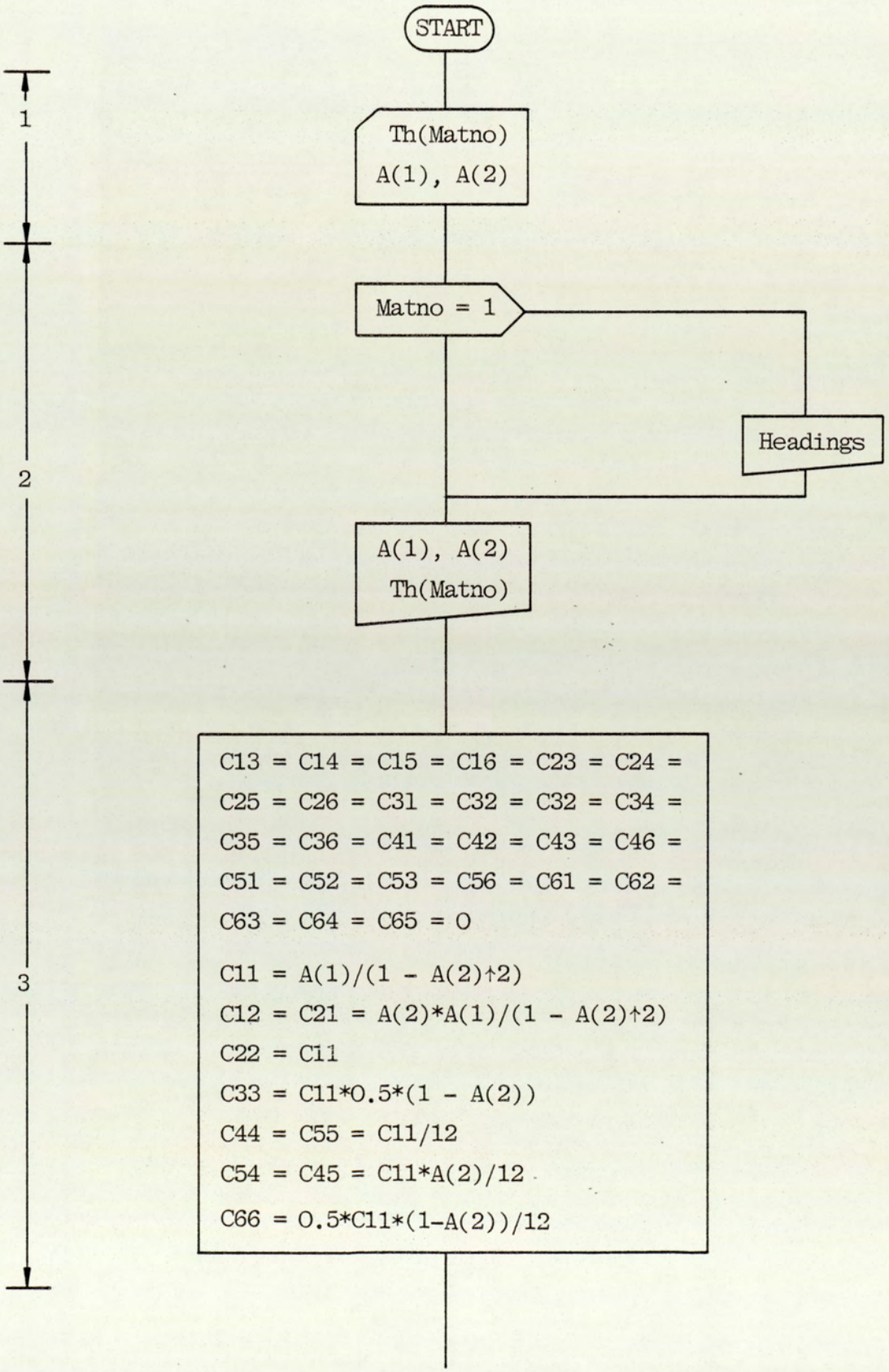
This subprogram contains three subroutines, namely "Assembly", "Haloof" and "Skewedcon".

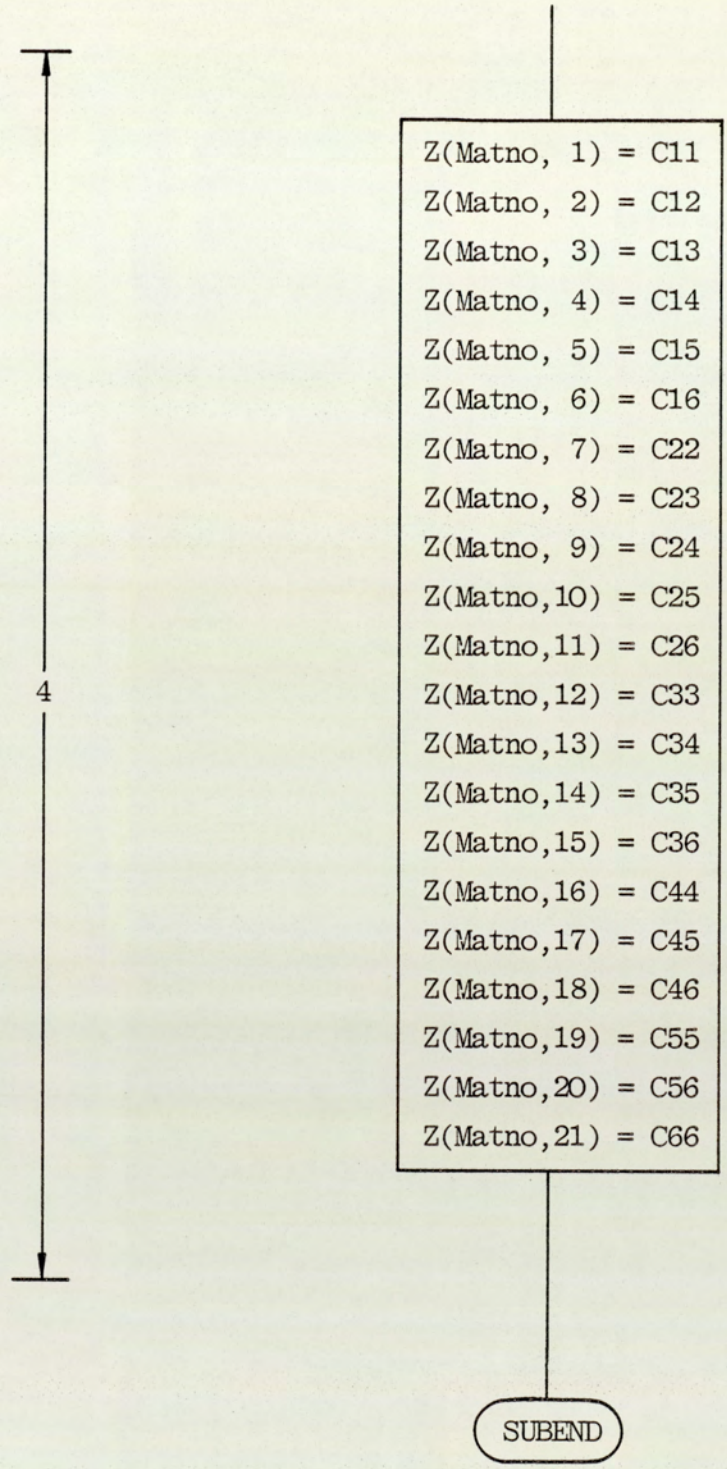
(a) "Assembly"

In this respect, the discussion will cover the individual element stiffness matrices and their assembly into the overall one-dimensional stiffness array. Having the strain-element nodal displacements relationship [B] constructed, the equation (4.40) is integrated numerically

Fig. 4.14 Flow Chart of Subroutine"Constrel "

SUB Constrel (Z(*), Th(*), INTEGER Matno)




4
Z(Matno, 1) = C11
Z(Matno, 2) = C12
Z(Matno, 3) = C13
Z(Matno, 4) = C14
Z(Matno, 5) = C15
Z(Matno, 6) = C16
Z(Matno, 7) = C22
Z(Matno, 8) = C23
Z(Matno, 9) = C24
Z(Matno,10) = C25
Z(Matno,11) = C26
Z(Matno,12) = C33
Z(Matno,13) = C34
Z(Matno,14) = C35
Z(Matno,15) = C36
Z(Matno,16) = C44
Z(Matno,17) = C45
Z(Matno,18) = C46
Z(Matno,19) = C55
Z(Matno,20) = C56
Z(Matno,21) = C66
SUBEND

to obtain the element stiffness matrix. The integrations are carried out using the five point Gauss integration rule (described in equation (4.43) for the quadrilateral semiloof shell element) or the three point Gauss rule for equation (4.40) appropriate to the triangular element.

The overall stiffness matrix [K] is obtained by considering each element in turn and locating the coefficients of the element stiffness matrix in their correct positions in the overall stiffness matrix as shown in Fig. (4.12), for which the address sequence, discussed in Section 4.3.2.2, has been used.

The process is repeated for all the elements, summing contributions from each one into the overall system equations.

(b) "Haloof"

This subroutine is called by the main subroutine "Assembly". No flow chart is included as the subroutine was taken as a package from Ref. (64). It has been transferred from FORTRAN language to suit the BASIC of the HP9845B computer. This subroutine calls for each integrating point at the element of interest in order to form [B] matrix. The input to this subroutine comprises:

(i) [Elxyzt] : this is an array of (7+2*Qort,4) size. It is constructed from the point nodal coordinates and the thickness of each node in the element as follows:

$$\{Elxyzt\} = \begin{bmatrix} X_1 & Y_1 & Z_1 & t_1 \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ X_{7+2*Qort} & Y_{7+2*Qort} & Z_{7+2*Qort} & t_{7+2*Qort} \end{bmatrix}$$

where X_1, Y_1, Z_1 and t_1 etc. are the x,y,z coordinates and the thickness of node number 1 etc.

The $(7+2*Qort)^{th}$ node is the central node. Its coordinates and the thickness are found by interpolation.

$t_1 = t_2 = \ldots = t_{6+2*Qort}$ = thickness of the current element (on the assumption that the element thickness is constant).

(ii) Element nodal connections.

(iii) The natural coordinates ξ and η of the current integrating point.

The output of this subroutine comprises:

(a) poin(3) : The interpolated x,y,z of the current integrating point.

(b) Thik : the interpolated local thickness of the current integrating point.

(c) Area: the integrated area over the face of the element.

(d) Ro(3,3): whose columns give unit local axes.

(e) Wshel (13,45): This provides useful shape function output in its 24 or 32 columns for the triangular or quadrilateral element. The following useful rows give:

1.	U	The displacement components in local axes
2.	V	
3.	W	
4.	$\partial U/\partial X$	
5.	$\partial U/\partial Y$	The in-plane deflection derivatives in
6.	$\partial V/\partial X$	the local axes.
7.	$\partial V/\partial Y$	
10.	$\partial^2 U/\partial X \partial Z$	The bending terms.
11.	$\partial^2 U/\partial Y \partial Z$	
12.	$\partial^2 V/\partial Y \partial Z$	
13.	$\partial^2 V/\partial Y \partial Z$	

The numbers refer to the row number in Wshel array.

(c) "Skewedcon"

This subroutine is called if skewed conditions prevail at the nodes of the current element. This is discussed in Section 4.3.2.6.

4.3.2.5 Application of Body Forces and Distributed Loads

During the construction of the subprogram "SMASBL", the subroutines are written in such a way that they are able to take the following types of loading into consideration. First, the body forces per unit volume due to gravity and rotational forces, and secondly, the normal distributed loading.

(a) Body Forces

If B_x, B_y and B_z are the body forces per unit volume along the global axes x,y and z respectively, then the element load vector $\{Felvec\}^e$ is given by:

$$\{Felvec\}^e = \int_V \{Bf\}^t [Ro][N]\, dvol \qquad \ldots\ (4.45)$$

where {Bf} is a (3,1) array and $= \begin{Bmatrix} B_x \\ B_y \\ B_z \end{Bmatrix}$,

[Ro] is (3,3) matrix, this contains the direction cosine of the unit vectors; i.e.

$$[Ro] = \begin{bmatrix} X_x & Y_x & Z_x \\ X_y & Y_y & Z_y \\ X_z & Y_z & Z_z \end{bmatrix} \qquad \ldots\ (4.46)$$

and [N] is a (3,24 + 8*Qort) shape function matrix. Its three rows correspond to the in-plane and out-of-plane

displacements (U, V and W). This is represented in the first three rows of Wshel array.

The {Felvec} is a vector with (24+8*Qort) coefficients. This gives the contribution of the body forces to the nodes of the current element.

(b) Distributed Load

For the distributed load, the shape function corresponding to the displacement normal to the surface is used. {Felvec}is as follows:

$$\{Felvec\}^e = \int_A [N(\xi,\eta)]pdA \qquad \ldots \quad (4.47)$$

where A represents the area of the middle surface and p is a function of natural coordinates. The numerical integration of equations (4.45) and (4.47) is performed simultaneously with the formation of the element stiffness matrices governed by the same rules of numerical integration. After summing for all points of integration for the current element, the $\{Felvec\}^e$ coefficients are transferred to their proper positions in the overall load vector {Q} , as demonstrated in the following discussion of the flow chart of the "Assembly" subroutine.

Summary of the Flow Chart of the "SMASBL" Subprogram

After the subprogram "SMASBL" is linked to the main program SMILOF using the LINK statement, the "Assembly" subroutine is called through the master program "SMILOF". The following flow chart is drawn for the "Assembly" subroutine. It constructs each element stiffness matrix and assembles the overall stiffness as a one-dimensional array using the address sequence. Furthermore, it constructs the element load vector described in Section 4.3.2.5.

With reference to the flow chart shown in Fig. 4.15, the following steps summarise the subroutine "Assembly".

1. The definition of the different arrays and variables. This step also involves an optional ASSIGN of files. The ASSIGN statement sets up or references an existing internal file table and allows the utilisation of data files (with PRINT# and READ# statements). The data files are created in the master program SMILOF. The optional ASSIGN is decided by the control variables (see Appendix A).

2. Coefficients of array W are set to the coordinates of the integrating points (natural coordinates), while the coefficients of array A are set to weighting functions.

3. This is a preparation step to call the "Haloof" subroutine.

This subroutine needs the element nodal connections in the following sequence (see Fig. 4.6(b)): (1 2 3 4 5 6 7 8) whereas in the input data, the following sequence is normally used: (1 3 5 7 2 4 6 8). The INTEGER array {Mod} defined here, is used as a reference for the sequence of the nodes required by the "Haloof" subroutine. For example: the element nodal connections for the current element Z is changed to the sequence required by the "Haloof" as follows:

(Node(Z,Mod(1)), Node(Z,Mod(2)),....,Node(Z,Mod(8)))

i.e. (1 3 5 7 2 4 6 8) as required by "Haloof".

4. A loop is constructed round the number of elements in the mesh. This step involves the initialisation of different arrays and the declaration of some variables in order to call the "Haloof " subroutine. This contains all the input requirements to this subroutine.

5. A loop is constructed on the number of integrating points.

6. The "Haloof" subroutine is called in order to evaluate the coefficients of the strain-element nodal displacement array [B].

7. The matrix multiplications of equation (4.40) are

performed using the coefficients of the elasticity matrix [D] to calculate the element stiffness matrix.

8. This comprises the optional PRINT≠ of the [B] array, Thik , [Ro] array and the interpolated coordinates x,y and z of the current integrating point as required by the control variables. It is a continuation of step 1.

9. The element load vector $\{Felvec\}^e$ is calculated (as explained in Section 4.3.2.5) if the body forces and/or normal distribution are applied.

Steps 5-9 are repeated for the rest points of integration.

10. The subroutine "Skewedcon" is called if skewed conditions prevail at the node(s) of the current element. If skewed node constants are found, the element stiffness matrix is transformed to the skewed directions before starting the assembly of the system stiffness matrix [K]. This will be illustrated later.

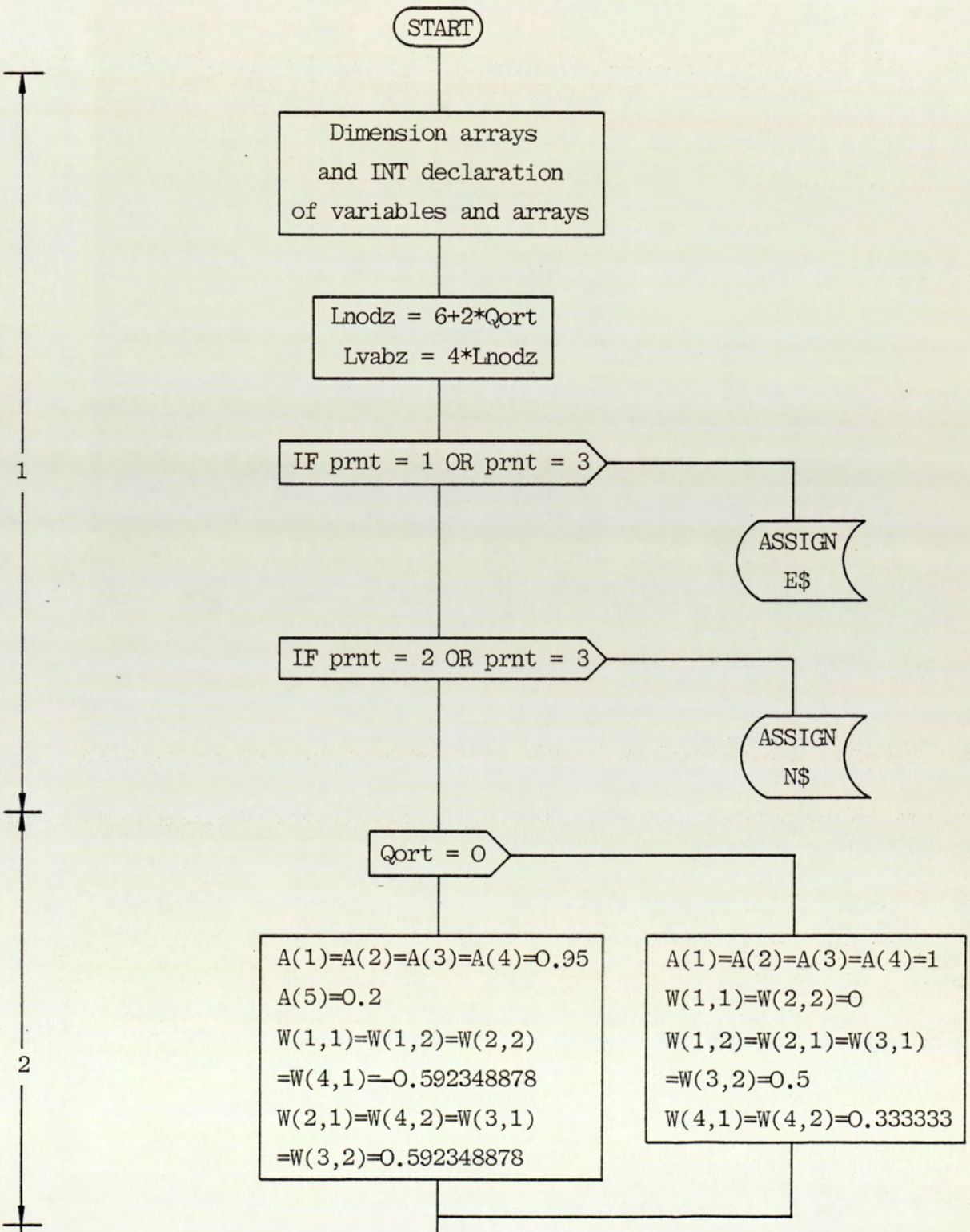
11. The element stiffness coefficients are located in their appropriate positions in the overall stiffness matrix as a one-dimensional array using the element nodal connections matrix. This stage has been programmed to accept the element nodal connections matrix sequence as it is in the "Haloof" subroutine (see Step 3).

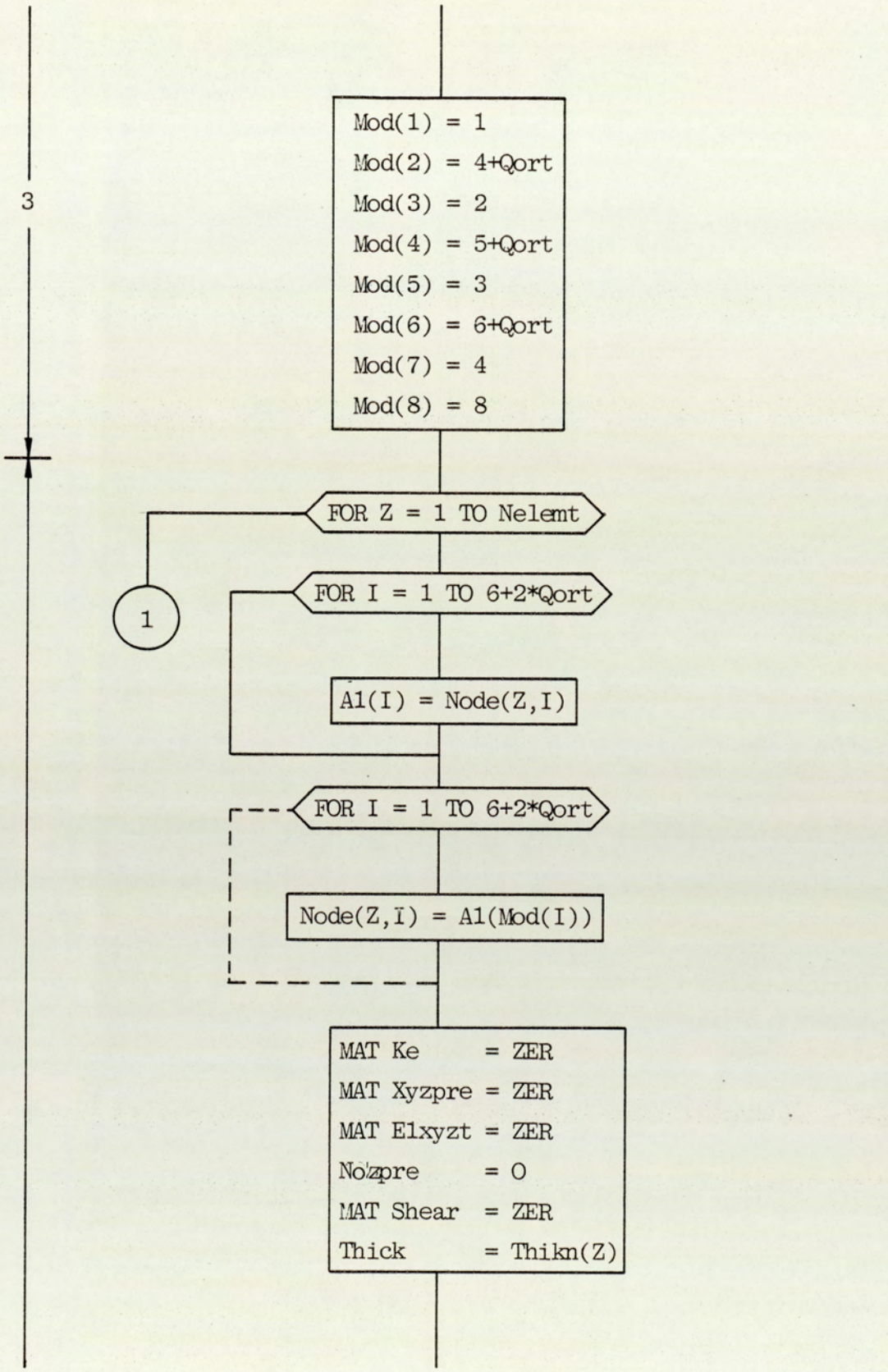
12. The element load vector $\{Felvec\}^e$ coefficients are transferred into the system load vector {Q} in their proper positions.

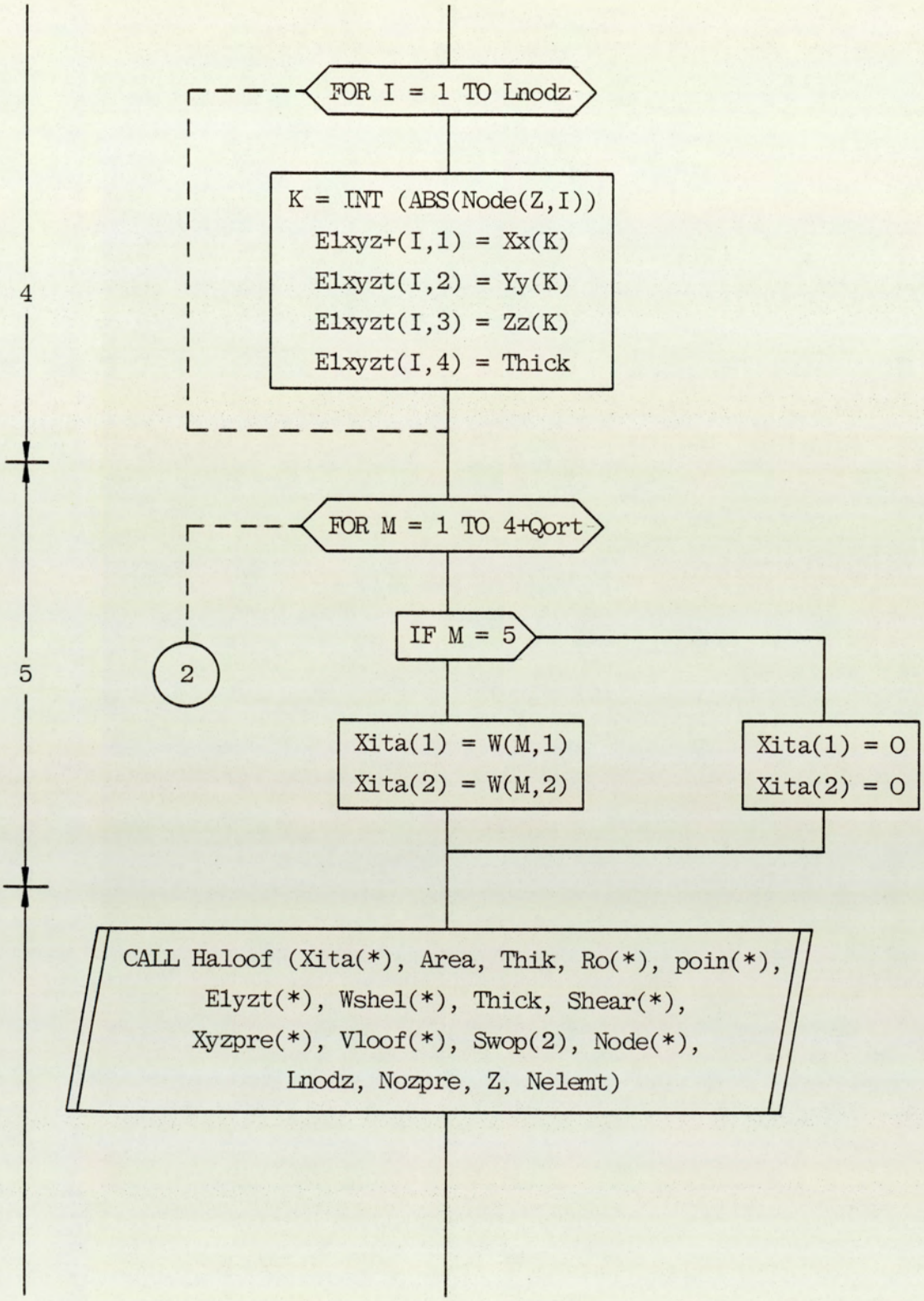
Steps 4-12 are repeated for the other elements of the mesh.

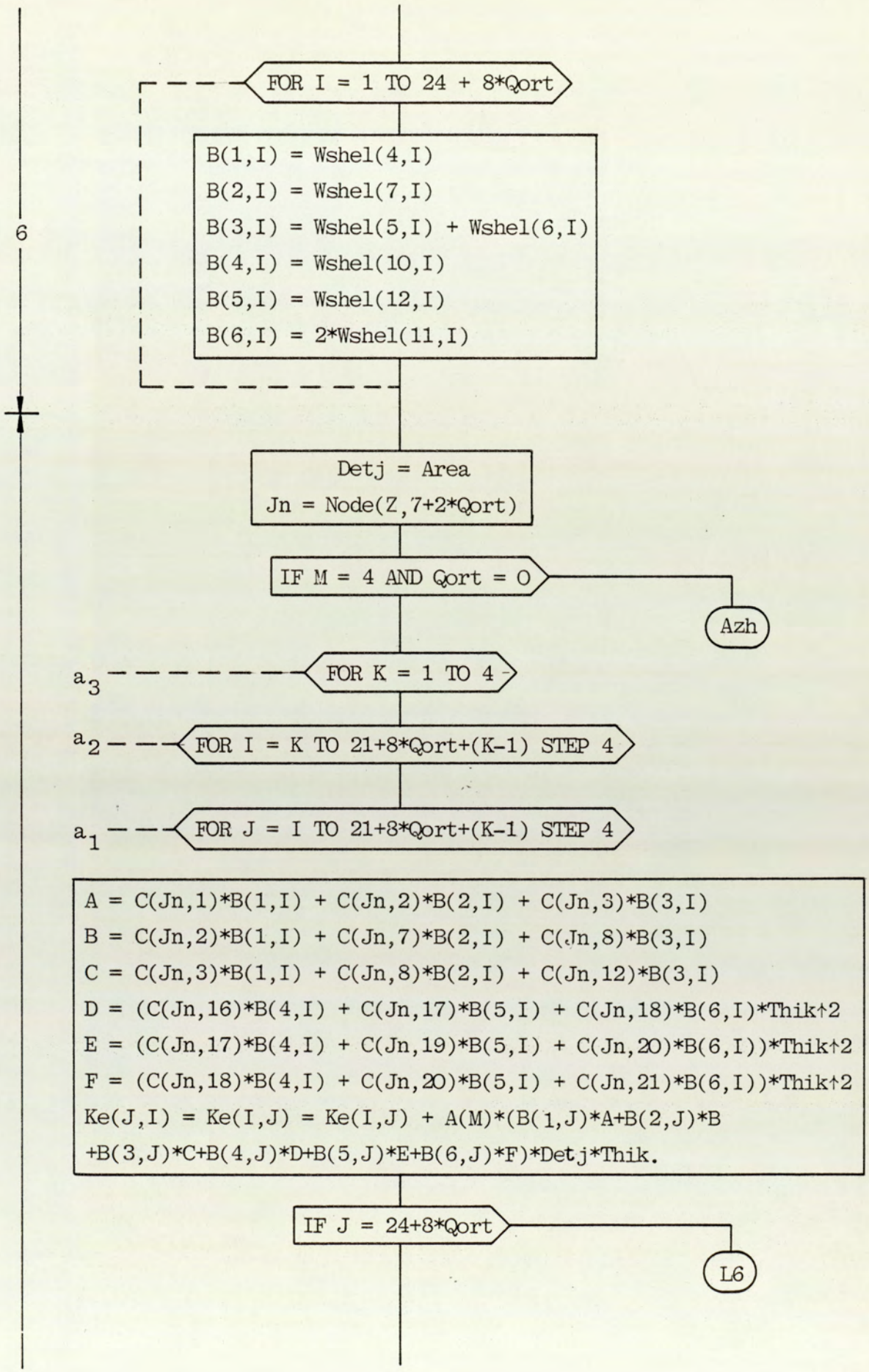
Fig. 4.15 Flow Chart of Subroutine"Assembly "

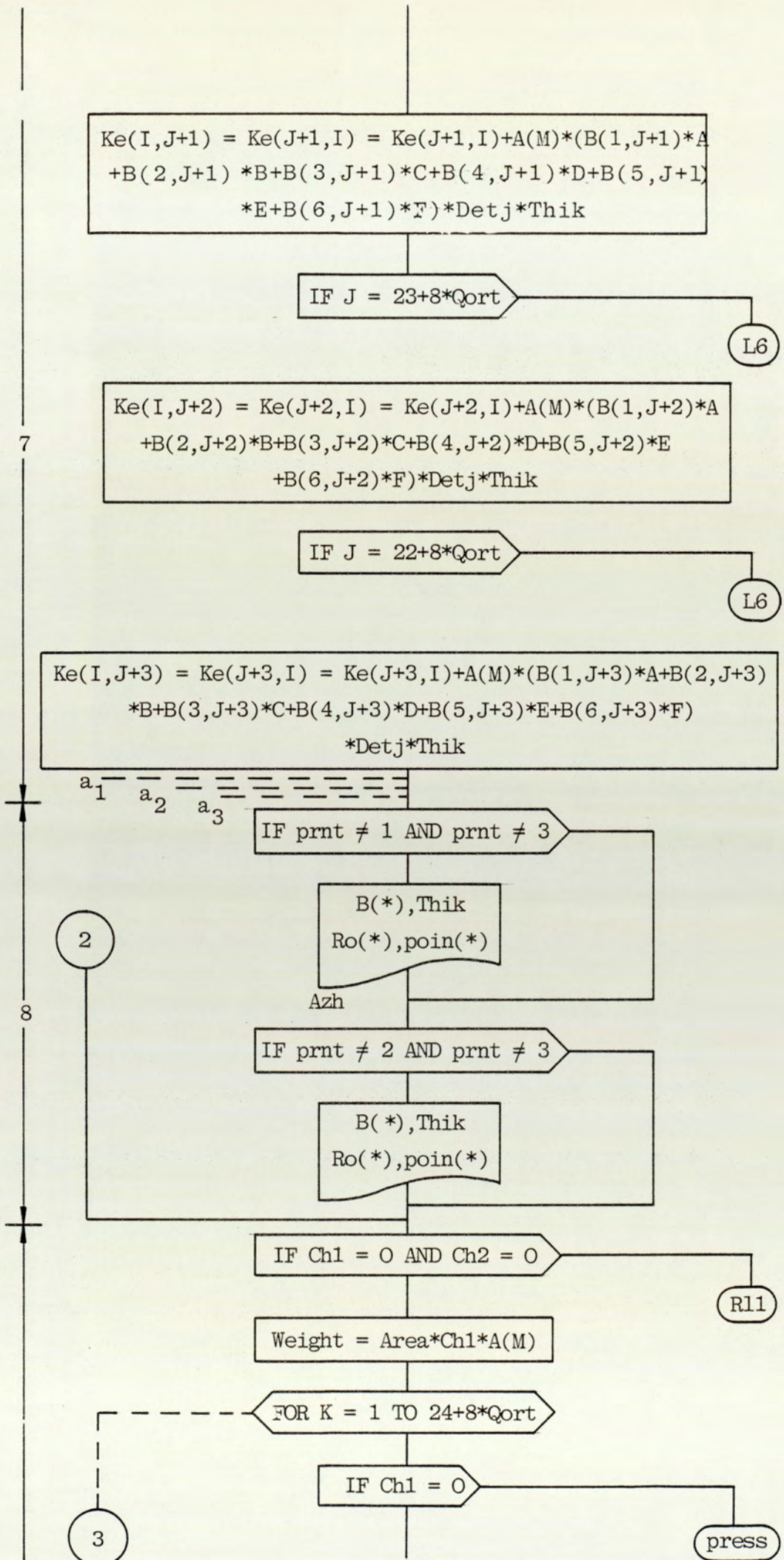
SUB Assembly (Q(*), K(*), Xx(*), Yy(*), Zz(*), B(*), Th(*), Omg, Ch1, Ch2, Thikn(*), Angsk (*), N$, E$, INTEGER Node(*), Test(*), Nelemt, Add(*), Nfree, Nskew, Nosk(*), Qort, prnt).

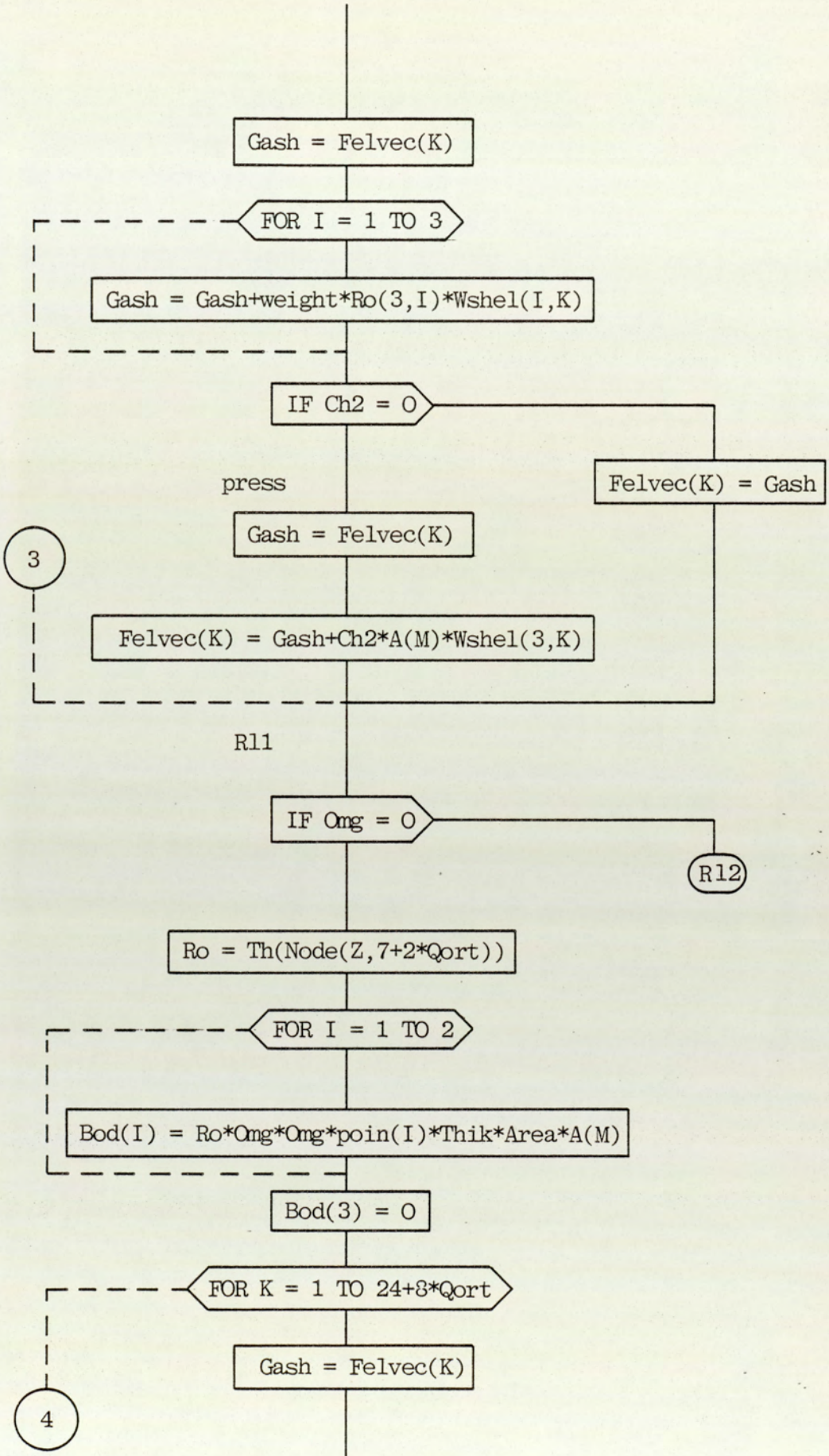


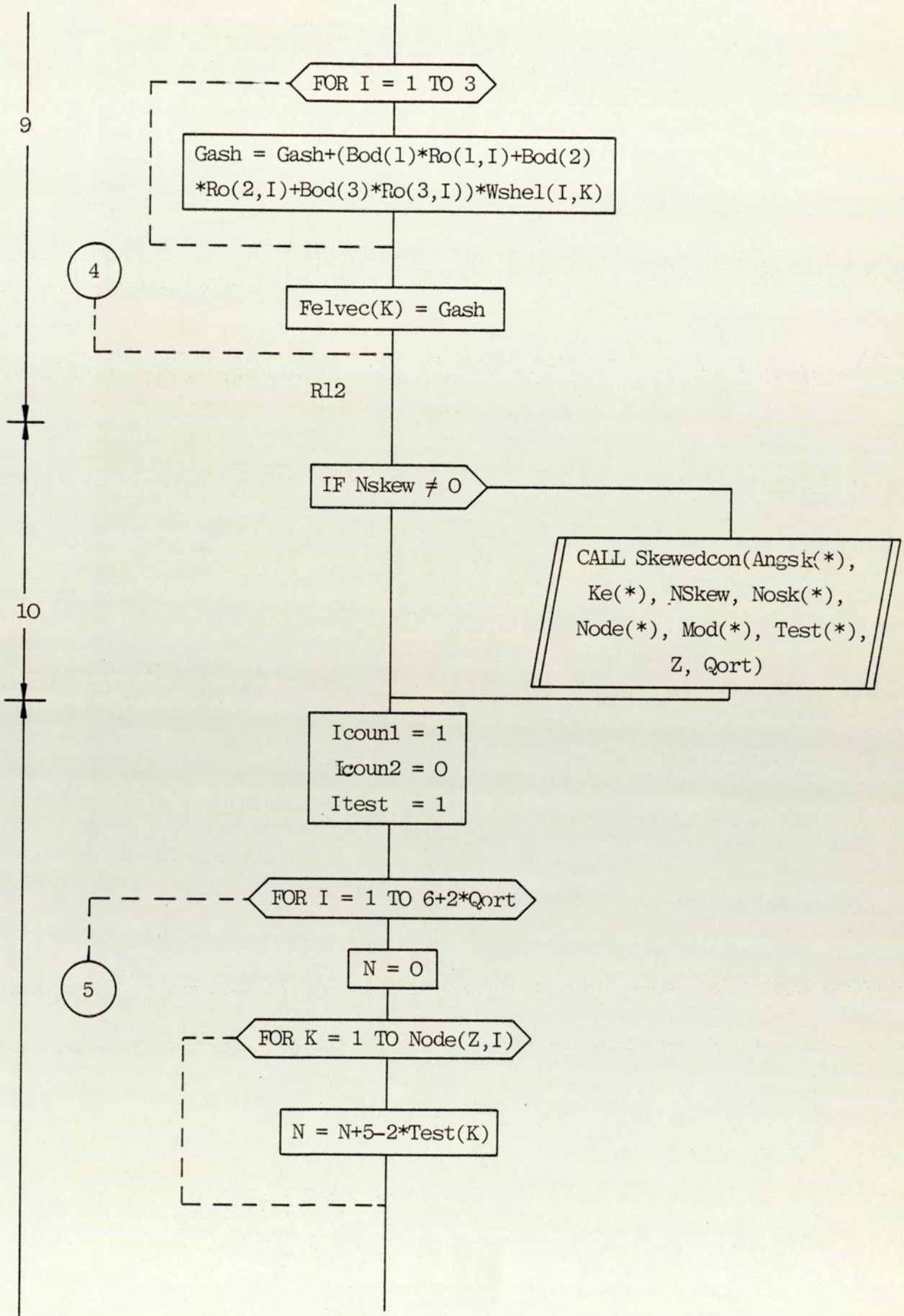

3
Mod(1) = 1
Mod(2) = 4+Qort
Mod(3) = 2
Mod(4) = 5+Qort
Mod(5) = 3
Mod(6) = 6+Qort
Mod(7) = 4
Mod(8) = 8
FOR Z = 1 TO Nelemt
1
FOR I = 1 TO 6+2*Qort
A1(I) = Node(Z,I)
FOR I = 1 TO 6+2*Qort
Node(Z,I) = A1(Mod(I))
MAT Ke = ZER
MAT Xyzpre = ZER
MAT E1xyzt = ZER
No'zpre = 0
MAT Shear = ZER
Thick = Thikn(Z)

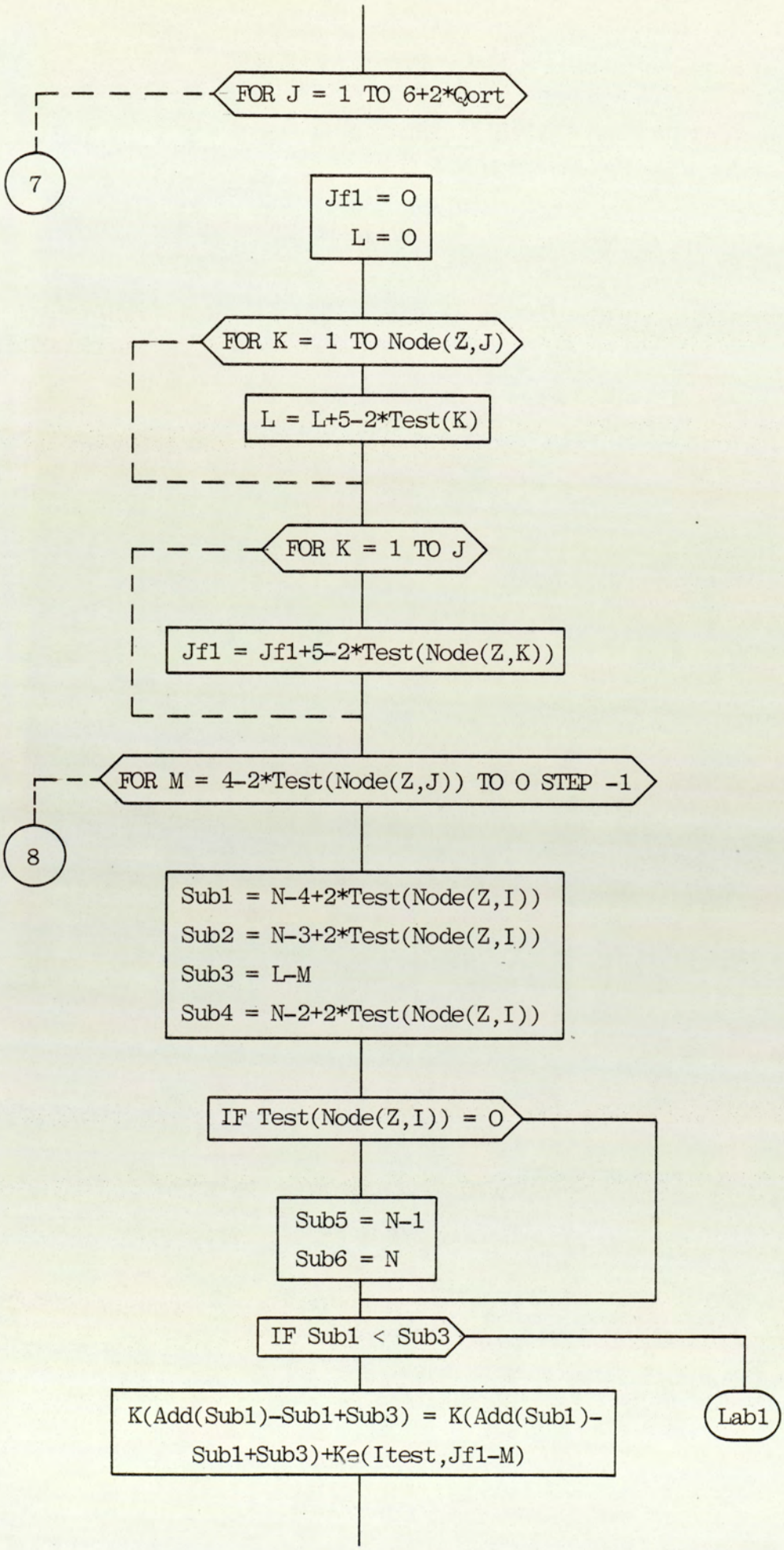

4
5
FOR I = 1 TO Lnodz
K = INT (ABS(Node(Z,I))
E1xyz+(I,1) = Xx(K)
E1xyzt(I,2) = Yy(K)
E1xyzt(I,3) = Zz(K)
E1xyzt(I,4) = Thick
FOR M = 1 TO 4+Qort
2
IF M = 5
Xita(1) = W(M,1)
Xita(2) = W(M,2)
Xita(1) = 0
Xita(2) = 0
CALL Haloof (Xita(*), Area, Thik, Ro(*), poin(*),
E1yzt(*), Wshel(*), Thick, Shear(*),
Xyzpre(*), Vloof(*), Swop(2), Node(*),
Lnodz, Nozpre, Z, Nelemt)

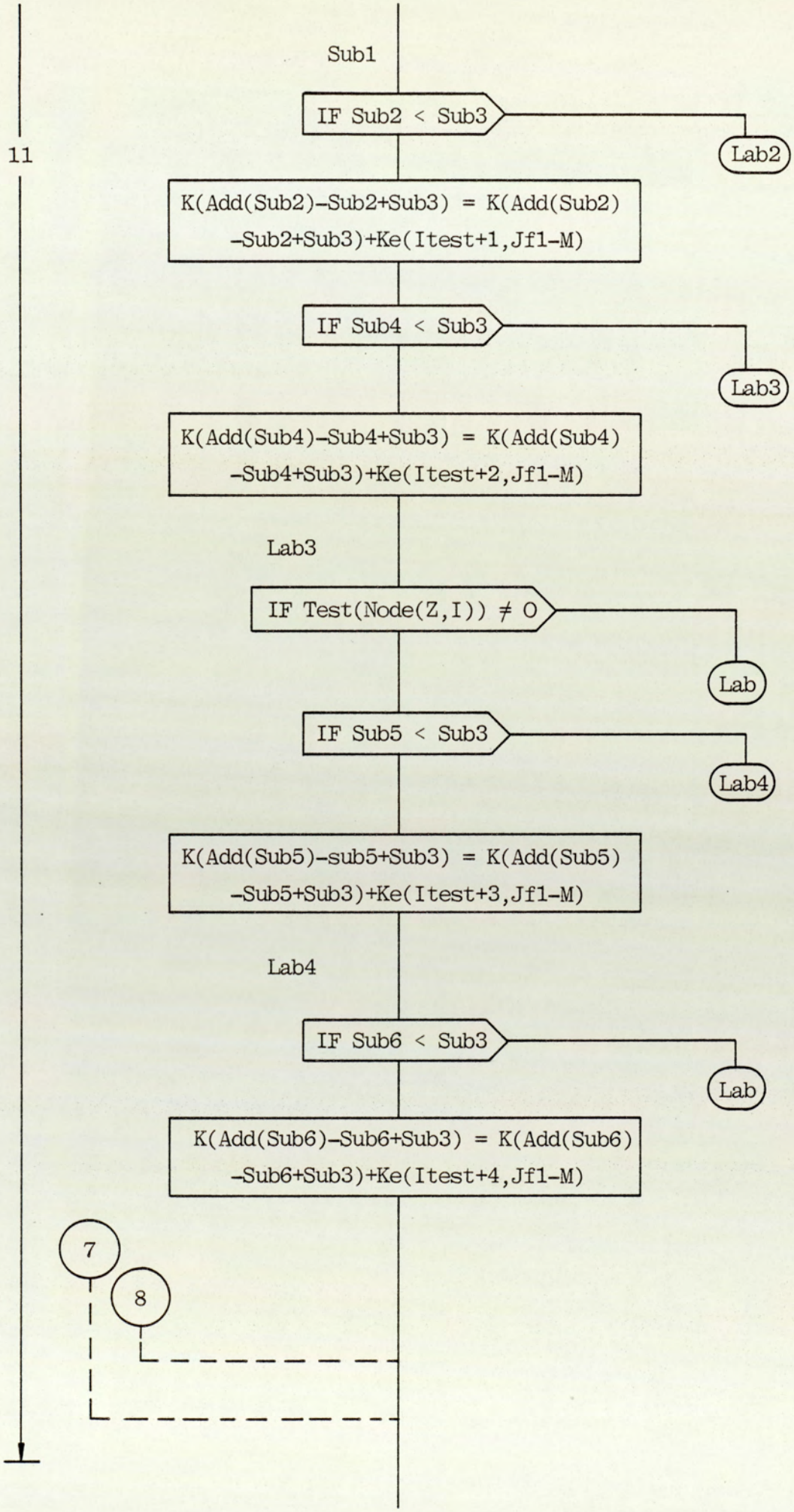

6
FOR I = 1 TO 24 + 8*Qort
B(1,I) = Wshel(4,I)
B(2,I) = Wshel(7,I)
B(3,I) = Wshel(5,I) + Wshel(6,I)
B(4,I) = Wshel(10,I)
B(5,I) = Wshel(12,I)
B(6,I) = 2*Wshel(11,I)
Detj = Area
Jn = Node(Z,7+2*Qort)
IF M = 4 AND Qort = 0
Azh
a3
FOR K = 1 TO 4
a2
FOR I = K TO 21+8*Qort+(K-1) STEP 4
a1
FOR J = I TO 21+8*Qort+(K-1) STEP 4
A = C(Jn,1)*B(1,I) + C(Jn,2)*B(2,I) + C(Jn,3)*B(3,I)
B = C(Jn,2)*B(1,I) + C(Jn,7)*B(2,I) + C(Jn,8)*B(3,I)
C = C(Jn,3)*B(1,I) + C(Jn,8)*B(2,I) + C(Jn,12)*B(3,I)
D = (C(Jn,16)*B(4,I) + C(Jn,17)*B(5,I) + C(Jn,18)*B(6,I)*Thik↑2
E = (C(Jn,17)*B(4,I) + C(Jn,19)*B(5,I) + C(Jn,20)*B(6,I))*Thik↑2
F = (C(Jn,18)*B(4,I) + C(Jn,20)*B(5,I) + C(Jn,21)*B(6,I))*Thik↑2
Ke(J,I) = Ke(I,J) = Ke(I,J) + A(M)*(B(1,J)*A+B(2,J)*B
+B(3,J)*C+B(4,J)*D+B(5,J)*E+B(6,J)*F)*Detj*Thik.
IF J = 24+8*Qort
L6

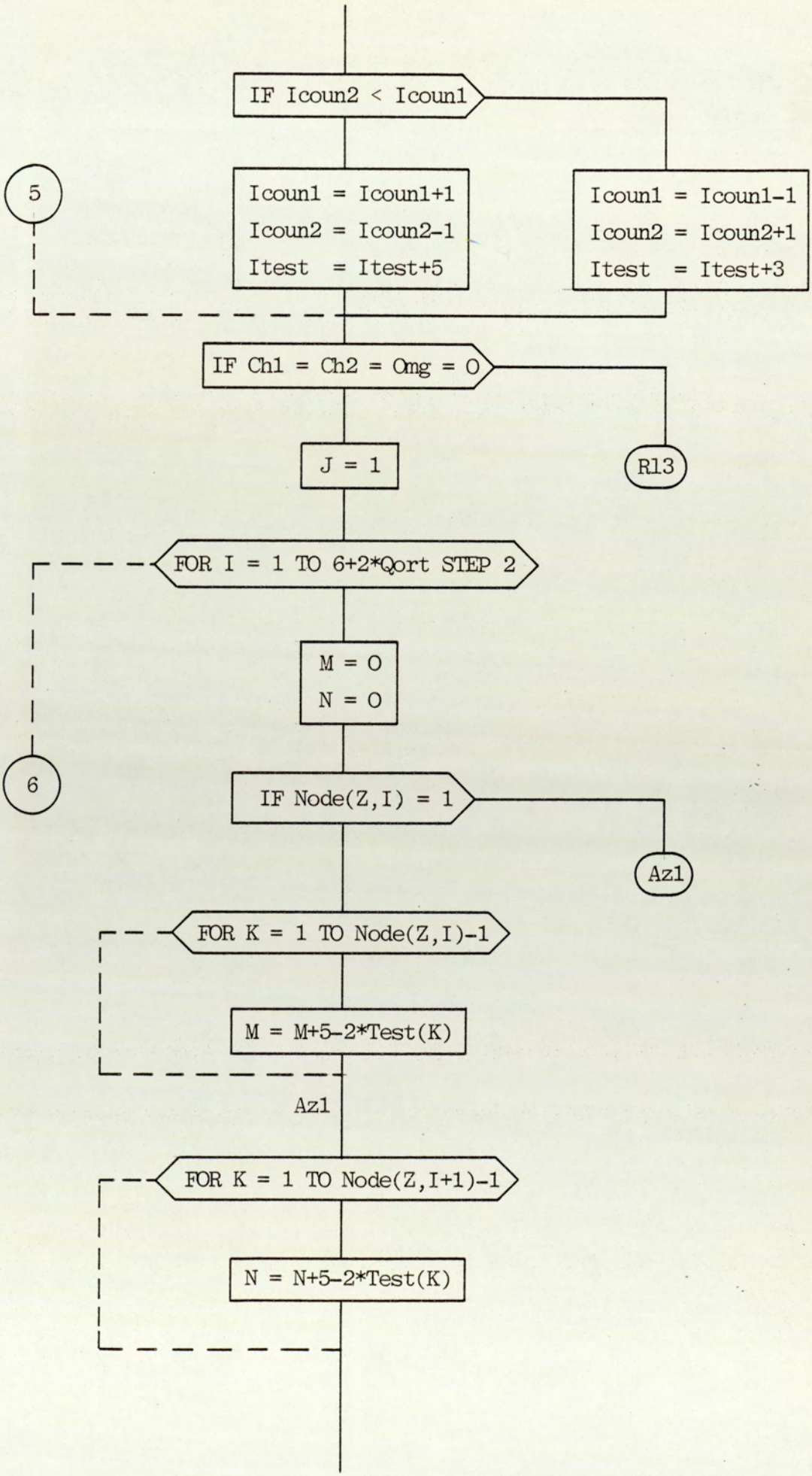

Ke(I,J+1) = Ke(J+1,I) = Ke(J+1,I)+A(M)*(B(1,J+1)*A
+B(2,J+1) *B+B(3,J+1)*C+B(4,J+1)*D+B(5,J+1)
*E+B(6,J+1)*F)*Detj*Thik
IF J = 23+8*Qort
L6
Ke(I,J+2) = Ke(J+2,I) = Ke(J+2,I)+A(M)*(B(1,J+2)*A
+B(2,J+2)*B+B(3,J+2)*C+B(4,J+2)*D+B(5,J+2)*E
+B(6,J+2)*F)*Detj*Thik
7
IF J = 22+8*Qort
L6
Ke(I,J+3) = Ke(J+3,I) = Ke(J+3,I)+A(M)*(B(1,J+3)*A+B(2,J+3)
*B+B(3,J+3)*C+B(4,J+3)*D+B(5,J+3)*E+B(6,J+3)*F)
*Detj*Thik
a1
a2
a3
IF prnt ≠ 1 AND prnt ≠ 3
B(*),Thik
Ro(*),poin(*)
Azh
2
8
IF prnt ≠ 2 AND prnt ≠ 3
B(*),Thik
Ro(*),poin(*)
IF Ch1 = 0 AND Ch2 = 0
R11
Weight = Area*Ch1*A(M)
FOR K = 1 TO 24+8*Qort
IF Ch1 = 0
3
press

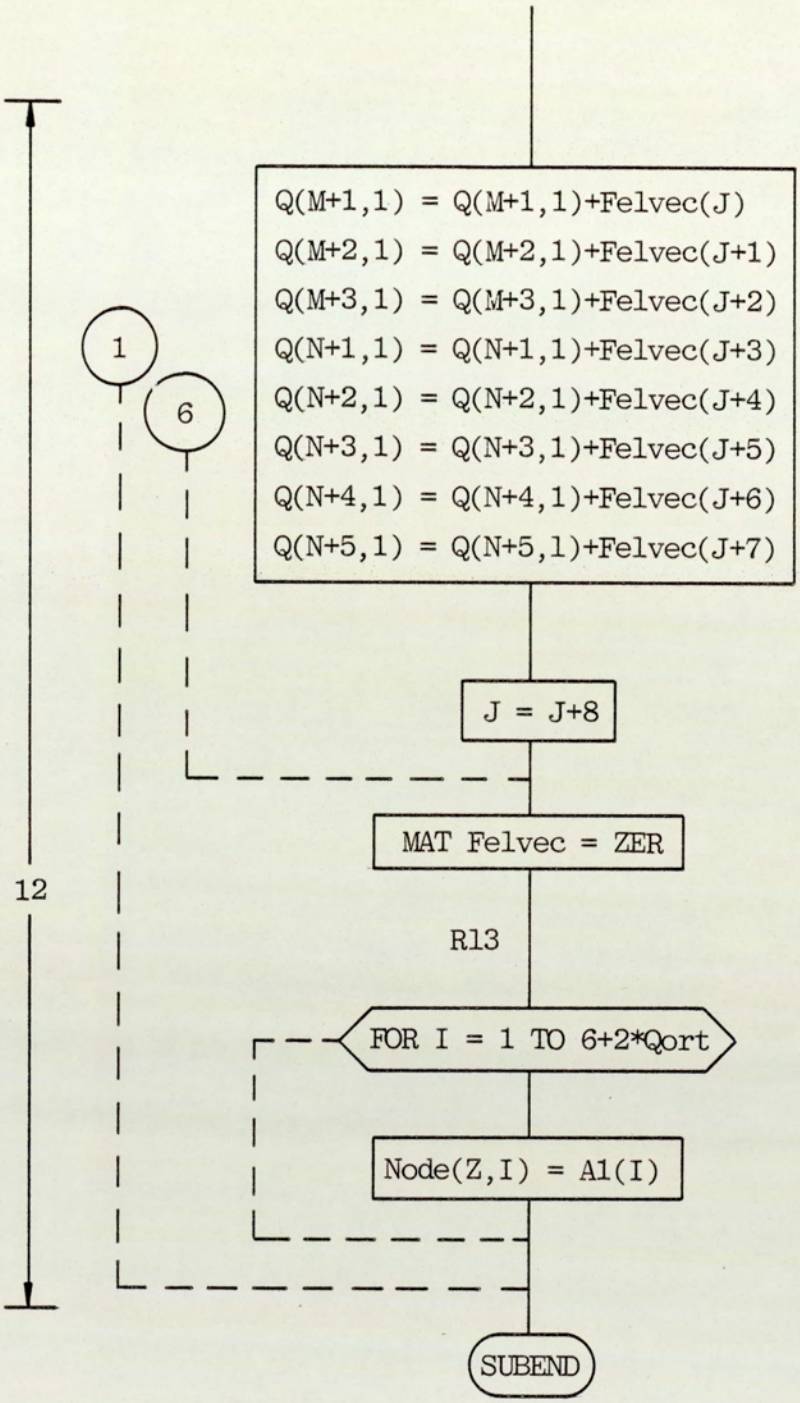

Gash = Felvec(K)
FOR I = 1 TO 3
Gash = Gash+weight*Ro(3,I)*Wshel(I,K)
IF Ch2 = 0
press
Felvec(K) = Gash
Gash = Felvec(K)
3
Felvec(K) = Gash+Ch2*A(M)*Wshel(3,K)
R11
IF Omg = 0
R12
Ro = Th(Node(Z,7+2*Qort))
FOR I = 1 TO 2
Bod(I) = Ro*Omg*Omg*poin(I)*Thik*Area*A(M)
Bod(3) = 0
FOR K = 1 TO 24+8*Qort
Gash = Felvec(K)
4


9
FOR I = 1 TO 3
Gash = Gash+(Bod(1)*Ro(1,I)+Bod(2)
*Ro(2,I)+Bod(3)*Ro(3,I))*Wshel(I,K)
4
Felvec(K) = Gash
R12
10
IF Nskew ≠ 0
CALL Skewedcon(Angsk(*),
Ke(*), NSkew, Nosk(*),
Node(*), Mod(*), Test(*),
Z, Qort)
Icoun1 = 1
Icoun2 = 0
Itest = 1
FOR I = 1 TO 6+2*Qort
5
N = 0
FOR K = 1 TO Node(Z,I)
N = N+5-2*Test(K)


FOR J = 1 TO 6+2*Qort
7
Jf1 = 0
L = 0
FOR K = 1 TO Node(Z,J)
L = L+5-2*Test(K)
FOR K = 1 TO J
Jf1 = Jf1+5-2*Test(Node(Z,K))
FOR M = 4-2*Test(Node(Z,J)) TO 0 STEP -1
8
Sub1 = N-4+2*Test(Node(Z,I))
Sub2 = N-3+2*Test(Node(Z,I))
Sub3 = L-M
Sub4 = N-2+2*Test(Node(Z,I))
IF Test(Node(Z,I)) = 0
Sub5 = N-1
Sub6 = N
IF Sub1 < Sub3
K(Add(Sub1)-Sub1+Sub3) = K(Add(Sub1)-
Sub1+Sub3)+Ke(Itest,Jf1-M)
Lab1


11
Sub1
IF Sub2 < Sub3
Lab2
K(Add(Sub2)-Sub2+Sub3) = K(Add(Sub2)
-Sub2+Sub3)+Ke(Itest+1,Jf1-M)
IF Sub4 < Sub3
Lab3
K(Add(Sub4)-Sub4+Sub3) = K(Add(Sub4)
-Sub4+Sub3)+Ke(Itest+2,Jf1-M)
Lab3
IF Test(Node(Z,I)) ≠ 0
Lab
IF Sub5 < Sub3
Lab4
K(Add(Sub5)-sub5+Sub3) = K(Add(Sub5)
-Sub5+Sub3)+Ke(Itest+3,Jf1-M)
Lab4
IF Sub6 < Sub3
Lab
K(Add(Sub6)-Sub6+Sub3) = K(Add(Sub6)
-Sub6+Sub3)+Ke(Itest+4,Jf1-M)
7
8


IF Icoun2 < Icoun1
5
Icoun1 = Icoun1+1
Icoun2 = Icoun2-1
Itest = Itest+5
Icoun1 = Icoun1-1
Icoun2 = Icoun2+1
Itest = Itest+3
IF Ch1 = Ch2 = Omg = 0
J = 1
R13
FOR I = 1 TO 6+2*Qort STEP 2
M = 0
N = 0
6
IF Node(Z,I) = 1
Az1
FOR K = 1 TO Node(Z,I)-1
M = M+5-2*Test(K)
Az1
FOR K = 1 TO Node(Z,I+1)-1
N = N+5-2*Test(K)


Q(M+1,1) = Q(M+1,1)+Felvec(J)
Q(M+2,1) = Q(M+2,1)+Felvec(J+1)
Q(M+3,1) = Q(M+3,1)+Felvec(J+2)
Q(N+1,1) = Q(N+1,1)+Felvec(J+3)
Q(N+2,1) = Q(N+2,1)+Felvec(J+4)
Q(N+3,1) = Q(N+3,1)+Felvec(J+5)
Q(N+4,1) = Q(N+4,1)+Felvec(J+6)
Q(N+5,1) = Q(N+5,1)+Felvec(J+7)
1
6
J = J+8
MAT Felvec = ZER
12
R13
FOR I = 1 TO 6+2*Qort
Node(Z,I) = A1(I)
SUBEND

4.3.2.6 Subroutine "Skewedcon"

In applications, the structures are commonly constrained to move in the direction of the global axes. This means that the prescribed displacements are in the direction of the global coordinates. However, the prescribed displacements can also be in other directions (as shown in Fig.4.16), in which the skewed rectangular plate is constrained to move in the directions X, Y and Z, while the global axes are x, y and z.

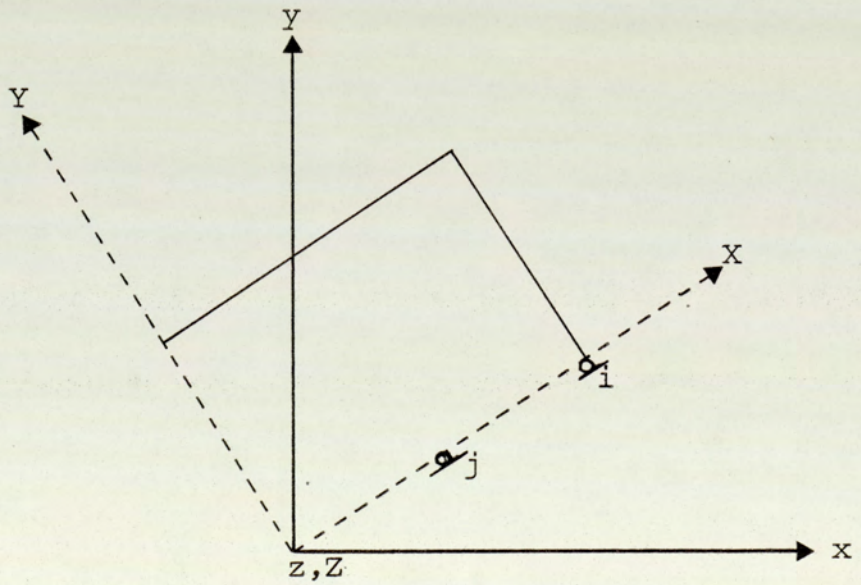


Fig.4.16 Skewed rectangular plate

This situation needs a transformation of the matrices for the nodes where those skewed conditions are present. In fact, it is necessary to transform the element stiffness matrices, and the load vector to refer the skewed directions for the skewed nodes.

For a typical corner node i it is easy to construct the transformation matrix $[T]$ as follows:

$$\begin{Bmatrix} u \\ v \\ w \end{Bmatrix}_{glo} = \begin{bmatrix} \cos\phi & -\sin\phi & 0 & U \\ \sin\phi & \cos\phi & 0 & V \\ 0 & 0 & 1 & W \end{bmatrix}_{skewed} \quad \ldots \quad (4.48)$$

and for the midside node j:

$$\begin{Bmatrix} u \\ v \\ w \\ \theta^1_{xz} \\ \theta^2_{xz} \end{Bmatrix}_{glo} = \begin{bmatrix} \cos\phi & -\sin\phi & 0 & 0 & 0 & U \\ \sin\phi & \cos\phi & 0 & 0 & 0 & V \\ 0 & 0 & 1 & 0 & 0 & W \\ 0 & 0 & 0 & 1 & 0 & \theta^2_{XZ} \\ 0 & 0 & 0 & 0 & 1 & \theta^2_{XZ} \end{bmatrix} \quad \ldots \quad (4.49)$$

or

$$\{d\} = [T]\{d'\} \quad \ldots \quad (4.50)$$

where

{d} = is the nodal displacement vector associated with global axes

{d'} = is the nodal displacement vector associated with skewed axes.

and [T] = the transformation matrix connecting the two sets of degrees of freedom.

For the entire nodal displacement vector, the transformation matrix may be constructed as follows:

$$[S] = \begin{bmatrix} [I] & & & & \\ & [I] & & & \\ & & [T] & & \\ & & & [I] & \\ & & & & [I] \end{bmatrix} \quad \ldots \quad (4.51)$$

where [T] corresponds to the nodes which are prescribed in the skewed directions. [I] is the identity matrix, which is (3×3) if it corresponds to a corner node, and (5×5) if it corresponds to a midside node.
The number of diagonals = 6+2*Qort.

The transformation of the element stiffness matrix and the element load vector are found in the following way:

The work done by the force applied in the skewed and unskewed conditions must be the same, as shown by:

$$\{F\}^t\{d\}^e = \{F'\}^t\{d'\}^e \quad \ldots \quad (4.52)$$

The element displacement vector $\{u\}^e$ in the global direction is as follows:

$$\{d\}^e = [S]\{d'\}^e \quad \ldots \quad (4.53)$$

where $\{d'\}^e$ represents the element displacement vector in the skewed directions and $[S]$ is given by equation (4.51).

Substitution equation (4.53) for $\{u\}_e$ into equation (4.52) results in the following equation:

$$\{F'\}^e = [S]^t\{F\}^e \quad \ldots \quad (4.54)$$

The equilibrium equations for the unskewed system is:

$$[Ke]\{u\}^e = \{F\}^e$$

pre-multiplying by $[S]^t$ gives:

$$[S]^t[Ke]\{u\}^e = [S]^t\{F\}^e$$

or

$$[S]^t[Ke][S]\{u'\}^e = [S]^t\{F\}^e$$

i.e. $[K'_e]\{u'\}^e = \{F'\}^e$

where

$$[K'_e] = [S]^t [Ke] [S] \quad \ldots (4.55)$$

and

$$\{F'\}^e = [S]^t \{F\} \quad \ldots (4.56)$$

This transformation is applied individually for the element stiffness matrices using the subroutine "Skewedcon" before the assembly of the overall stiffness matrix.

The flow chart of the "skewedcon" illustrates the steps involved. With reference to the flow chart, shown in Fig. 4.17, these steps are as follows:

1. Coefficients of the transformation array [S] according to equation (4.51) are constructed for the current element.

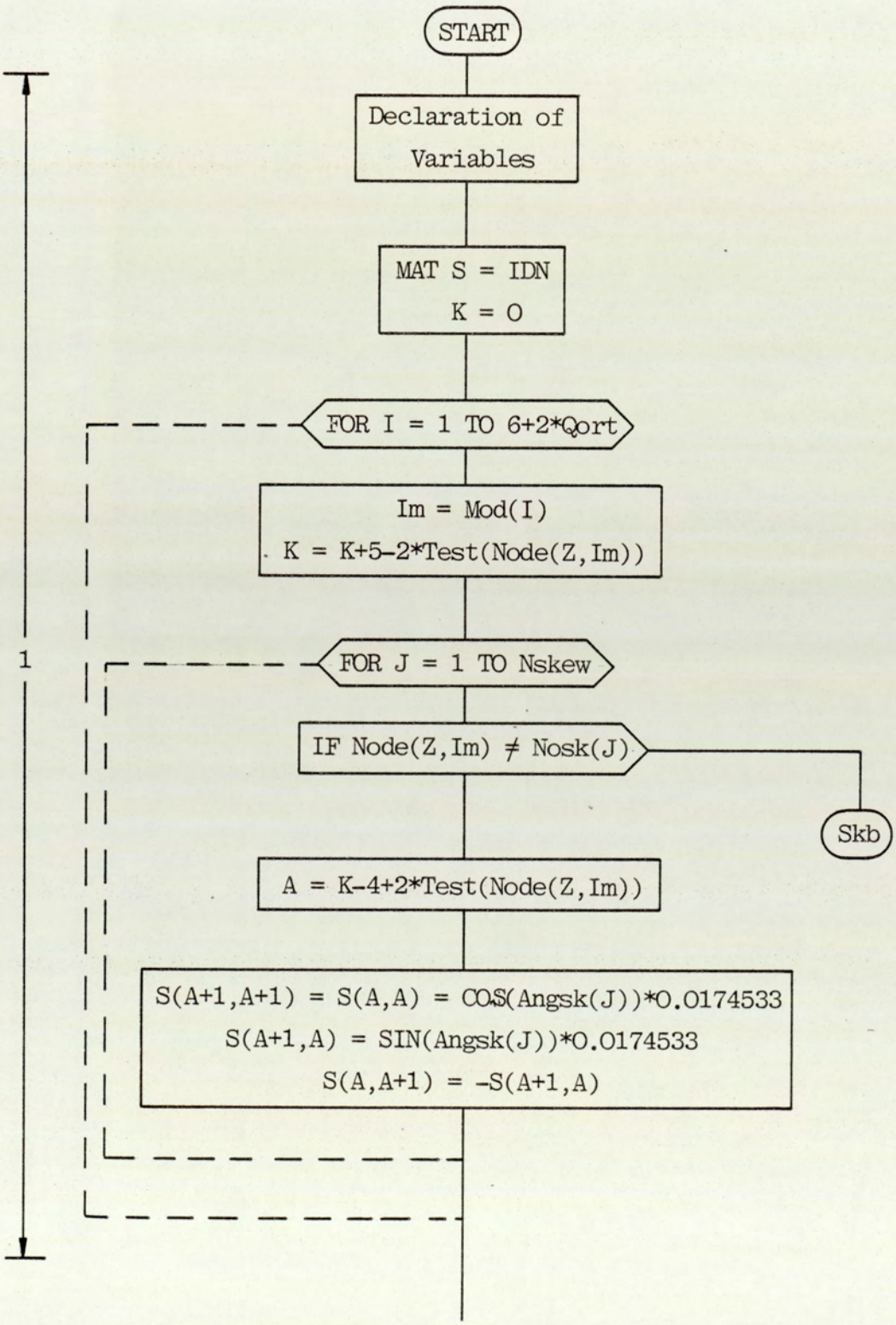
2. Matrix multiplications of equation (4.55) are performed on the element stiffness matrix.

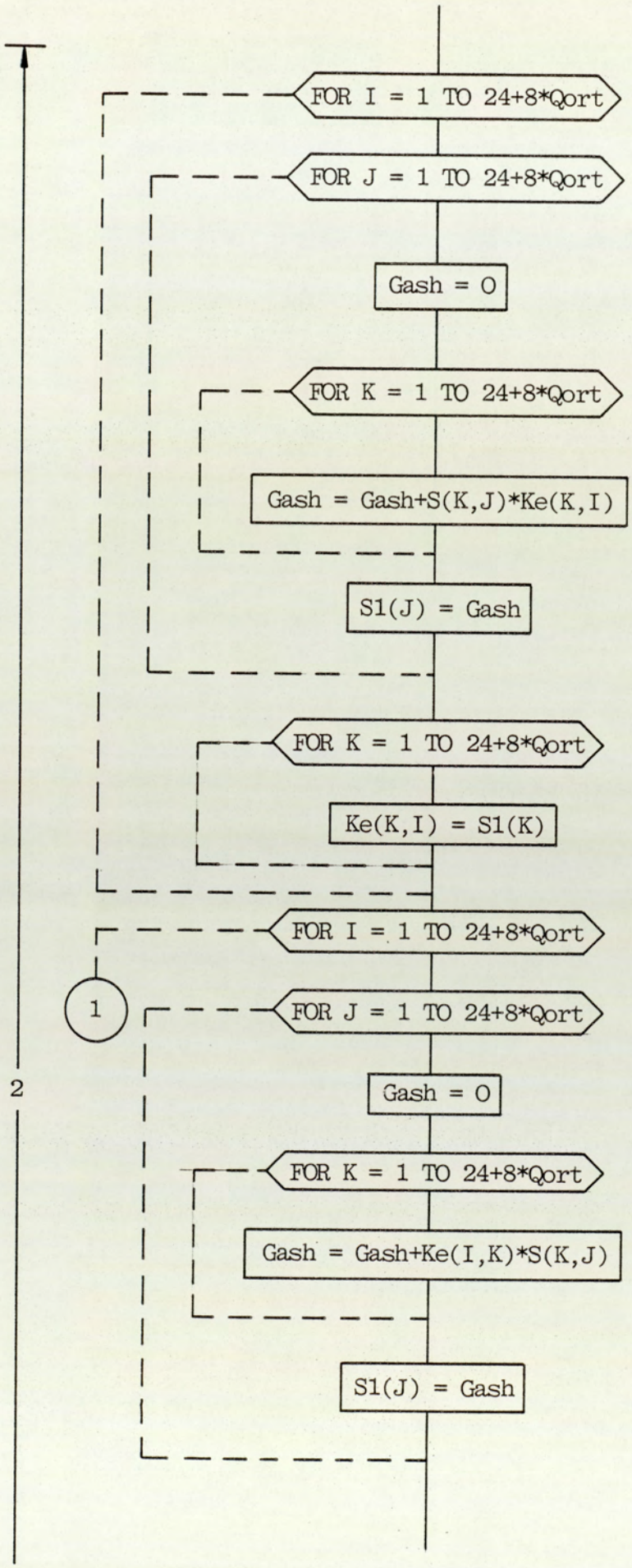
4.3.2.7 Subprogram SMLDAP

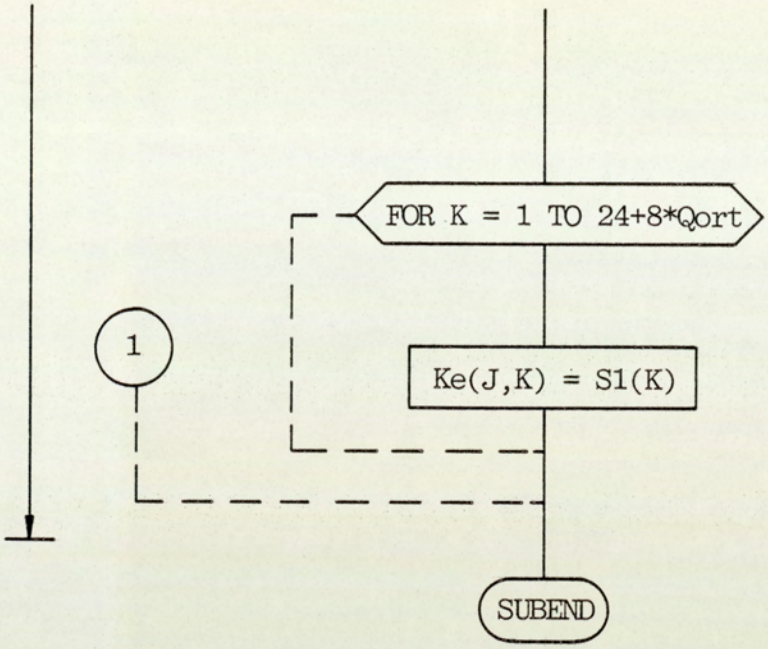
The subroutine "Loadapp" has been written in this subprogram to take into account the application of concentrated forces and normal moments at the loof nodes.

Fig. 4.17 Flow Chart of Subroutine'Skewedcon''

SUB Skewedcon (Angsk(*), Ke(*), INTEGER Nskew, Nosk(*), Node(*), Mod(*), Test(*), Z, Qort).




FOR I = 1 TO 24+8*Qort
FOR J = 1 TO 24+8*Qort
Gash = 0
FOR K = 1 TO 24+8*Qort
Gash = Gash+S(K,J)*Ke(K,I)
S1(J) = Gash
FOR K = 1 TO 24+8*Qort
Ke(K,I) = S1(K)
FOR I = 1 TO 24+8*Qort
1
FOR J = 1 TO 24+8*Qort
Gash = 0
FOR K = 1 TO 24+8*Qort
Gash = Gash+Ke(I,K)*S(K,J)
S1(J) = Gash
2


FOR K = 1 TO 24+8*Qort
1
Ke(J,K) = S1(K)
SUBEND

This needs to establish the degree of freedom in the direction of the required load at the node of interest. The subroutine "loadapp" does precisely this and adds these types of loadings (if they exist) to the system load vector {Q}.

The following flow chart illustrates the steps involved which are not more than checking the code of the current node (see Appendix A). On this basis the direction is defined to correspond to the degree(s) of freedom on which the concentrated force or normal moment is applied.

4.3.2.8 Subprogram SMSKEW

As explained in Section 4.3.2.5, the overall load vector {Q}needs to be modified to correspond to skewed coordinates. For a typical skewed node, the modified load vector for this node follows equation (4.5 6)

$$\{F'\}_e = [S]^t \{F\}_e$$

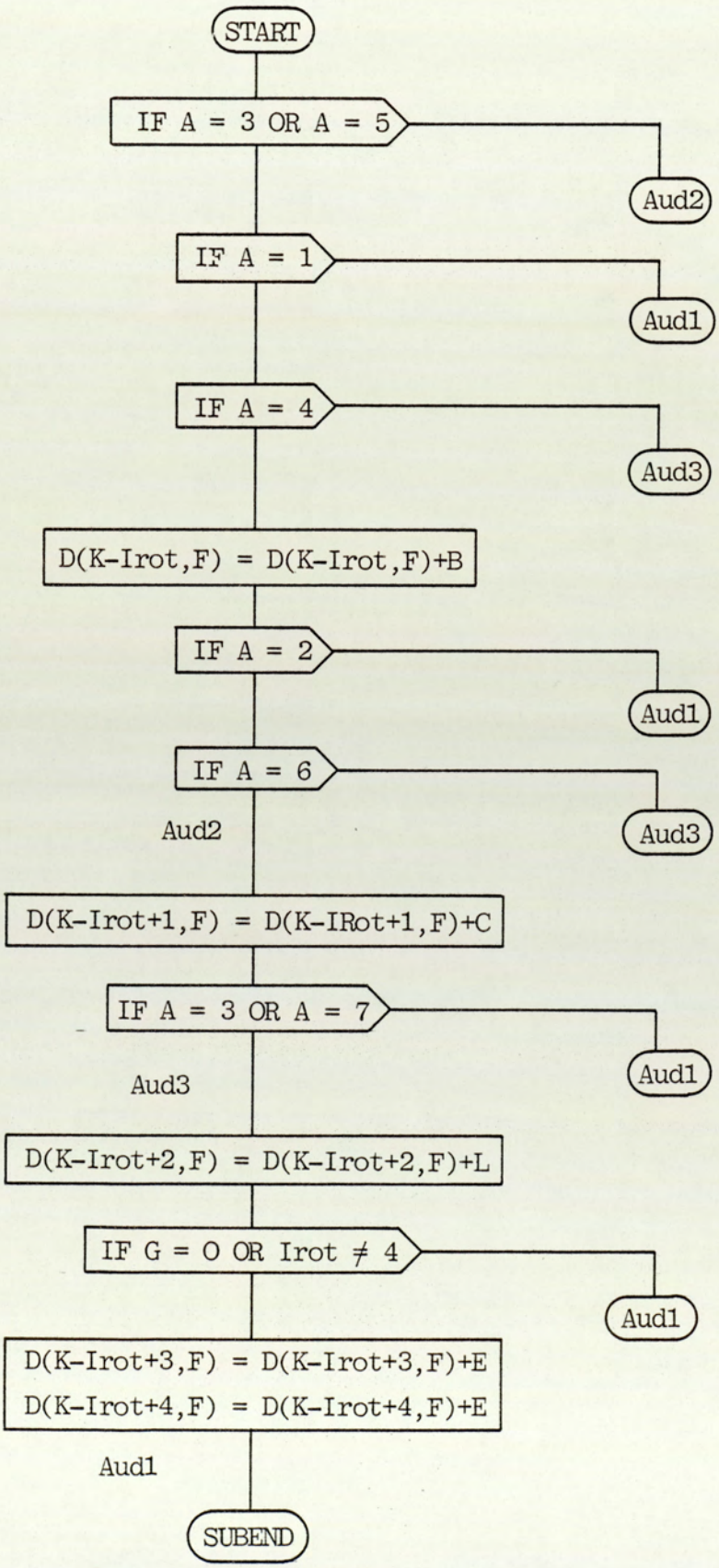
The steps involved are presented with reference to the flow chart shown in Fig. 4.19:

1. A loop is set for the number of skewed nodes.

2. The coefficients of the transformation matrix are

Fig. 4.18 Flow Chart of Subroutine "Loadapp"

SUB Loadapp (B, C, L, E, D(*), A, G, K, F, Irot).



calculated for the current skewed node.

3. The load vector {Q} is modified using step 2.

Steps 2-3 are repeated for the other skewed nodes.

4.3.2.9 Subprogram SMYVBS

This subprogram contains two subroutines. They are called through the main program "SMILOF" as required. The first one, "Geombc", is used to adjust the overall stiffness matrix [K] and the load vector {Q} in allocating the prescribed displacements, while the second, "Symvbsol" is a solving routine to give the nodal displacements in the global directions.

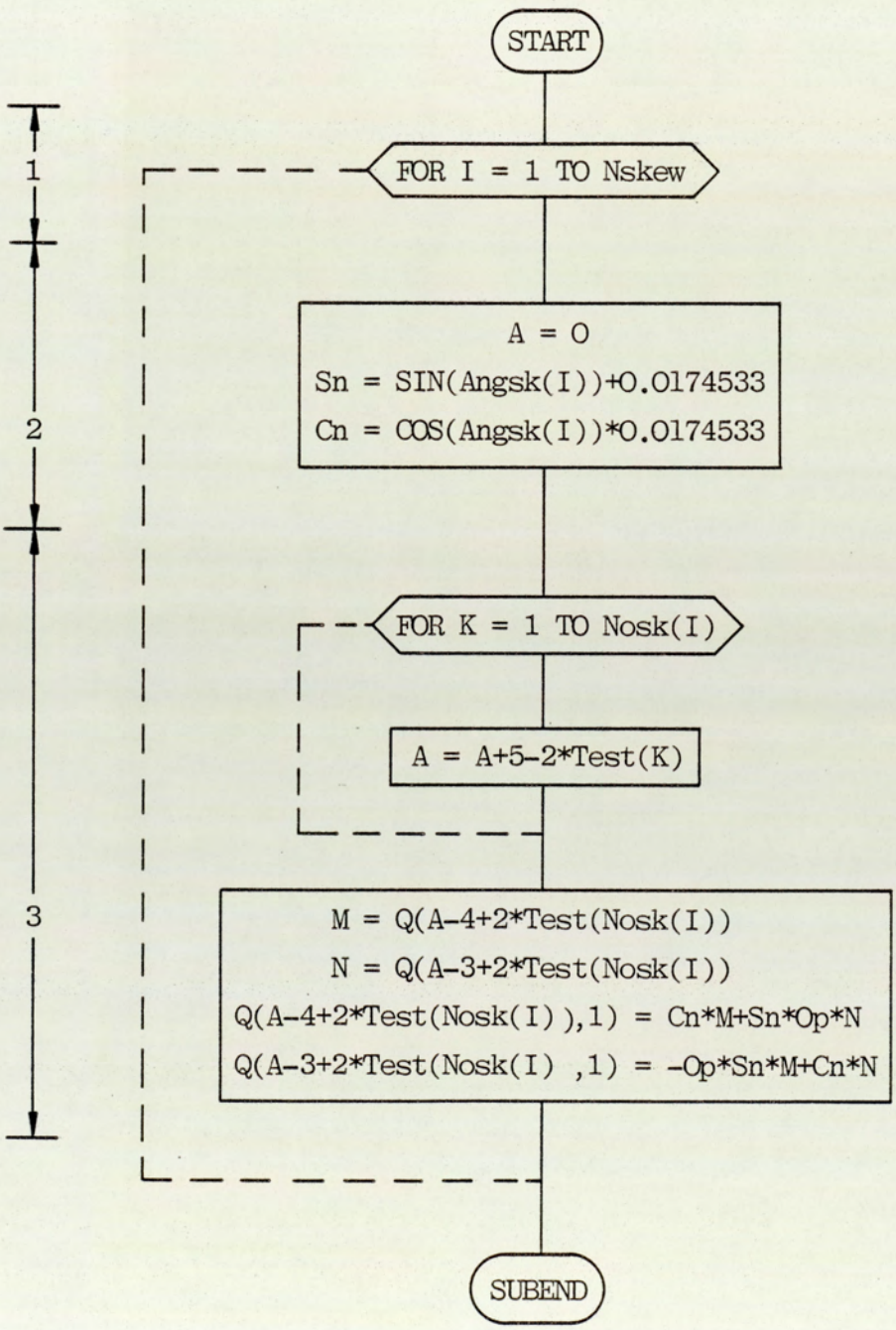
(a) "Geombc"

In Section 4.3.2.6, the skewed conditions have been explained. It was shown that the necessary transformations must be done for any skewed boundary conditions.

At this subroutine, the values of such prescribed displacements are applied. The principle for the application of such kinematic displacement is explained by partitioning and re-arranging the overall stiffness equations (see Ref. (65). This can be summarised as follows:

Fig. 4.19 Flow Chart of Subroutine "Skewload"

SUB Skewload (Q(*), Angsk(*), Test(*), Nosk(*), Op)



The following system of equations is considered

$$\left[\begin{array}{c|c} [K_{11}] & [K_{21}]^t \\ \hline [K_{21}] & [K_{22}] \end{array}\right] \left\{\begin{array}{c} \{u_1\} \\ \hline \{u_2\} \end{array}\right\} = \left\{\begin{array}{c} \{Q_1\} \\ \hline \{Q_2\} \end{array}\right\}$$

where $\{u_1\}$ is a vector of free nodal displacements and $\{u_2\}$ is a vector of specified displacements. This may be written as:

$$[K_{11}]\{u_1\} = \{Q_1\} - [K_{21}]^t\{u_2\}$$

and

$$\{Q_2\} = [K_{21}]\{u_1\} + [K_{22}]\{u_2\}$$

where $\{Q_2\}$ represents the reactions at the constrained nodes. For the particular case where the values of the specified displacements are zero, the rows and columns of the system equations corresponding to the constrained degrees of freedom are eliminated. For the general case the system of equations may be written as:

$$\left[\begin{array}{c|c} [K_{11}] & [O] \\ \hline [O] & [I] \end{array}\right] \left\{\begin{array}{c} \{u_1\} \\ \hline \{u_2\} \end{array}\right\} = \left\{\begin{array}{c} \{Q_1\} - [K_{21}]^t\{u_2\} \\ \hline \{u_2\} \end{array}\right\}$$

In fact, the rows and columns of $|K|$ corresponding to the constrained degrees of freedom are made zero, and the diagonal is made unity. Finally, the prescribed values of displacements are inserted in the load vector.

(b) "Symvbsol"

This solving routine for the system of equations is given by Jennings and Tuff, Ref(63) using Cholesky triangular factorisation.

The equations of equilibrium are of the form:

$$[K]\{q\} = \{Q\} \qquad \ldots (4.57)$$

The solution vector $\{q\}$ is obtained by performing a triangular decomposition on the matrix $[K]$ by the square root method of Cholesky, which requires that $[K]$ be symmetric positive definite. The Cholesky triangular factorisation method is suitable for large problems. It requires no storage facilities other than that available for the stiffness matrix $[K]$.

Using Cholesky factorisation, $[K]$, can be written as follows:

$$[K] = [L][L]^t \qquad \ldots (4.58)$$

where $[L]$ is upper triangular matrix.

Now equation (4.58) becomes:

$$[L][L]^t\{q\} = \{Q\} \qquad \ldots \quad (4.59)$$

This represents two triangular systems;

by defining $[L]^t\{q\} = \{r\}$ ... (4.60)

equation (4.59) becomes $[L]\{r\} = \{Q\}$ ... (4.61)

Equation (4.61) is solved for {r} by forward elimination while equation (4.60) is solved for {q} by back substitution. This type of solving the equilibrium equations eliminates the need for complicated inversion of the stiffness matrix. The matrix [L] overwrites in the store by using the recursive relationships.

$$L_{ij} = (K_{ij} - \sum_{K=1}^{j-1} L_{iK} L_{jK}) / L_{jj} \quad , \text{ for } j<i \qquad \ldots \quad (4.62)$$

$$L_{ii} = \sqrt{K_{ii} - \sum_{K=1}^{i-1} (L_{iK})^2} \quad , \text{ for diagonal terms}$$

... (4.63)

The subroutine "Symvbsol" together with "Geombc" have been used by Ref. (65). In this work these subroutines have been translated from the ALGOL language to suit the desk top computer HP9845B. Therefore, no flow charts are included as they were taken directly from Ref.(65) . However, the listing of these subroutines is given in Appendix (B).

4.3.2.10 Subprogram SMELST

The subroutine "Elstrs" which is written in this subprogram is designed to evaluate the strains and stresses at the centroid of each element. Once the nodal displacements have been determined, the element strains and stresses may be obtained using equation (4.3):

$$\{\varepsilon\} = [B]\{d\}$$

The order of $\{\varepsilon\}$ matrix is 6: three membrane strains ε_X^m, ε_Y^m and ε_{XY}^m, while the other three refer to the bending strains ε_X^b, ε_Y^b and ε_{XY}^b. The element stresses are calculated using equation (4.44):

$$\{\sigma\} = [D]\{\varepsilon\}$$

$\{\sigma\}$ contains 3 membrane stresses: N_X, N_Y are the forces per unit length acting normal to the sides of the elements; and N_{XY} is the shear force per unit length acting in the plane of the sides of the element. The three bending terms: M_X, M_Y are the bending moments per unit length, and M_{XY}: is the torsional moment per unit length. As mentioned in Section 4.3.2.1, the strain-element nodal displacement matrices are read from the file prepared to print these matrices in the stage of formulating the element stiffness matrices. The reason for storing these matrices is that the evaluation of [B] matrices takes a significant time in the desk top computer HP9845B,

especially for quadrilateral semiloof shell element: storing therefore, means saving computing time.

The output of this subprogram as required by the control variables are as follows:

(i) Strains and stresses at each element centroid in the direction of local axes. These strains and stresses are transferred to the directions of the global axes by using the direction cosine matrix [Ro].

(ii) Principal strains and stresses at each element centroid.

The transformation matrix of strains and stresses from local axes directions (X, Y, Z) to the global axes directions (x, y, z) is explained as follows:

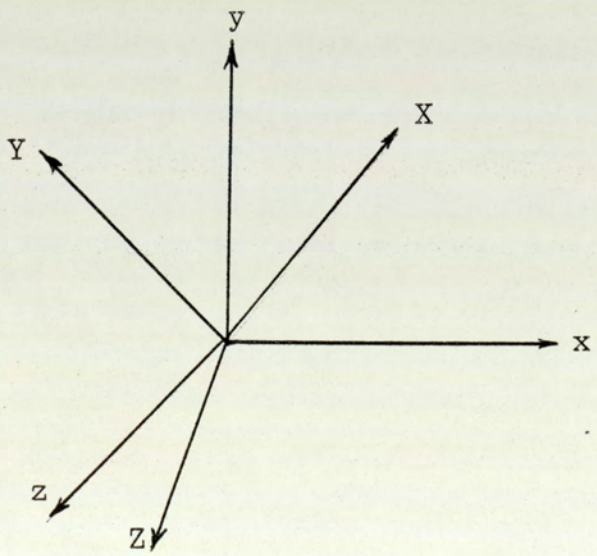


Fig. 4.20 Direction Cosine

Consider Fig. 4.20, suppose the global axes are related to the local axes by the following direction cosine matrix:

	X	Y	Z
x	ℓ_1	ℓ_2	ℓ_3
y	m_1	m_2	m_3
z	n_1	n_2	n_3

i.e.

$$|Ro| = \begin{bmatrix} \ell_1 & \ell_2 & \ell_3 \\ m_1 & m_2 & m_3 \\ n_1 & n_2 & n_3 \end{bmatrix} \quad \ldots \quad (4.64)$$

The stresses and strains are transformed by the following operations (see Ref. (66))

$$\{\sigma\}_{loc} = [T_\sigma]\{\delta\}_{glo} \;, \quad \{\sigma\}_{glo} = [T_\sigma]^{-1}\{\sigma\}_{loc}$$

$$\ldots \quad (4.65)$$

$$\{\varepsilon\}_{loc} = [T_\varepsilon]\{\varepsilon\}_{glo} \;, \quad \{\varepsilon\}_{glo} = [T_\varepsilon]^{-1}\{\varepsilon\}_{loc}$$

Matrices $[T_\sigma]$ and $[T_\varepsilon]$ each have the property that the inverse of one equals the transpose of the other. Robert Cook, Ref(66) has written them in matrix form as follows:

$$[T_\sigma] = \begin{array}{|c|c|c|c|c|c|} \hline \ell_1^2 & m_1^2 & n_1^2 & 2\ell_1 m_1 & 2m_1 n_1 & 2n_1 \ell_1 \\ \hline \ell_2^2 & m_2^2 & n_2^2 & 2\ell_2 m_2 & 2m_2 n_2 & 2n_2 \ell_2 \\ \hline \ell_3^2 & m_3^2 & n_3^2 & 2\ell_3 m_3 & 2m_3 n_3 & 2n_3 \ell_3 \\ \hline \ell_1\ell_2 & m_1 m_2 & n_1 n_2 & (\ell_1 m_2 + \ell_2 m_1) & (m_1 n_2 + m_2 n_1) & (n_1 \ell_2 + n_2 \ell_1) \\ \hline \ell_2\ell_3 & m_2 m_3 & n_2 n_3 & (\ell_2 m_3 + \ell_3 m_2) & (m_2 n_3 + m_3 n_2) & (n_2 \ell_3 + n_3 \ell_2) \\ \hline \ell_3\ell_1 & m_3 m_1 & n_3 n_1 & (\ell_3 m_1 + \ell_1 m_3) & (m_3 n_1 + m_1 n_3) & (n_3 \ell_1 + n_1 \ell_3) \\ \hline \end{array} \quad \ldots \quad (4.66)$$

If the above matrix $[T_\sigma]$ is partitioned into 3×3 submatrices, matrix $[T_\varepsilon]$ may be written in terms of the submatrices as follows:

$$[T_\sigma] = \left[\begin{array}{c|c} [T_{11}] & [T_{12}] \\ \hline [T_{21}] & [T_{22}] \end{array}\right], \quad |T_\varepsilon| = \left[\begin{array}{c|c} [T_{11}] & [\tfrac{1}{2}T_{12}] \\ \hline [2T_{21}] & [T_{22}] \end{array}\right] \quad \ldots \quad (4.67)$$

As a result of using equations (4.65), the stress and strain components (at each element centroid) in the direction of the global axe, are as follows:

$$
\begin{aligned}
\{\varepsilon\}^m &= \{\varepsilon_x \quad \varepsilon_y \quad \varepsilon_z \quad \varepsilon_{xy} \quad \varepsilon_{xz} \quad \varepsilon_{yz}\}^m \\
\{\sigma\}^m &= \{N_x \quad N_y \quad N_z \quad N_{xy} \quad N_{xz} \quad N_{yz}\}^m \\
\{\varepsilon\}^b &= \{\varepsilon_x \quad \varepsilon_y \quad \varepsilon_z \quad \varepsilon_{xy} \quad \varepsilon_{xz} \quad \varepsilon_{yz}\}^b \\
\{\sigma\}^b &= \{M_x \quad M_y \quad M_z \quad M_{xy} \quad M_{xz} \quad M_{yz}\}^b
\end{aligned}
\qquad \ldots \quad (4.68)
$$

where the superscripts m and b refer to the membrane and bending components respectively.

When the global z-axis is in the same direction with local z-axis, then, the z-components of strains and stresses in equations (4.68) vanish.

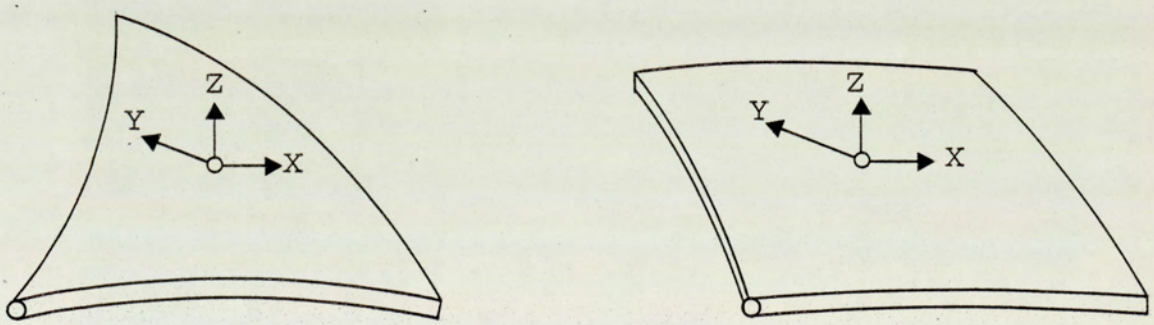


Fig.4.21 Orthogonal axes at the element centroid (stress component definition)

Fig. 4.21 shows the orthogonal local axes X, Y. The X, Y are in the tangent plane to the shell at the point considered, while Z is perpendicular to them. If other points on the same element are taken, the local axes are approximately parallel, except for the fact when the curvature is changed, this results in a change in Z-direction.

In conclusion, the output mentioned in item (i) refers to these directions. This can be transferred to the global axes using equation (4.66). The steps involved are described with reference to the following flow chart shown in Fig. 4.22.

1. Dimension arrays and INT declaration of arrays and variables. The data file E$ is opened by using the ASSIGN≠ statement. Thus the recorded data at the Assembly stage are retrieved by using READ≠ statement.

2. A loop is set for the number of elements. For the first element, the strain-element nodal displacements, the interpolating thickness, the direction cosine of the local axes and the interpolating coordinates of the centre of the first element are read using READ≠ statement.

3. "Eldisp1" subroutine is called to arrange the array containing the displacements of the node, for the current element from the nodal displacements vector {Q}. This is discussed in Section 4.3.2.11.

4. Matrix multiplications are carried out using equations (4.39) and (4.44) to determine the element strains and stresses in the direction of local axes, as well as to print them on the line printer chosen.

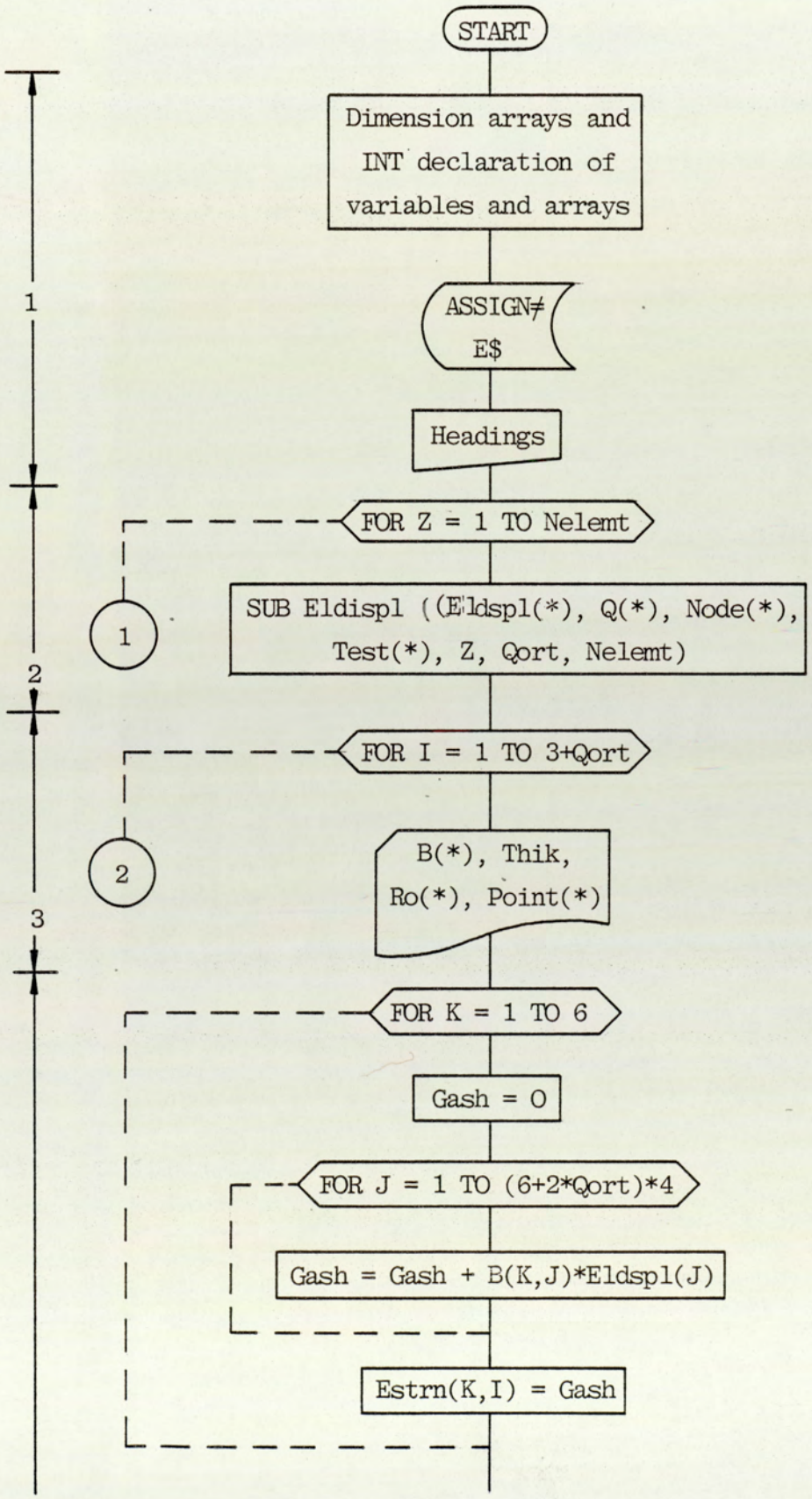
5. The element strains and stresses are transferred to the global axes using the transformation matrices (4.66) and are then printed.

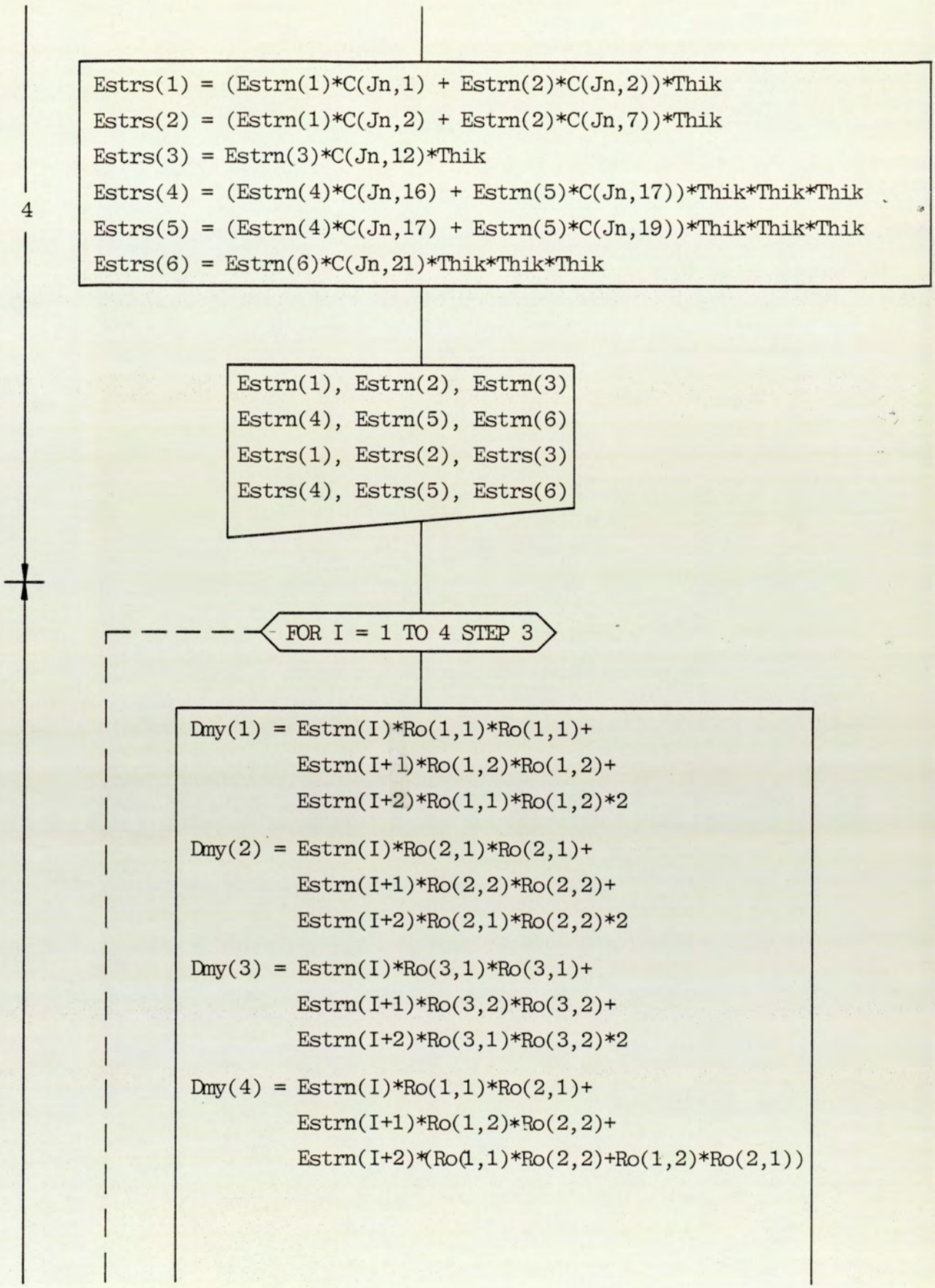
6. Principal stresses are calculated (optionally), as required by the control variables.

Steps 2-6 are repeated for each element of the mesh used.

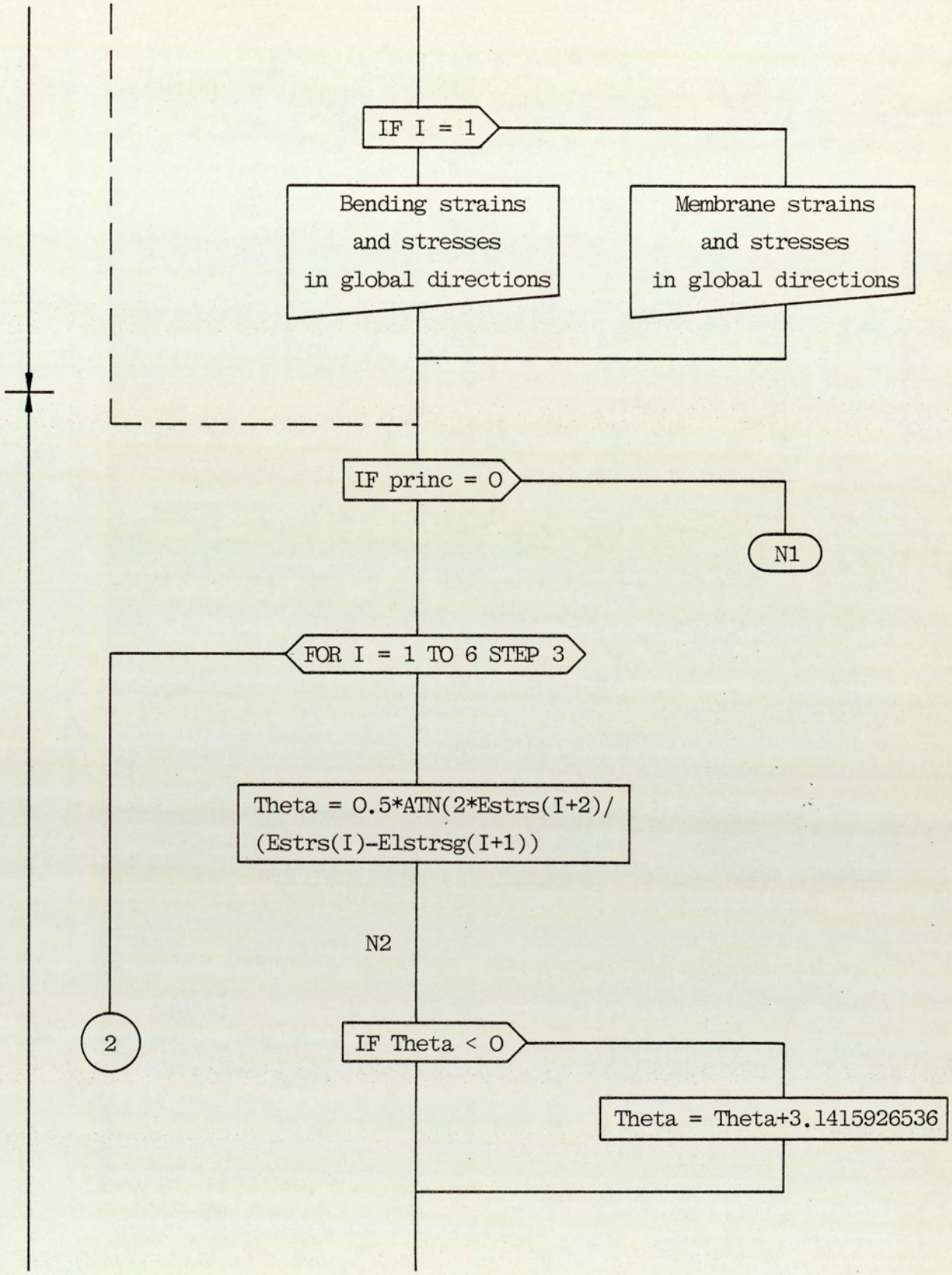
Fig. 4.22 Flow Chart of Subroutine"Elstrs"

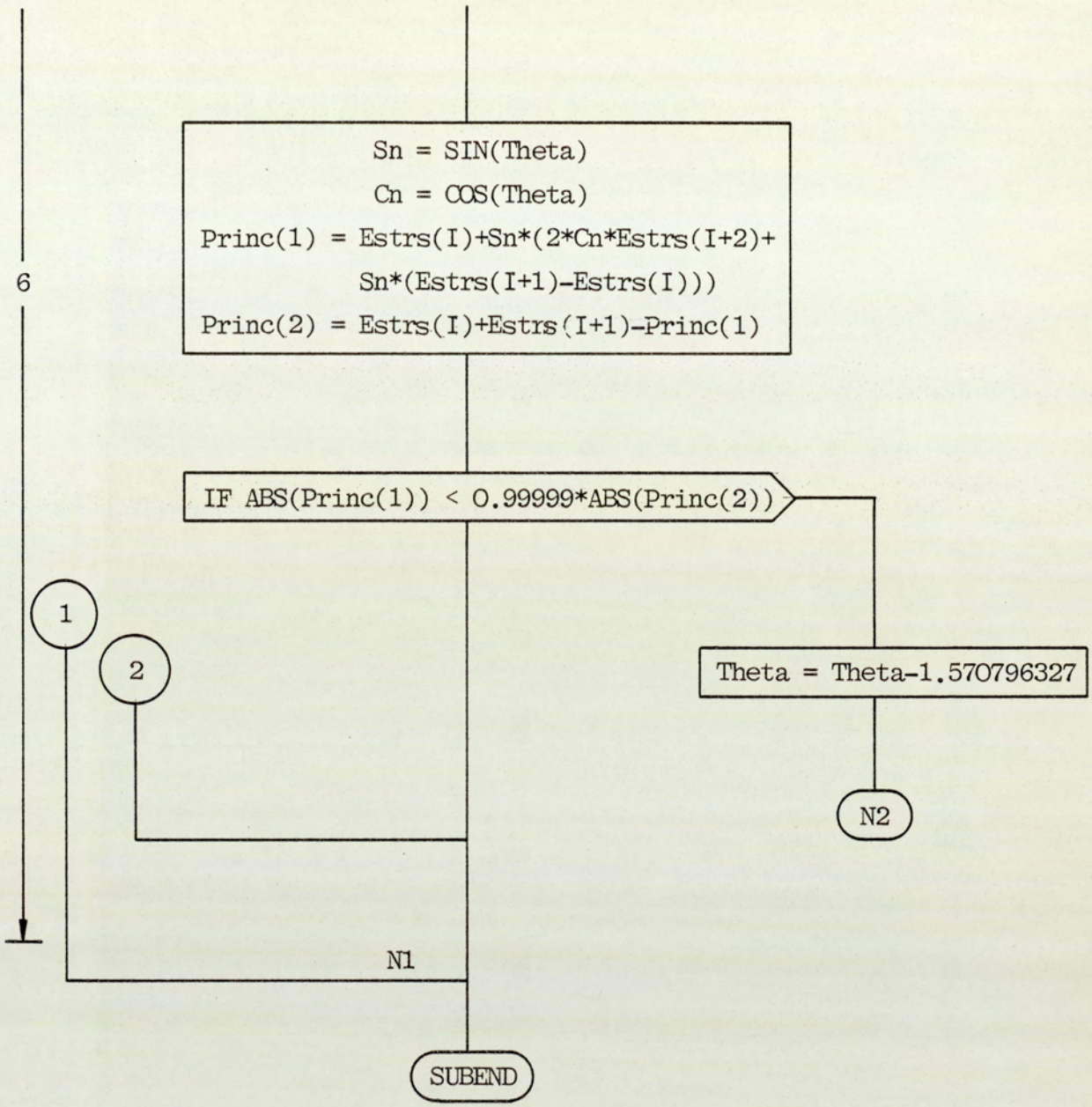
SUB Elstrs (E$,Q(*),B(*),C(*),INTEGER Node(*), Test(*), Nelemt, Nnode, Qort, Princ).




4
Estrs(1) = (Estrn(1)*C(Jn,1) + Estrn(2)*C(Jn,2))*Thik
Estrs(2) = (Estrn(1)*C(Jn,2) + Estrn(2)*C(Jn,7))*Thik
Estrs(3) = Estrn(3)*C(Jn,12)*Thik
Estrs(4) = (Estrn(4)*C(Jn,16) + Estrn(5)*C(Jn,17))*Thik*Thik*Thik
Estrs(5) = (Estrn(4)*C(Jn,17) + Estrn(5)*C(Jn,19))*Thik*Thik*Thik
Estrs(6) = Estrn(6)*C(Jn,21)*Thik*Thik*Thik
Estrn(1), Estrn(2), Estrn(3)
Estrn(4), Estrn(5), Estrn(6)
Estrs(1), Estrs(2), Estrs(3)
Estrs(4), Estrs(5), Estrs(6)
FOR I = 1 TO 4 STEP 3
Dmy(1) = Estrn(I)*Ro(1,1)*Ro(1,1)+
Estrn(I+1)*Ro(1,2)*Ro(1,2)+
Estrn(I+2)*Ro(1,1)*Ro(1,2)*2
Dmy(2) = Estrn(I)*Ro(2,1)*Ro(2,1)+
Estrn(I+1)*Ro(2,2)*Ro(2,2)+
Estrn(I+2)*Ro(2,1)*Ro(2,2)*2
Dmy(3) = Estrn(I)*Ro(3,1)*Ro(3,1)+
Estrn(I+1)*Ro(3,2)*Ro(3,2)+
Estrn(I+2)*Ro(3,1)*Ro(3,2)*2
Dmy(4) = Estrn(I)*Ro(1,1)*Ro(2,1)+
Estrn(I+1)*Ro(1,2)*Ro(2,2)+
Estrn(I+2)*(Ro(1,1)*Ro(2,2)+Ro(1,2)*Ro(2,1))

```
Dmy(5) = Estrn(I)*Ro(2,1)*Ro(3,1)+
         Estrn(I+1)*Ro(2,2)*Ro(3,2)+
         Estrn(I+2)*(Ro(2,1)*Ro(3,2)+Ro(3,1)*Ro(2,2)

Dmy(6) = Estrn(I)*Ro(3,1)*Ro(1,1)+
         Estrn(I+1)*Ro(3,2)*Ro(1,2)+
         Estrn(I+2)*(Ro(3,1)*Ro(1,2)+Ro(3,2)*Ro(1,1))

Dmy(7) = Estrs(I)*Ro(1,1)*Ro(1,1)+
         Estrs(I+1)*Ro(1,2)*Ro(1,2)+
         Estrs(I+2)*Ro(1,1)*Ro(1,2)

Dmy(8) = Estrs(I)*Ro(2,1)*Ro(2,1)+
         Estrs(I+1)*Ro(2,2)*Ro(2,2)+
         Estrs(I+2)*Ro(2,1)*Ro(2,2)

Dmy(9) = Estrs(I)*Ro(3,1)*Ro(3,1)+
         Estrs(I+1)*Ro(3,2)*Ro(3,2)+
         Estrs(I+2)*Ro(3,1)*Ro(3,2)

Dmy(10)= Estrs(I)*Ro(1,1)*Ro(2,1)*2+
         Estrs(I+1)*Ro(1,2)*Ro(2,2)*2+
         Estrs(I+2)*(Ro(1,1)*Ro(2,2)+Ro(1,2)*Ro(2,1) )

Dmy(11)= Estrs(I)*Ro(2,1)*Ro(3,1)*2+
         Estrs(I+1)*Ro(2,2)*Ro(3,2)*2+
         Estrs(I+2)*(Ro(2,1)*Ro(3,2)+Ro(2,2)*Ro(3,1))

Dmy(12)= Estrs(I)*Ro(3,1)*Ro(1,1)*2+
         Estrs(I+1)*Ro(3,2)*Ro(1,2)*2+
         Estrs(I+2)*(Ro(3,1)*Ro(1,2)+Ro(3,2)*Ro(1,1))
```


IF I = 1
Bending strains and stresses in global directions
Membrane strains and stresses in global directions
IF princ = 0
N1
FOR I = 1 TO 6 STEP 3
Theta = 0.5*ATN(2*Estrs(I+2)/ (Estrs(I)-Elstrsg(I+1))
N2
2
IF Theta < 0
Theta = Theta+3.1415926536


6
Sn = SIN(Theta)
Cn = COS(Theta)
Princ(1) = Estrs(I)+Sn*(2*Cn*Estrs(I+2)+
Sn*(Estrs(I+1)-Estrs(I)))
Princ(2) = Estrs(I)+Estrs(I+1)-Princ(1)
IF ABS(Princ(1)) < 0.99999*ABS(Princ(2))
1
2
Theta = Theta-1.570796327
N2
N1
SUBEND

4.3.2.11 Subroutine "Eldips1"

This subroutine is intended to adjust the array {Eldsp1} which contains the displacements of the nodes for the current element from the nodal displacements array {q}. The {Eldsp1} should be adjusted so that the degrees of freedom are ordered according to the sequence of equation (4.36). This is because the strain-element nodal displacements matrix resulted from the "Haloof" subroutine has been formulated according to this sequence and in doing so, the multiplications of equation (4.39) are easily carried out in the subroutine "Elstrs".

The flow chart shows the steps needed to establish the {Eldsp1} array. With reference to the flow chart shown in Fig. 4.23, these steps are as follows:

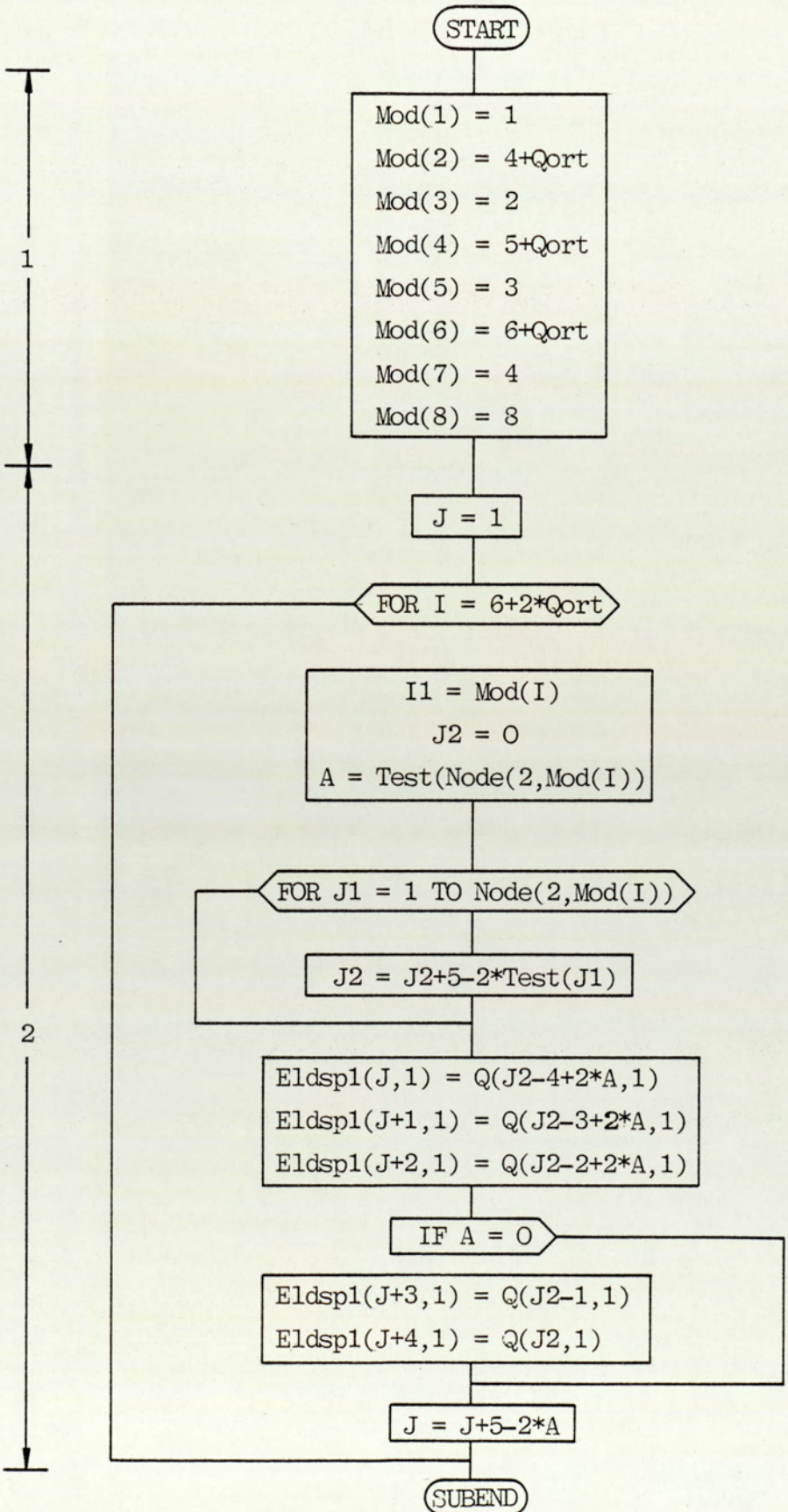
1. This is similar to step 3 mentioned in describing the flow chart of the "Assembly" subroutine. It assists to arrange the coefficients of the {Eldisp1} array according to the sequence of equation (4.36).

2. A loop is set for the number of nodes in the element. For the first node, the displacements are allocated in their proper positions in the array {Eldsp1}.

Step 2 is repeated for the other nodes of the element.

Fig. 4. 23 Flow Chart of Subroutine"Eldisp1"

SUB Eldisp1(Eldisp1(*), Q(*), INTEGER Node(*), Test(*), Z, Qort, Nelemt)



4.3.2.12 Subprogram SMNDST

The subroutine "Nodstr" is written in this subprogram in such a way that the element stresses and strains are calculated at the integrating points within each element and are printed at the element output stage. The strains and stresses are given in terms of local element coordinates, as shown in Fig. 4.24, and are defined in Section 4.3.2.10.

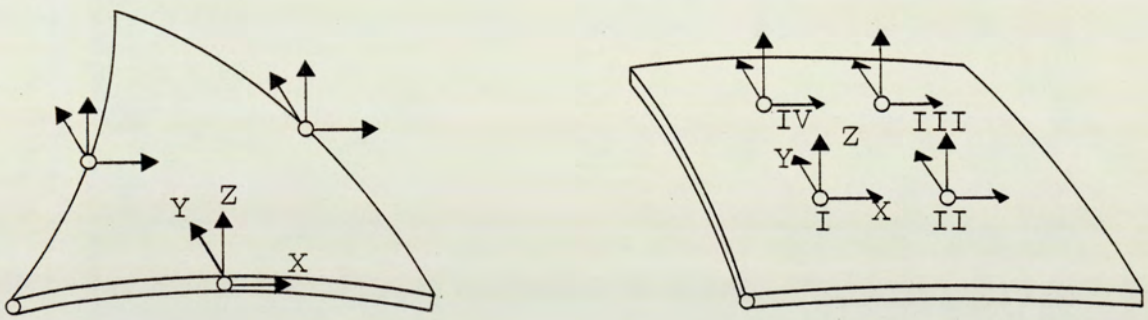


Fig.4.24 Orthogonal axes at the integrating points (stress component definition)

The transformation described in Section 4.3.2.10 for element strains and stresses has been programmed for the integrating points in order to produce them in terms of the global element coordinates. The smoothed nodal values for stresses are calculated using the smoothed stresses technique which is presented in Ref (67) . This makes it possible to obtain the smoothed nodal stresses. This technique can be summarised as follows:

For two-dimensional problems, the stresses are assumed to have a bilinear variation over the element:

$$\sigma(\xi,\eta) = \begin{bmatrix} 1 & \xi & \eta & \xi\eta \end{bmatrix} \begin{Bmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \end{Bmatrix} \qquad \ldots \quad (4.69)$$

If the four integrating points in Fig. 4.25 in the quadrilateral type are considered, then:

$$\begin{Bmatrix} \sigma_I \\ \sigma_{II} \\ \sigma_{III} \\ \sigma_{IV} \end{Bmatrix} = \begin{bmatrix} 1 & -0.592348878 & -0.592348878 & 0.350877 \\ 1 & 0.592348878 & -0.592348878 & 0.350877 \\ 1 & 0.592348878 & 0.592348878 & 0.360877 \\ 1 & -0.592348878 & 0.592348878 & 0.350877 \end{bmatrix} \begin{Bmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \end{Bmatrix}$$

$$\ldots \quad (4.70)$$

The smoothed nodal values for stresses may now be calculated by substituting equation (4.70) into equation (4.69). This gives:

$$\begin{Bmatrix} \sigma_1 \\ \sigma_2 \\ \sigma_3 \\ \sigma_4 \end{Bmatrix} = \begin{bmatrix} 1.806607 & -0.4625098 & 0.118412707 & -0.4625098 \\ -0.4625098 & 1.806607 & -0.4265048 & 0.118412707 \\ 0.118412707 & -0.4625098 & 1.806607 & -0.4625098 \\ -0.4625098 & 0.118412707 & -0.4625098 & 1.806607 \end{bmatrix} \begin{Bmatrix} \sigma_I \\ \sigma_{II} \\ \sigma_{III} \\ \sigma_{IV} \end{Bmatrix}$$

or

$$\{\sigma\}_{\text{smoothed}} = [A]\ \{\sigma\}_{\text{intg. points}} \qquad \ldots \quad (4.71)$$

where $[A]$ is the smoothing matrix based on the mean square average.

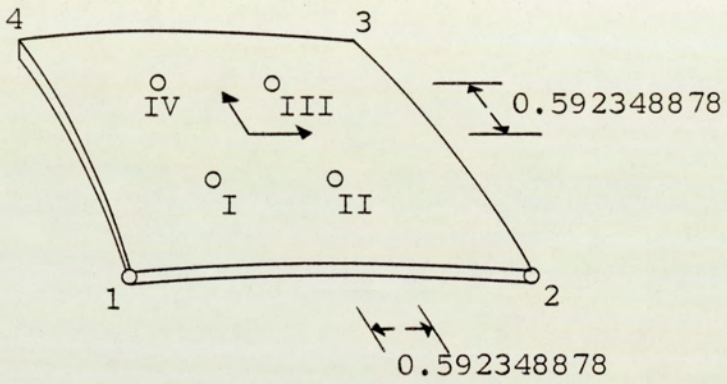


Fig.4.25 Location of integrating points

Now, stresses calculated at the four integrating points are multiplied by the smoothing matrix $[A]$ to give smoothed values at the vertices, i.e. matrix multiplications of equation (4.71). Smoothed values from adjacent elements are then averaged at the element vertices, while the midside node smoothed stresses may be found by averaging the smoothed stresses for corner nodes.

The flow chart, show in Fig. 4.26, and the steps involved are as follows:

1. Dimension arrays and INT declaration of arrays and variables. The data files N$ is opened by using the ASSIGN# statement. Thus the recorded data at the Assembly stage are retrieved by using the READ# statement.

2. This step is similar to step 2 in the flow chart which was drawn in Section 4.3.2.10, except that a new loop for the number of integrating points is set inside the number of elements loop.

Steps 3, 4, 5 and 6 are similar to those in Section 4.3.2.10.

7. Smoothed strains and stresses at the corner nodes for the current element are evaluated. This implies carrying out the matrix multiplications of equation (4.71).

8. A loop is set round the number of nodes in the mesh. For the first corner node each element is searched to find if it contains this node, so as to divide the values of smoothed strains and stresses by the number of elements containing this node. At the end of this step all the smoothed strains and stresses are found for the corner nodes only.

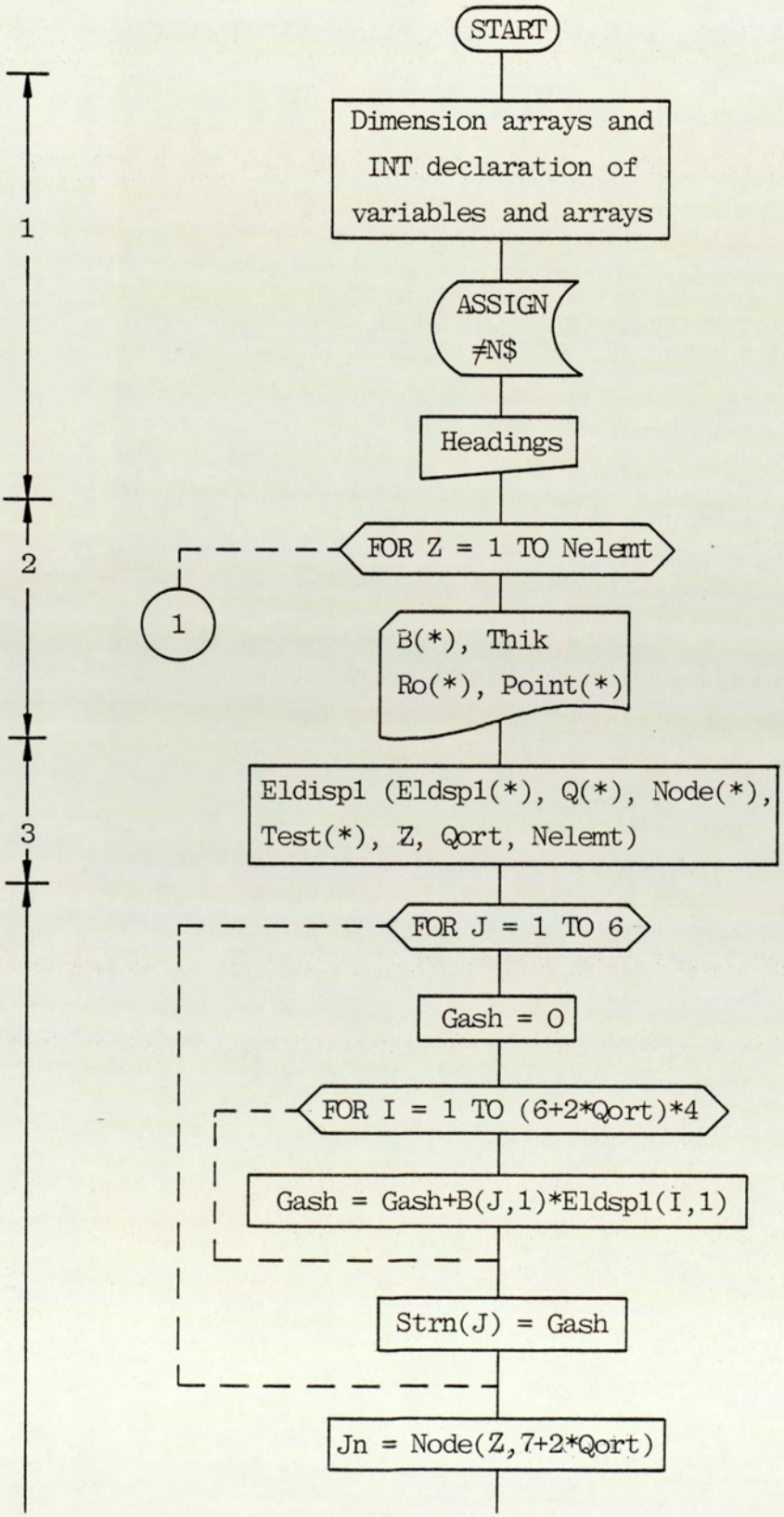
9. This step is designed to evaluate the smoothed strains and stresses for the midside nodes in a process similar

to that applied in step 8.

10. The smoothed strains and stresses are printed on the line printer chosen.

Fig. 4.26 Flow Chart of Subroutine"Nodstr"

SUB Nodstr (N$, Q(*), B(*), C(*), INTEGER Node(*), Test(*), Nelemt, Nnode, Qort, Princ)



```
Strs(1,I) = (Strn(1,I)*C(Jn,1)+Strn(2,I)*
            C(Jn,2))*Thik
Strs(2,I) = (Strn(1,I)*C(Jn,2)+Strn(2,I)*
            C(Jn,7))*Thik
Strs(3,I) = Strn(3,I)*C(Jn,12)*Thik
Strs(4,I) = (Strn(4,I)*C(Jn,16)+Strn(5,I)*
            C(Jn,17))*Thik*Thik*Thik
Strs(5,I) = (Strn(4,I)*C(Jn,17)+Strn(5,I)*
            C(Jn,19))*Thik*Thik*Thik
Strs(6,I) = Strn(6,I)*C(Jn,21)*Thik*Thik*Thik
```

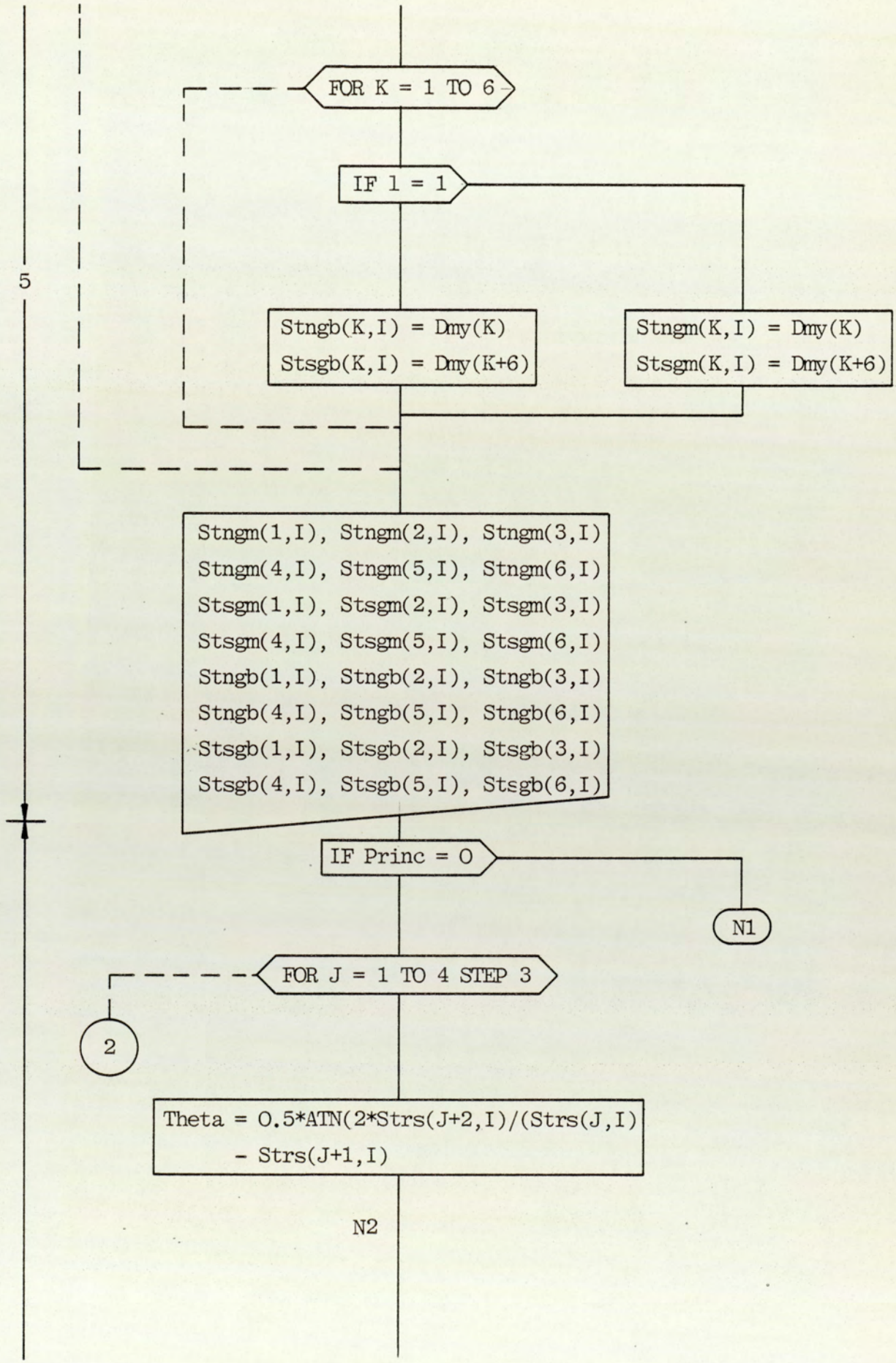
4

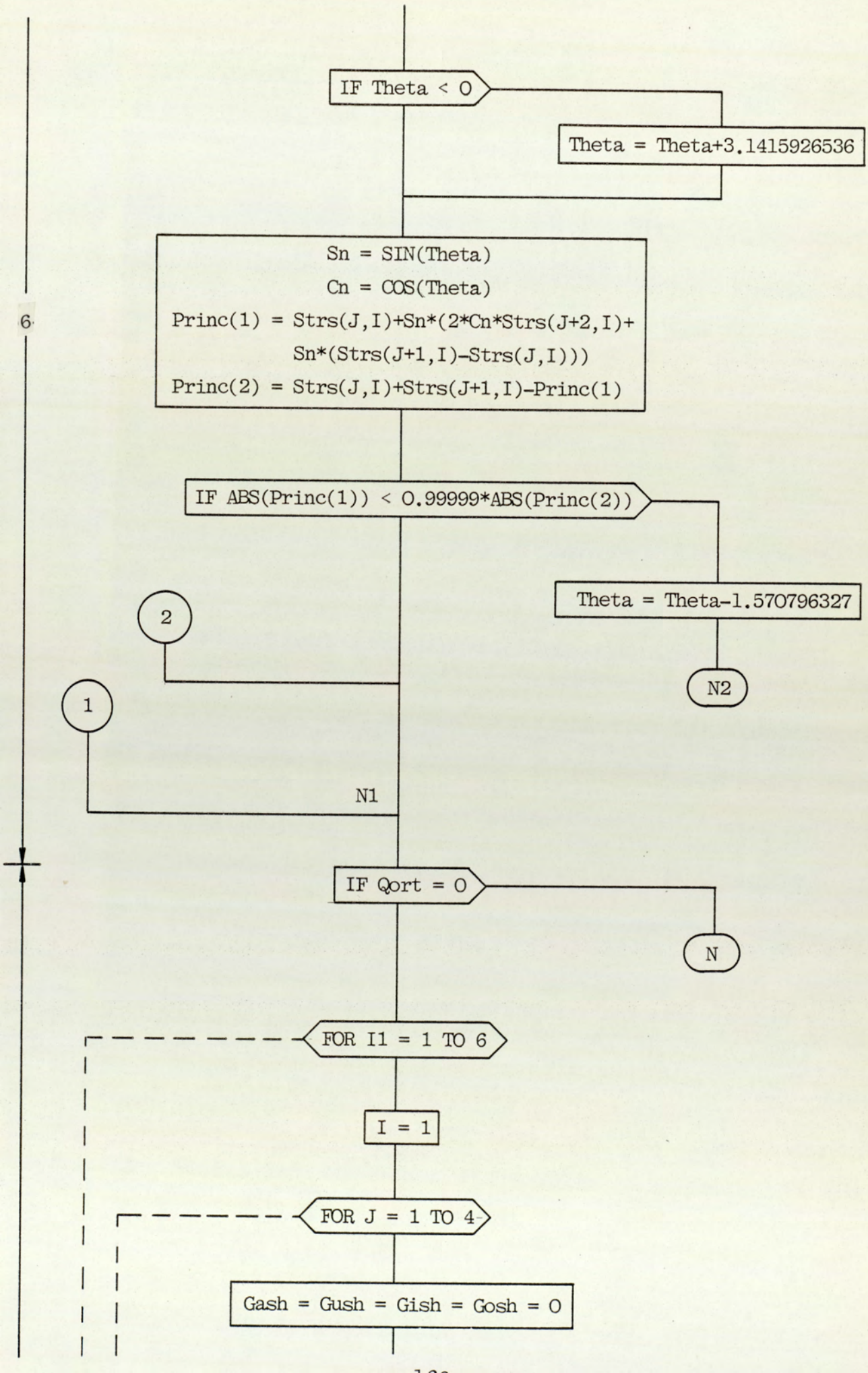
```
Strn(1,I), Strn(2,I), Strn(3,I)
Strn(4,I), Strn(5,I), Strn(6,I)
Strs(1,I), Strs(2,I), Strs(3,I)
Strs(4,I), Stres(5,I),
Strs(6,I)
```

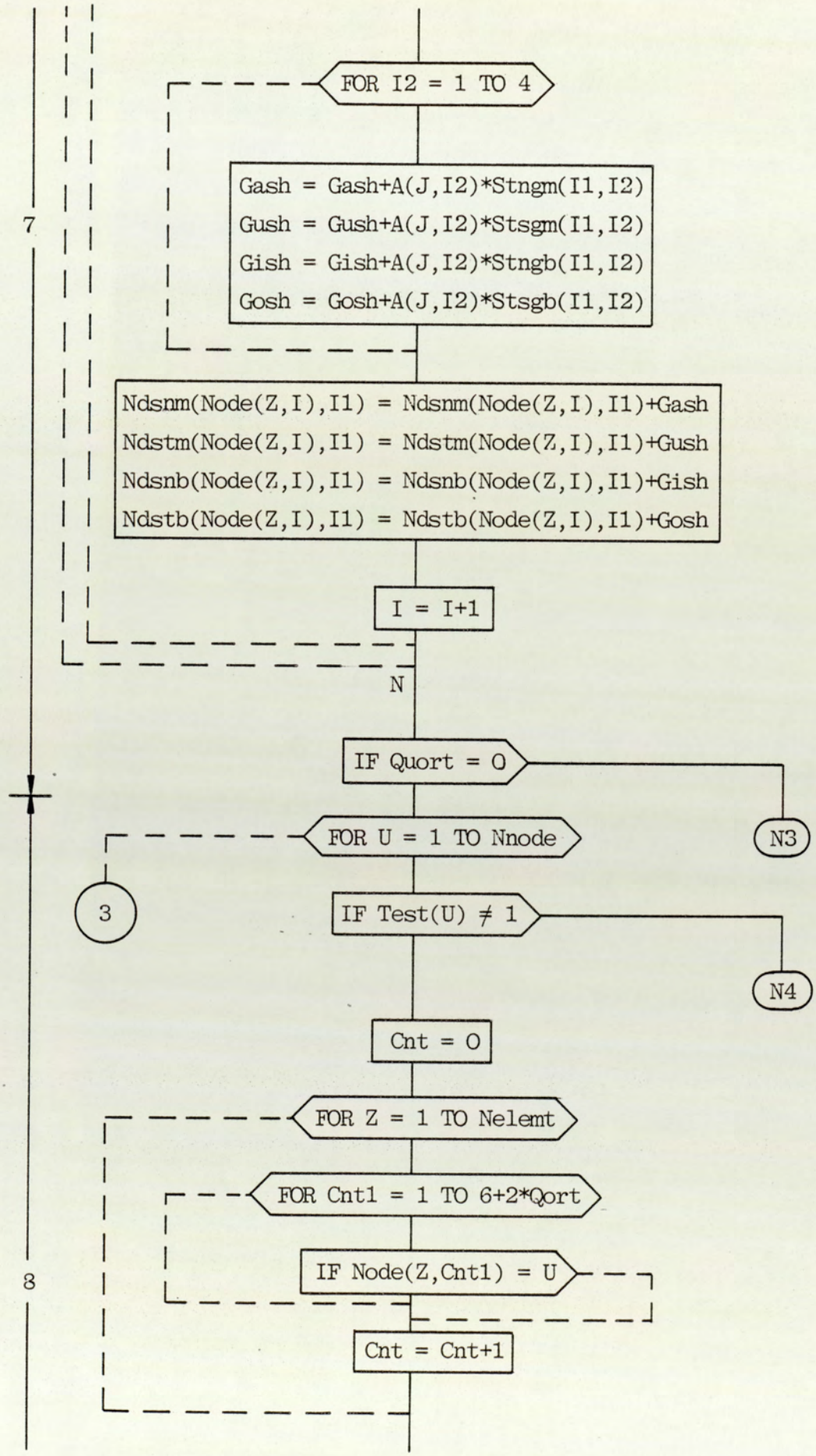
```
FOR L = 1 TO 4 STEP 3
```

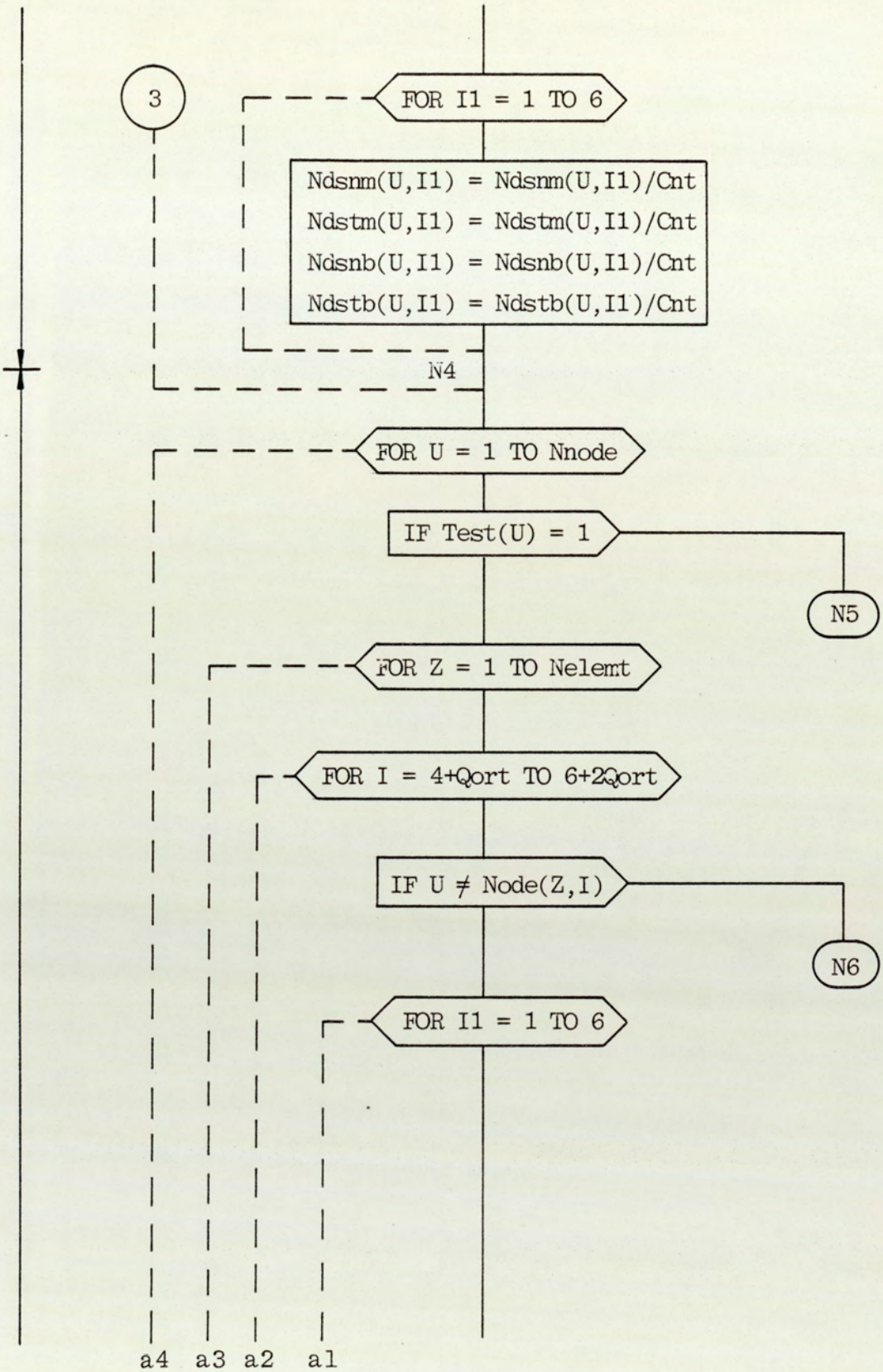
```
Dmy(1) = Strn(L,I)*Ro(1,1)*Ro(1,1)+Strn(L+1,I)*
         Ro(1,2)*Ro(1,2)+Strn(L+2,I)*Ro(1,1)*
         Ro(1,2)*2
Dmy(2) = Strn(L,I)*Ro(2,1)*Ro(2,1)+
         Strn(L+2,I)*Ro(2,2)*Ro(2,2)+
         Strn(L+2,I)*Ro(2,1)*Ro(2,2)*2

Dmy(3) = Strn(L,I)*Ro(3,1)*Ro(3,1)+
         Strn(L+1,I)*Ro(3,2)*Ro(3,2)+
         Strn(L+2,I)*Ro(3,1)*Ro(3,2)*2
```

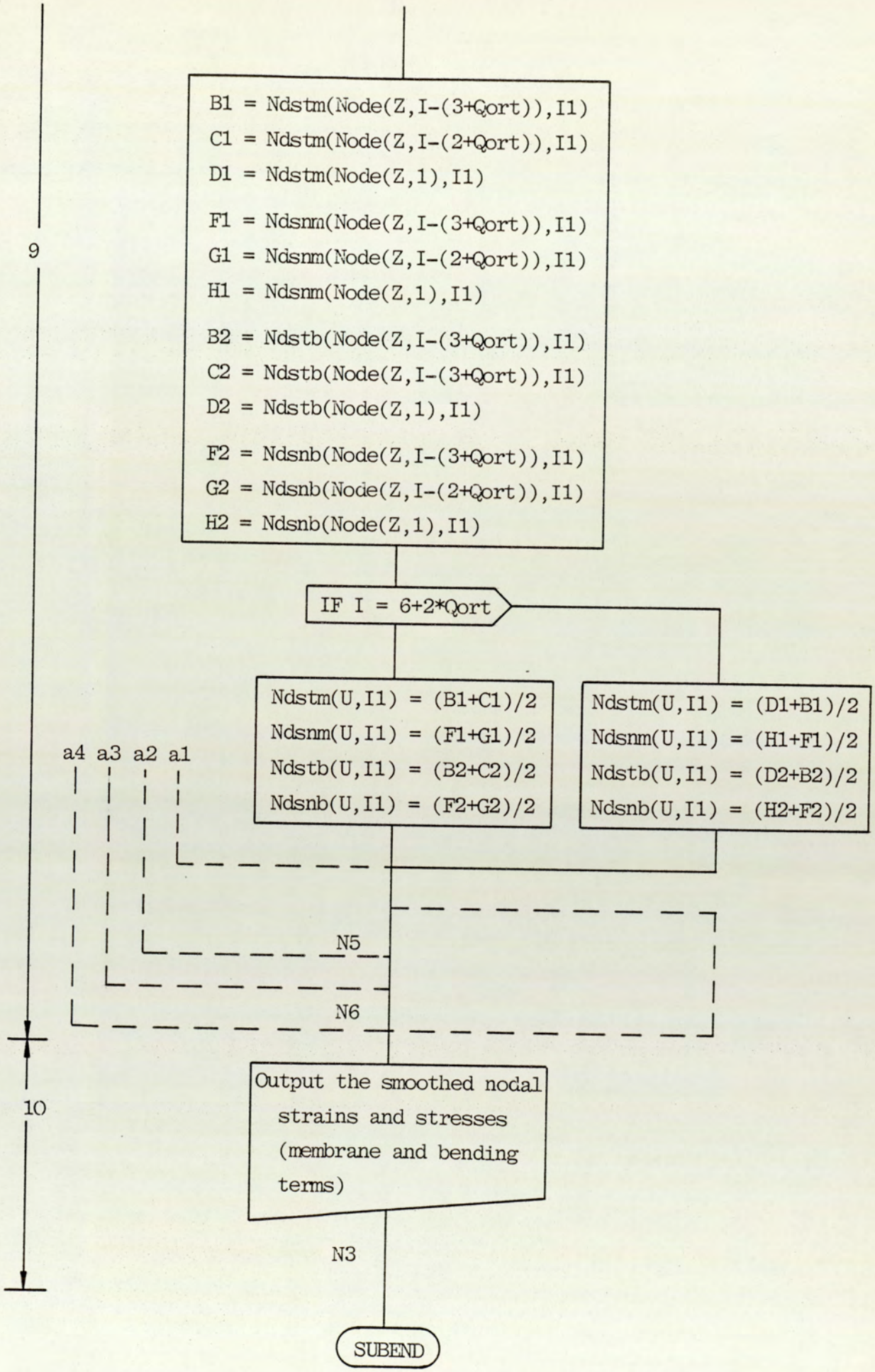
```
Dmy(4) =  Strn(L,I)*Ro(1,1)*Ro(2,1)+
          Strn(L+1,I)*Ro(1,2)*Ro(2,2)+
          Strn(L+2,I)*(Ro(1,1)*Ro(2,2)+
          Ro(1,2)*Ro(2,1))

Dmy(5) =  Strn(L,I)*Ro(2,1)*Ro(3,1)+
          Strn(L+1,I)*Ro(2,2)*Ro(3,2)+
          Strn(L+2,I)*(Ro(2,1)*Ro(3,2)+
          Ro(3,1)*Ro(2,2))

Dmy(6) =  Strn(L,I)+Ro(3,1)*Ro(1,1)+
          Strn(L+1,I)*Ro(3,2)+Ro(1,2)+
          Strn(L+2,I)*(Ro(3,1)*Ro(1,2)+
          Ro(3,2)*Ro(1,1))

Dmy(7) =  Strs(L,I)*Ro(1,1)*Ro(1,1)+
          Strs(L+1,I)*Ro(1,2)*Ro(1,2)+
          Strs(L+2,I)*Ro(1,1)*Ro(1,2)

Dmy(8) =  Strs(L,I)*Ro(2,1)*Ro(2,1)+
          Strs(L+1,I)*Ro(2,2)*Ro(2,2)+
          Strs(L+2,I)*Ro(2,1)*Ro(2,2)

Dmy(9) =  Strs(L,I)*Ro(3,1)*Ro(3,1)+
          Strs(L+1,I)*Ro(3,2)*Ro(3,2)+
          Strs(L+2,I)*Ro(3,1)*Ro(3,2)

Dmy(10)=  Strs(L,I)*Ro(1,1)*Ro(2,1)*2+
          Strs(L+1,I)*Ro(1,2)*Ro(2,2)*2+
          Strs(L+2,I)*(Ro(1,1)*Ro(2,2)+
          Ro(1,2)*Ro(2,1))

Dmy(11)=  Strs(L,I)*Ro(2,1)*Ro(3,1)*2+
          Strs(L+1,I)*Ro(2,2)*Ro(3,2)*2+
          Strs(L+2,I)*(Ro(2,1)*Ro(3,2)+
          Ro(2,2)*Ro(3,1))

Dmy(12)=  Strs(L,I)*Ro(3,1)*Ro(1,1)*2+
          Strs(L+1,I)*Ro(3,2)*Ro(1,2)*2+
          Strs(L+2,I)*(Ro(3,1)*Ro(1,2)+
          Ro(3,2)*Ro(1,1))
```


5
FOR K = 1 TO 6
IF 1 = 1
Stngb(K,I) = Dmy(K)
Stsgb(K,I) = Dmy(K+6)
Stngm(K,I) = Dmy(K)
Stsgm(K,I) = Dmy(K+6)
Stngm(1,I), Stngm(2,I), Stngm(3,I)
Stngm(4,I), Stngm(5,I), Stngm(6,I)
Stsgm(1,I), Stsgm(2,I), Stsgm(3,I)
Stsgm(4,I), Stsgm(5,I), Stsgm(6,I)
Stngb(1,I), Stngb(2,I), Stngb(3,I)
Stngb(4,I), Stngb(5,I), Stngb(6,I)
Stsgb(1,I), Stsgb(2,I), Stsgb(3,I)
Stsgb(4,I), Stsgb(5,I), Stsgb(6,I)
IF Princ = 0
N1
FOR J = 1 TO 4 STEP 3
2
Theta = 0.5*ATN(2*Strs(J+2,I)/(Strs(J,I)
- Strs(J+1,I)
N2


IF Theta < 0
Theta = Theta+3.1415926536
Sn = SIN(Theta)
Cn = COS(Theta)
Princ(1) = Strs(J,I)+Sn*(2*Cn*Strs(J+2,I)+
Sn*(Strs(J+1,I)-Strs(J,I)))
Princ(2) = Strs(J,I)+Strs(J+1,I)-Princ(1)
IF ABS(Princ(1)) < 0.99999*ABS(Princ(2))
Theta = Theta-1.570796327
N2
2
1
N1
IF Qort = 0
N
FOR I1 = 1 TO 6
I = 1
FOR J = 1 TO 4
Gash = Gush = Gish = Gosh = 0


FOR I2 = 1 TO 4
Gash = Gash+A(J,I2)*Stngm(I1,I2)
Gush = Gush+A(J,I2)*Stsgm(I1,I2)
Gish = Gish+A(J,I2)*Stngb(I1,I2)
Gosh = Gosh+A(J,I2)*Stsgb(I1,I2)
Ndsnm(Node(Z,I),I1) = Ndsnm(Node(Z,I),I1)+Gash
Ndstm(Node(Z,I),I1) = Ndstm(Node(Z,I),I1)+Gush
Ndsnb(Node(Z,I),I1) = Ndsnb(Node(Z,I),I1)+Gish
Ndstb(Node(Z,I),I1) = Ndstb(Node(Z,I),I1)+Gosh
I = I+1
N
IF Quort = 0
FOR U = 1 TO Nnode
N3
3
IF Test(U) ≠ 1
N4
Cnt = 0
FOR Z = 1 TO Nelemt
FOR Cnt1 = 1 TO 6+2*Qort
IF Node(Z,Cnt1) = U
Cnt = Cnt+1
7
8


3
FOR I1 = 1 TO 6
Ndsnm(U,I1) = Ndsnm(U,I1)/Cnt
Ndstm(U,I1) = Ndstm(U,I1)/Cnt
Ndsnb(U,I1) = Ndsnb(U,I1)/Cnt
Ndstb(U,I1) = Ndstb(U,I1)/Cnt
N4
FOR U = 1 TO Nnode
IF Test(U) = 1
N5
FOR Z = 1 TO Nelemt
FOR I = 4+Qort TO 6+2Qort
IF U ≠ Node(Z,I)
N6
FOR I1 = 1 TO 6
a4
a3
a2
a1


B1 = Ndstm(Node(Z,I-(3+Qort)),I1)
C1 = Ndstm(Node(Z,I-(2+Qort)),I1)
D1 = Ndstm(Node(Z,1),I1)
F1 = Ndsnm(Node(Z,I-(3+Qort)),I1)
G1 = Ndsnm(Node(Z,I-(2+Qort)),I1)
H1 = Ndsnm(Node(Z,1),I1)
B2 = Ndstb(Node(Z,I-(3+Qort)),I1)
C2 = Ndstb(Node(Z,I-(3+Qort)),I1)
D2 = Ndstb(Node(Z,1),I1)
F2 = Ndsnb(Node(Z,I-(3+Qort)),I1)
G2 = Ndsnb(Node(Z,I-(2+Qort)),I1)
H2 = Ndsnb(Node(Z,1),I1)
IF I = 6+2*Qort
Ndstm(U,I1) = (B1+C1)/2
Ndsnm(U,I1) = (F1+G1)/2
Ndstb(U,I1) = (B2+C2)/2
Ndsnb(U,I1) = (F2+G2)/2
Ndstm(U,I1) = (D1+B1)/2
Ndsnm(U,I1) = (H1+F1)/2
Ndstb(U,I1) = (D2+B2)/2
Ndsnb(U,I1) = (H2+F2)/2
a4 a3 a2 a1
N5
N6
Output the smoothed nodal strains and stresses (membrane and bending terms)
N3
SUBEND
9
10

4.3.3 Modifications of plate and shell structure programs to accept a segmental solving routine

In complicated plate and shell structures, the mesh used should be fine, in order to obtain reasonable results. The sizes of the stiffness matrices resulting from such problems are large and cannot be held in the memory of the desk top computer such as HP9845B.

From equations (4.62) and (4.63), it is apparent that to form any coefficient L_{ij}, only rows (i) and (j) of the main sequence need to be in the memory of the computer.

On this basis Jennings and Tuff, Ref(63) proposed a method of storing the system stiffness matrix $[K]$ in a segmental form in some mass storage facility. Only two segments are required in the memory at any one time. Wood, Ref. (68)has constructed a solving routine based on Jennings and Tuff's idea to solve fracture mechanics problems. The backing store in the desk top computer HP9845B are; two cartridges and one flexible disc. More storage devices may be connected if necessary. The maximum number of segments cannot exceed 10. This is a limitation of the operating system (see Appendix A). There are also time penalties in adopting this method of storage. Large segments are more efficient as less time would be used in transfer operations.

Inspite of all these difficulties, the solving routine "Segsol" (see Ref. (68)) has been developed as a means of overcoming the problem of limited memory facility. Consequently, the subroutine "Addarray" is modified to be used with backing store as well as the subroutine "Assembly". The subroutine "Addarray" modification involves the division of the Add array into segments, each containing approximately the same number of coefficients and also an integral number of rows. The largest segment has been limited to 10000 coefficients. The number of segments may be found by dividing the number of coefficients of [K] into subarrays each containing approximately 10000 coefficients. The limits of each segment is computed. The maximum segment size is found in order that sufficient computer memory can be made available.

The subroutine "Assembly" modification involves a check to determine whether the coefficient lies within the bound of the segment in store. This is because only one segment is held in the memory of the computer. The subroutine "Test" does this and also exchanges the segment in store with the required segment in the backing store.

A comprehensive discussion of the backing store technique together with the flow chart of the solving routine is given in (Ref.(68)). The listing of the modified programs is given in Appendix B. These are

presented jointly with the programs dealing with the sectorially symmetric structures, which are discussed in Chapter 6.

4.4 CLOSING REMARKS

In this chapter, the theory of the semiloof shell element as developed by Irons, Ref.(53), has been discussed. The development of a general finite element package for plate and shell structures using the semiloof shell element has been described. This package was implemented on the HP9845B desk top computer. The master Program SMILOF was developed and stored in a separate file together with the following subprograms, which are independently saved into data files in order to save some memory storage. (SMINPT, SMCNTR, SMASBL, SMLDAP, SMSKEW, SMYVBS, SMELST, SMNDST). Only one subprogram is linked to the main program to perform a special task at a time using the "LINK" statement.

For larger problems, the programs were modified to be used together with the P.Wood segmental solving routine "Segsol".

The package has been tested against some known solutions and this is the subject of the next chapter.

CHAPTER FIVE

NUMERICAL EXAMPLES

5.1 INTRODUCTION

The semiloof shell element described in Chapter 4 has been applied to the solution of a number of thin plate and shell structures. Various examples are tested to establish the generality and validity of this element to deal with such structures. In order to see the performance of the computer programs, the selected problems do not include complex shell geometry, due to the need to provide comparison with available solutions. On the other hand, these examples cover different types of loading and boundary conditions, such as the structures may contain:

(i) curved boundaries

(ii) clamped edges, simply supported edges,...etc.

(iii) uniformly distributed loads, concentrated loads, gravity effects and in-plane tensile loading.

A number of complex shell geometries with multiple junctions were tested. These do not have exact solutions. Two of the above structures have been compared with the experimental results which are presented in Chapter 7. In this chapter two complex shell geometries with multiple junctions are examined by the developed finite element programs. These are to show the rate of convergence of the semiloof shell element in dealing with sharp corners

and multiple junctions.

In the given examples, except the first one, the semiloof shell element quadrilateral type was used. This is because the smoothed nodal stresses are needed to be compared with the published results. Triangular type may be used, but the comparison will be with fewer points since the stresses here are obtained at the midside nodes only (integrating points).

In all the examples tested, the output of the computer programs are as follows:

(i) The nodal displacements (u, v and w) in the direction of the global axes (x, y and z) respectively and the loof nodes rotations. These resulted from the solving routine.

(ii) The subprogram SMELST output are:

(a) The membrane strains and stresses (ε_X^m, ε_Y^m, ε_{XY}^m, N_X, N_Y and N_{XY}) and, the bending strains and stresses (ε_X^b, ε_Y^b, ε_{XY}^b, M_X, M_Y and M_{XY}). These are calculated and printed at each element centroid in terms of the local element coordinates (see Section 4.3.2.10).

(b) The results of (a) are calculated in terms of the global element coordinates.

(c) The principal stresses (tensions and bending moments and their directions)are computed at each element centroid.

(iii) The subprogram "SMNDST" outputs are:

(a) The membrane and bending strains and stresses as defined in item (ii) for each integrating point within each element in terms of the local element coordinates. Direction cosines of the local axes with respect to the global ones and the nodal coordinates for the integrating point are also printed.

(b) The results of (a) are computed in terms of the global element coordinates.

(c) For each integrating point within each element, the principal stresses (membrane and bending moments) and their directions are determined.

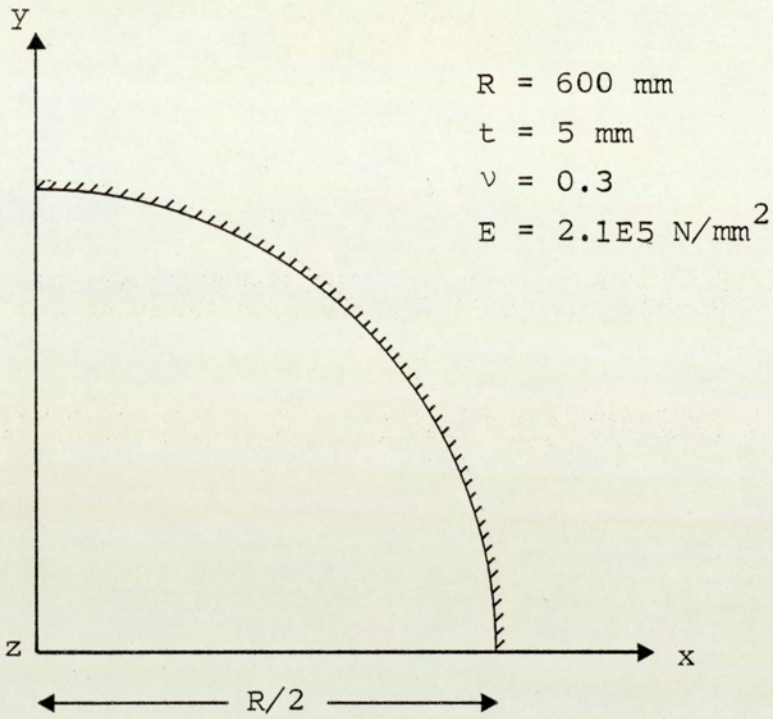
(d) If the quadrilateral element is used, the smoothed nodal strains and stresses are calculated and printed in terms of the global element coordinates.

In the following chapter several examples are presented to demonstrate the accuracy obtainable from the finite element programs developed in Chapter 4.

5.2 CLAMPED DISC UNDER UNIFORM TRANSVERSE LOAD

This is a simple example to demonstrate the usefulness of the semiloof shell element when used to idealise curved boundaries in structures. Because of symmetry conditions only one quarter of the disc is idealised. Figure 5.1 shows the geometry and meshes used. The deflection and bending moment profiles were compared with the exact solution of Timoshenko, Ref. (1). Accurate prediction of the bending moments and the displacements were obtained using only three quadrilateral elements or five triangular elements as shown in Fig. 5.2.

This demonstrates the ability of the semiloof shell element adequately to represent the curved boundaries of thin plates.



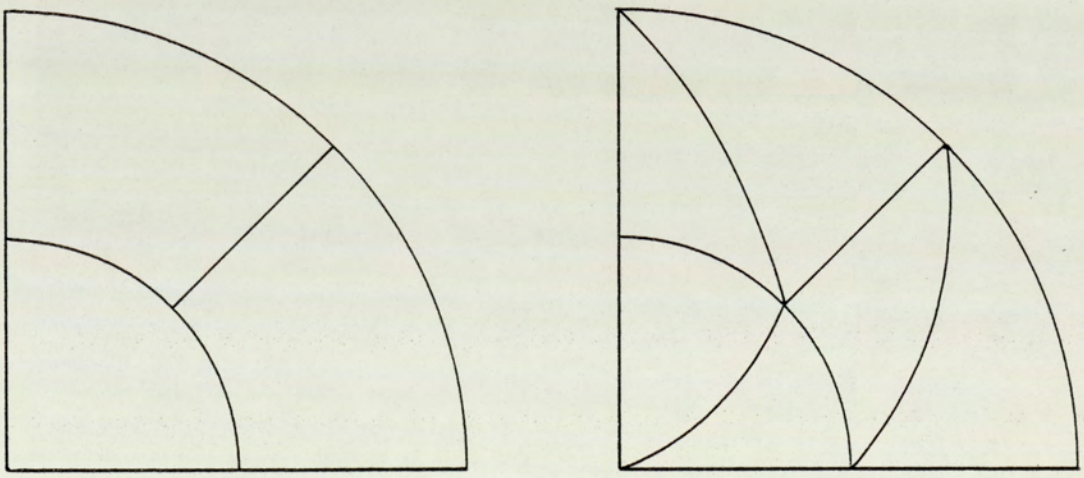


Fig.5.1 Geometry and the finite element meshes used for clamped disc under uniform load

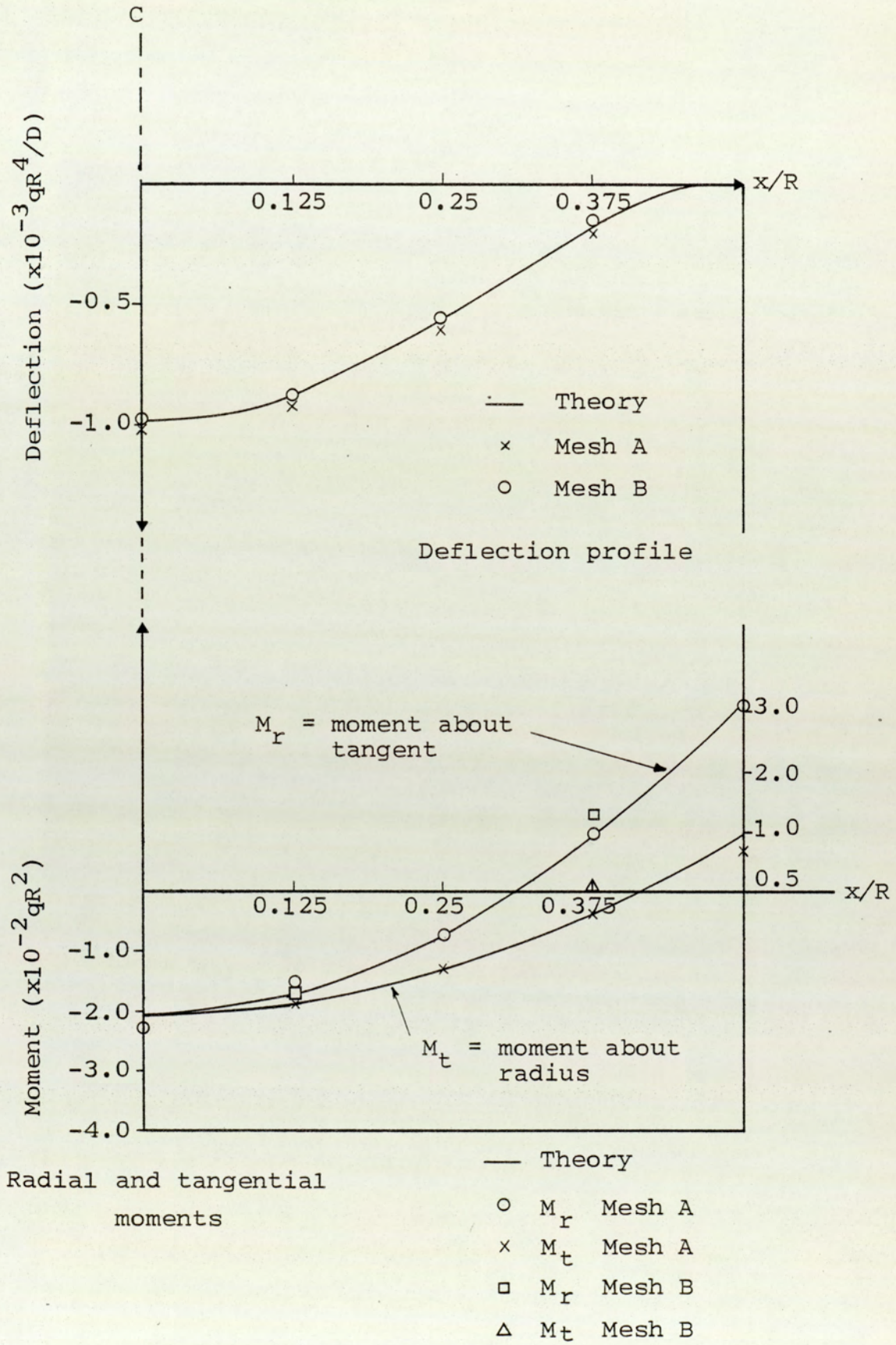


Fig. 5.2 Clamped disc under uniform distributed load. Deflection and bending moments along a diametral section.

5.3 SQUARE PLATE CONVERGENCE STUDIES

The following convergence studies have been carried out for a simple case of a thin plate of side ℓ, Poissons ratio ν, and flexural rigidity, D. (see Fig. 5.3). The boundary conditions are taken as:

(i) simply supported edges.

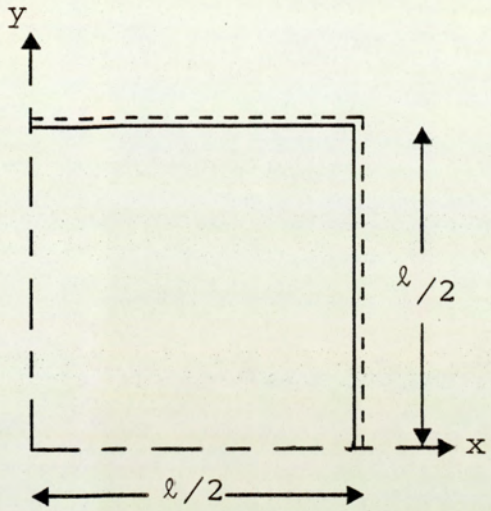
(ii) clamped edges

and the load cases are:

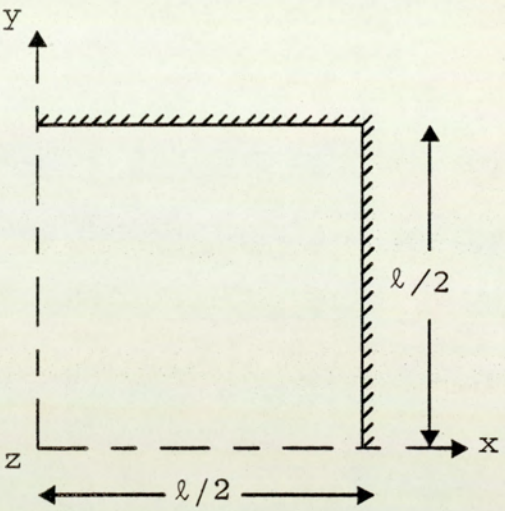
(i) uniformly distributed load

(ii) central concentrated load.

Because of symmetry conditions, only one quarter of the plate needed to be considered. Four uniform meshes were used. These were: (1×1), (2×2), (3×3), and (4×4). In order to study the rate of convergence, some of the results are plotted, others are compared numerically with the exact solution given by Timoshenko, Ref.(1) as shown in Tables (5.1) and (5.2). In all cases, the numerical results presented in Tables (5.1) and (5.2) indicate a rapid rate of convergence to the exact solution given by Timoshenko for both displacements and stresses. Also, it is clear that the comparison drawn in Figs. 5.4 - 5.6 shows that the semiloof shell element approaches the exact solution with a minimum number of degrees of freedom.



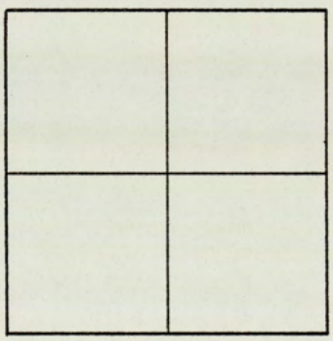
Simply supported plate



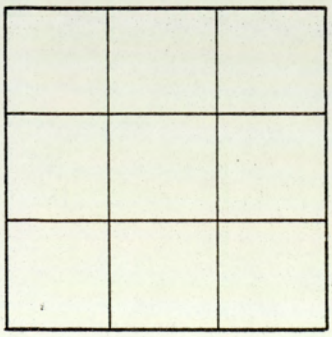
Clamped square plate

Flexural rigidity $D = \frac{E t^3}{12(1 - \nu^2)}$

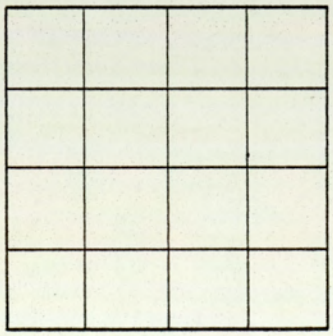
(1x1) Mesh



(2x2) Mesh



(3x3) Mesh



(4x4) Mesh

Fig.5.3 Square plate convergence studies

Mesh in Symmetric Quarter		(1×1)	(2×2)	(3×3)	(4×4)	Theory Ref(1)
Uniformly Distributed Load (q)	Central deflection $\times 10^6 \times \frac{D}{q\ell^4}$	1442	1264	1263	1262	1260
	Edge Bending Moment M_y $\times 10^5 \times \frac{1}{q\ell^2}$	3543	4638	4883	4953	5130
	Central Bending Moment M_x $\times 10^5 \times \frac{1}{q\ell^2}$	2484	2674	2454	2379	2310
Concentrated Central Load (P)	Central deflection $\times 10^6 \times \frac{D}{P\ell^2}$	7025	6071	5859	5768	5600
	Edge Bending Moment M_y $\times 10^4 \times \frac{1}{P}$	1564	1187	1280	1258	1257

Table (5.1) Clamped Square Plate Under Uniformly Distributed Load (q) and Concentrated Load (P)

Mesh in Symmetric Quarter		(1×1)	(2×2)	(3×3)	(4×4)	Theory Ref(1)
Uniformly distributed load (q)	Central deflection $\times 10^5 \frac{D}{q\ell^4}$	4085	4046	4054	4057	4060
	Central bending moment, M_x $\times 10^5 \frac{1}{q\ell^2}$	6835	5134	4937	4871	4791
Concentrated Central load (p)	Central deflection $\times 10^5 \frac{D}{p\ell^2}$	1315	1212	1188	1177	1160

Table 5.2 Simply Supported Square Plate Under Uniformly Distributed Load (q) and Concentrated Load (p)

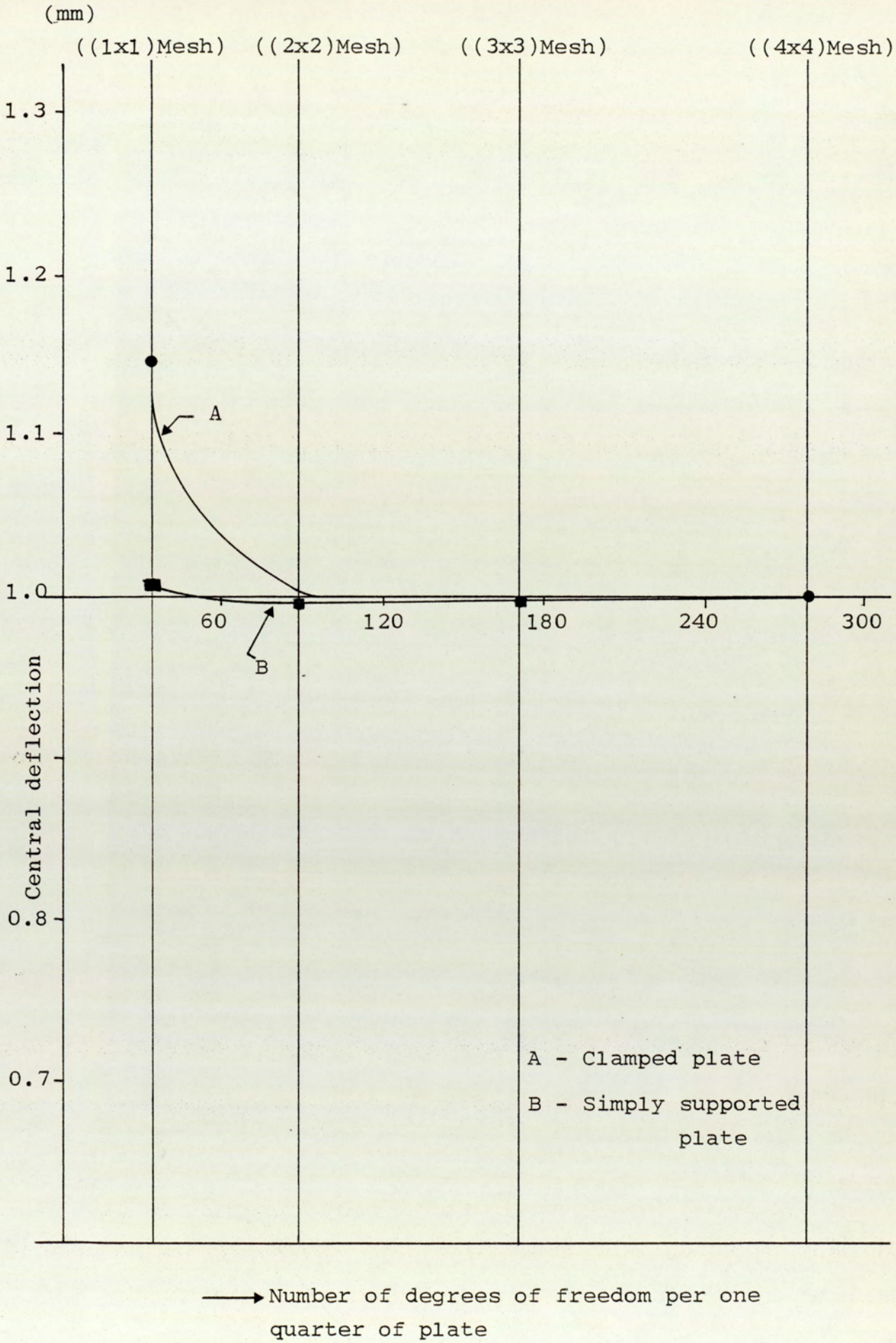


Fig.5.4 Convergence of deflections for a square plate with uniform loading

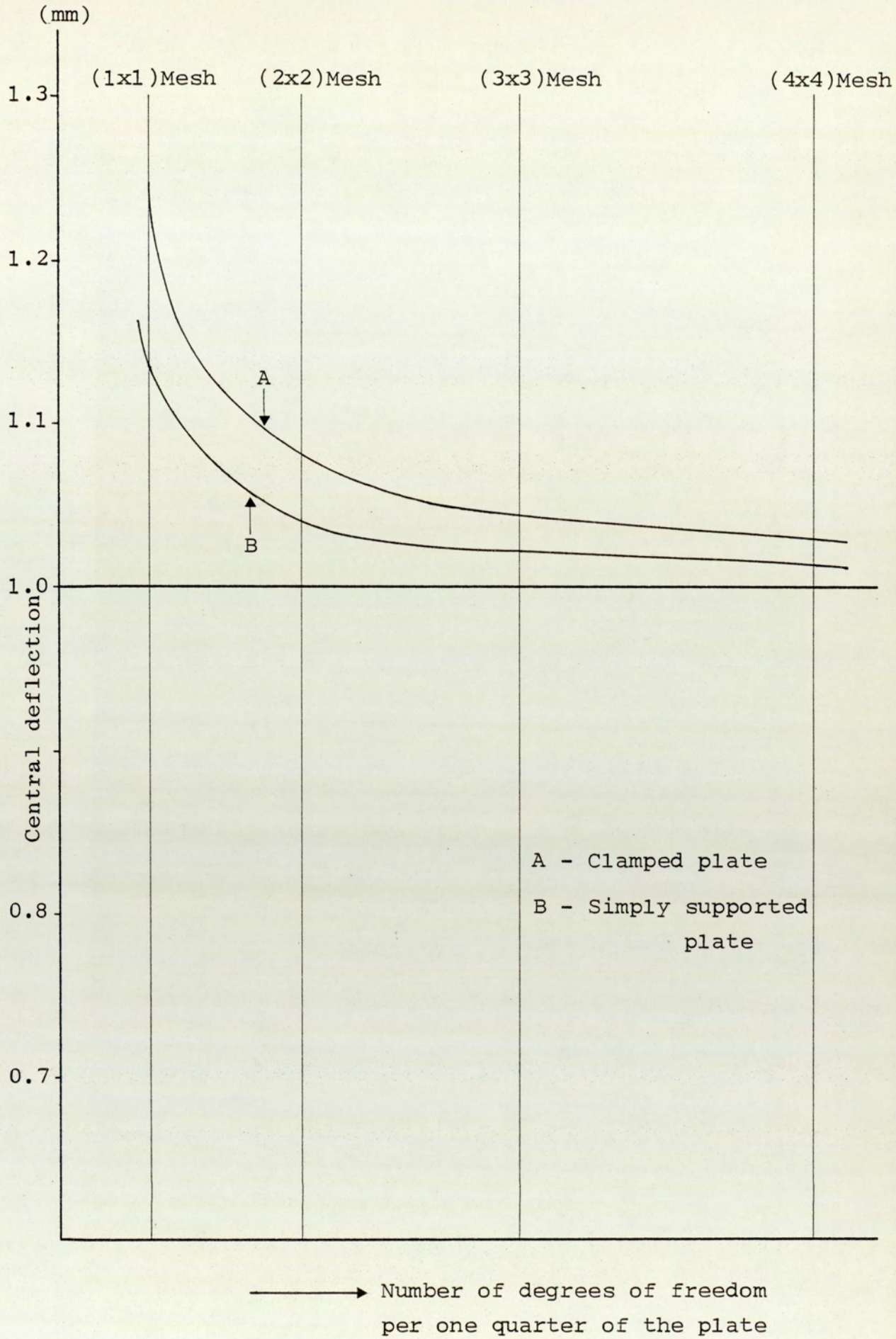


Fig.5.5 Convergence of deflections for a square plate under concentrated load at centre

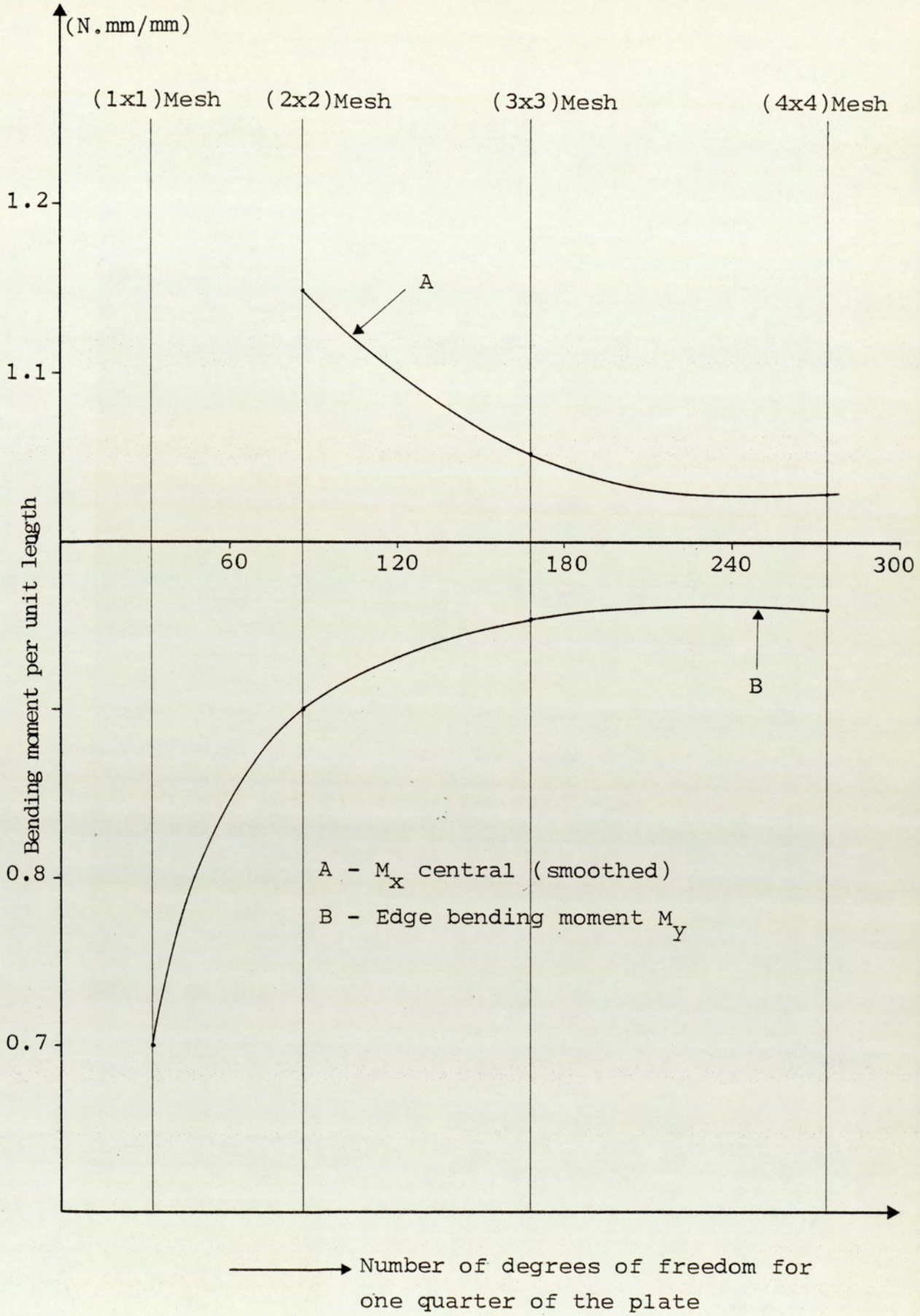


Fig.5.6 Convergence of bending moments for a square plate under uniform loading

5.4 IN-PLANE TENSILE LOADING OF A FLAT PLATE

The problem considered here is a square flat plate loaded on two opposite sides by a parabolically distributed normal traction, while the other two are stress free. Results are compared with exact solution published in Ref. (40) . The equivalent nodal forces due to the normal distribution pressure are calculated using the principle of virtual work (see Ref. (19)).

Figure 5.7 shows the details of the goemetry and the meshes used. The excellent agreement of the computed displacements and stresses gives a confidence into the performance of the semiloof shell element to idealise the thin plate and shell structures subjected to in-plane (membrance) loading. Some of these results are presented in Table 5.3.

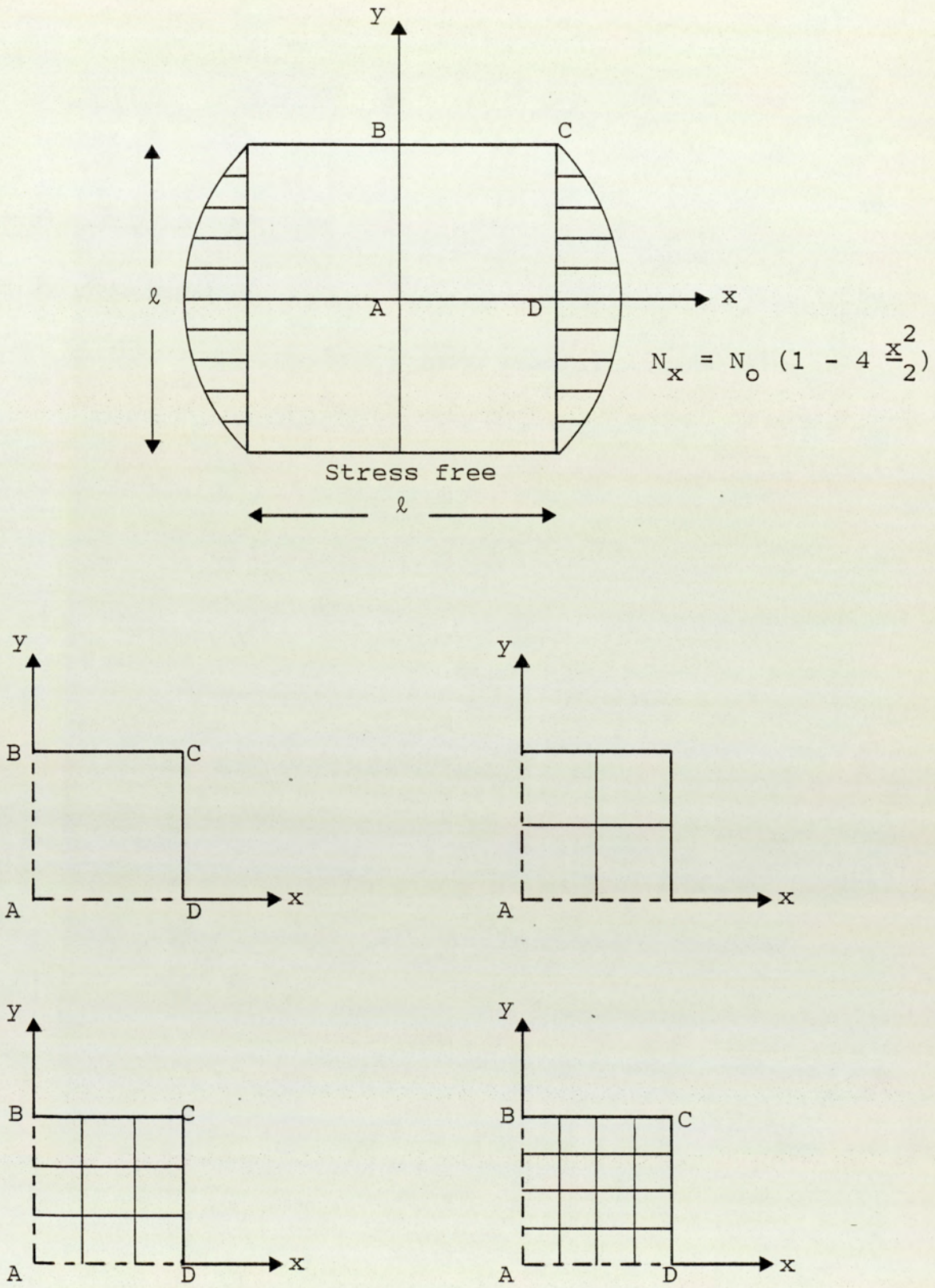


Fig.5.7 Rectangular plate with parabolically distributed edge tractions

Finite Element Mesh	$\frac{10Etv_B}{(1-\nu^2)N_o\ell}$	$\frac{10Etu_C}{(1-\nu^2)N_o\ell}$	$\frac{10Etu_D}{(1-\nu^2)N_o\ell}$
(1×1)	-1.4125	1.194	5.068
(2×2)	-1.4759	1.255	5.023
(3×3)	-1.470	1.2628	5.0157
(4×4)	-1.474	1.266	5.038
Exact	-1.519928	1.27727	5.073478

Finite Element Mesh	$\frac{10\ N_y A}{N_o}$	$\frac{10\ N_x A}{N_o}$	$\frac{N_y B}{N_o}$	$\frac{10\ N_x B}{N_o}$	$\frac{N_y C}{N_o}$
(1×1)	-3.0285	8.9088	-0.045	4.3893	0.0499
(2×2)	-1.6505	8.7663	0.00562	4.1611	-0.0034
(3×3)	-1.5014	8.638	0.00588	4.0959	-0.0112
(4×4)	-1.4064	8.637	0.004156	4.0911	-0.0106
Exact	-1.40954	8.59046	0	4.1067	0

Table 5.3 Numerical Results for rectangular plate with parabolically distributed edge tractions

5.5 CYLINDRICAL SHELL ROOF

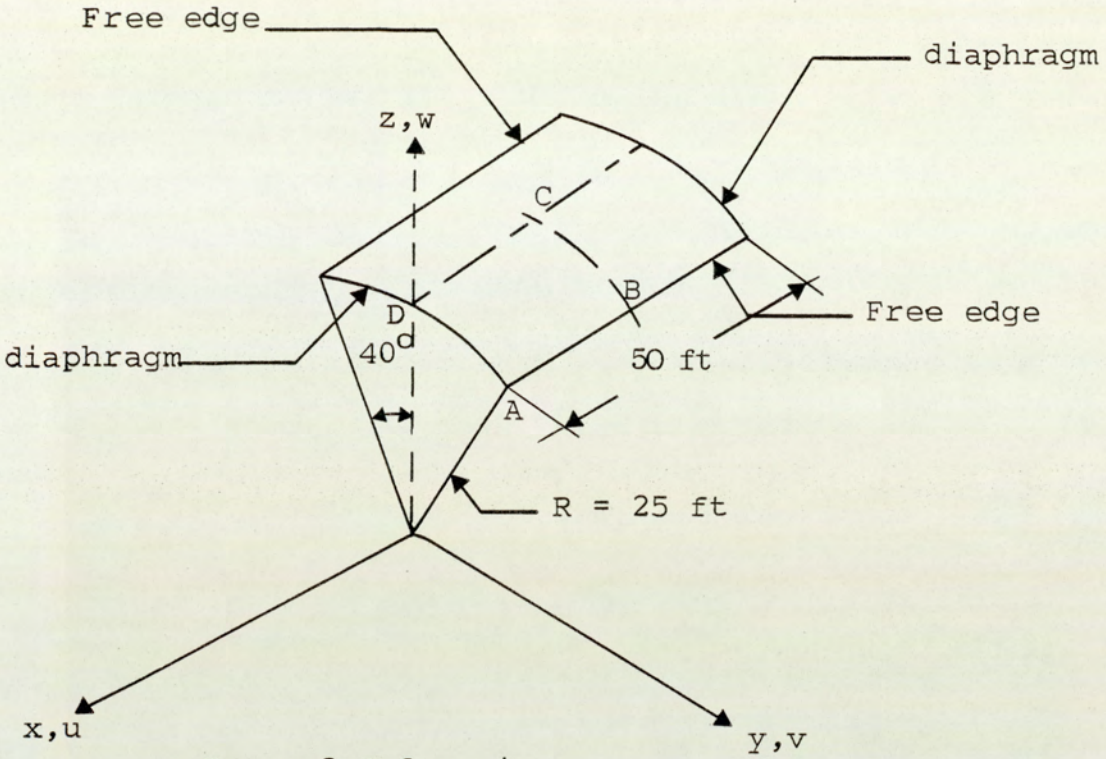
The programs were tested on the problem of a cylindrical shell roof loaded by its own weight. The details and dimensions of this system are shown in Fig. 5.8. The straight edges supported by diaphragms are assumed to be infinitely rigid in their plane and infinitely flexible out of it. This is one of the most typical examples used as a performance test for the finite shell elements programs; membrane and bending effects are of comparable importance in the solution. It has been used by many finite element investigators to test their elements.

Some of the results are given in Table 5.4 together with the results of other elements for comparison purposes. Meshes (A, B and C) are drawn on the exact solution as given in Ref. (69). By observing the results tabulated in Table 5.4, it can be seen that the finite element results obtained using the semiloof shell element are accurate even with the coarsest mesh and the convergence is rapid. Also, it yields the best accuracy for the minimum number of degrees of freedom. The comparison between the results of the finite element approach using the meshes A, B and C and of the analysis of Scordelis and Lo (Ref. 69), is given in Figs. 5.9-5.11. The displacements, membrane stresses, and bending moments are considered in these figures. Good agreement

between the two sets of results have been obtained. Also, a high rate of convergence to the exact solution was noticed.

Element Grid	Net no.of equations	u_A in	w_B in	u_B in	w_C in	
2×2	30	-0.0735	-4.571	2.375	0.601	Ref(29)
3×3	63	-0.1049	-3.629	1.912	0.5281	simple flat
4×4	108	-0.1201	-3.530	1.861	0.5234	triangular
5×5	165	-0.1283	-3.527	1.860	0.5275	shell
10×10	630	-0.1417	-3.564	1.881	0.5414	element
4×5	-	-	-3.56	-	0.48	Ref(70)
8×12	-	-	-3.71	-	0.54	curved shell
12×18	-	-	-	-	-	element
4×5	-	-	-2.49	-	0.398	Ref(71)
8×12	-	-	-3.324	-	0.535	curved shell
12×18	-	-	-3.531	-	0.54	element
1×2	53	-0.1288	-3.0396	1.6493	0.4056	Present
2×3	121	-0.1418	-3.437	1.8159	0.4964	analysis
3×4	212	-0.1489	-3.5834	1.8889	0.52087	semiloof
4×4	275	-0.1488	-3.5913	1.8942	0.5248	shell element
Exact		-0.15133	-3.7033	1.9637	0.5249	Ref(40)
Exact			-3.696		0.552	Ref(69)

Table 5.4 Numerical Results for Cylindrical Shell Roof Problem



E = 3 E 6 psi

ν = 0.0

t = 3 inch

shell weight = 90 lb/ft^2

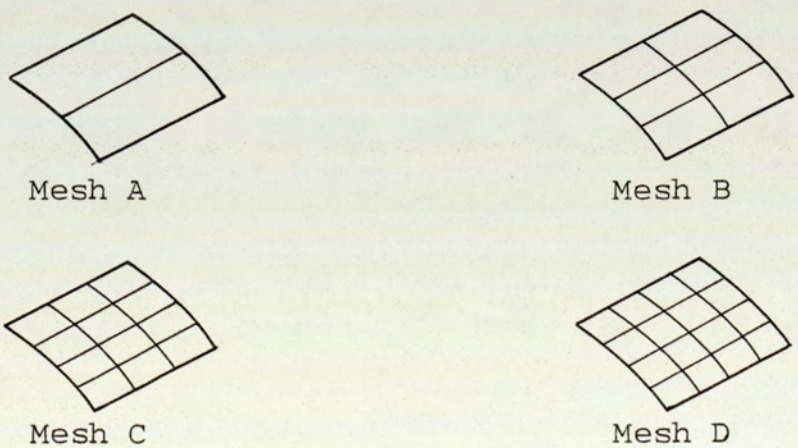


Fig.5.8 Cylindrical shell roof under its own weight

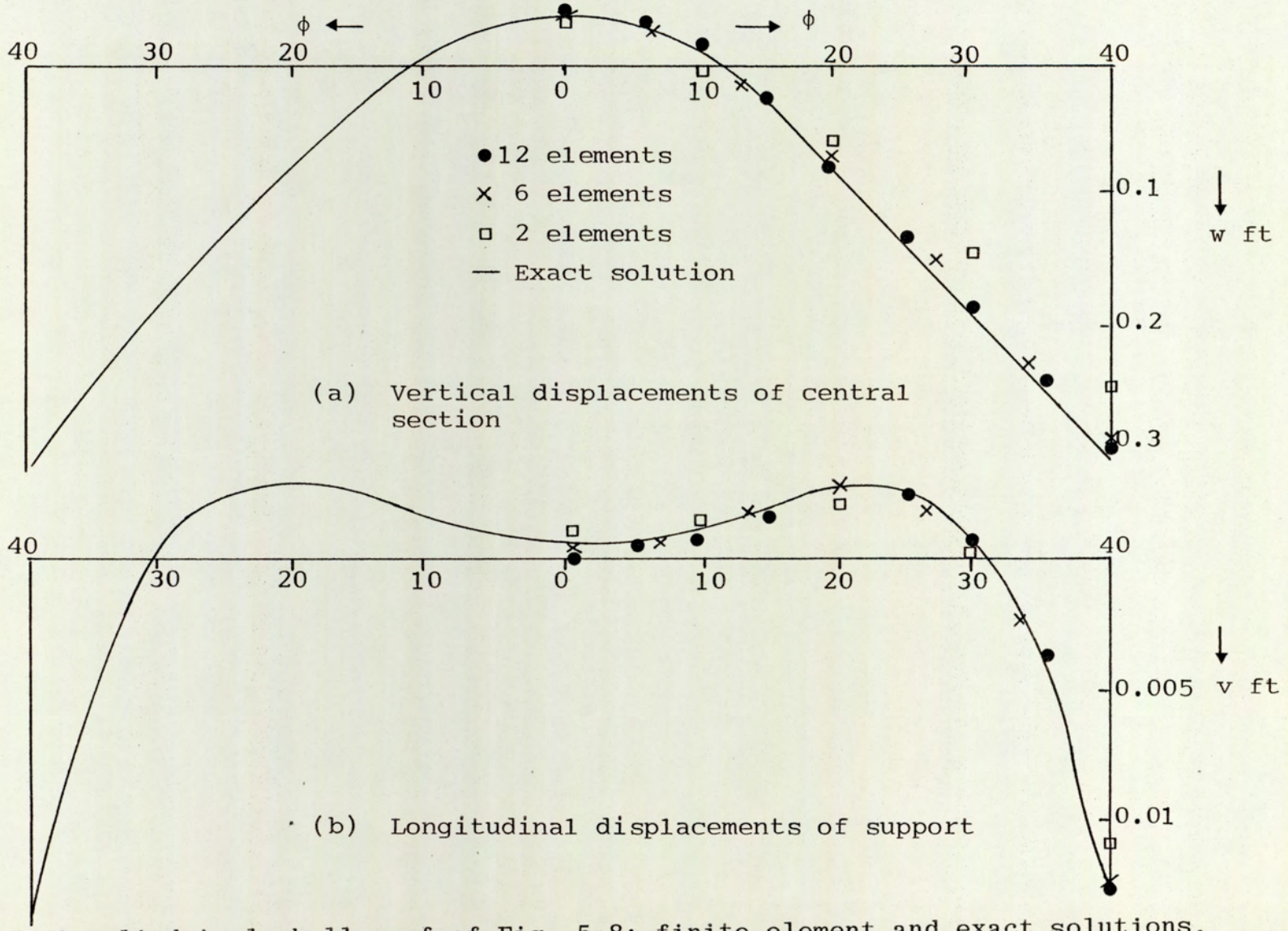


Fig. 5.9 A cylindrical shell roof of Fig. 5.8: finite element and exact solutions.

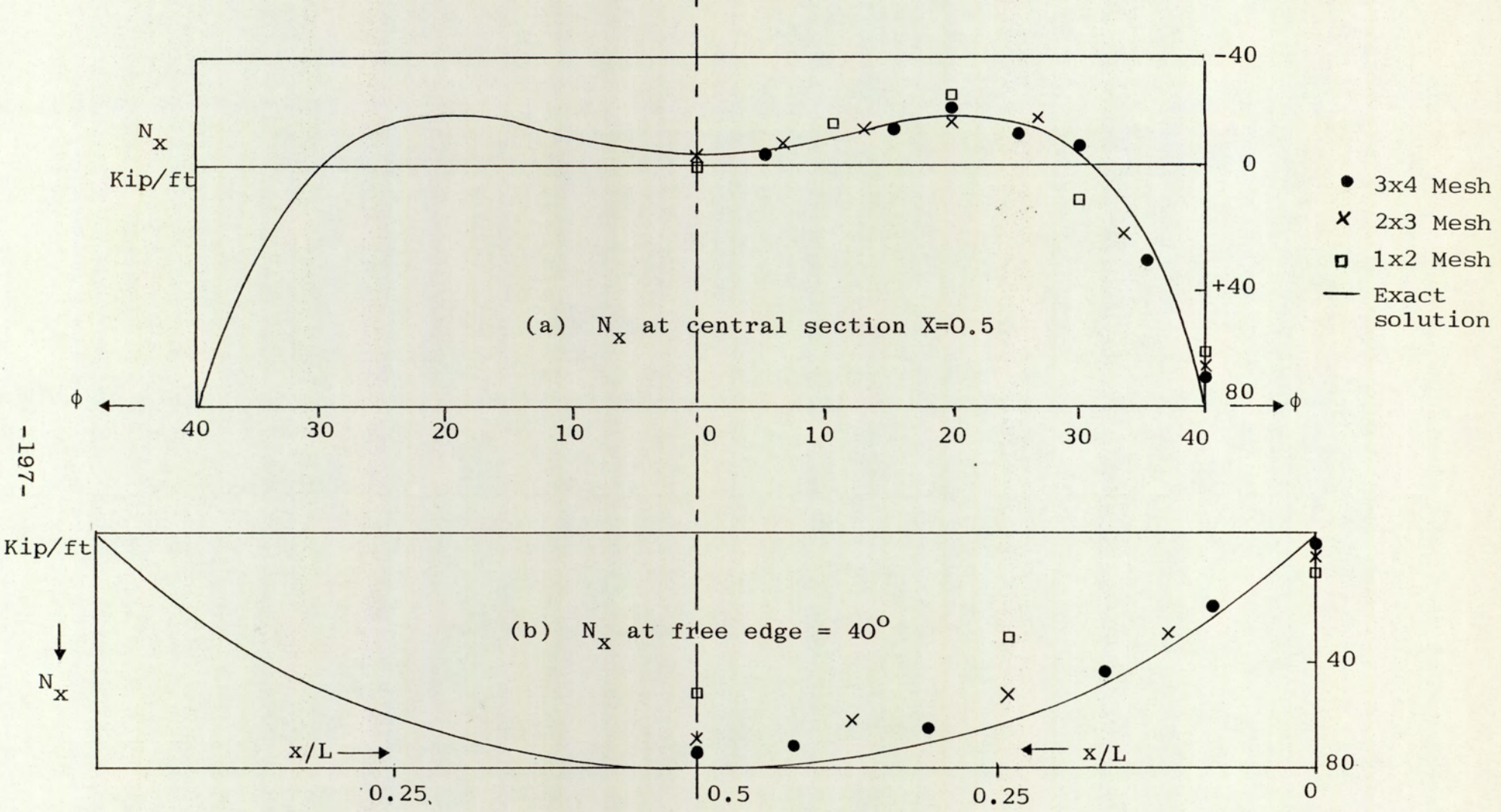


Fig.5.10 A cylindrical shell roof of Fig.5.8 finite element and exact solutions.

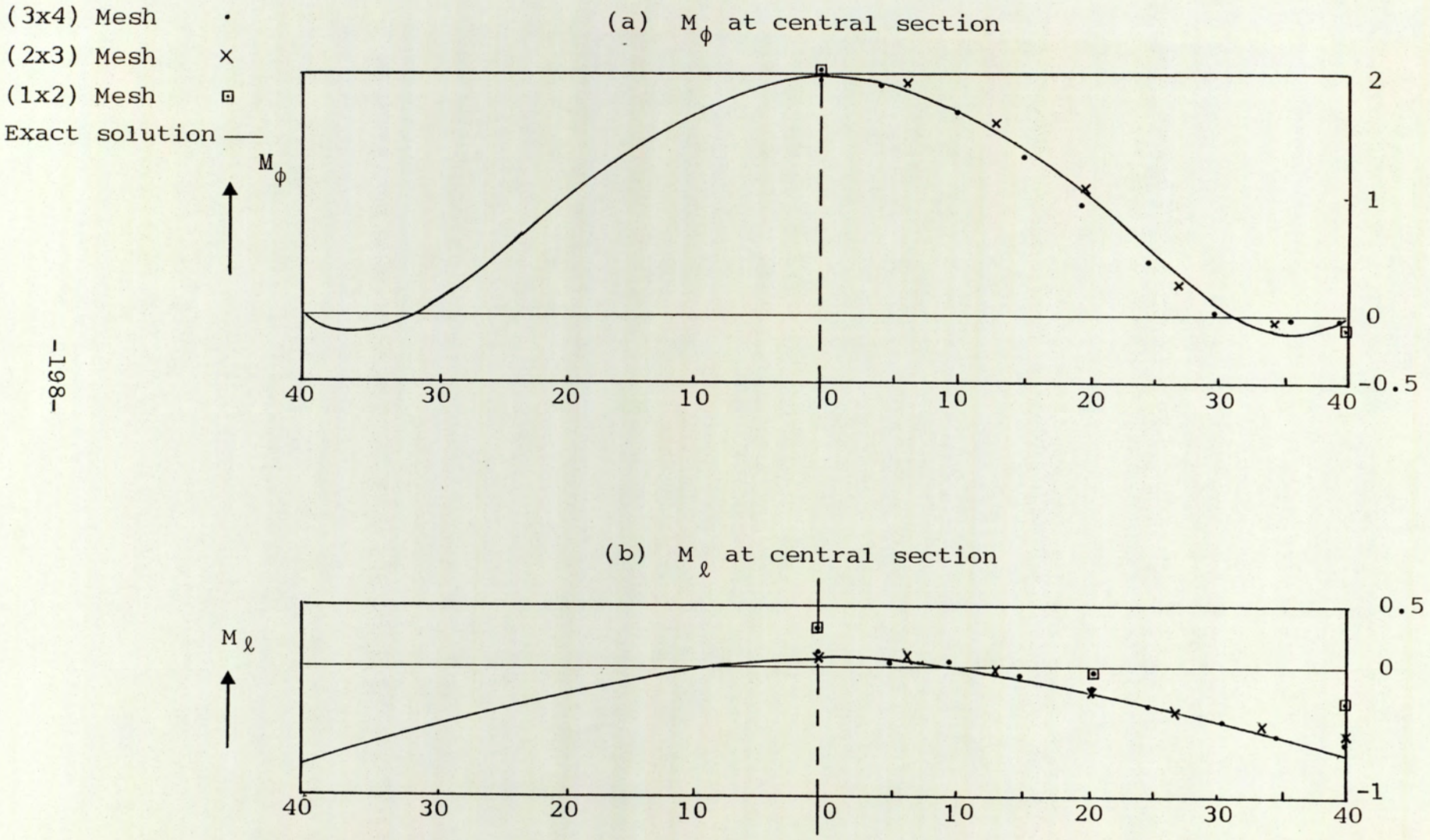


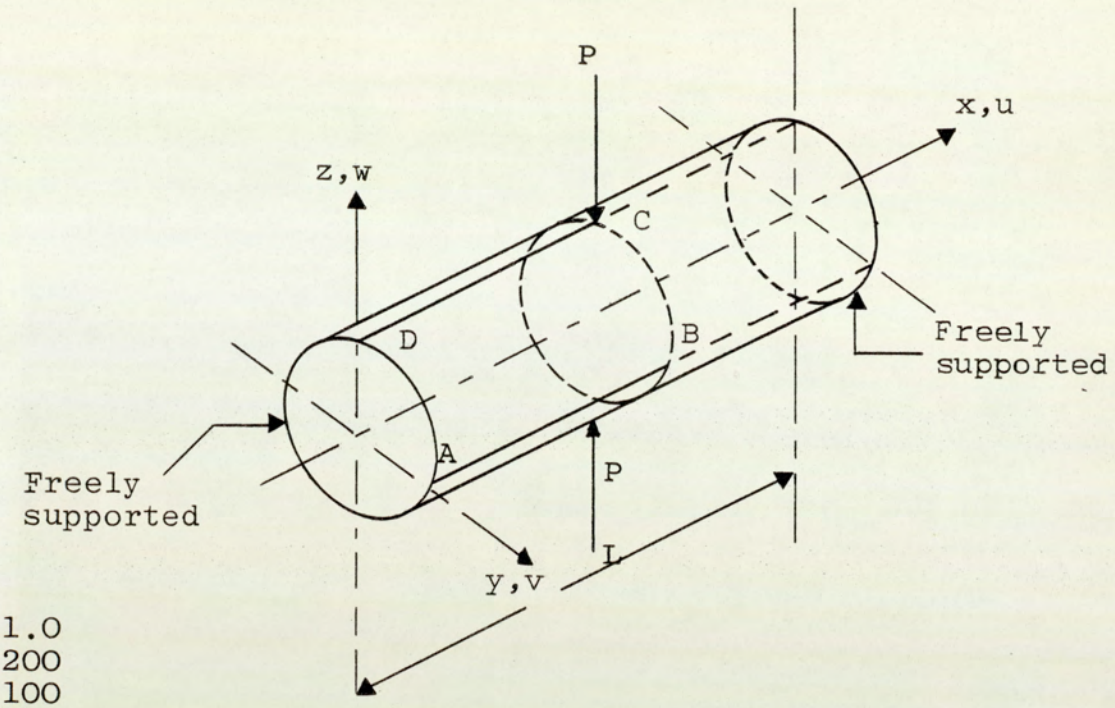
Fig.5.11 A cylindrical shell roof of Fig.5.8 : finite element and exact solutions

5.6 A DEEP PINCHED CYLINDRICAL SHELL

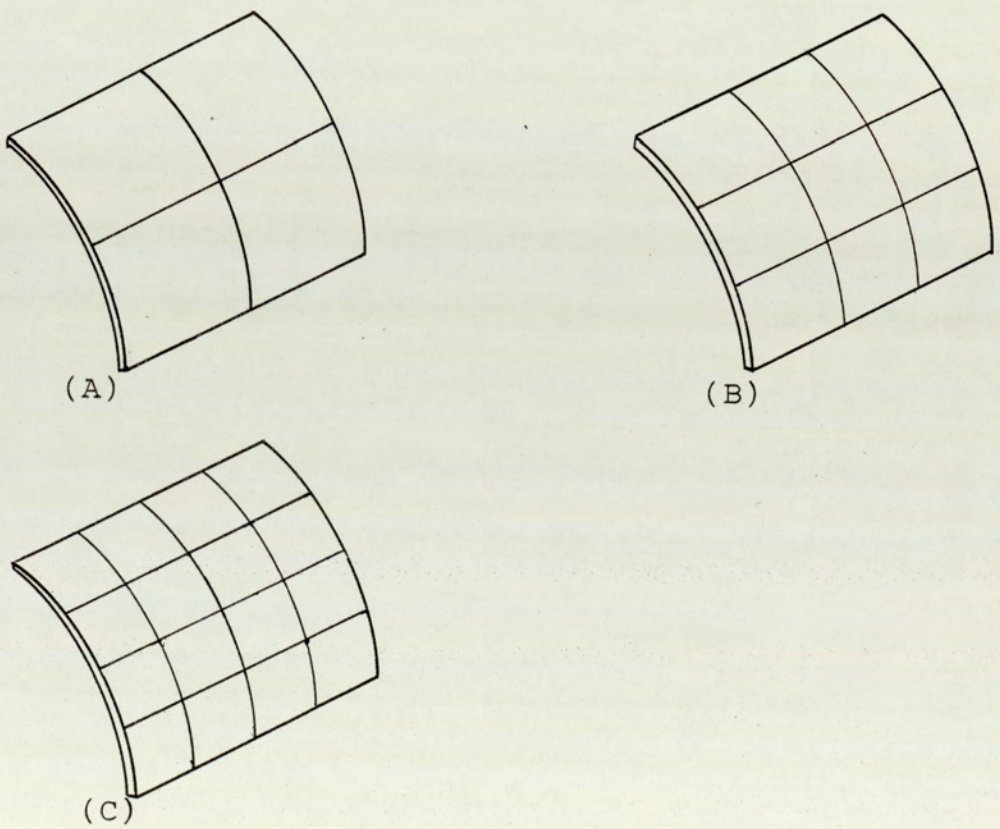
As a complex problem to test the programs developed, a deep pinched cylindrical shell is analysed. Figure 5.12 shows geometry, loads, material properties, and meshes used. The cylinder is freely supported at both ends, and subjected to two central loads acting at diametrically opposed points. Because of symmetry conditions only one-eighth of the shell need be considered in an analysis: in the present test, this has been done using three different meshes; these are designated (2×2), (3×3) and (4×4) .

The displacement and stress resultsants are presented and compared with exact results from a double Fourier series solution by Flugge, Ref. (72).

The non-dimensional results for the displacements, stress resultants and the bending moments are given in Figs. 5.13 - 5.15. They are in good agreement with the Flugge solution and indicate a high rate of convergence. This example makes it evident that the semiloof shell element can also tackle even a deep shell problem.



t = 1.0
L = 200
R = 100
ν = 0.3



Mesh used

Fig.5.12 Geometry and the finite element meshes used for pinched cylindrical shell

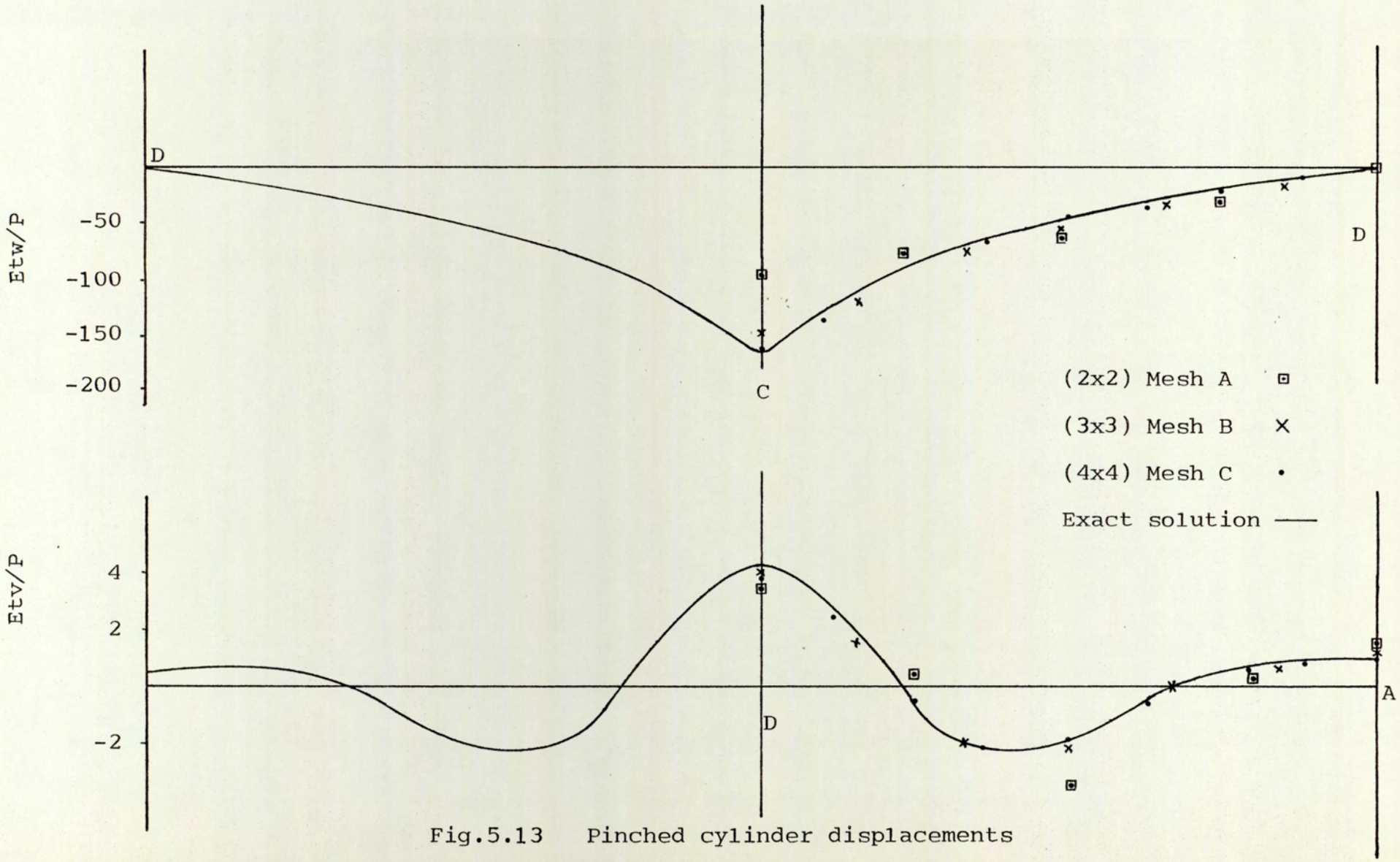


Fig.5.13 Pinched cylinder displacements

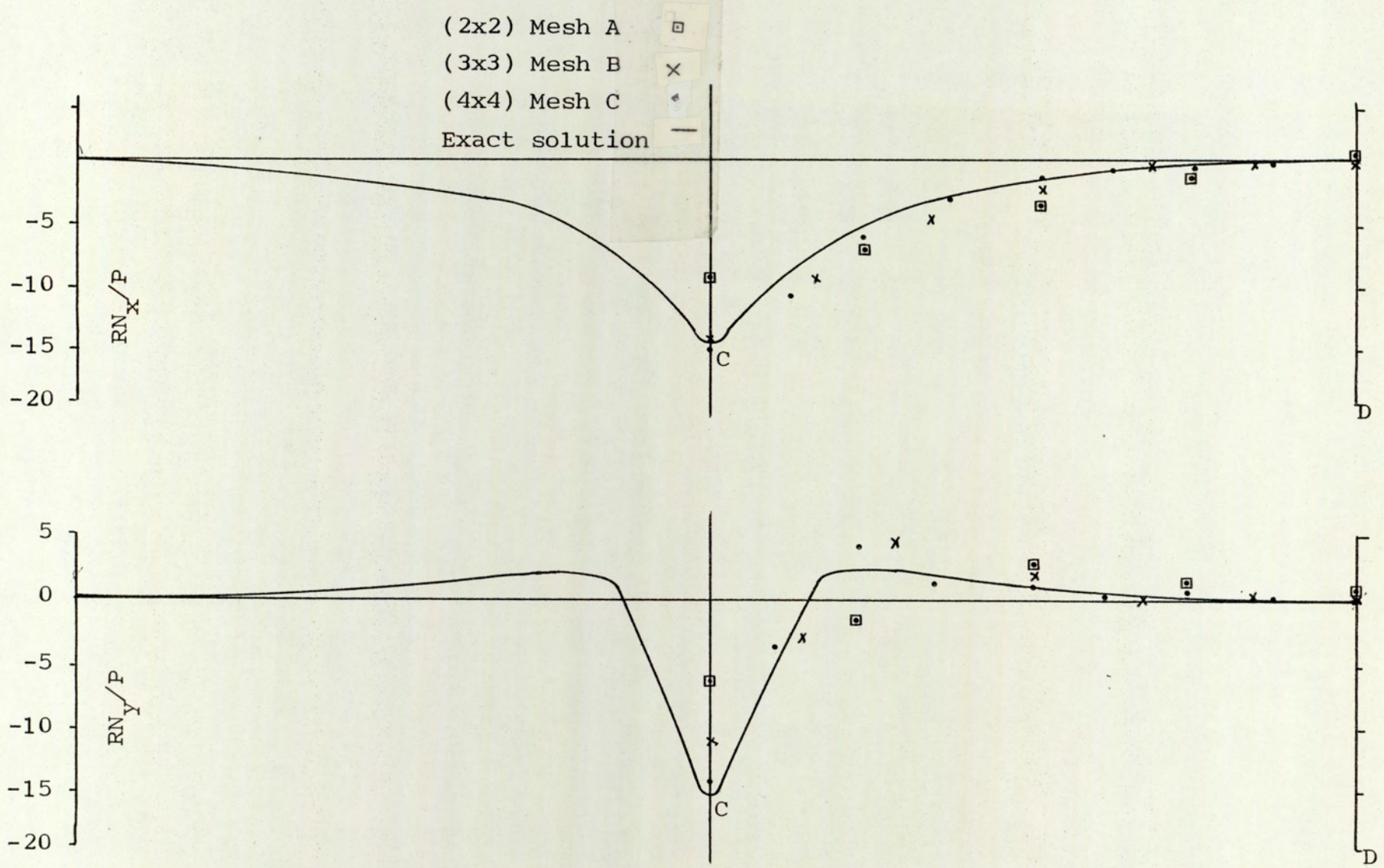


Fig.5.14 Membrame stress resultants along D-C of the pinched cylindrical shell .

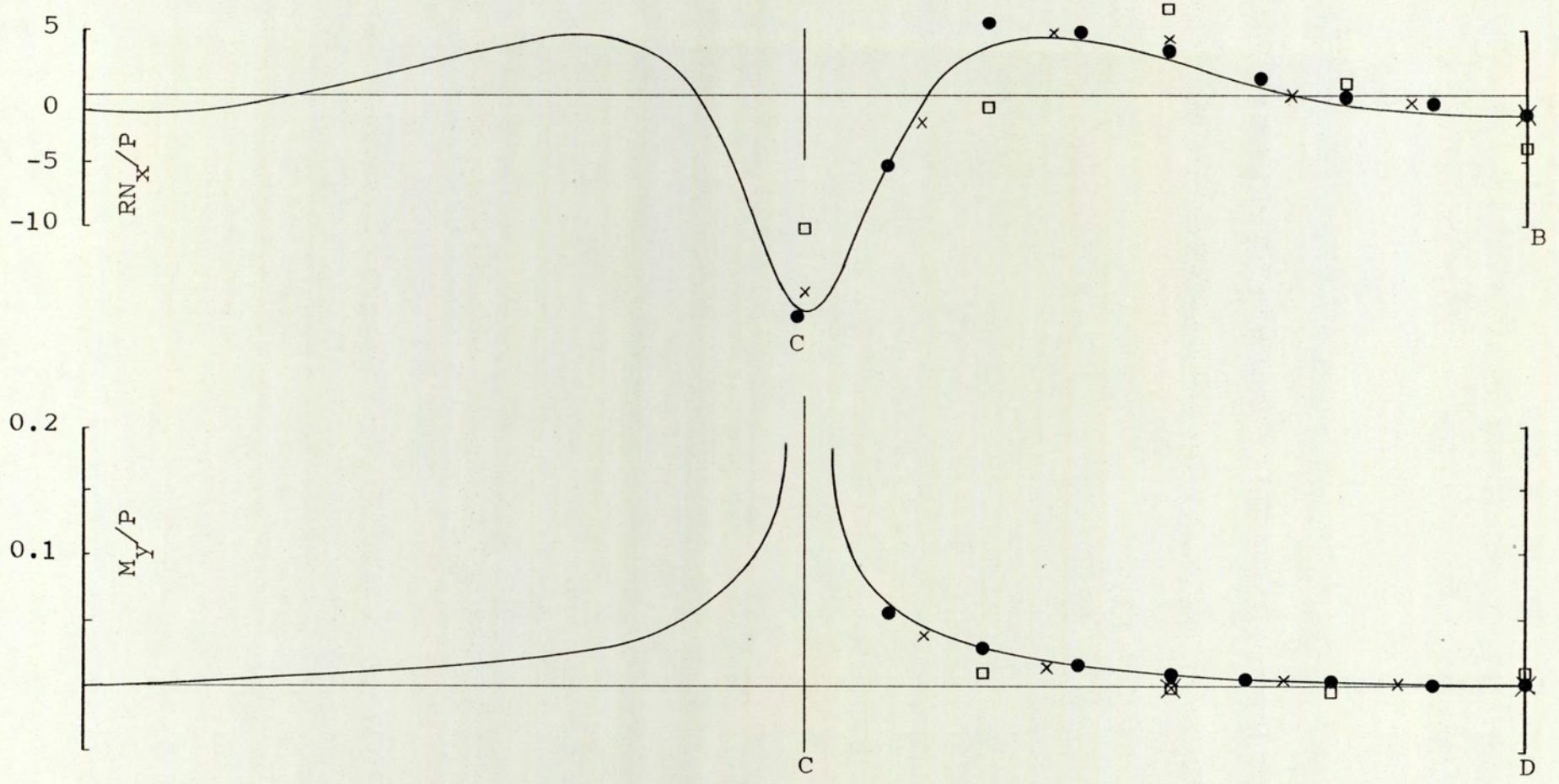


Fig.5.15 Bending moment distributions along C-D and membrane stresses along C-B of the pinched cylindrical shell

5.7 FOLDED PLATES

These problems concern plates with intersections under a point load. Two examples are presented here. Details and geometry are given in Figs. 5.16(a) and 5.22(a). The finer mesh for each structure was used as a reference for comparison with other used meshes in order to predict the rate of convergence.

Various displacements and bending moments along different cross-sections are presented. These structures are as follows:

(a) Three intersecting plates at right angles to each other; details of the geometry, loading and the meshes used are shown in Figs. 5.16(a),(b),(c),(d), (e). Displacements and bending moments are plotted in Figs. 5.17 to 5.21.

(b) Two intersecting plates at right angles to each other, another plate is welded to them. Details of the geometry, loading and the meshes used are shown in Figs. 5.22(a),(b),(c),(d). Displacements and bending moment distributions are plotted in Figs. 5.23 to 5.27.

In both cases (a) and (b), the semiloof shell element has proved to be an efficient element to tackle structures with sharp corners and multiple junctions. High rate of convergence for the displacements and bending moments

were obtained. These indicate the good performance of the semiloof shell element in the idealisation of such complicated structures, which is a lead into the actual fan impeller structures of interest in this work, since the blade/backsheet and blade/conesheet junctions of the fan impellers are similar to those found in the above examples.

Another two examples of multiple junctions were tested experimentally. These are presented in Chapter 7, where a comparison of finite element predictions and experimental results are given.

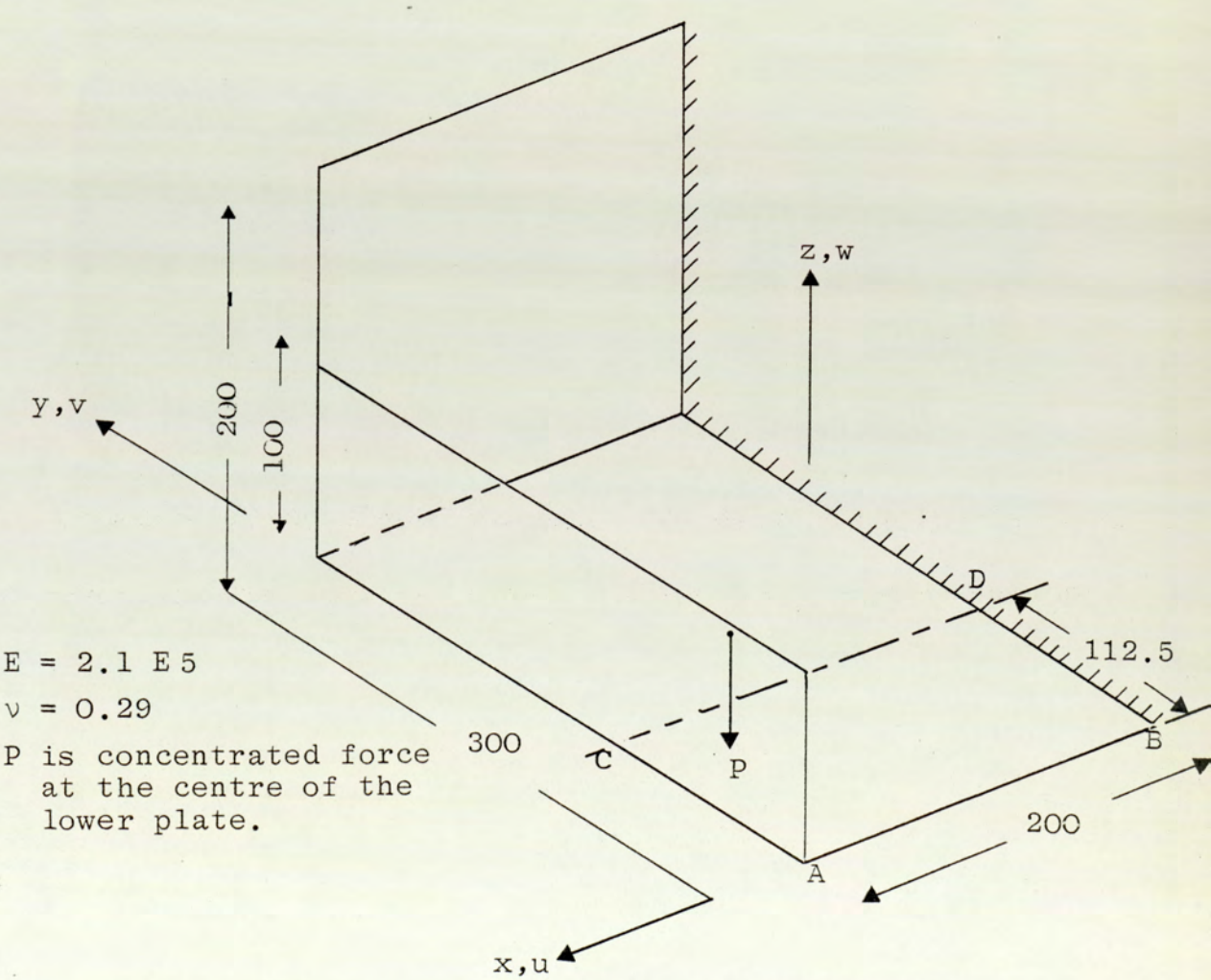
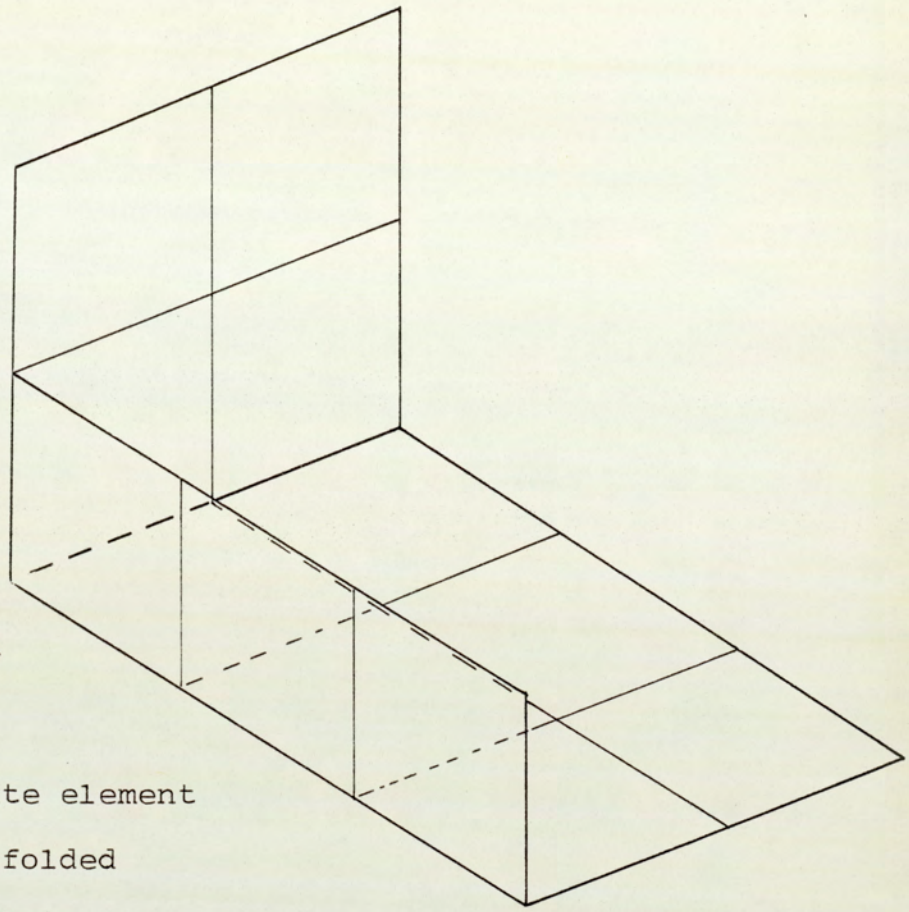
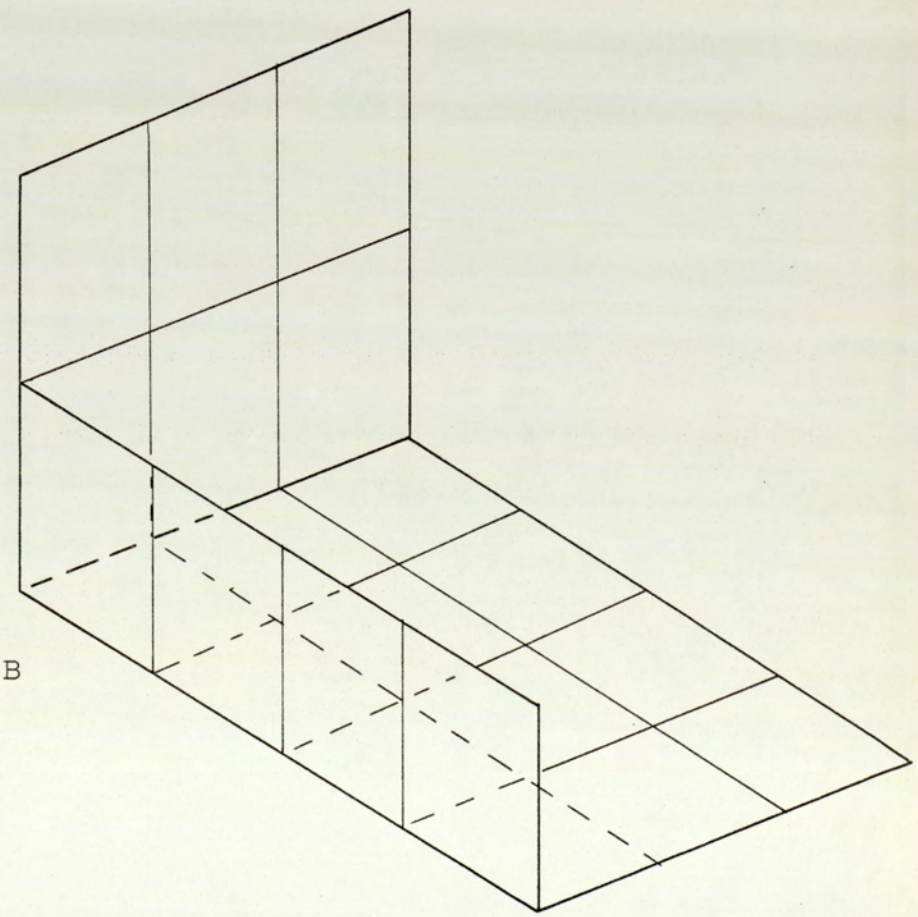


Fig. 5.16(a) Geometry of folded plate structure No. 1.

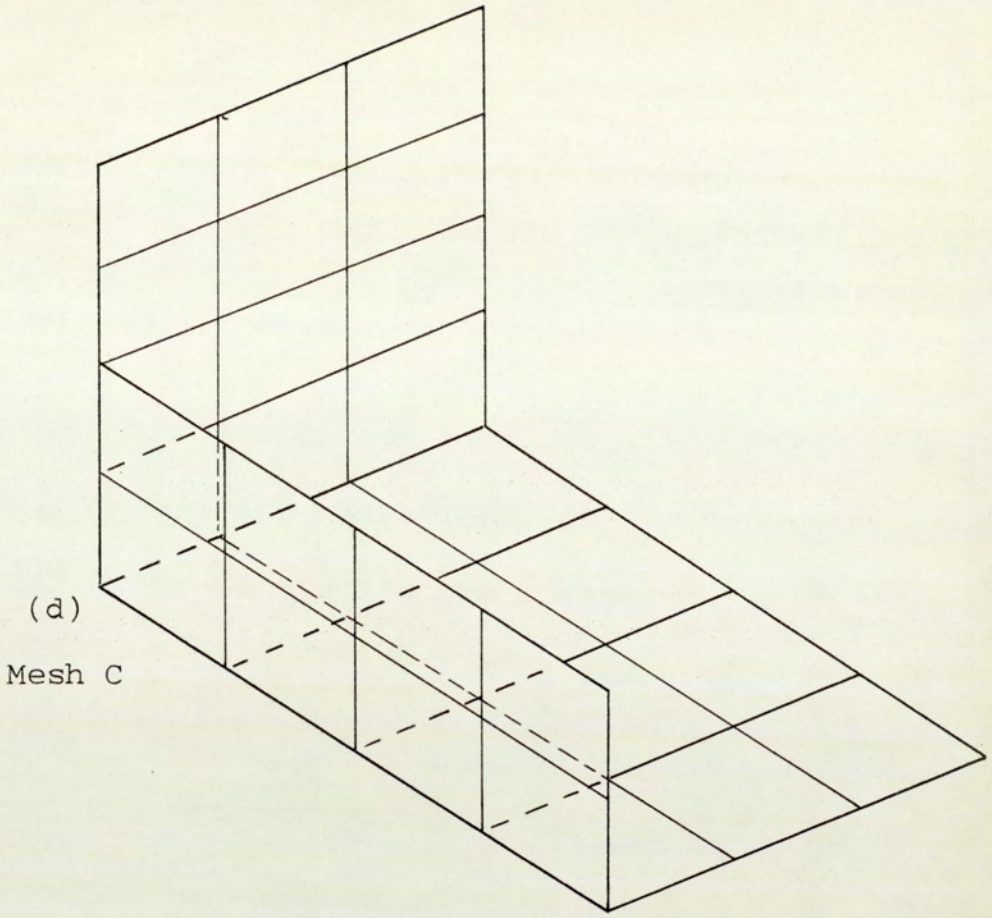


(b)
Mesh A

Figs.5.16b,c Finite element meshes A and B for folded plate structure No.1

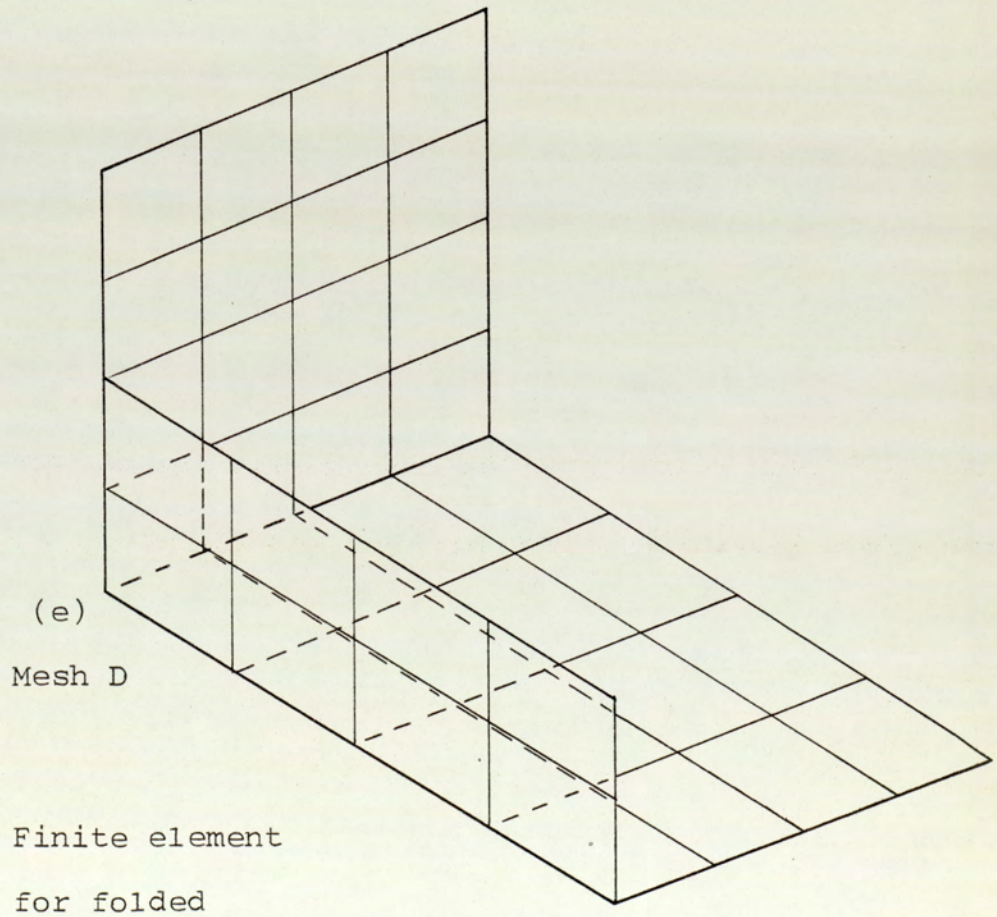


(c)
Mesh B



(d)

Mesh C



(e)

Mesh D

Figs.5.16d,e Finite element meshes C and D for folded plate structure No.1

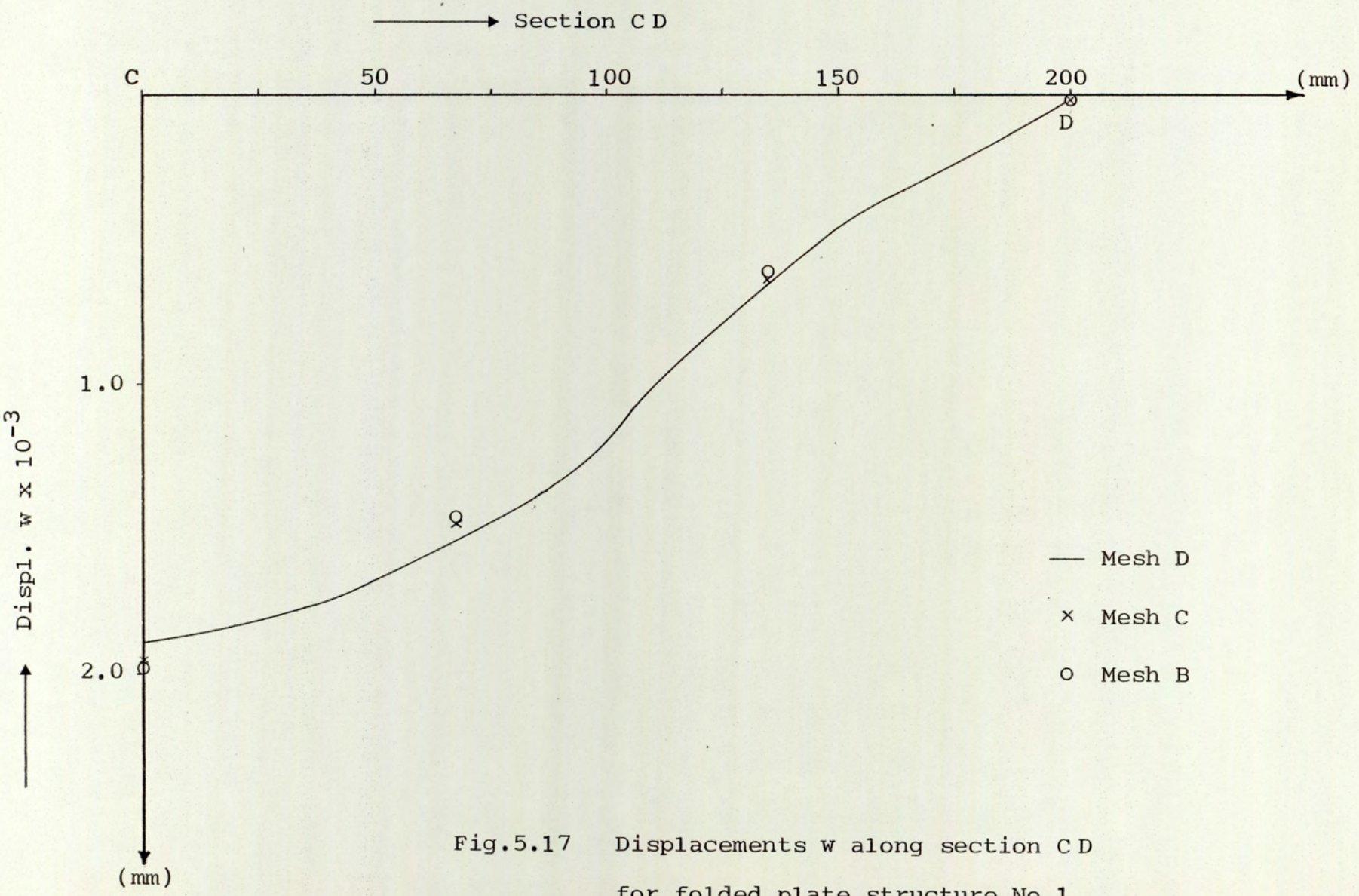


Fig.5.17 Displacements w along section C D for folded plate structure No.1

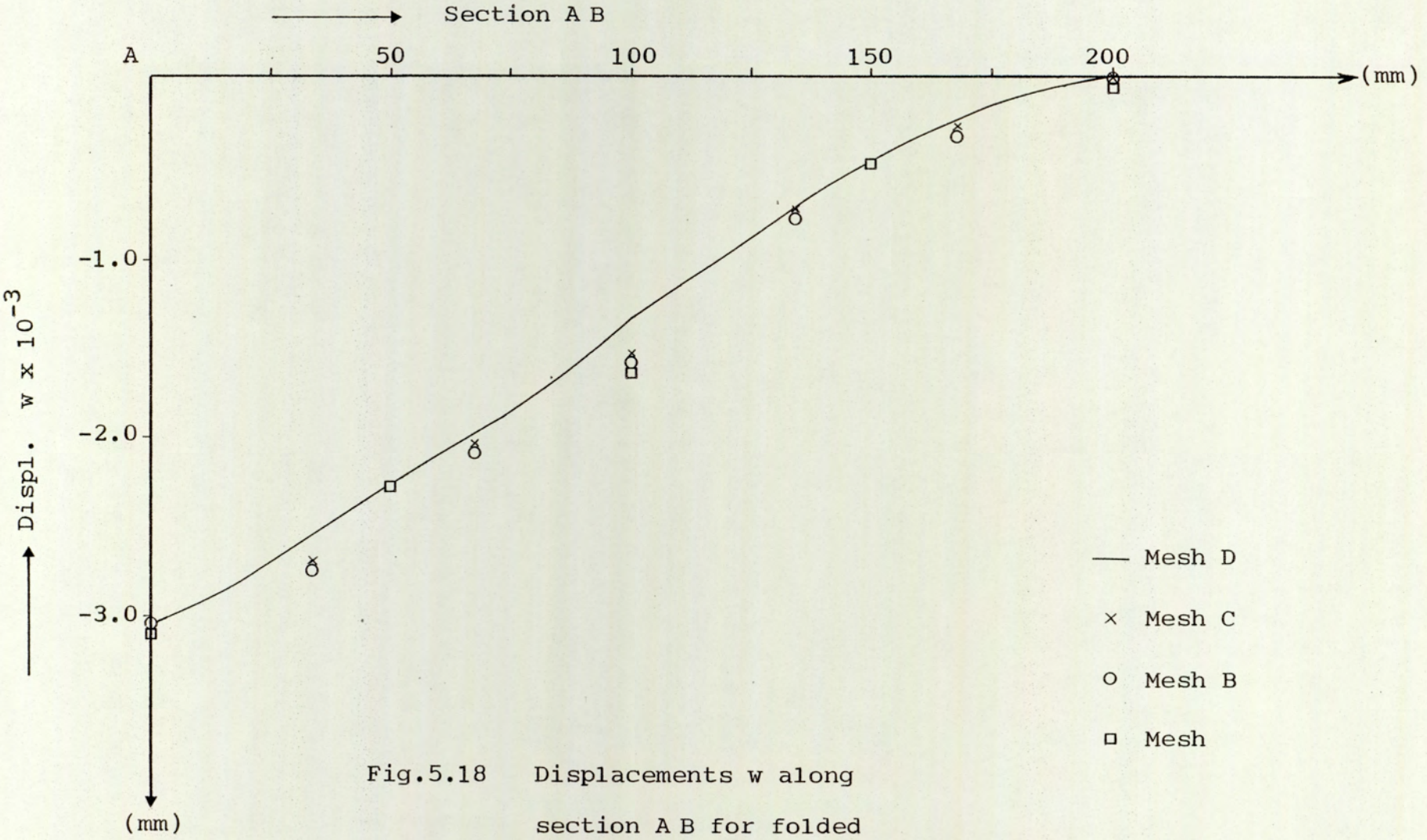


Fig.5.18 Displacements w along section A B for folded plate structure No.1

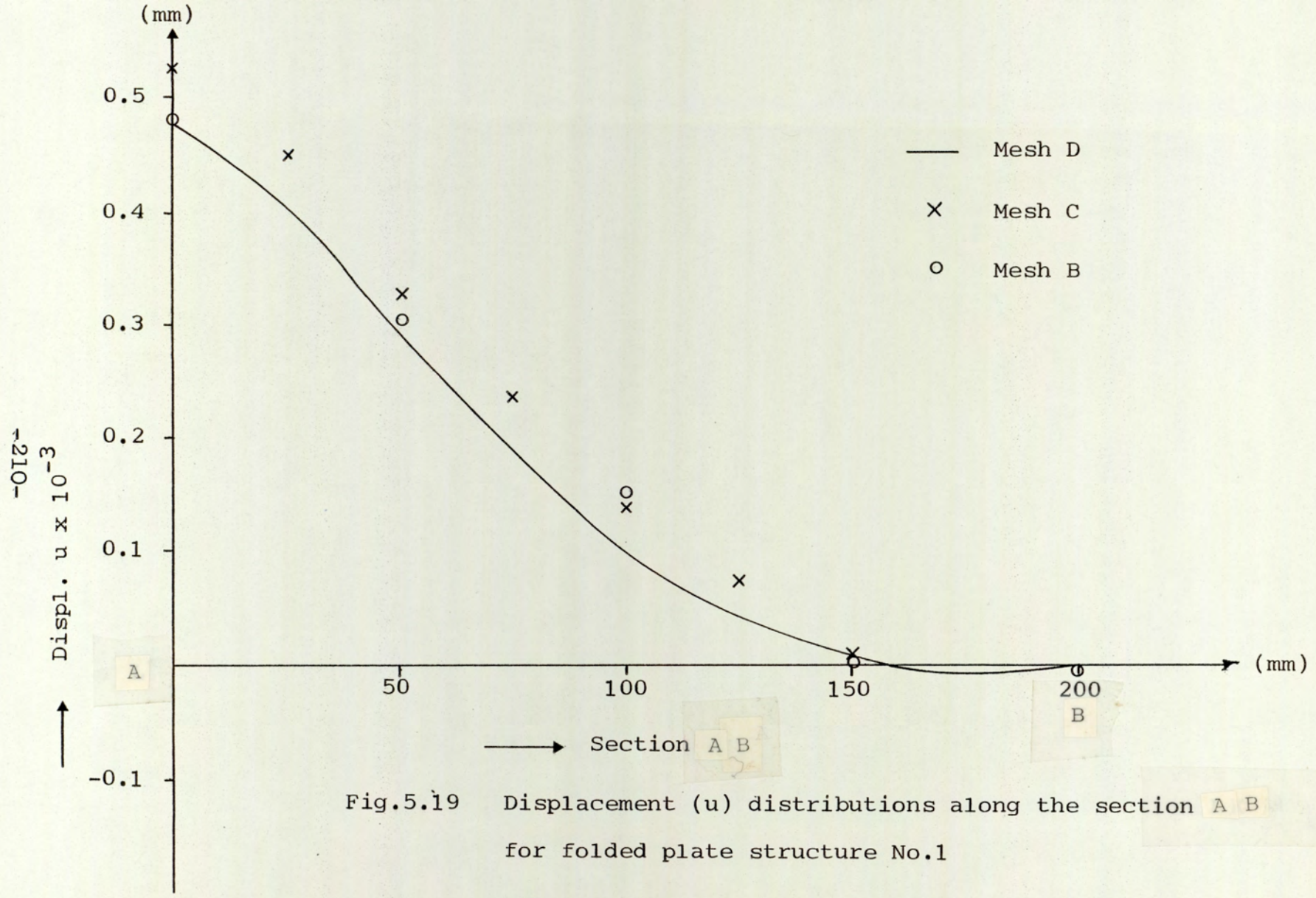


Fig.5.19 Displacement (u) distributions along the section A B for folded plate structure No.1

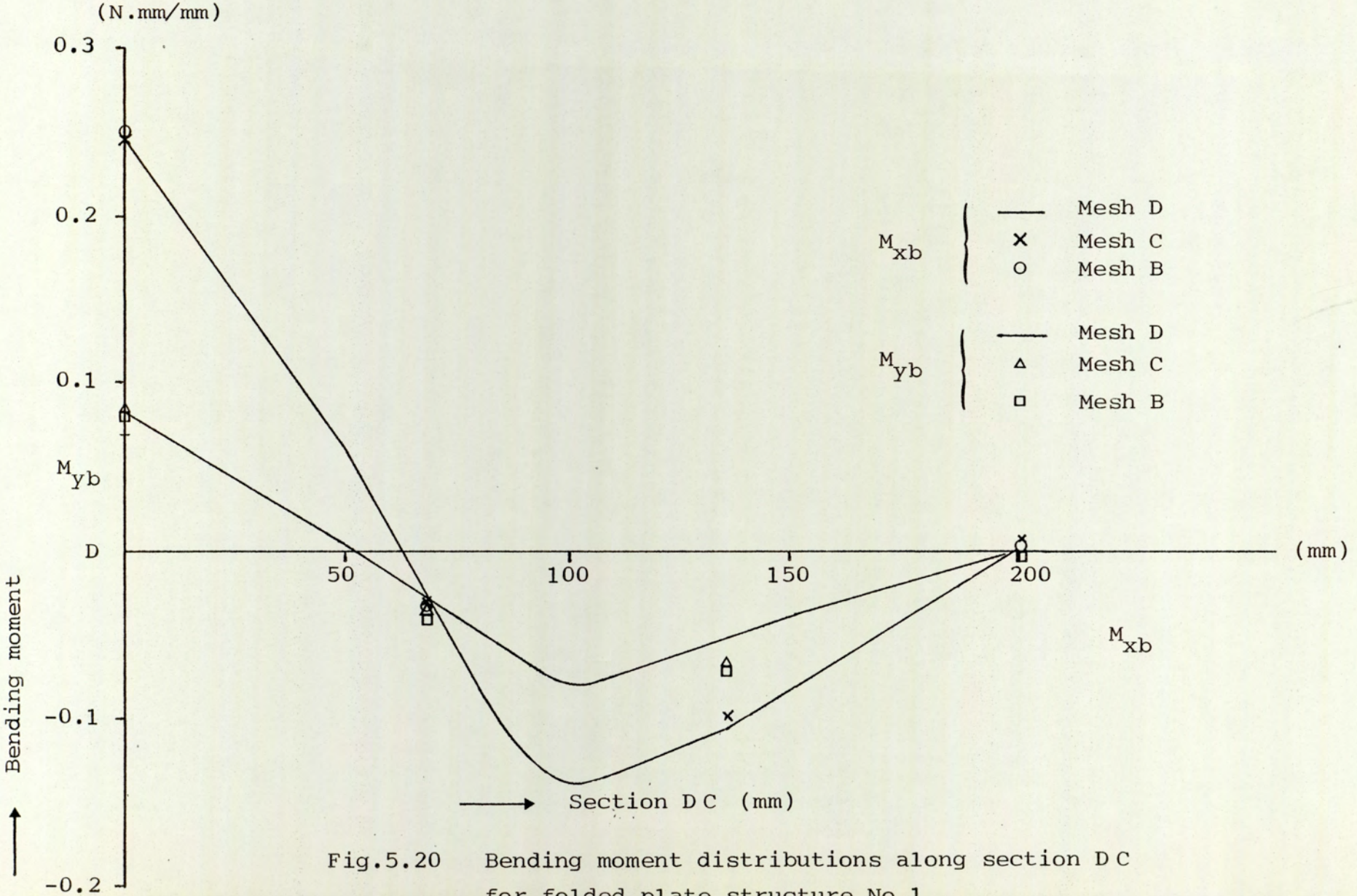


Fig.5.20 Bending moment distributions along section D C for folded plate structure No.1

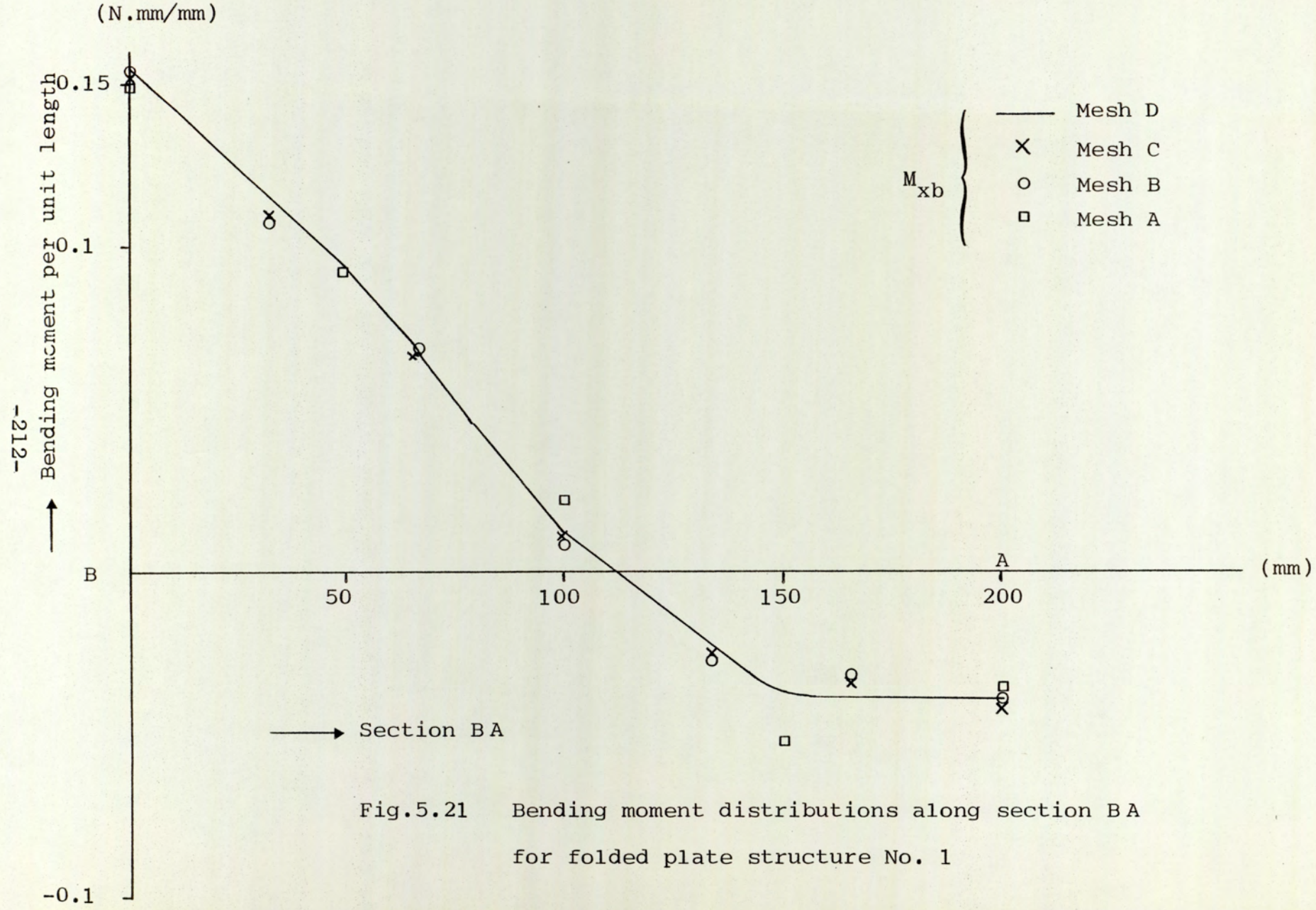


Fig.5.21 Bending moment distributions along section B A for folded plate structure No. 1

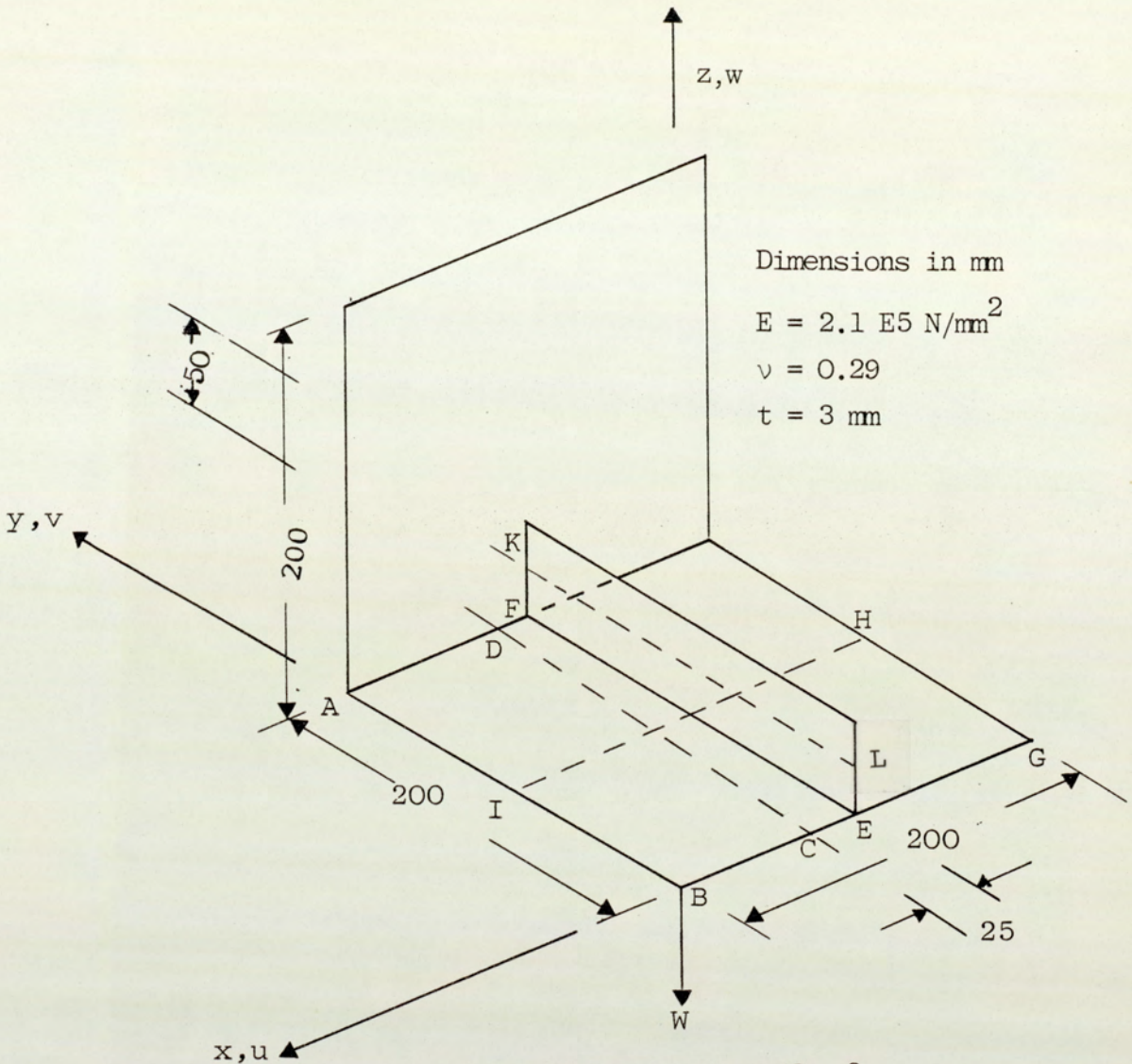
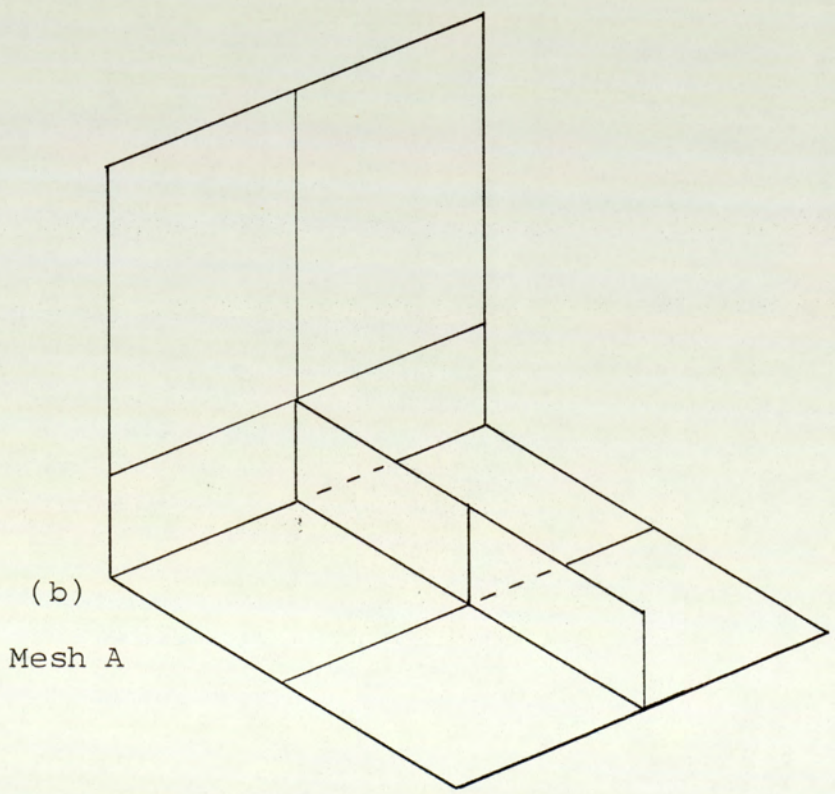
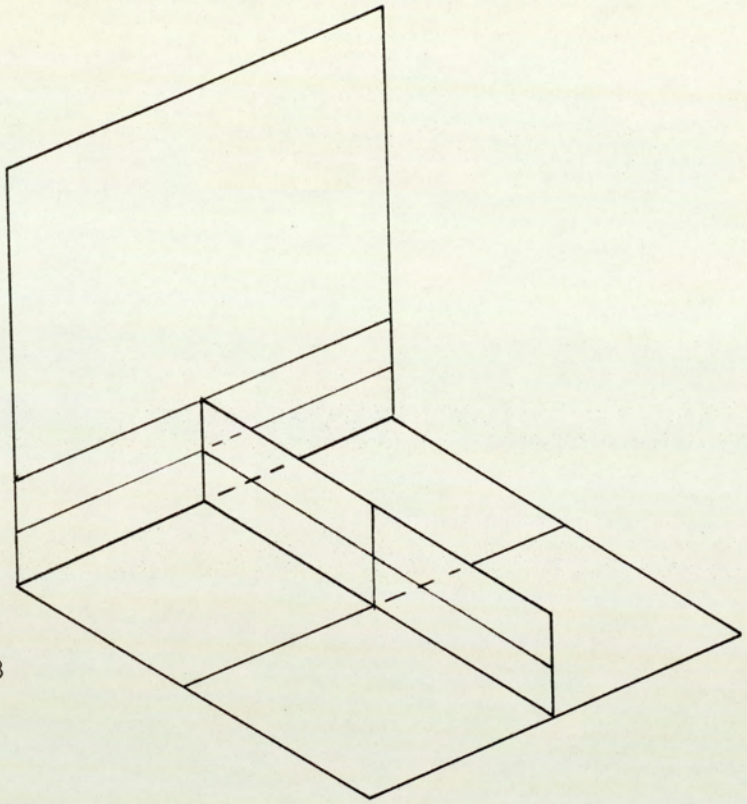


Fig. 5.22(a) Geometry of folded plate structure No. 2.



(b)

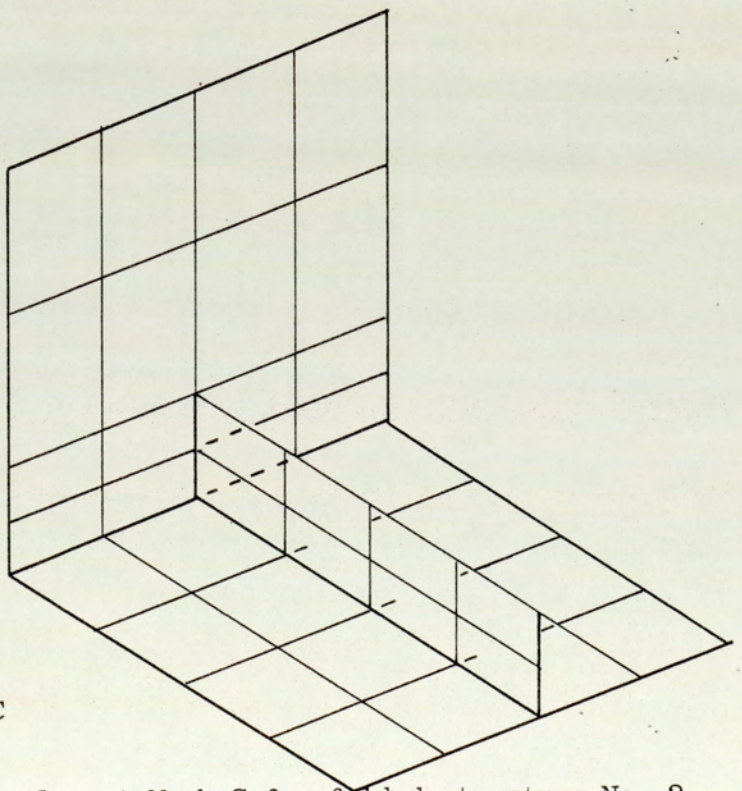
Mesh A



(c)

Mesh B

Fig.5.22(c) Finite element meshes B
folded plate structure No.2



(d)

Mesh C

Fig. 5.22 (d) Finite element Mesh C for folded structure No. 2.

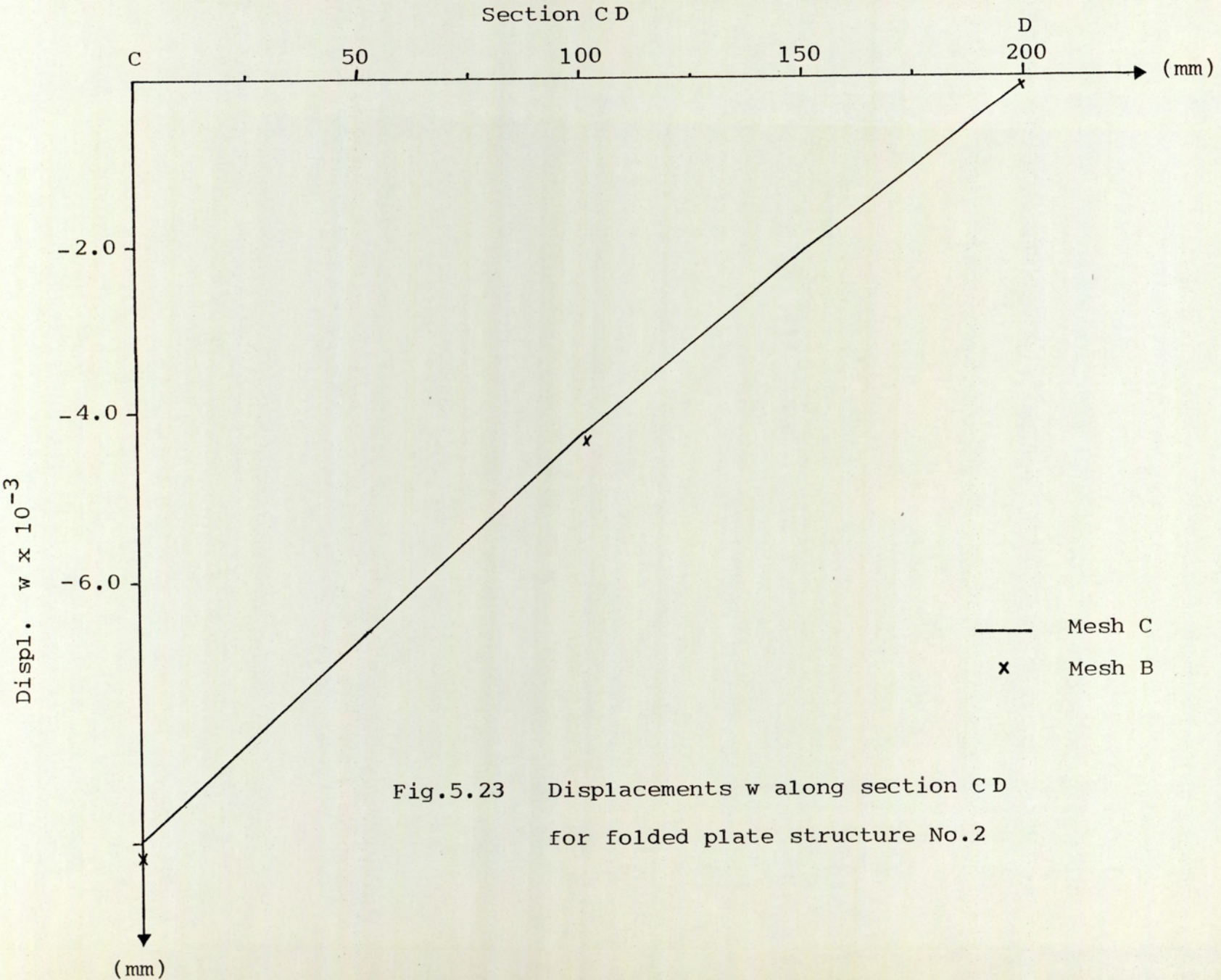


Fig.5.23 Displacements w along section C D for folded plate structure No.2

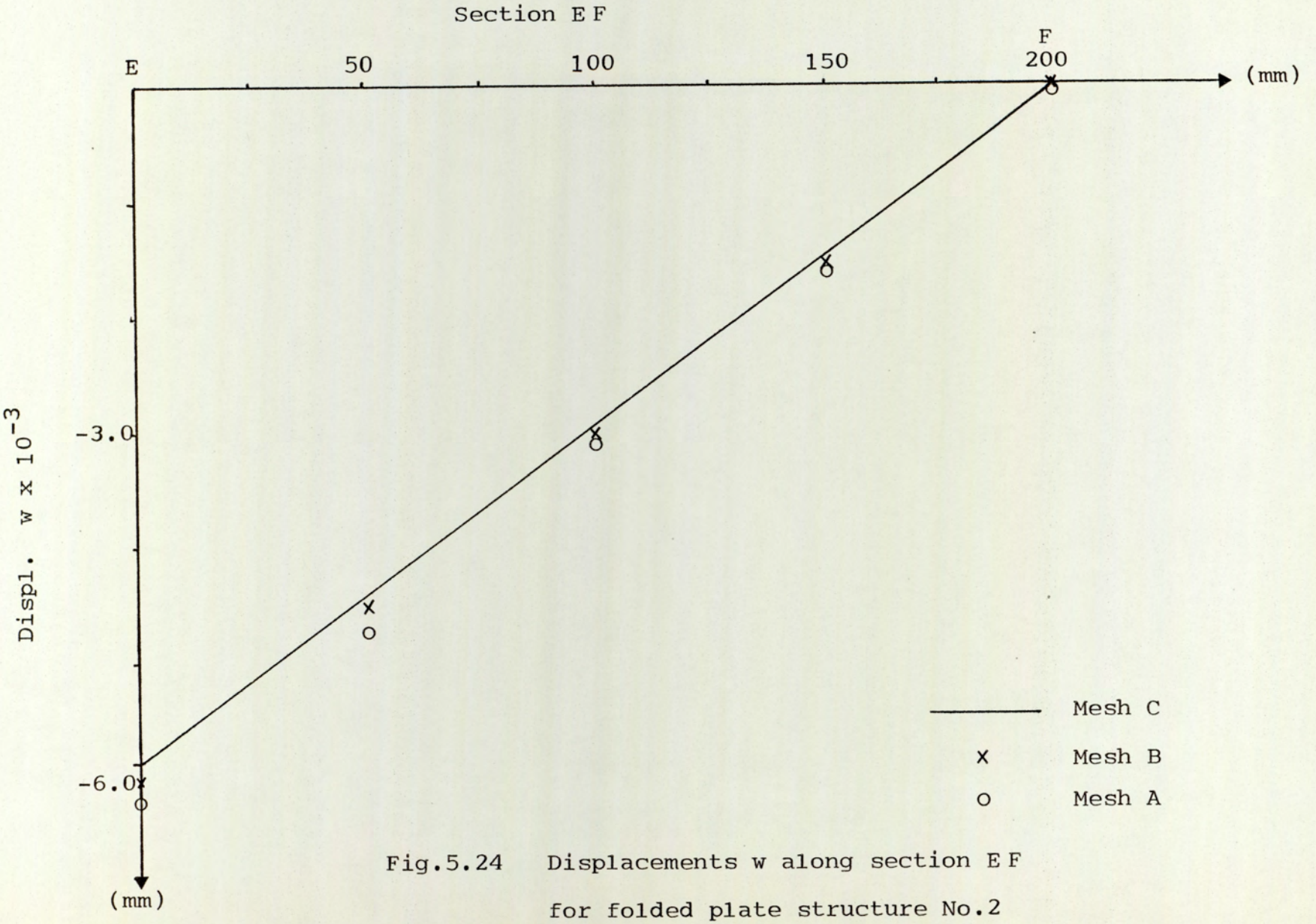


Fig.5.24 Displacements w along section E F for folded plate structure No.2

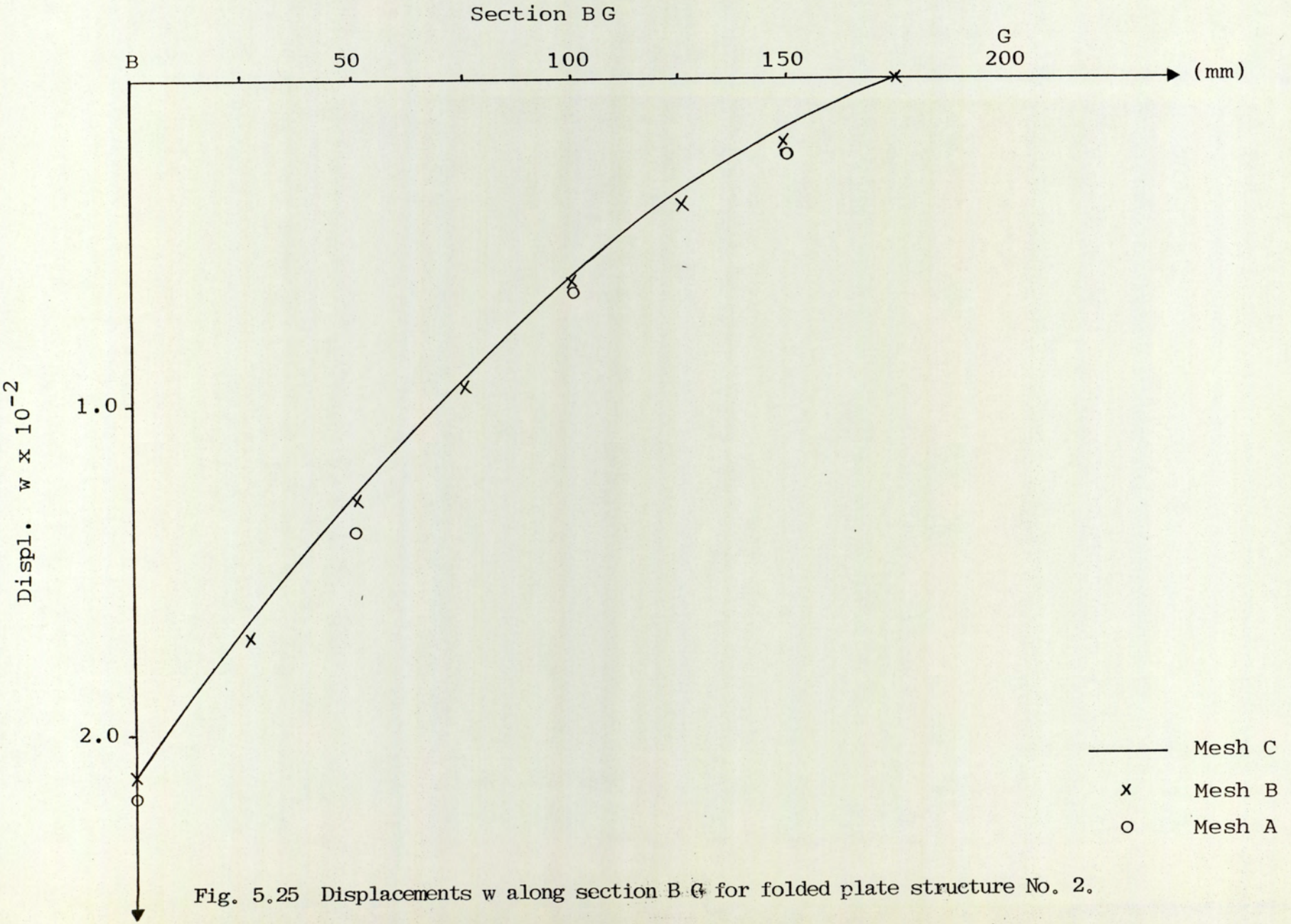


Fig. 5.25 Displacements w along section B G for folded plate structure No. 2.

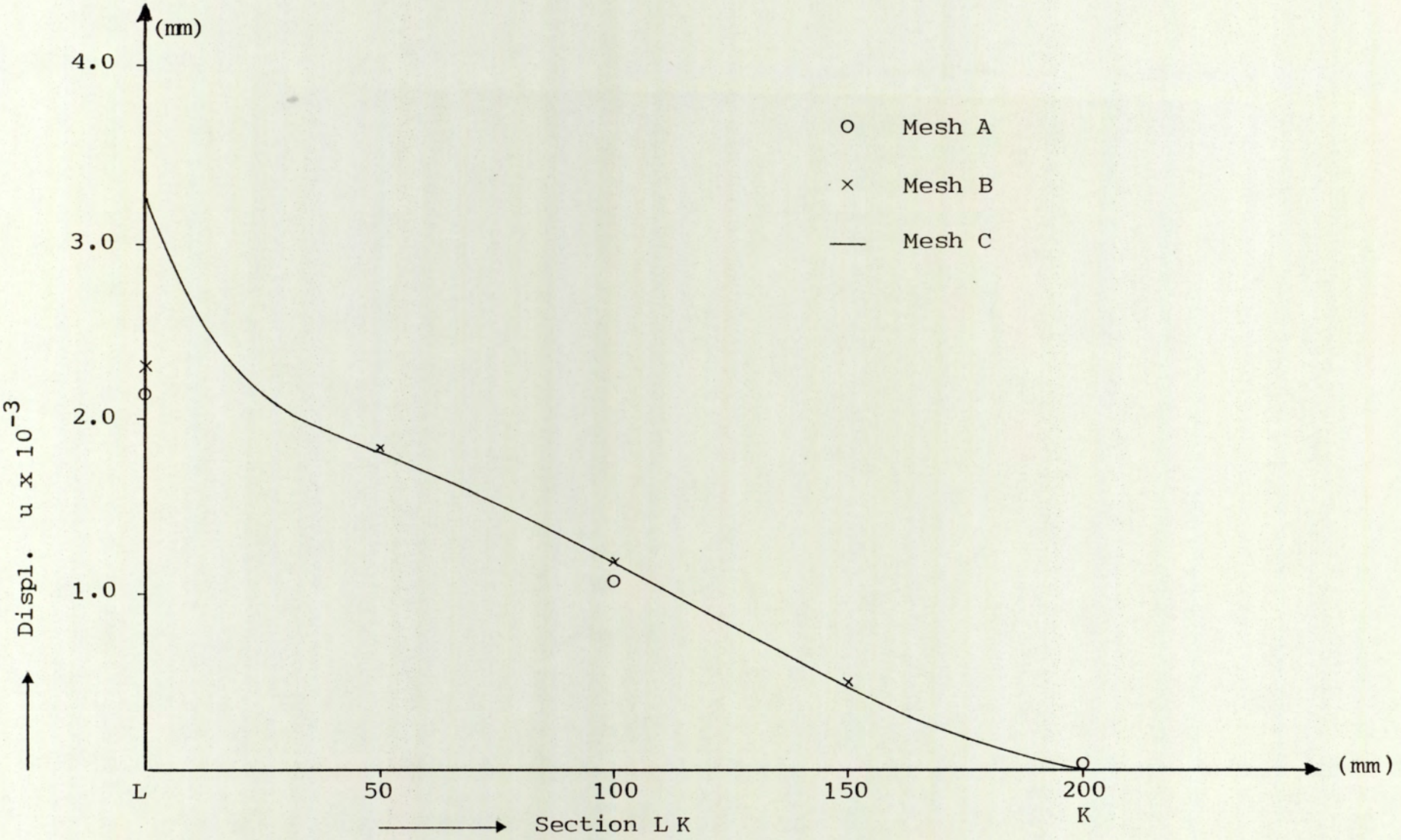


Fig.5.26 The displacements (u) along the section L K of folded plate structure No.2

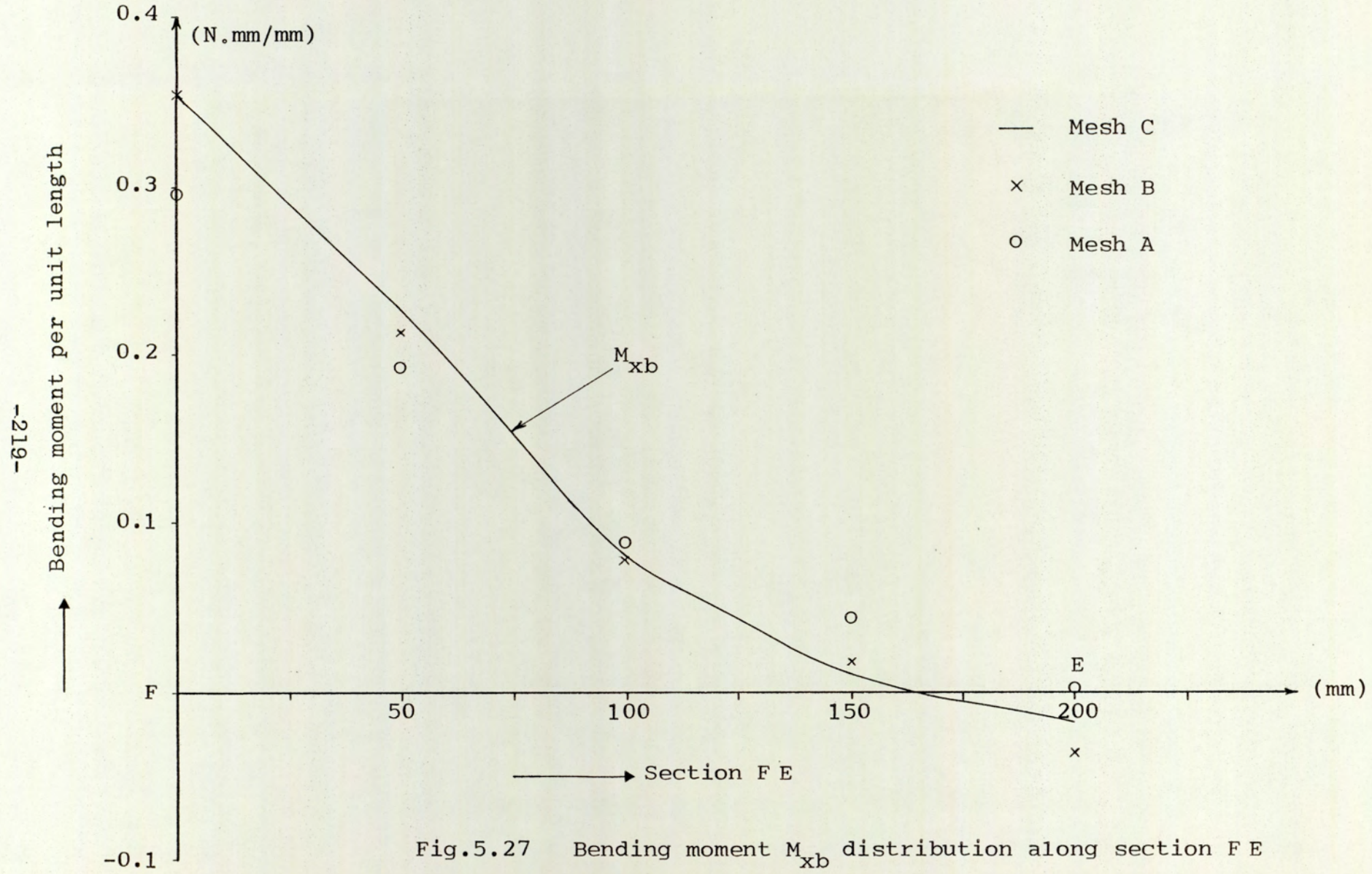


Fig.5.27 Bending moment M_{xb} distribution along section F E for folded plate structure No.2

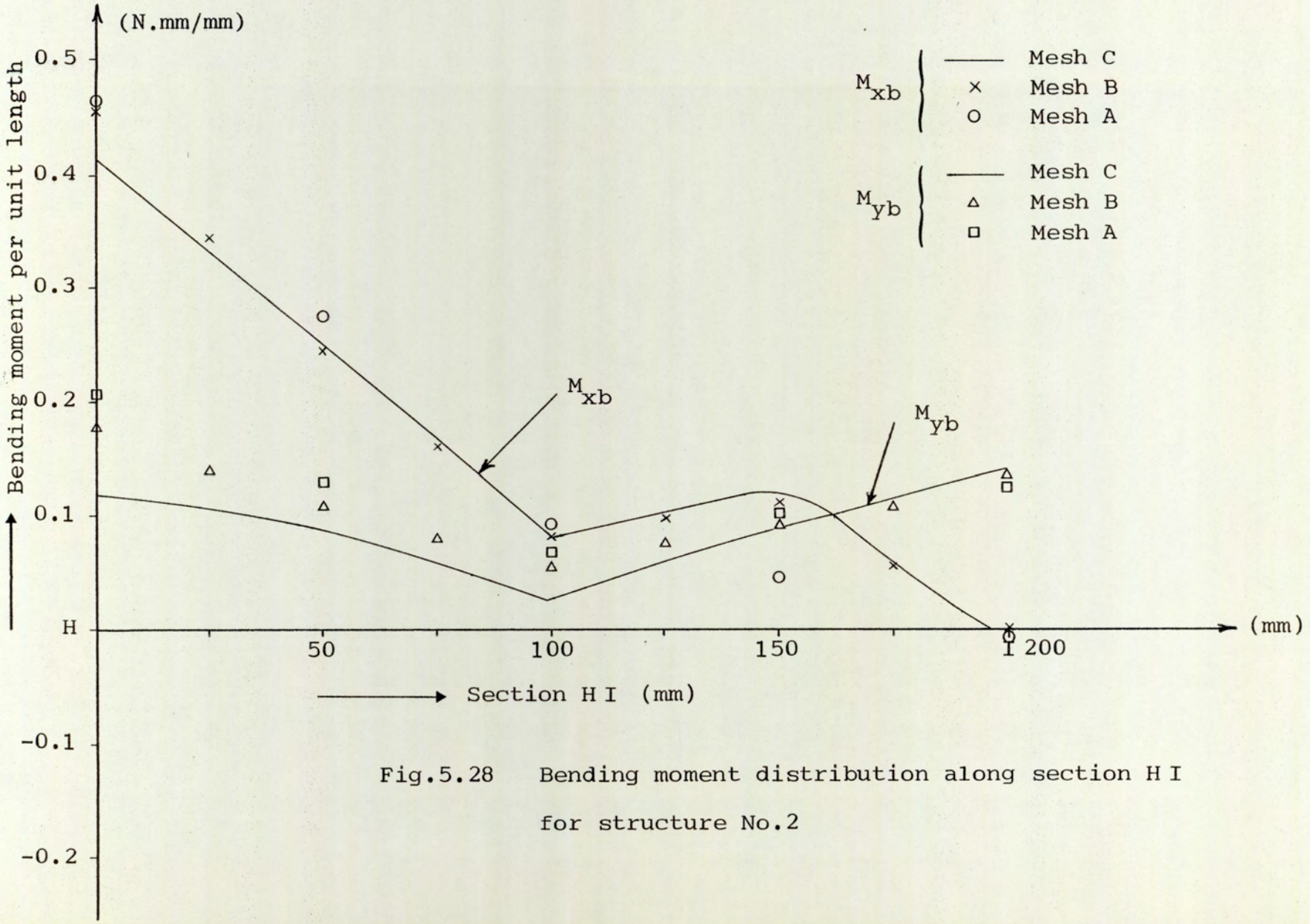
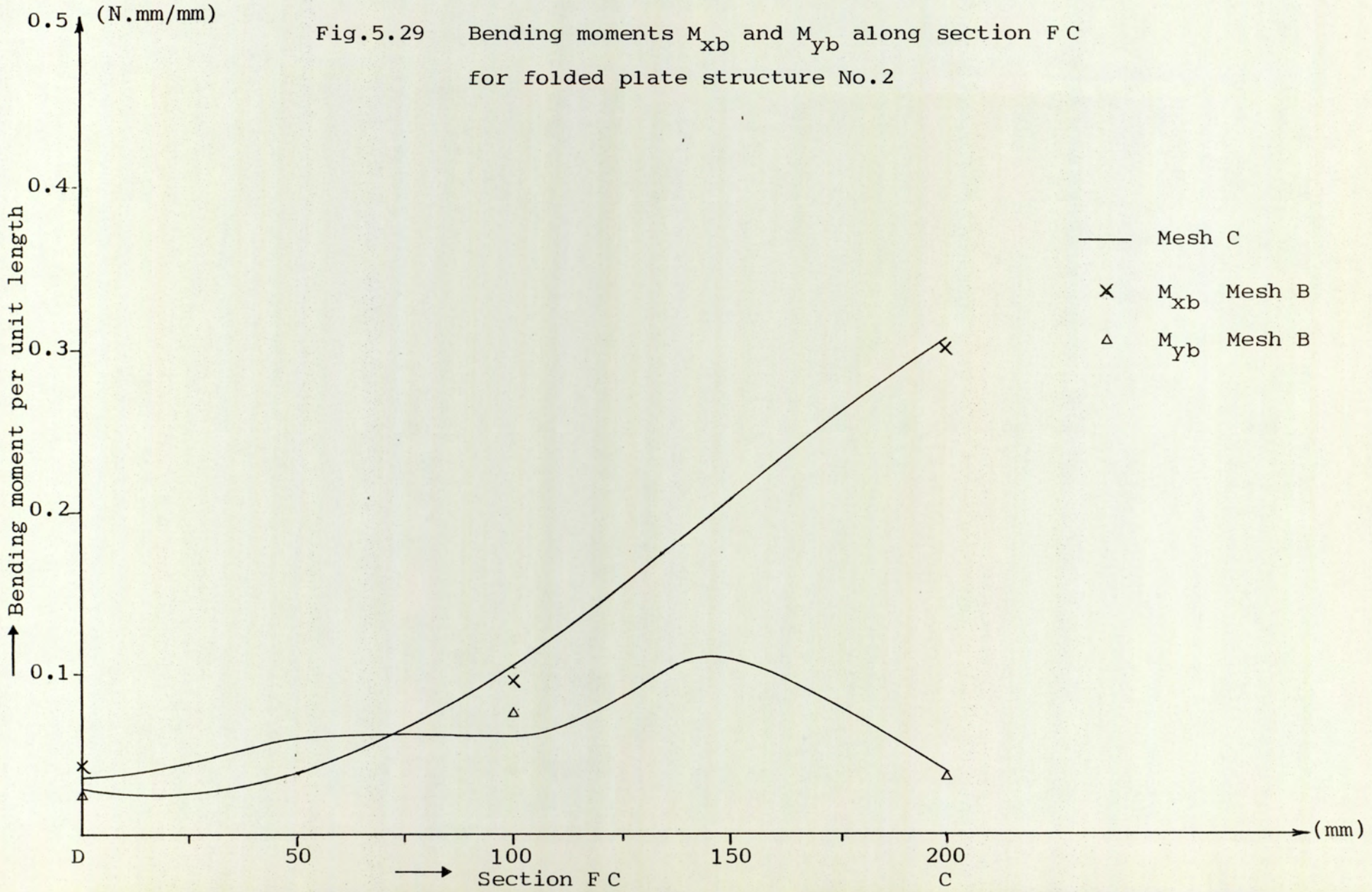


Fig.5.28 Bending moment distribution along section H I for structure No.2

Fig.5.29 Bending moments M_{xb} and M_{yb} along section F C for folded plate structure No.2



5.8 CLOSING REMARKS

The semiloof shell element has been used to solve plate and shell structures with different boundary shapes. Very good results were obtained confirming the value of the semiloof shell element and showing that the associated computer program developed in this work functional correctly.

In the next chapter, the programs developed in Chapter 4 are modified to exploit sectorial (or rotationally periodic) symmetry in structures: a significant feature in rotating fan impellers.

CHAPTER SIX

ROTATIONALLY PERIODIC STRUCTURES

6.1 INTRODUCTION

Axial symmetry exists in many engineering structures. A considerable range of computer programs have been developed by different researchers for analysing structures, taking full advantage of their axisymmetric behaviour in minimising computational effort in predicting performance.

If the structure is not exactly symmetric and cannot be represented by an axisymmetric idealisation, it is necessary to analyse the whole structure, which is often impracticable for computer storage and cost reasons. Some of these structures contain a series of identical structural elements, linked by identical junctions. Such types of construction with regularly repeated sections, has the property of periodicity (cyclic or sectorial symmetry). In this chapter, these rotationally periodic structures are considered with a view to analysing only one repeating sector. This approach would normally require much less computer time and storage than analysing the complete structure. The programs developed in Chapter 4 have been modified so as to match the cyclic symmetry of rotating fan impellers. This enables the designers efficiently to analyse a fan impeller using the developed finite element package. Numerical examples for testing the modified programs are also given. These are compared with known solutions.

6.2 CYCLIC SYMMETRY (ROTATIONALLY PERIODIC STRUCTURES)

Rotationally periodic structures consist of identical coupled substructures positioned symmetrically about an axis. In these structures, one can recognise a repetition of goemetry and loading. If the geometry of the structure is defined for any radial or axial position at some angle θ, it will be identical at $(\theta + n\,\phi_o)$, where ϕ_o is $\frac{2\pi}{N}$, and n and N are integers, N is structure dependent (the number of identical substructures that constitute the structure). Structures which possess the property of cyclic symmetry include rotating fan impellers, bladed turbine discs, centrifugal pumps and cooling towers.

Figure 6.1 shows a sector of a fan impeller (radial type) with 10 equispaced blades. If the geometry of the fan is completely defined for an angular segment of $\frac{2\pi}{10}$, such as the sector boundaries shown in Fig. 6.1, the rest of the fan can be generated by repeated rotation of the segment through $\frac{2\pi}{10}$ rad. Also the loading is repeated as the geometry is repeated. Stresses and deformations also display cyclic symmetry. Therefore, only one sector needs to be analysed. This technique has been used successfully by a number of investigators.

MacNeal, Harder and Mason, Ref. (73), have developed a method of simplifying the analysis of rotationally periodic

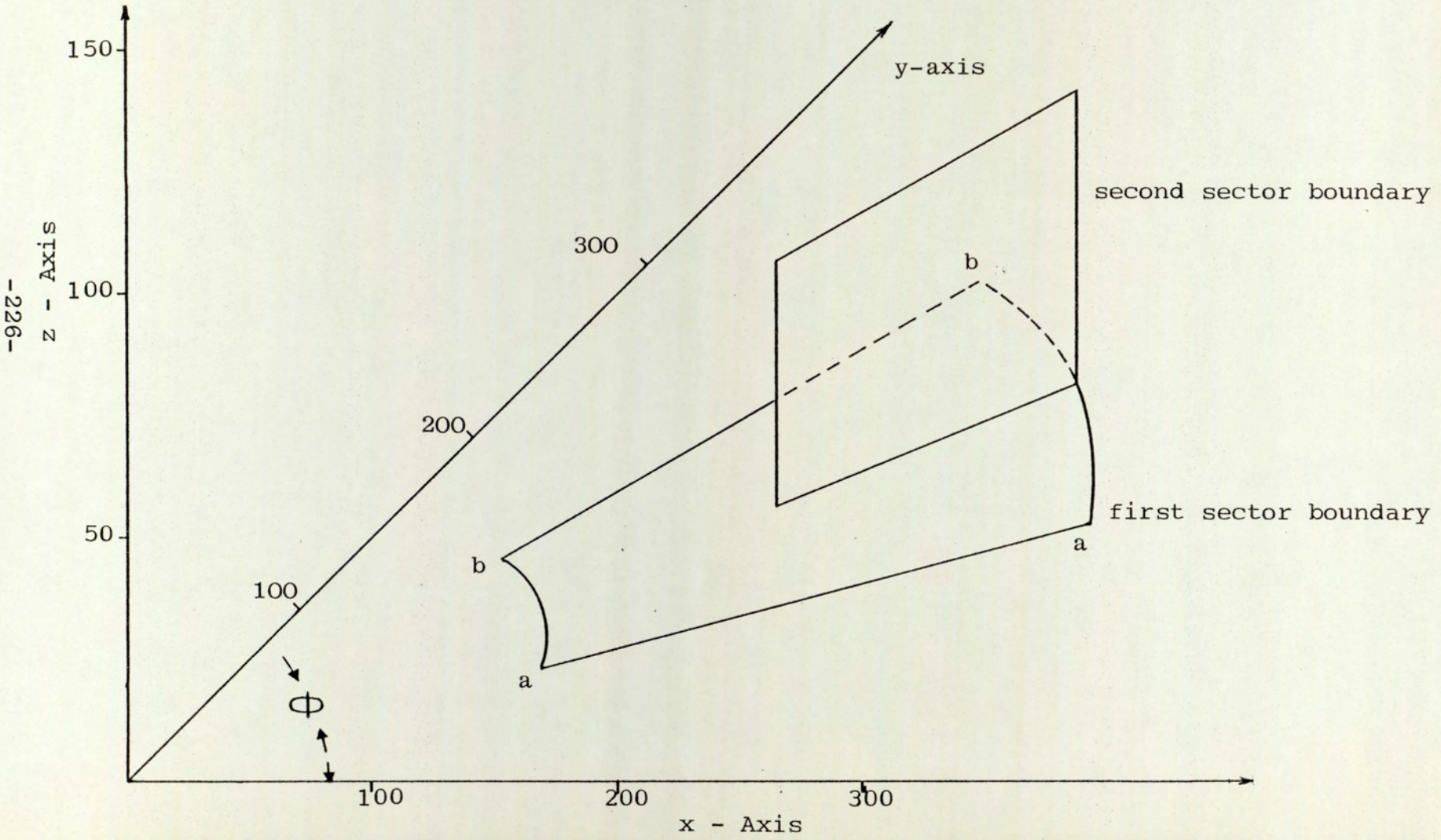


Fig.6.1 A typical sector of radial fan impeller

structures and have incorporated the technique in the finite element program NASTRAN, which can be used for static analysis, steady-state heat transfer analysis, and vibration analysis. Thomas, Ref.(74), has used the technique for studying natural frequency and mode shapes of rotationally periodic structures consisting of a finite number of identical substructures forming a closed ring.

Nelson, Ref.(75) has made use of the method adopted by Thomas in the investigation of cooling towers.

McEwan Ref.(76) has idealised curved bladed pump impellers as a sector bounded by two radial planes.

The implementation of cyclic symmetry, in the finite element programs is described in the following section. This will show how the analysis of a fan sector could be used to obtain an analysis of the complete fan impeller.

6.3 IMPLEMENTATION OF THE SECTORIAL SYMMETRY IN THE FINITE ELEMENT PROGRAMS

In this section, a method of exploiting the repeated similarities of geometry, material properties and loading of rotating fan impellers of complicated design, is introduced. This will reduce the amount of data preparation, memory storage and computing time required in evaluating mechanical design.

6.3.1 Theory of Repeated Structures

Figure 6.2 shows a rotating disc. This is an axisymmetric structure which can be regarded as a rotationally periodic structure where the periodicity tends to infinity; i.e. $N \to \infty$.

A typical segment between sections (a-a) and (b-b) is isolated. For anlaysis purposes a stiffness relationship can be written in the following form:

$$[K]\{q\} = \{Q\} \qquad \ldots \quad (6.1)$$

For the typical sector:

$$\begin{bmatrix} [K_{11}] & [K_{12}] & [K_{13}] \\ [K_{21}] & [K_{22}] & [K_{23}] \\ [K_{31}] & [K_{32}] & [K_{33}] \end{bmatrix} \begin{Bmatrix} \{q_1\} \\ \{q_2\} \\ \{q_3\} \end{Bmatrix} = \begin{Bmatrix} \{Q_1\} \\ \{Q_2\} \\ \{Q_3\} \end{Bmatrix} \qquad \ldots \quad (6.2)$$

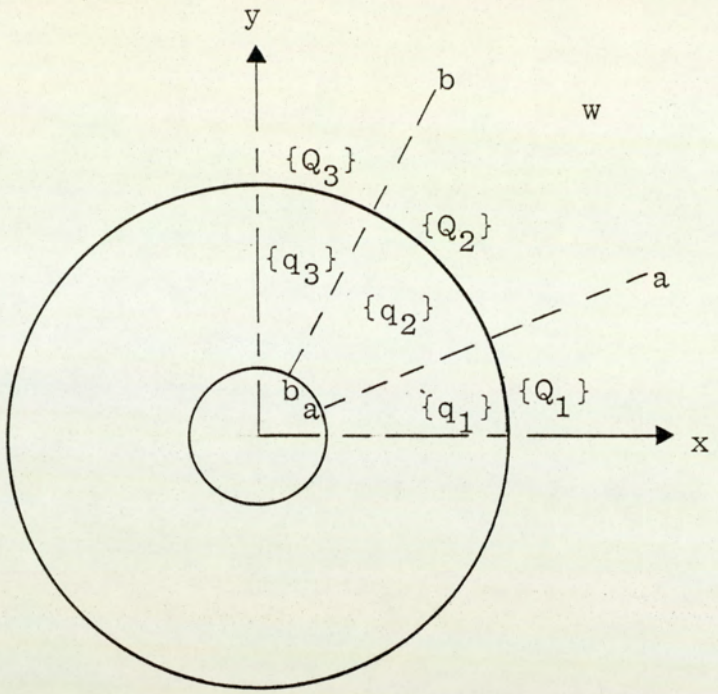


Fig. 6.2 Rotating Disc

where $\{q_1\}$ and $\{q_3\}$ represent degrees of freedom on edges (a-a) and (b-b) respectively, and $\{q_2\}$ represent all other degrees of freedom of the structure. Forces $\{Q_1\}$ and $\{Q_3\}$ are applied loads.

Subscript (1) refers to nodes on boundaries (a-a)

Subscript (2) refers to internal nodes

Subscript (3) refers to nodes on boundaries (b-b).

But $\{q_1\} = \{q_3\}$, therefore equation (6.2) can be written as follows:

$$\begin{bmatrix} [K_{11}]+[K_{33}]+[K_{13}]+[K_{31}] & [K_{11}]+[K_{31}] \\ [K_{11}]+[K_{13}] & [K_{11}] \end{bmatrix} \begin{Bmatrix} \{q_1\} \\ \{q_2\} \end{Bmatrix} = \begin{Bmatrix} \{Q_1\}+\{Q_3\} \\ \{Q_2\} \end{Bmatrix} \quad \ldots \quad (6.3)$$

Thus the complete repeated structure can be solved by solving one sector with corresponding contributions along the internal boundaries assembled into the same location and treated as one set of variables.

Equation (6.3) can be produced automatically by the element assembly process, which can be achieved by:

(i) Giving the same node number to cyclically symmetric nodes along (a-a) and (b-b). This numbering is only for element assembly purposes. Actual point co-ordinate must be used in the formulation of element stiffness matrices.

(ii) Ensuring the element division along each of the repeated boundaries is the same. This means that each node will have a corresponding node with exactly the same behaviour. If the disc in Fig. 6.2 is idealised, a small sector is used with symmetry conditions of zero circumferential displacement on the bounding radial planes (a-a) and (b-b). If the mesh in Fig. 6.3 is considered; then node 1 will have node 17 with exactly the same behaviour , ... etc. Also the number and spacing of elements (a-a) must be the same

along (b-b). Table [St] is set up to contain the pairs of similar behaviour nodes.

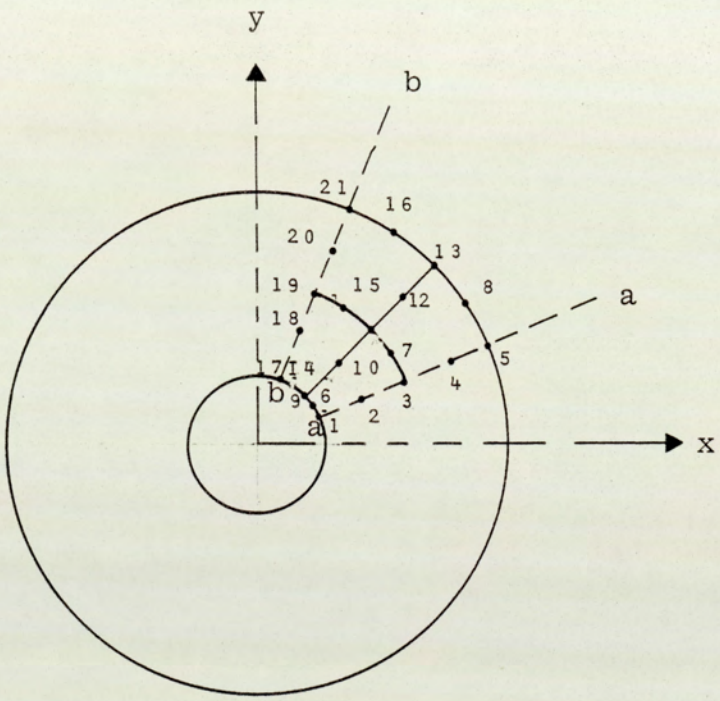


Fig.6.3 Sector Idealisation

For the mesh shown in Fig. 6.3, [St] is constructed as follows:

$$[St] = \begin{bmatrix} 1 & 17 \\ 2 & 18 \\ \cdot & \cdot \\ \cdot & \cdot \\ \cdot & \cdot \\ \cdot & \cdot \\ 7 & 21 \end{bmatrix}$$

This will be used during the assembly in order that the contribution to both similar nodes are assembled in the same location. After solution, this Table is used again to give the similar nodes the same displacements. The nodal stresses are calculated in the normal way.

6.3.2 Modifications of the Developed Finite Element Programs to Include Sectorial Symmetry

In cases of complicated plate and shell structures in which the resulting stiffness matrix is large and cannot be held in the computer memory, the segmental technique was suggested in Chapter 4 to solve these problems. This was implemented in a package called "IMPSMF". The fan impeller is an example of a complicated plate and shell structure in which the cyclic symmetry of substructures exist. As mentioned above, the resulting stiffness matrix is large, therefore, in adopting the cyclic symmetry in the finite element programs, it is essential to use the segmental technique. In the current discussions, the developed package "IMPSMF" is modified to include the sectorial symmetry of repeating structures. This is similar to the "SMILOF" package except that the modifications of cyclic symmetry and segmental technique are applied here. A code is introduced in "IMPSMF" for distinguishing between the problems of repeating structures and normal structures. The input data is very similar

to that described in Section 4.3.2.1 for the "SMILOF". A new variable (Nk) is introduced with the control variables. The (Nk) refers to the number of nodes on a side of repeated boundary (e.g. Nk=7 for the sector in Fig. 6.3). If Nk=0, then the programs execution is directed as normal complicated plate and shell structures. In order to include the cyclic symmetry in the "IMPSMF" package, the subroutines "Addarray", "Assembly" and "Skewedcon" are modified and the subroutines "Skewim1" and Skewim2) are introduced.

The flow charts of the modified subroutines are not drawn, owing to the similarity to the original subroutines. However, the listings are given in Appendix B. The modified subroutines are presented briefly below:

(a) Subroutine "Maddarray"

The subroutine "Addarray" described in Section 4.3.2.2. is modified to take into account the cyclic symmetry. A loop is set for the (Nnode-Nk). This is because, only the nodes on one repeated boundary are considered. Another new loop is set inside the main loop. Here, each node is checked to see if it is repeated on Section (b-b) of Fig. 6.3; the number of this node is made the same as the corresponding node on Section (a-a).. The other steps to form the subroutine "Maddarray" are as they were

described in Section 4.3.2.2. However, the listing of this subroutine is given in Appendix B.

(b) Subrountine "Massembly"

In this subroutine, the location of the element stiffness coefficients in a one-dimensional array [K] is done as explained in Section 4.3.2.4, using the modified address array sequence. The location is done automatically for the similar nodes on the repeated boundaries. The load vector is split into two vectors:

(i) {Q} vector is used to keep the nodal loads for the nodes of boundary (a-a) and the internal nodes, i.e. ($\{Q_1\}$ and $\{Q_2\}$).

(ii) $\{Q_3\}$ is used to keep the nodal loads for the nodes on the boundary (b-b).

These nodal loads are formed automatically for the body forces inside this subroutine in the direction of the global axes as explained in Section 4.3.2.5 . The listing of this subroutine is given in Appendix B.

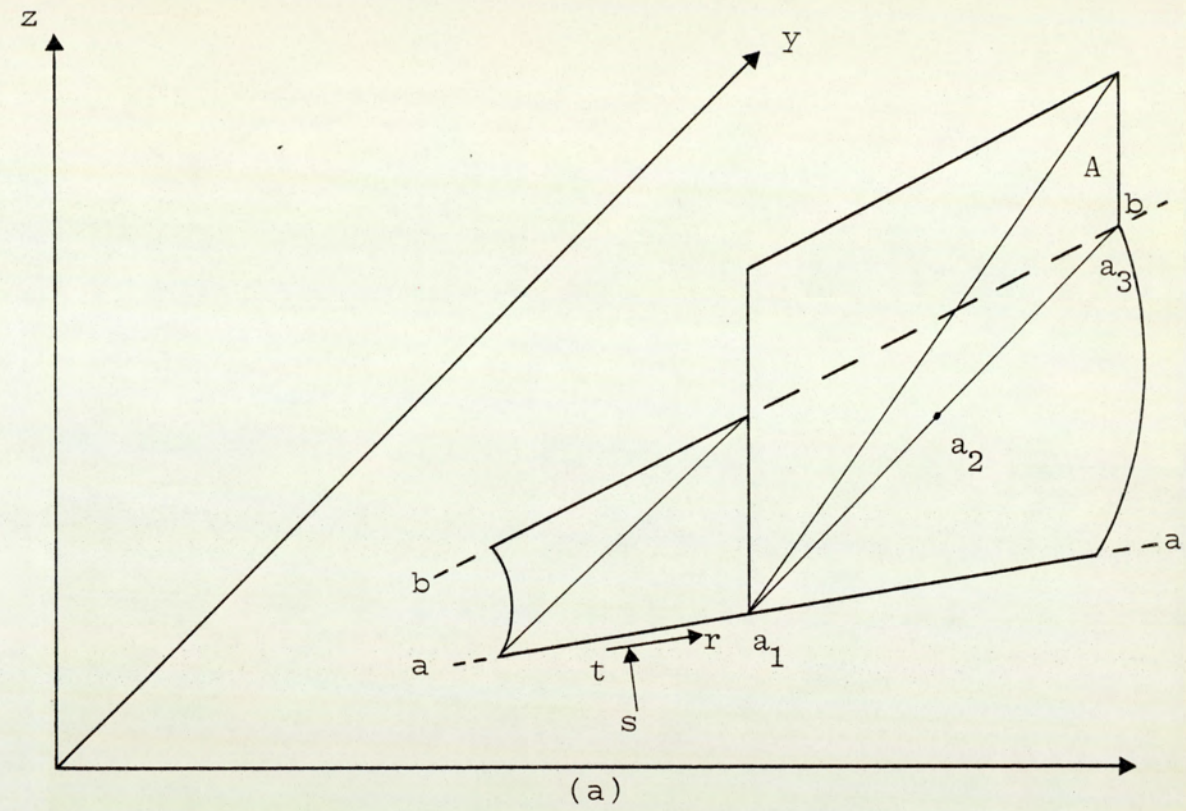
(c) Subroutine "Mskewedcon"

Following the discussions presented in Section 4.3.2.6, Skewed conditions prevail on the sector boundaries (a-a)

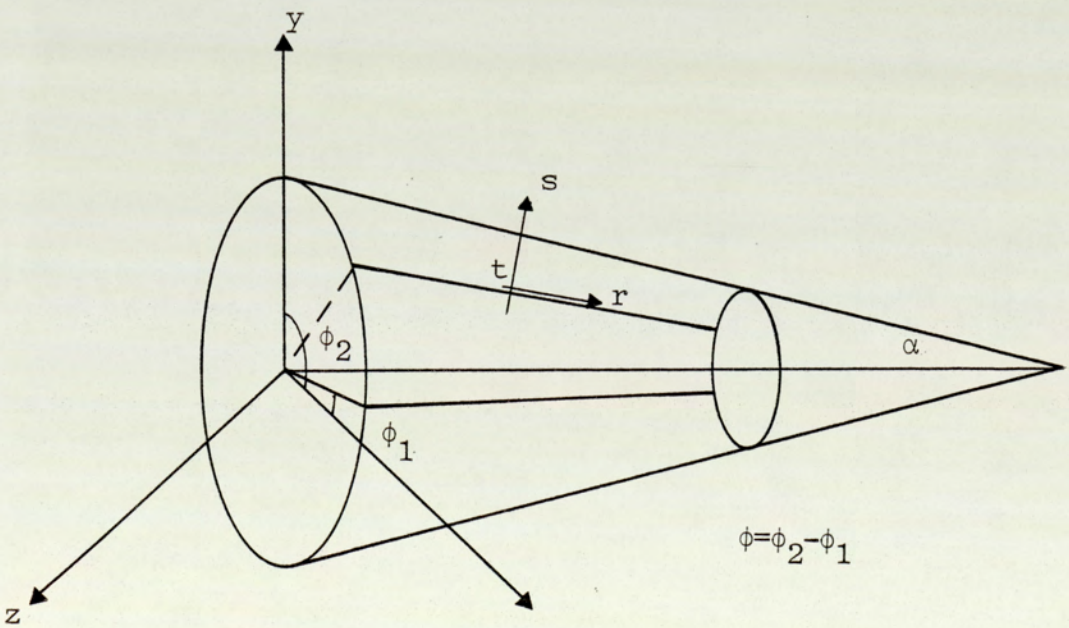
and (b-b). The method of constraining the nodes on the repeated boundaries involves prescribing the nodes on one section which is (a-a) with respect to local axes. The program automatically takes into consideration the concept of two similar nodes on sector boundaries having equal displacements. These displacements are prescribed in the directions of local axes, therefore, a transformation of the system matrices is required between the constructed local axes (r, s, t) and the global axes (x, y, z).

For a backsheet element of the fan impellers, the transformation matrix [T] is exactly the same as that defined in equation (4.48) and (4.49). To obtain [T] for a conesheet element, the cone of Fig. 6.4(b) is considered. A sector is defined between ϕ_1 and ϕ_2. The local axes are r, s and t. r is directed along the longitudinal direction, t tangent to the parallel circle, while s is perpendicular to both, and α is the semi-cone angle. From geometry, the relation between the global axes (x, y and x) and the local axes (r, s and t) may be written as follows:

$$\begin{Bmatrix} x \\ y \\ z \end{Bmatrix} = \begin{bmatrix} -\cos\phi\cos\alpha & -\sin\alpha\cos\phi & -\sin\phi \\ -\sin\phi\cos\alpha & -\sin\phi\sin\alpha & \cos\phi \\ -\sin\alpha & \cos\alpha & 0 \end{bmatrix} \begin{Bmatrix} r \\ s \\ t \end{Bmatrix} \quad \ldots \quad (6.4)$$



(a) Typical sector of radial fan impeller



(b) Typical sector of the conesheet

Fig.6.4 Typical repeated structures

A code is introduced to distinguish between the backsheet, blade and conesheet elements, in order to choose the proper transformation matrix [T]. This is achieved by the type of element. It is stored in the last column of the nodal connections matrix. The type of element is either 1, 2 or 3 respectively, referring to the backsheet, blade or conesheet element. If Fig. 6.4(a) is considered (intersection of backsheet and the blade), the blade element A contains the nodes a_1, a_3 on the sector boundaries. In this case the type of this element should be 1 (since a_1 and a_3 are common on the sector boundary between the blade and the backsheet).

In the case of intersection of the conesheet with the blade on the sector boundaries (if it exists), the blade element type will be 3. For a general plate and shell structure, the element type is 1.

Having the suitable transformation matrix [T], the element stiffness matrix is transformed from the global axes to the local axes using equation (4.54). This subroutine follows the same procedure of that explained in Section 4.3.2.6 and the listing is given in Appendix B.

(d) Subroutine "Skewim1"

As mentioned before, the body forces per unit volume due to gravity and rotational forces are evaluated automatically in the subroutine "Massembly". These are in the directions of the global axes. With cyclic symmetry, the nodes on the sector boundaries (a-a) and (b-b) are specified with respect to the directions of the constructed local axes. Their stiffnesses are transferred to these local axes in the subroutine "Mskewedcon". Therefore, the loads acting on these nodes i.e. $\{Q_1\}$ and $\{Q_3\}$ must be in the direction of these axes. This subroutine does this job and transfers the loading resulting from the subroutine "Massembly", to the local axes directions (r, s, t). The procedure used is similar to that explained in Section 4.3.2.7. A flow chart of this subroutine, shown in Fig. 6.5, follows and the steps involved are, with reference to the flow chart:

1. A loop is set for the number of skewed nodes.

2. Each node is checked to see if it is on the boundary (a-a) or (b-b) in order to define its position in $\{Q_3\}$ or in $\{Q\}$.

3. The proper coefficients of the transformation matrix are calculated.

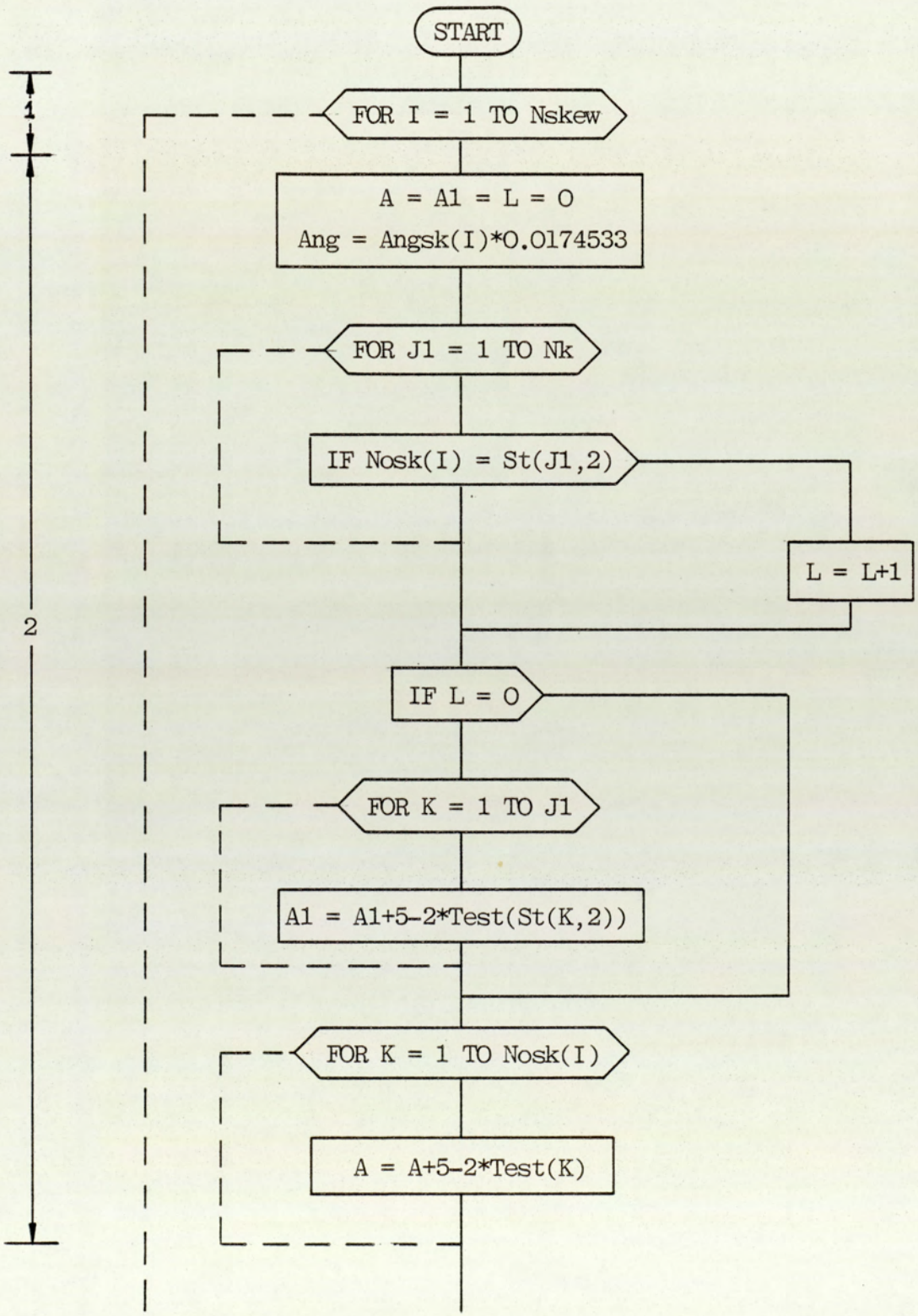
4. The load vector is modified for the current skewed node using equation (4.54).

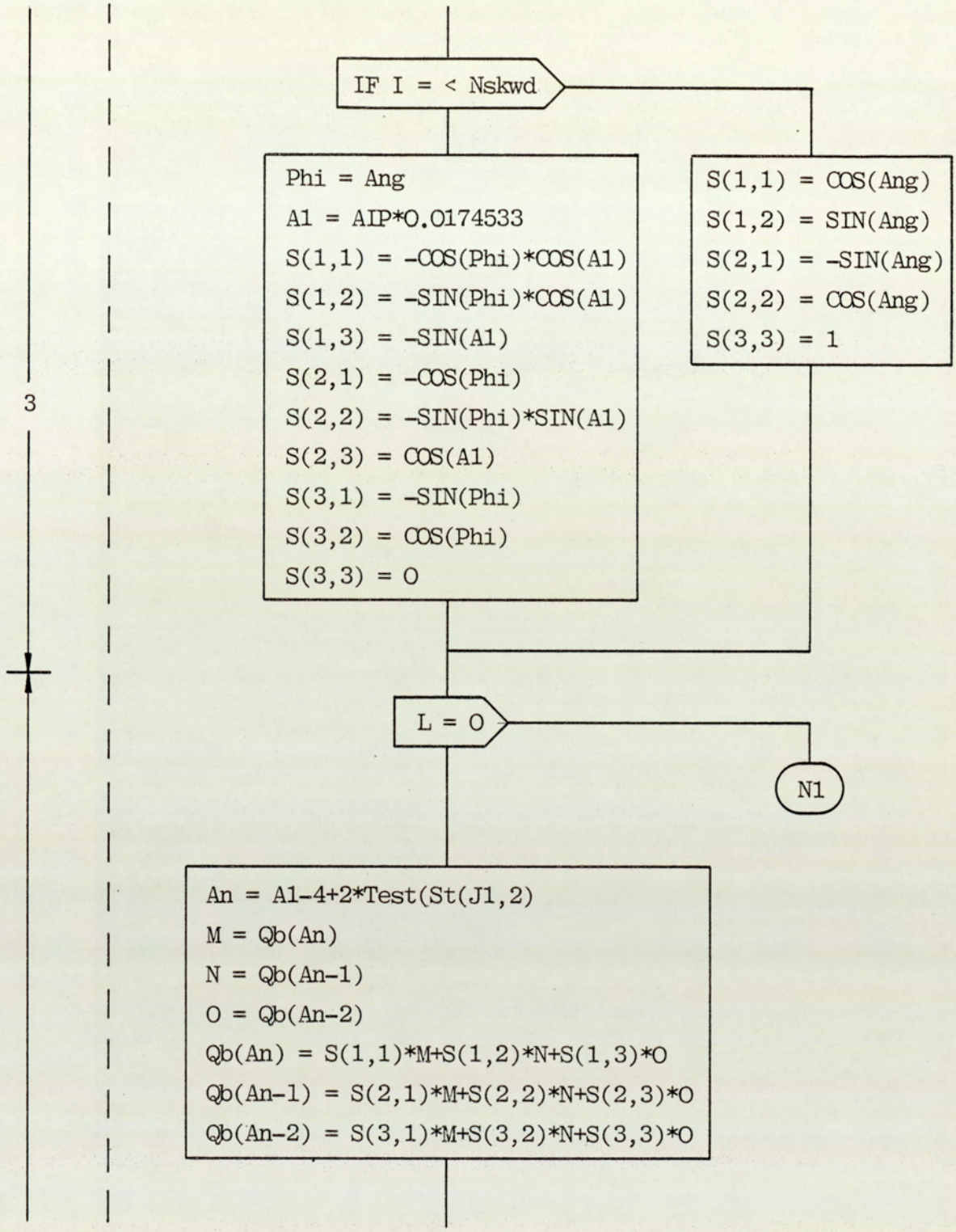
Steps 2-4 are repeated for the other skewed nodes.

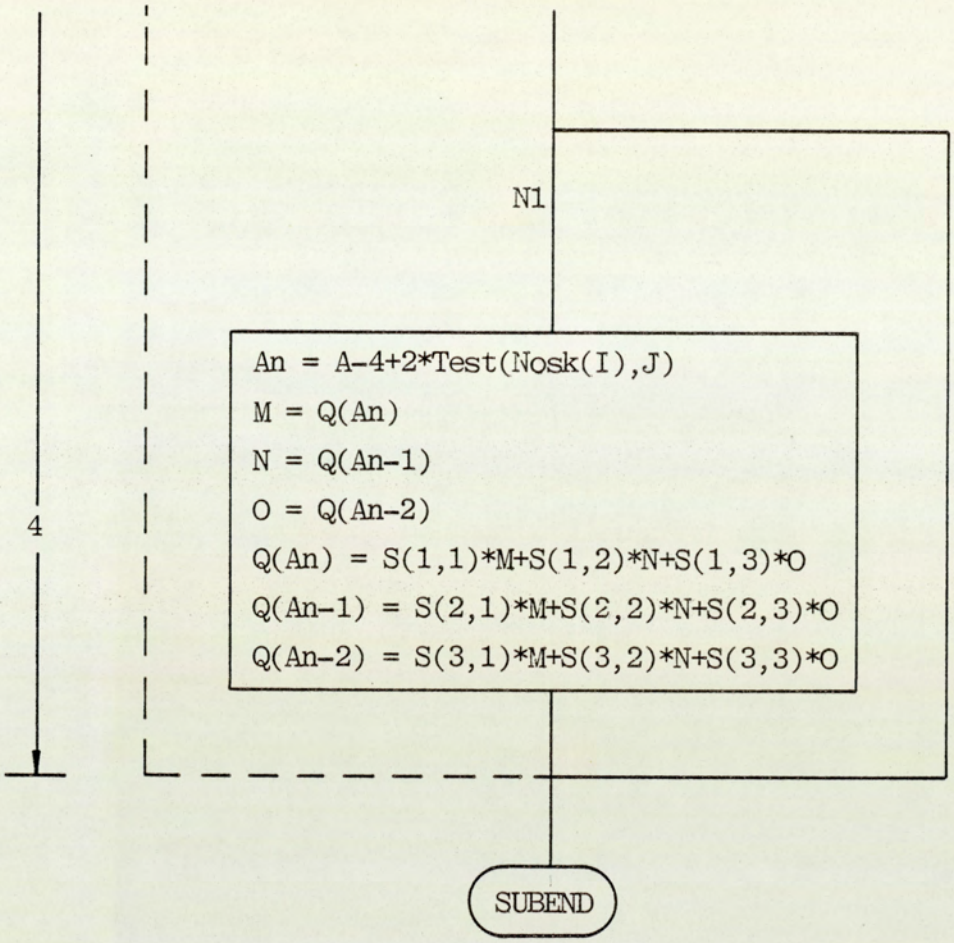
Having the loads in the local axes directions, the loads on Sections (a-a) and (b-b) are added together as defined in equation (6.3). This is performed in the main program. The prescribed displacements are applied using the subroutine "Geombc". The overall system equations can now be solved to determine values for nodal displacements. The resulting displacements on Section (a-a), which are the same as on (b-b) are in the local coordinate system.

Fig. 6.5 Flow Chart of Subroutine 'Skewim1"

SUB Skewim1 (Q(*), Angsk(*), Qb(*), AIP, INTEGER Test(*), J, Nosk(*), Nskew , St(*), Skewd, Skewc).




IF I = < Nskwd
Phi = Ang
A1 = AIP*0.0174533
S(1,1) = -COS(Phi)*COS(A1)
S(1,2) = -SIN(Phi)*COS(A1)
S(1,3) = -SIN(A1)
S(2,1) = -COS(Phi)
S(2,2) = -SIN(Phi)*SIN(A1)
S(2,3) = COS(A1)
S(3,1) = -SIN(Phi)
S(3,2) = COS(Phi)
S(3,3) = 0
S(1,1) = COS(Ang)
S(1,2) = SIN(Ang)
S(2,1) = -SIN(Ang)
S(2,2) = COS(Ang)
S(3,3) = 1
L = 0
N1
An = A1-4+2*Test(St(J1,2)
M = Qb(An)
N = Qb(An-1)
O = Qb(An-2)
Qb(An) = S(1,1)*M+S(1,2)*N+S(1,3)*O
Qb(An-1) = S(2,1)*M+S(2,2)*N+S(2,3)*O
Qb(An-2) = S(3,1)*M+S(3,2)*N+S(3,3)*O


N1
An = A-4+2*Test(Nosk(I),J)
M = Q(An)
N = Q(An-1)
O = Q(An-2)
Q(An) = S(1,1)*M+S(1,2)*N+S(1,3)*O
Q(An-1) = S(2,1)*M+S(2,2)*N+S(2,3)*O
Q(An-2) = S(3,1)*M+S(3,2)*N+S(3,3)*O
4
SUBEND

(e) Subroutine "Skwim2"

At this subroutine, the displacements on the sector boundaries (a-a) and (b-b) are converted back to the global axes. The final displacements are in the global directions. A flow chart, shown in Fig. 6.6, and the steps involved are, with reference to the flow chart:

1. A loop is set for the number of skewed nodes.

2. The table [St] is used to define the position of the current skewed node on the sector boundaries.

3. Using step 2, the coefficients of the displacements for the current skewed node are defined.

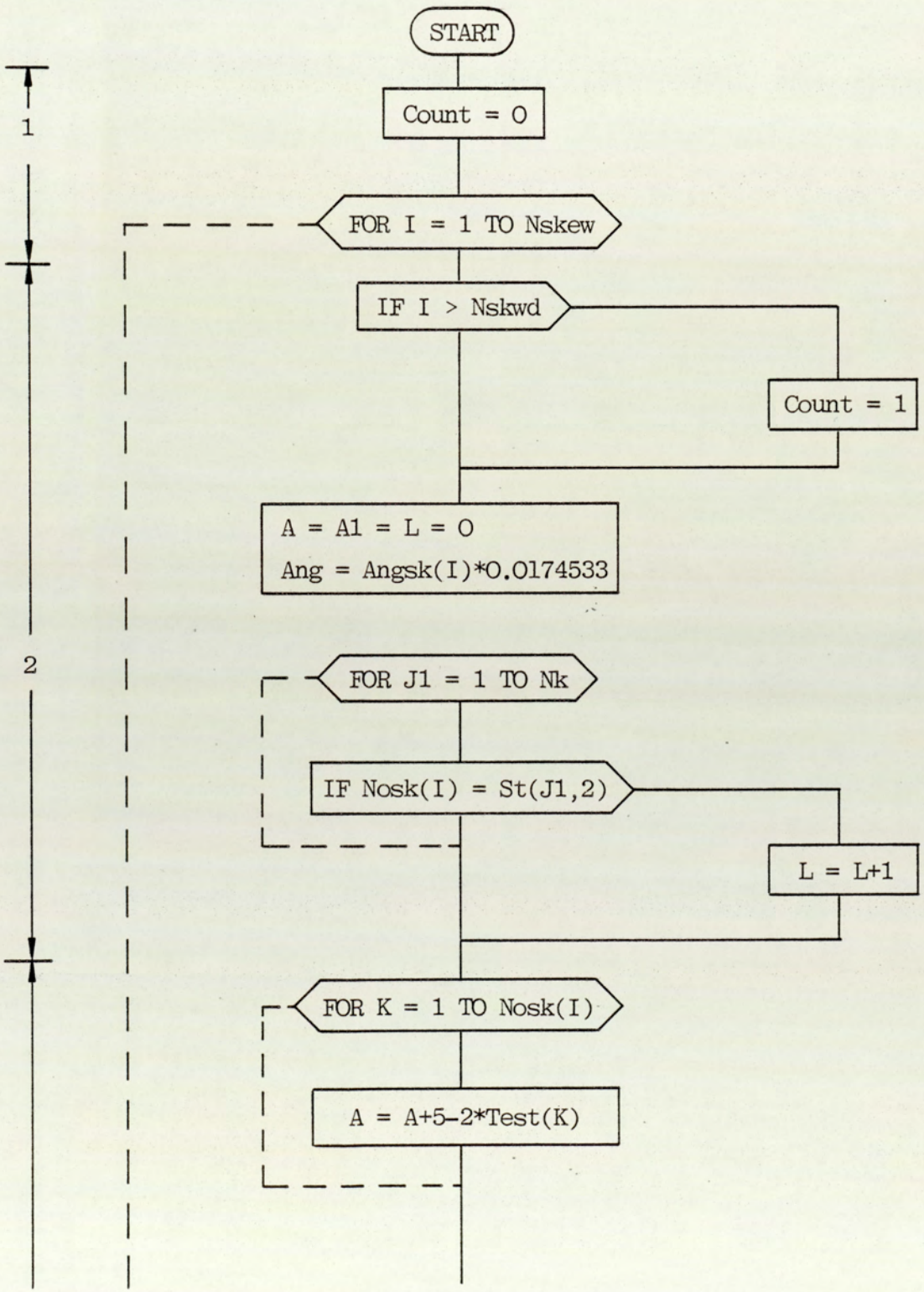
4. The proper coefficients of the trnasformation matrix are calculated.

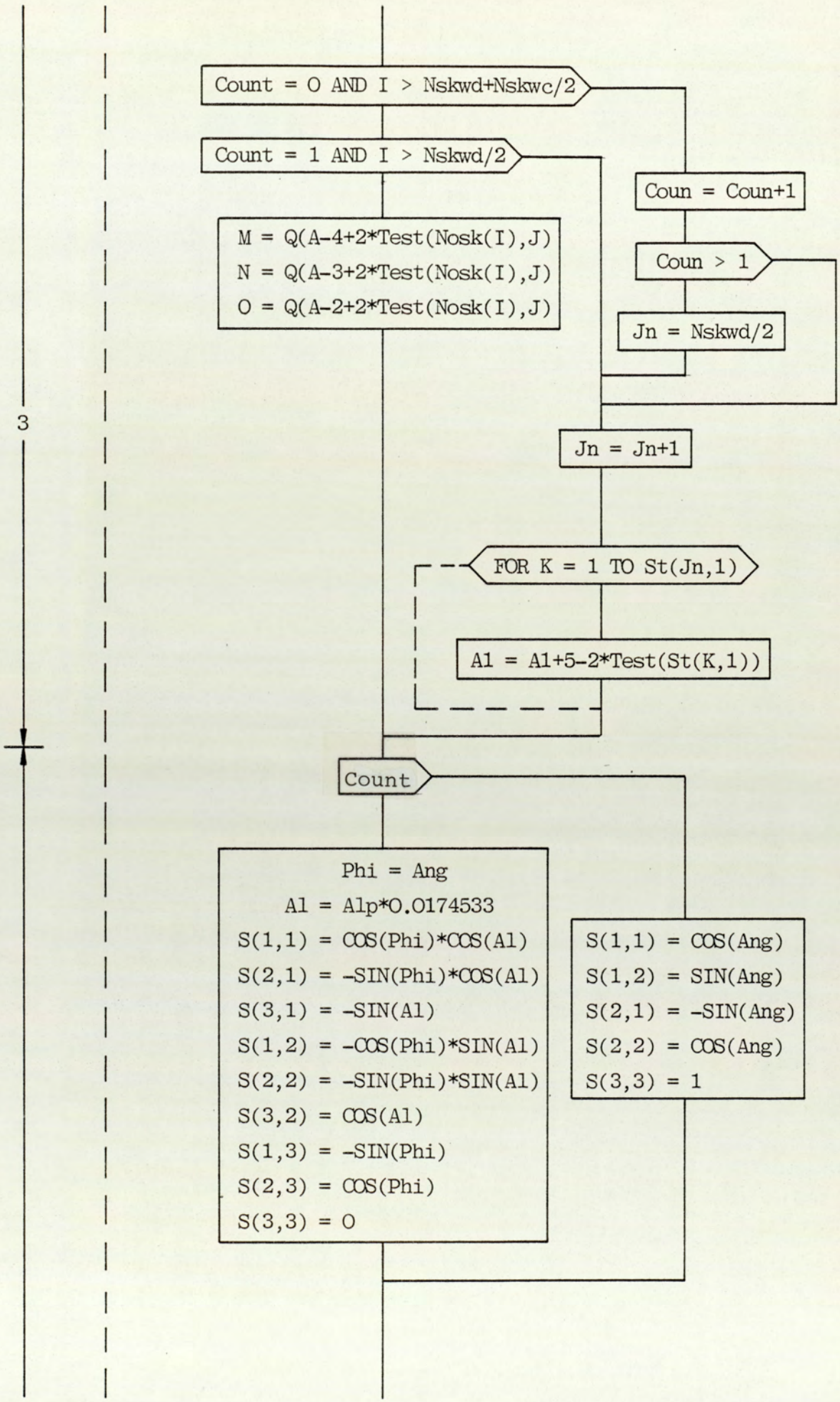
5. The displacement vector {Q} is modified.

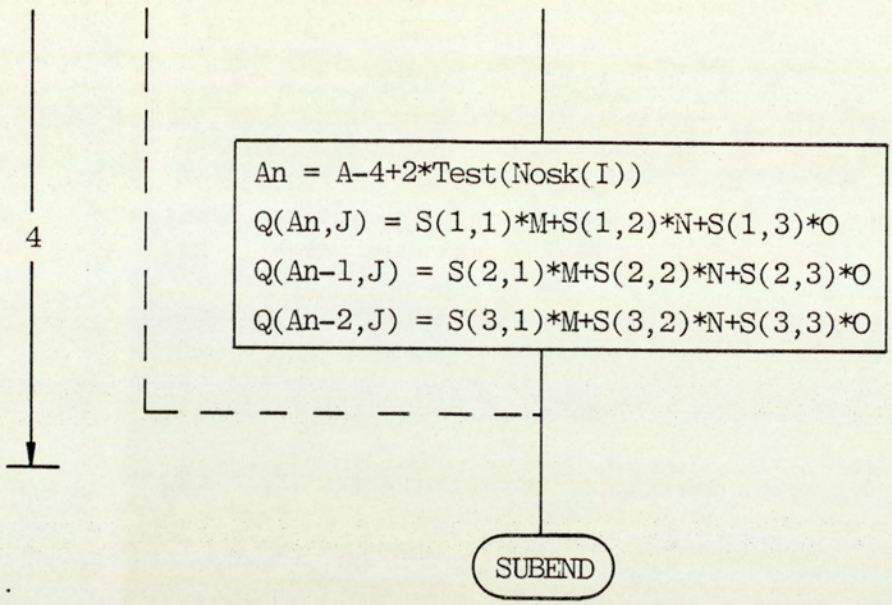
Steps 2-5 are repeated for the other skewed nodes. Final displacements are in the global directions.

Fig.6.6 Flow Chart of Subroutine "Skewim2"

SUB Skewim2 (Q(*),Angsk(*), Qb(*), AIP, INTEGER, Test(*), J, Nosk(*), Nskew, St(*), Nk, Nskwd, Nskwc).




Count = O AND I > Nskwd+Nskwc/2
Count = 1 AND I > Nskwd/2
Coun = Coun+1
M = Q(A-4+2*Test(Nosk(I),J)
N = Q(A-3+2*Test(Nosk(I),J)
O = Q(A-2+2*Test(Nosk(I),J)
Coun > 1
Jn = Nskwd/2
Jn = Jn+1
FOR K = 1 TO St(Jn,1)
A1 = A1+5-2*Test(St(K,1))
Count
Phi = Ang
Al = Alp*O.0174533
S(1,1) = COS(Phi)*COS(Al)
S(2,1) = -SIN(Phi)*COS(Al)
S(3,1) = -SIN(Al)
S(1,2) = -COS(Phi)*SIN(Al)
S(2,2) = -SIN(Phi)*SIN(Al)
S(3,2) = COS(Al)
S(1,3) = -SIN(Phi)
S(2,3) = COS(Phi)
S(3,3) = O
S(1,1) = COS(Ang)
S(1,2) = SIN(Ang)
S(2,1) = -SIN(Ang)
S(2,2) = COS(Ang)
S(3,3) = 1

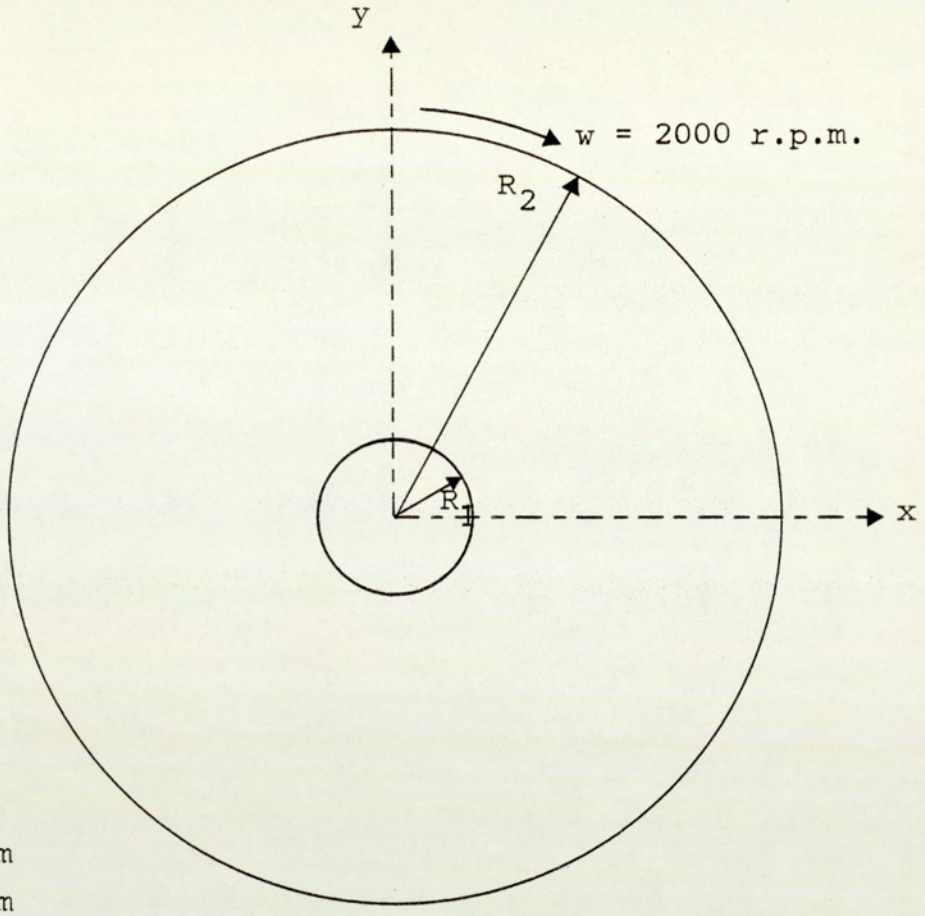

4
An = A-4+2*Test(Nosk(I))
Q(An,J) = S(1,1)*M+S(1,2)*N+S(1,3)*O
Q(An-1,J) = S(2,1)*M+S(2,2)*N+S(2,3)*O
Q(An-2,J) = S(3,1)*M+S(3,2)*N+S(3,3)*O
SUBEND

6.4 TESTED EXAMPLES

The choice of the examples aims at having a problem with a known solution, preferably an exact theoretical one for test purposes. This will make it possible to show the use of the constraints on the sector boundaries, which equals the displacements of two given identical nodes such as the rotating discs and the rotating cones. The convergence studies are carried on for another complicated structures such as the radial fan impeller with and without conesheet.

6.4.1 Rotating Disc

This example concernsthe use of the semiloof shell element in rotating parts. A simple disc is taken to establish the performance of the constraining technique explained in this Chapter, compared to a known solutions. For this purpose a sector of angle 20^{o} is used with symmetry conditions (tangential displacement v=0) on the bounding radial planes. The geometry and the meshes used are shown in Fig. 6.7. Figures 6.8 and 6.9 show the excellent agreement between the finite element results and the Timoshenko solution Ref.(1) for both radial and tangential stresses.



t = 4 mm
R_1 = 100 mm
R_2 = 510 mm
E = 2.1E5 N/mm^2
υ = 0.29
ρ = 7.8 E-9 N/mm^3

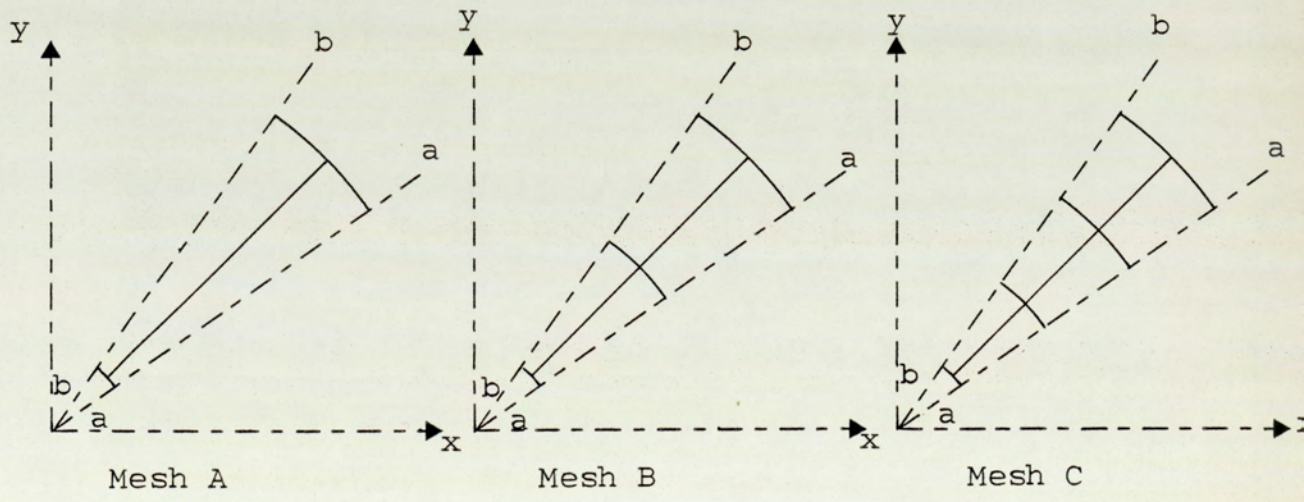
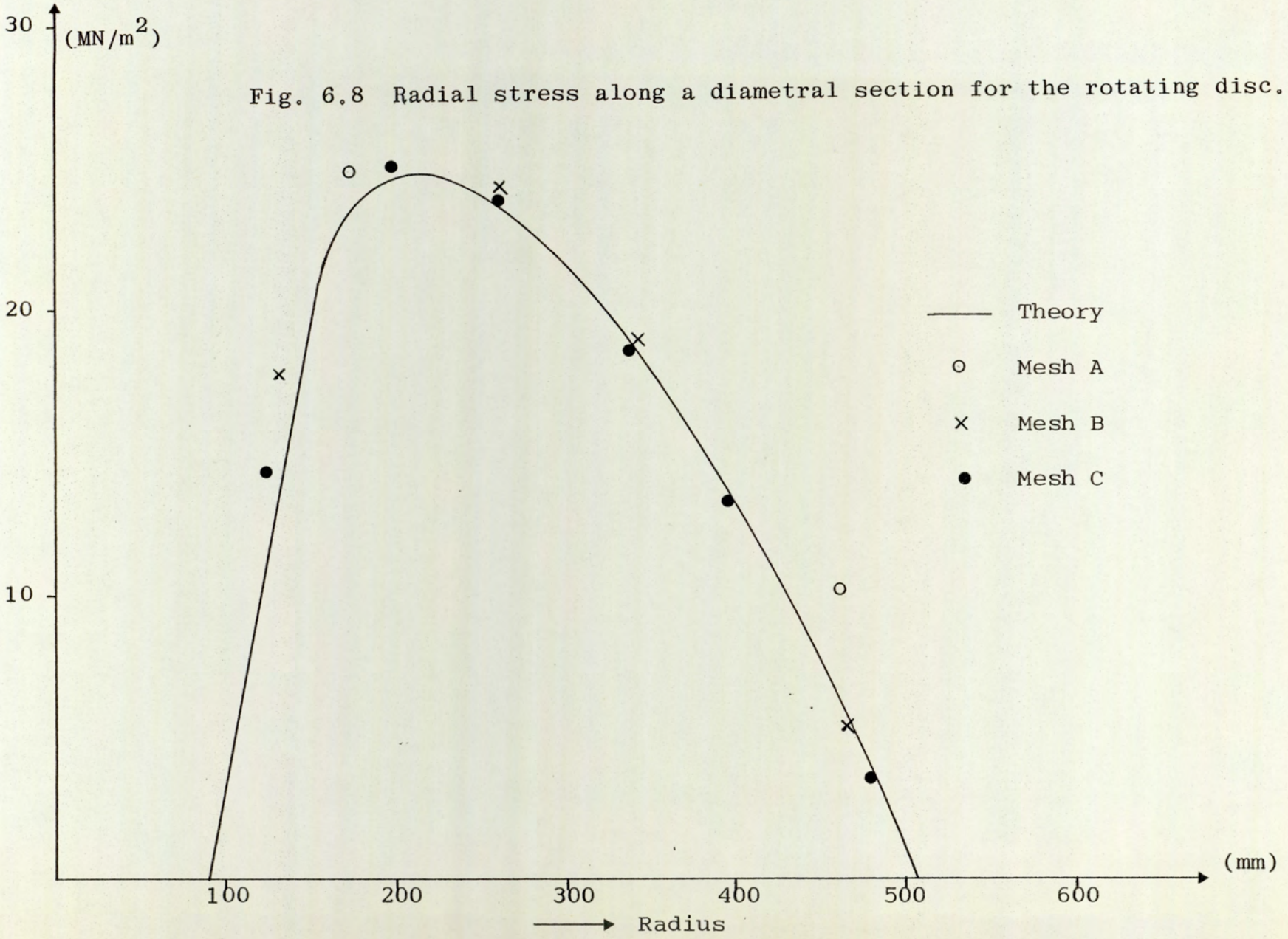


Fig.6.7 Geometry and meshes used for a rotating disc

Fig. 6.8 Radial stress along a diametral section for the rotating disc.



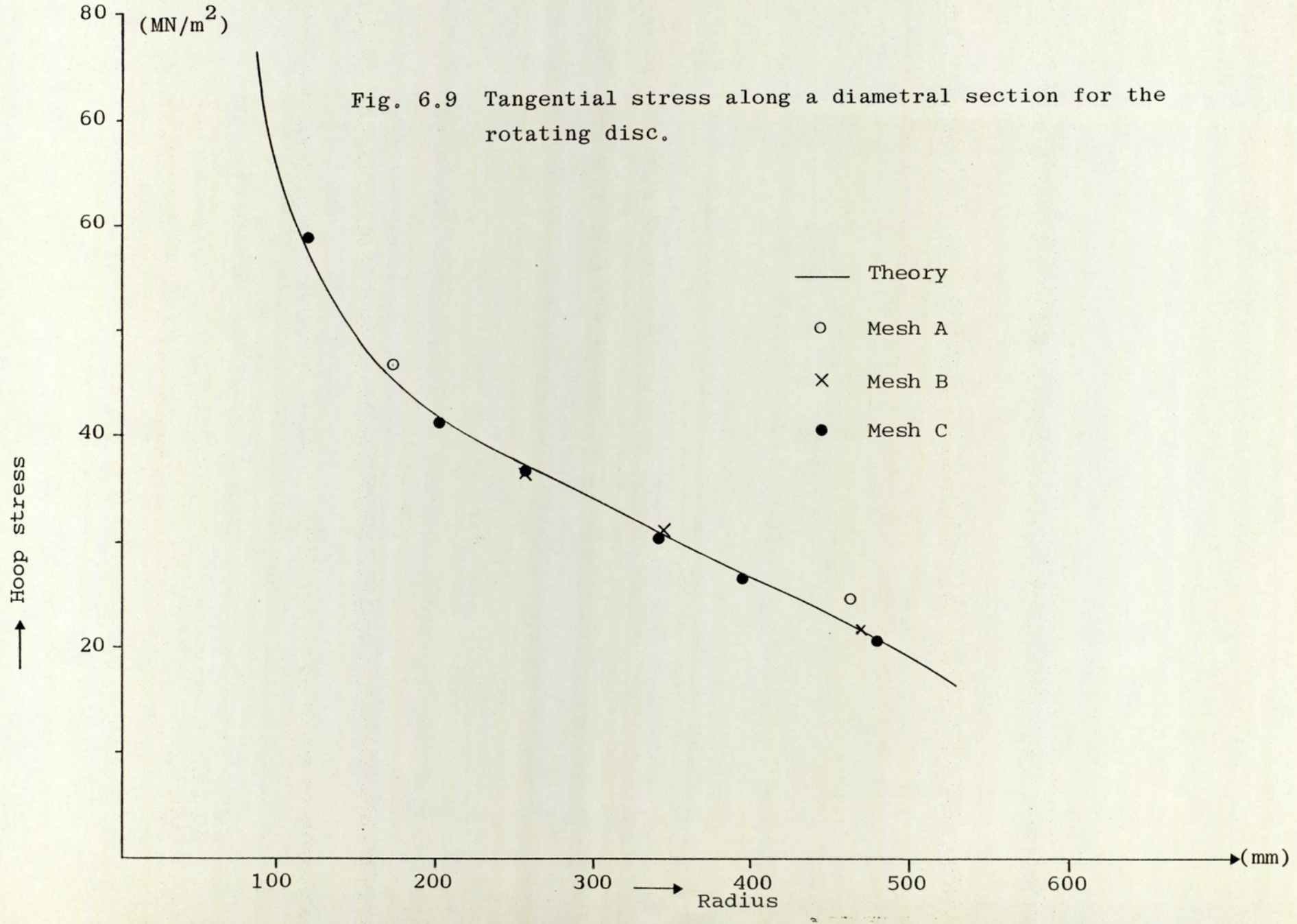
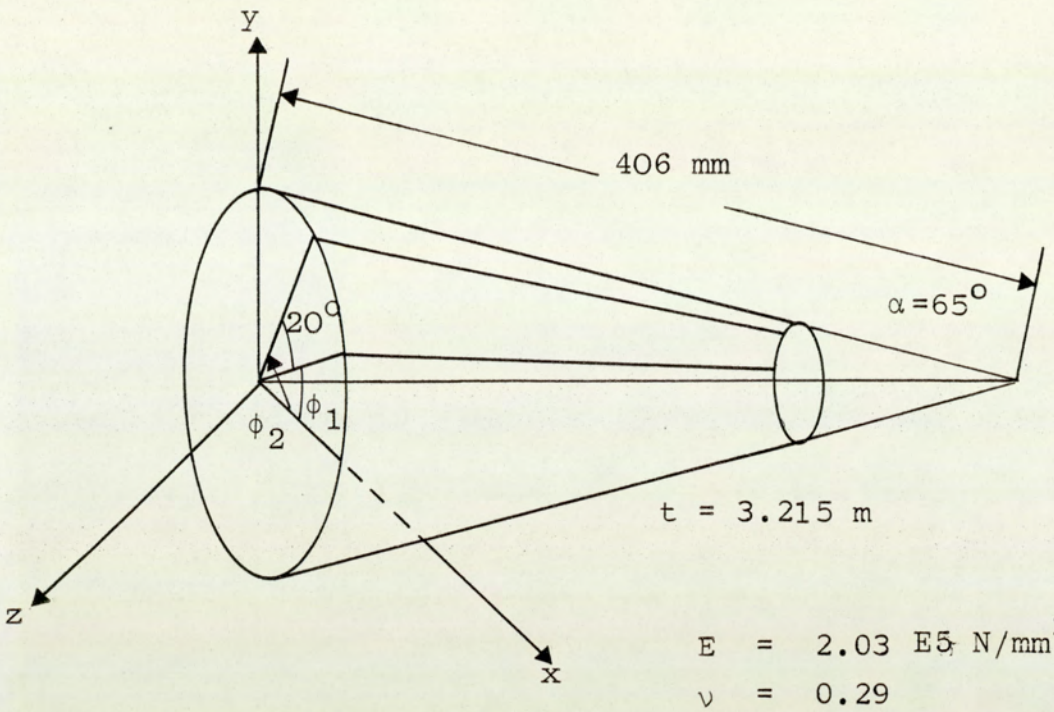


Fig. 6.9 Tangential stress along a diametral section for the rotating disc.

6.4.2 Rotating Cone

A rotating cone with the geometry shown in Fig. 6.10 has been solved analytically by Bell, Ref. (5) using the classical solution discussed in Ref. (77).

For comparison purposes, this problem is chosen to be solved and the Bell solution has been taken as a reference for the sector analysis discussed here. The inner edges are assumed to be built in. A sector of 20^{o} angle was used. The details of the geometry and the meshes used are shown in Fig. 6.10. The results are drawn on the exact solution published in Ref. (5) . The results of the stresses are taken through the longitudinal direction. The radial displacements and the membrane forces N_{θ} converge to the exact solution with few elements while the bending moment needed more refinement for the meshes used, as shown in Figs. 6.11, 6.12 and 6.13 respectively.

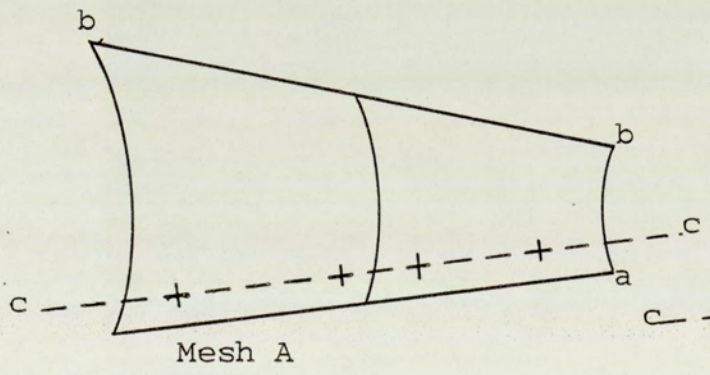


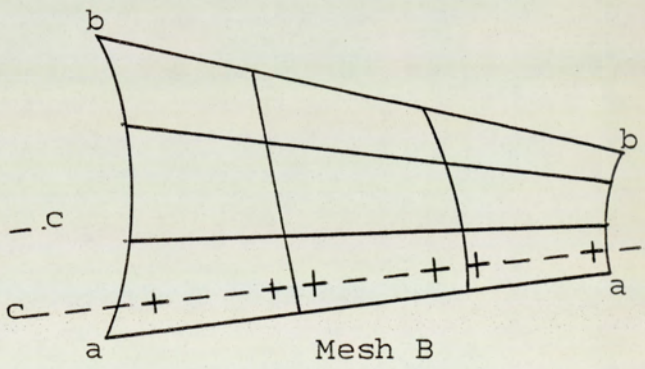
E = 2.03 E5 N/mm^2

ν = 0.29

ρ = 7.8 E-9 N/mm^3

ω = 3000 r.p.m.





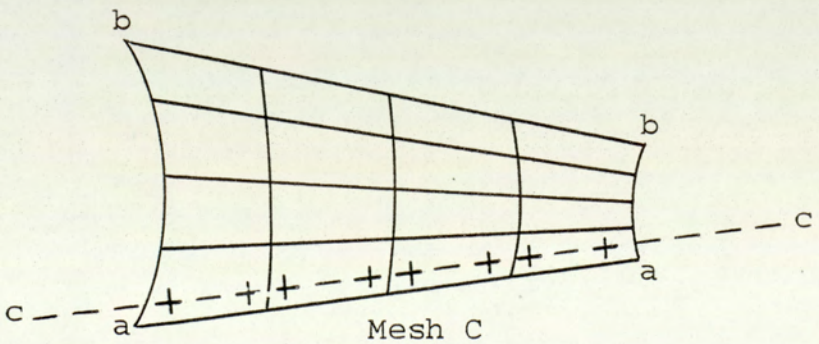


Fig.6.10 Geometry and meshes used in the analysis of the rotating cone

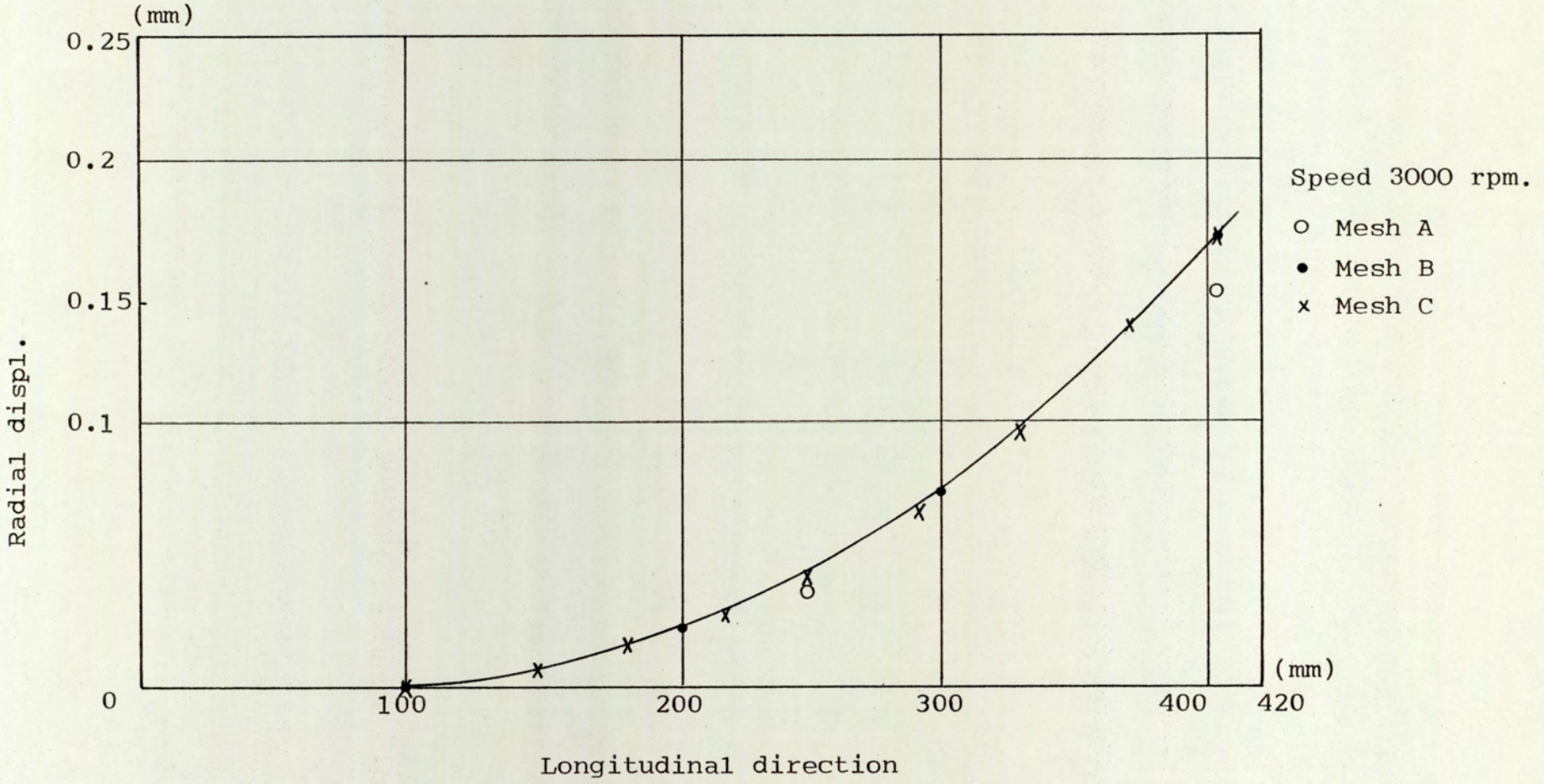


Fig.6.11 Radial displacement along the longitudinal direction for the rotating cone

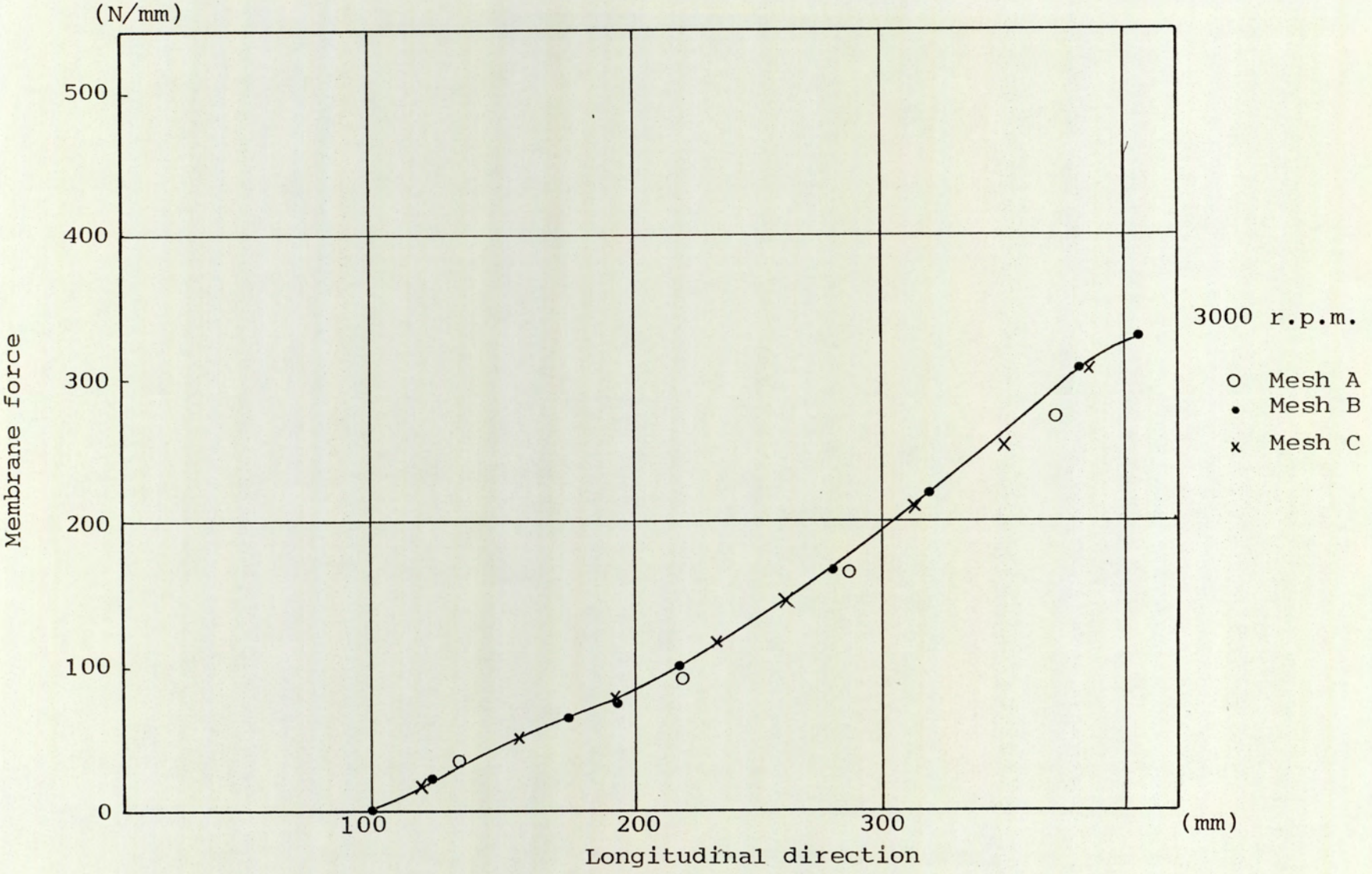


Fig.6.12 In-plane tangential force per unit length along the longitudinal direction for the rotating cone

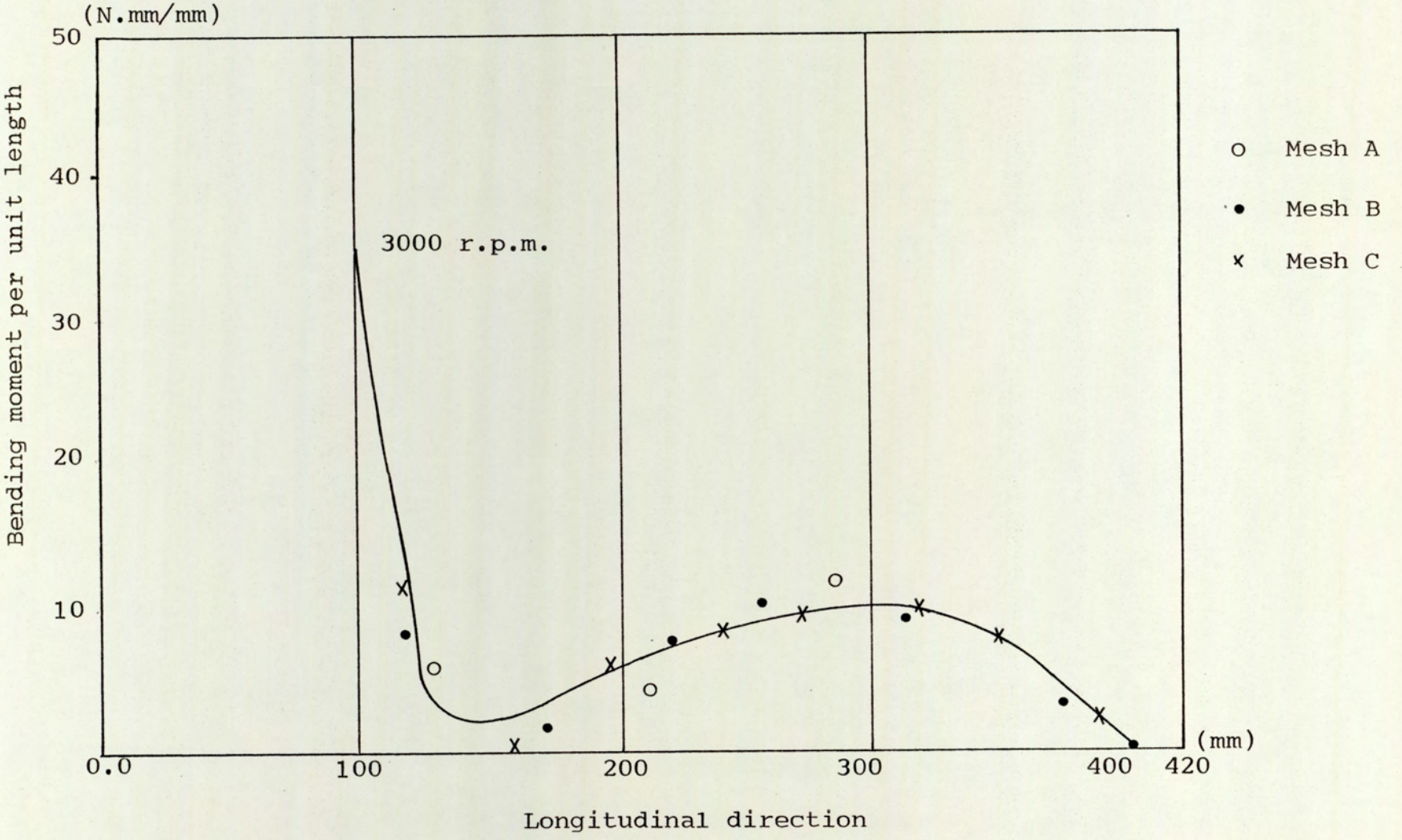


Fig.6.13 Bending moment along the longitudinal direction of the rotating cone

6.4.3 Simplified Fan Impellers

As the rotating fan impellers (simplified and actual) are examined experimentally in the next chapter, two examples are presented here in order to study the convergence of the solution for such intersecting structures and to show the validity of the implementation of cyclic symmetry in the developed computer programs. These are as follows:

(i) A rotating impeller with 10 equispaced radial blades (geometry and the meshes used are shown in Fig. 6.14(a), (b), (c) and (d)).

(ii) Impeller (i) plus a conesheet added to it (geometry and the meshes used are shown in Fig. 6.15(a), (b), (c) and (d)).

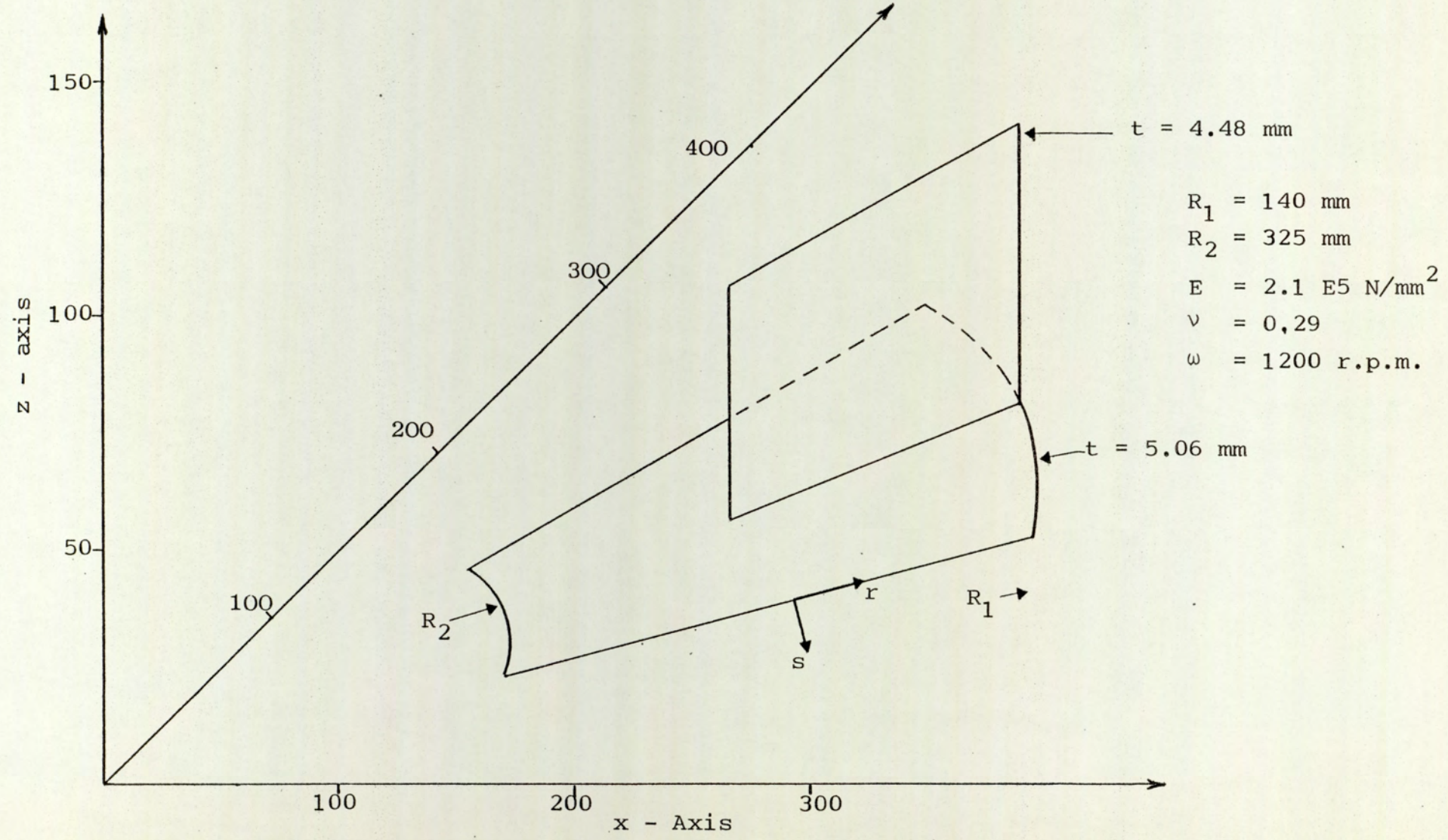
Using the repeatibility principles, the impeller can be analysed by dividing ($\frac{1}{10}$) of the structure into elements and imposing the equal displacements on the repeated boundaries (a-a) and (b-b). The sector has an included angle of 36^{o}.

Mesh C was drawn for both impellers (i) and (ii) for a reasonable numerical solution in order to study the convergence for the displacements and stresses. The results from mesh A shows that a finer mesh was required compared to the results from mesh C, especially for impeller (ii), therefore only the radial displacements

are drawn on the solution of mesh C.

Two sections on the backsheet have been investigated, these are at (a-a) and (d-d). Results are shown in Figs. 6.16 to 6.18 and Figs. 6.19 to 6.21 for both impellers (i) and (ii) respectively. Rapid convergence for the displacements, in-plane tensile forces and the radial bending moment are obtained. The results for the tangential bending moment shows that a finer mesh other than mesh B is required. The effects of adding the conesheet to impeller (i) are drawn in Figs. 6.22 and 6.23 on the backsheet at sections (c-c) and (d-d).

Fig. 6.14(a) Sector of the radial fan impeller (without conesheet)



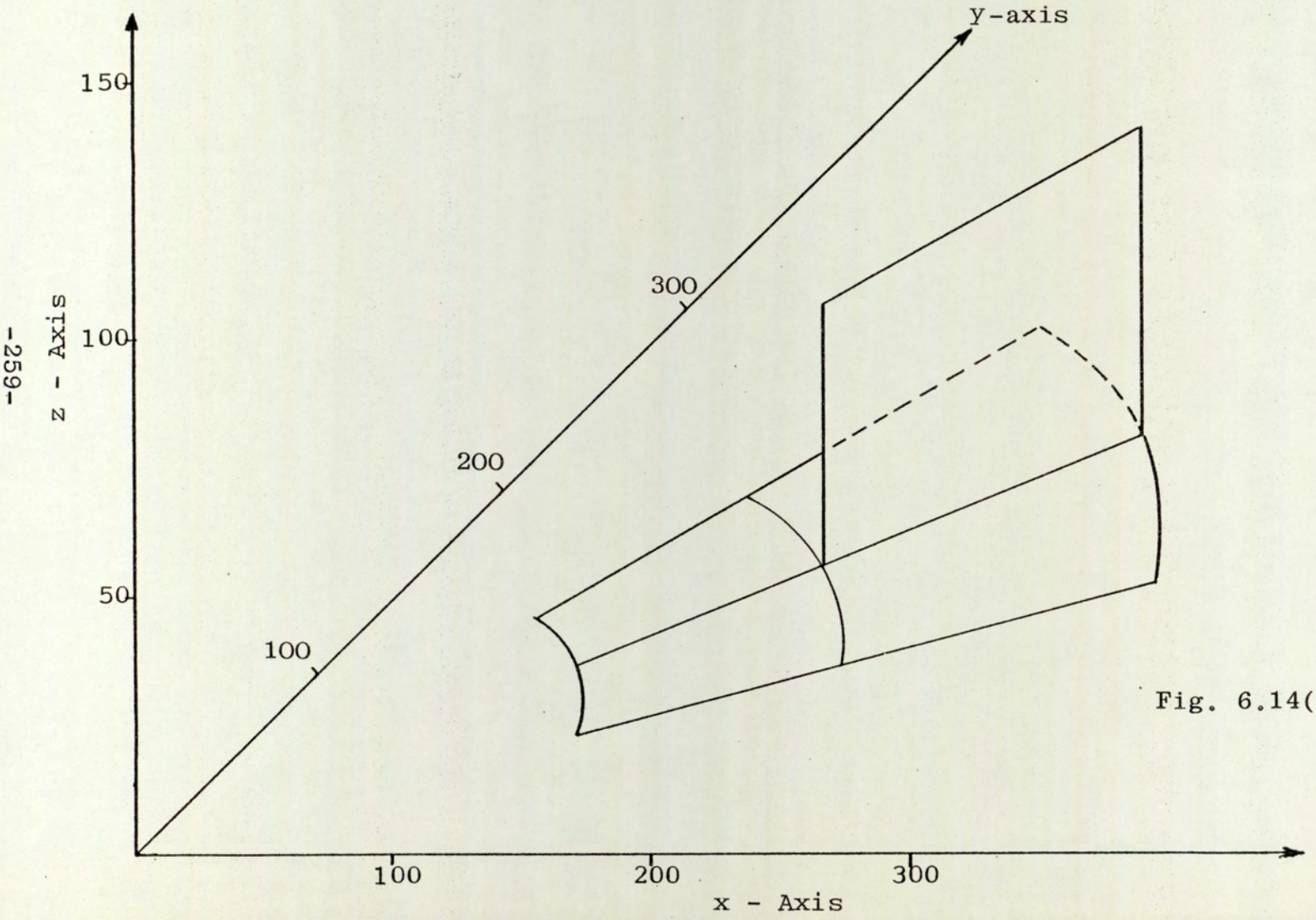


Fig. 6.14(b) Mesh A

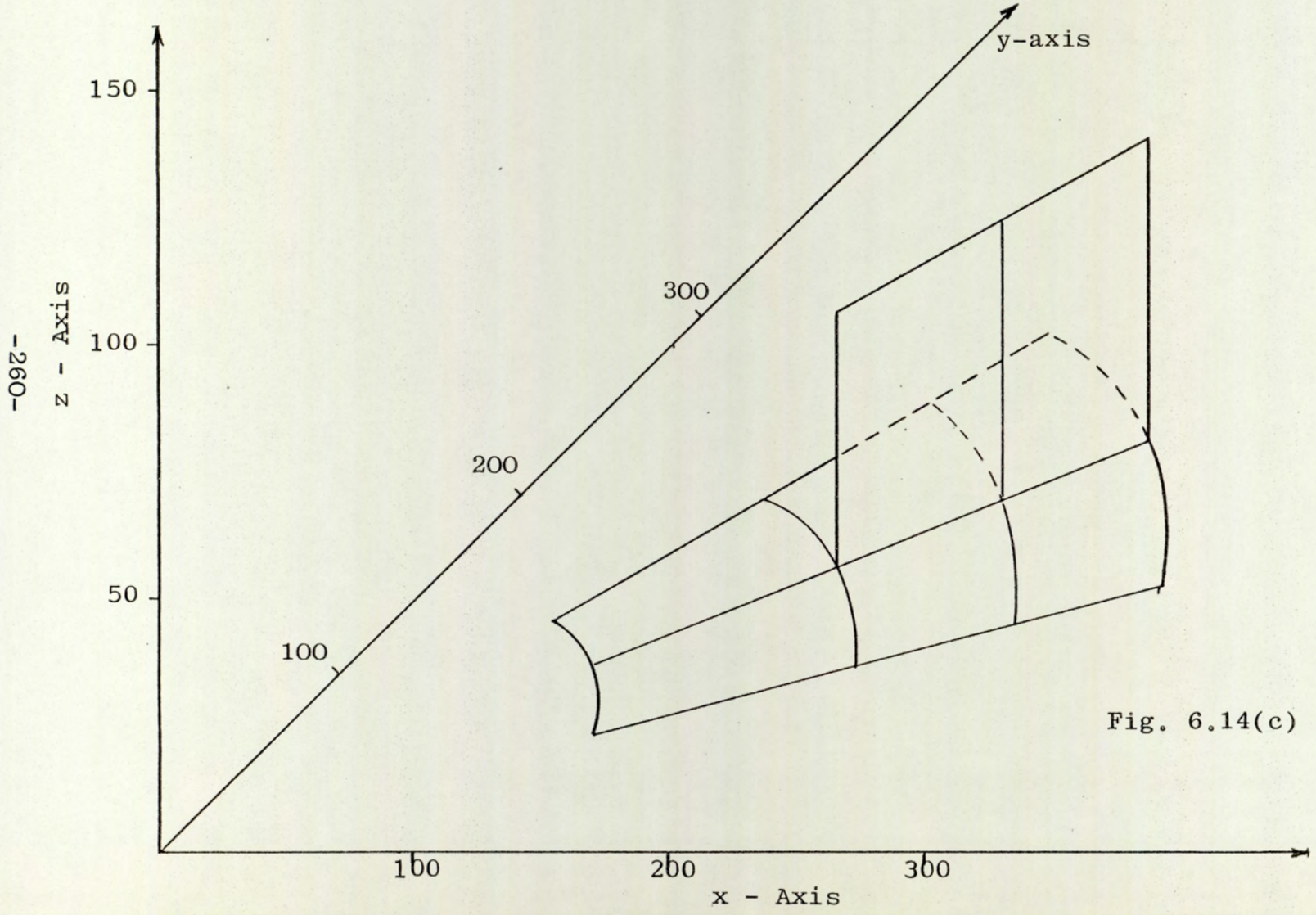


Fig. 6.14(c) Mesh B

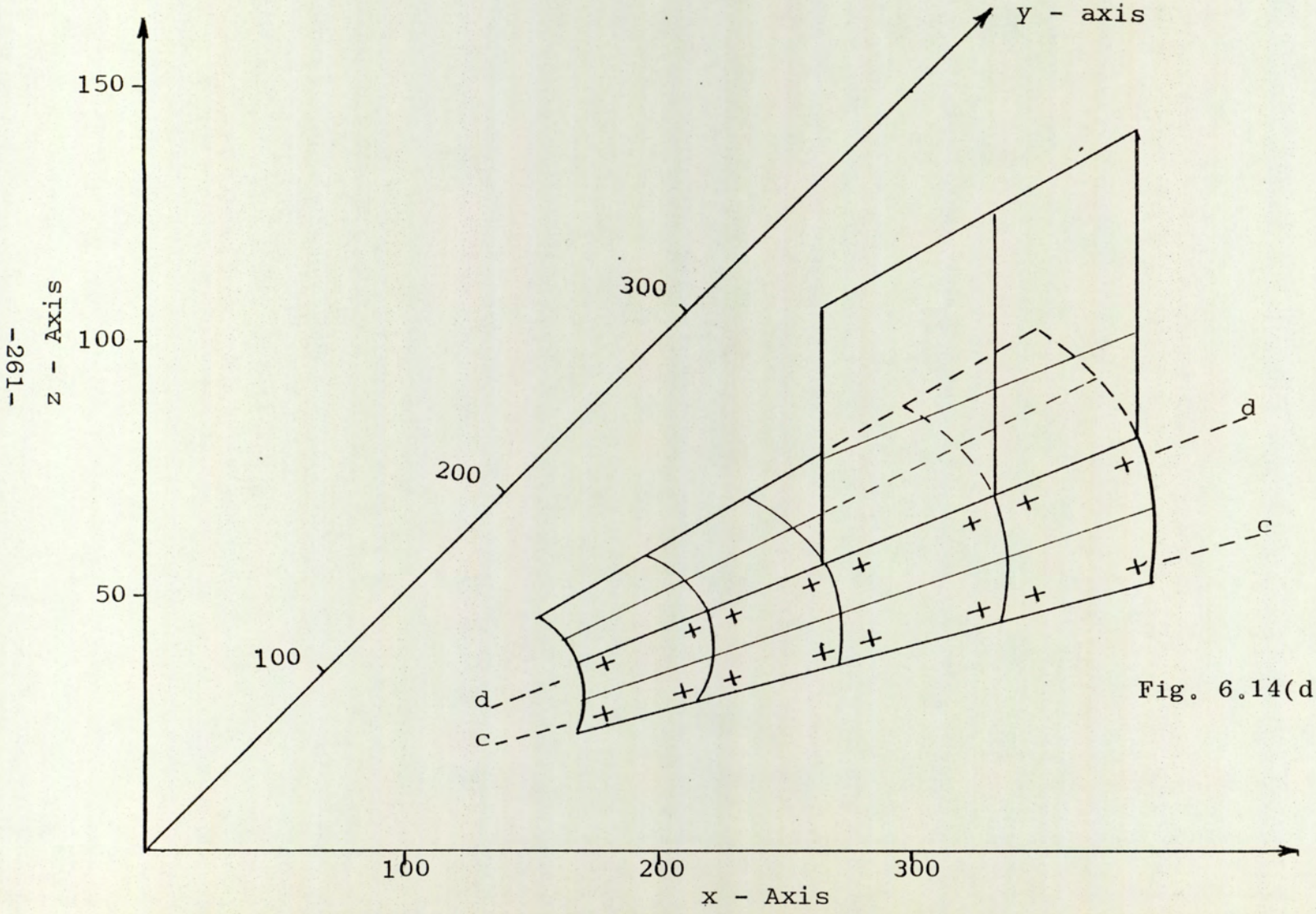


Fig. 6.14(d) Mesh C

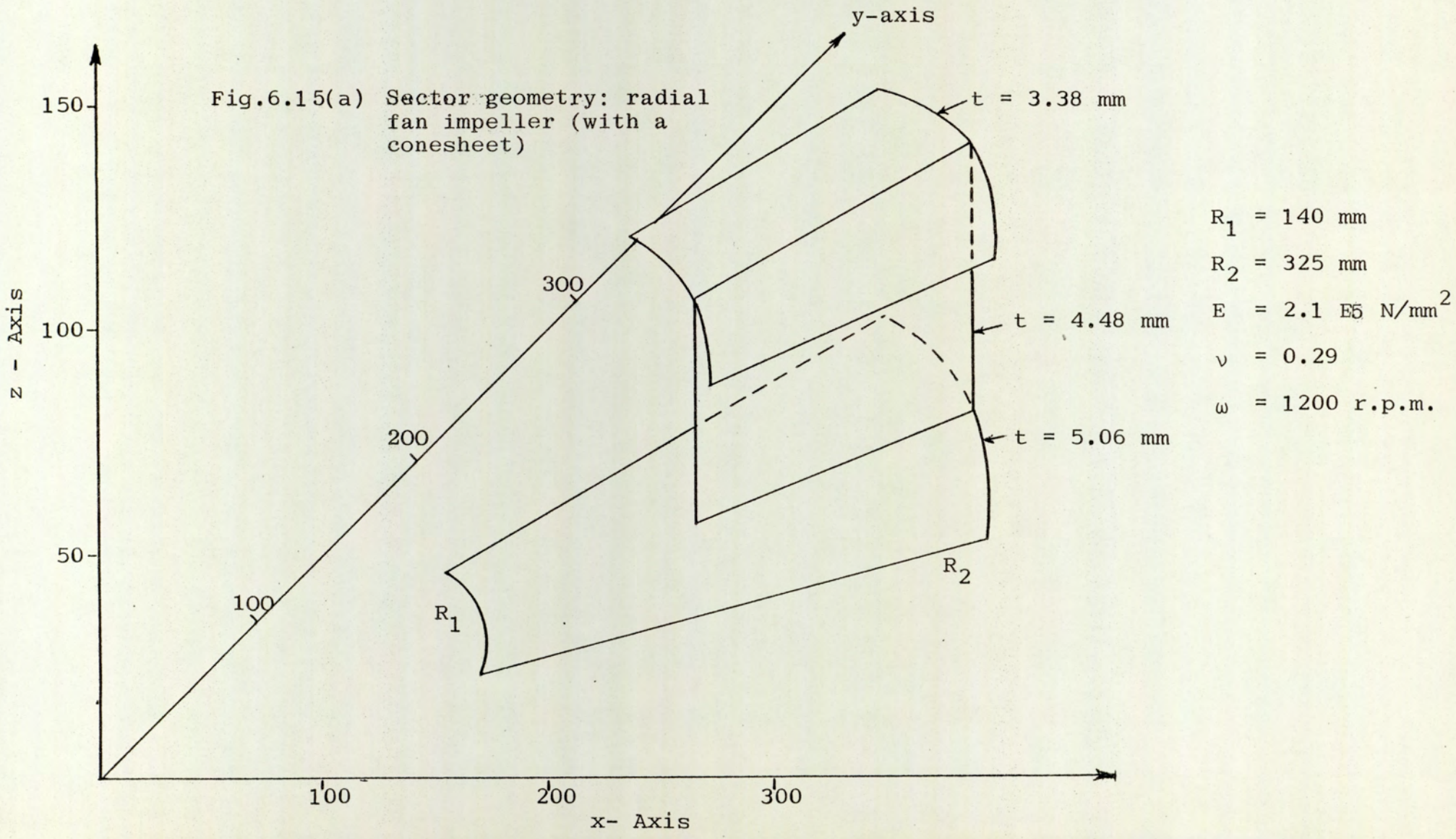


Fig.6.15(a) Sector geometry: radial fan impeller (with a conesheet)

R_1 = 140 mm

R_2 = 325 mm

E = 2.1 E5 N/mm^2

ν = 0.29

ω = 1200 r.p.m.

Fig. 6.15(b) Mesh A

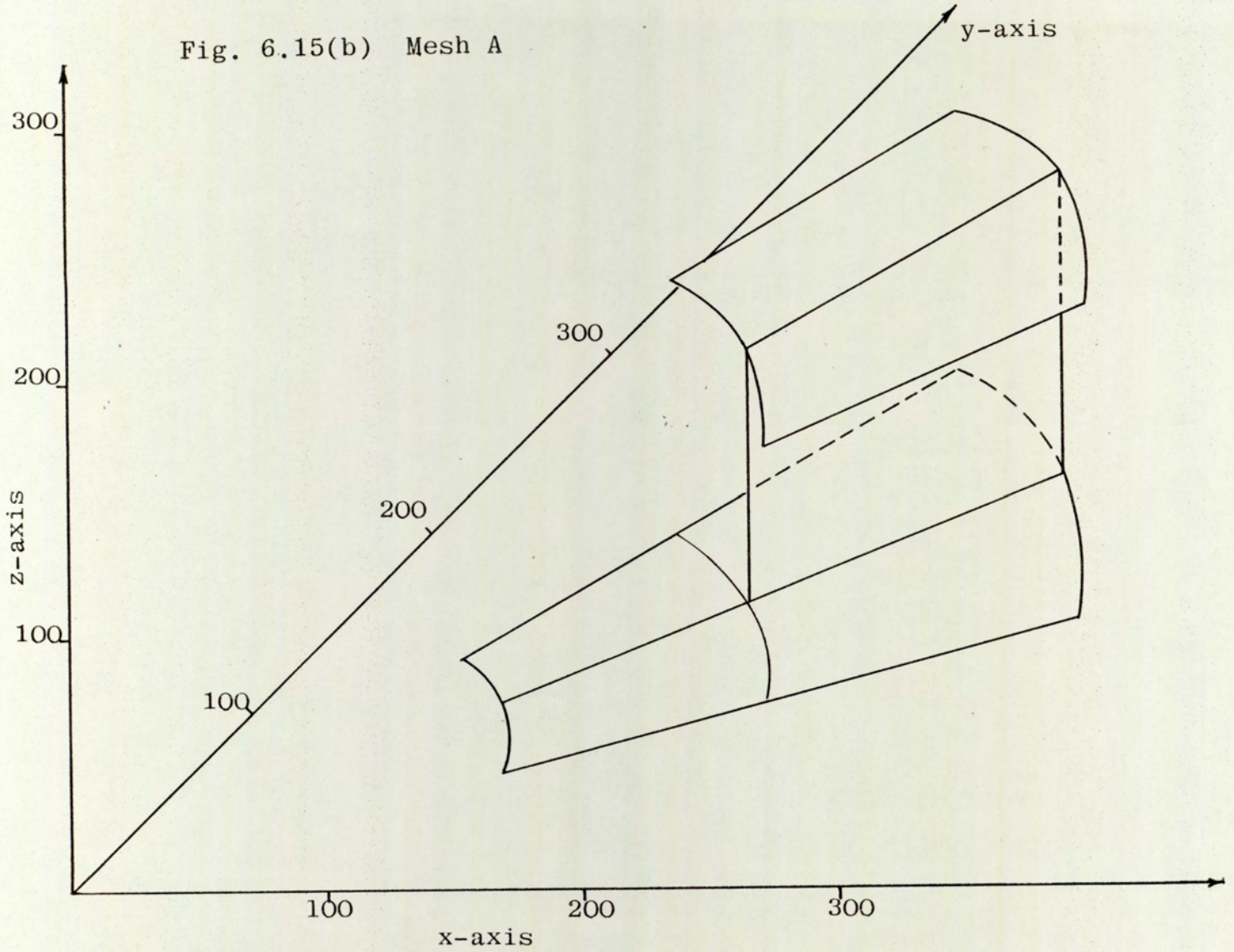


Fig. 6.15(c) Mesh B

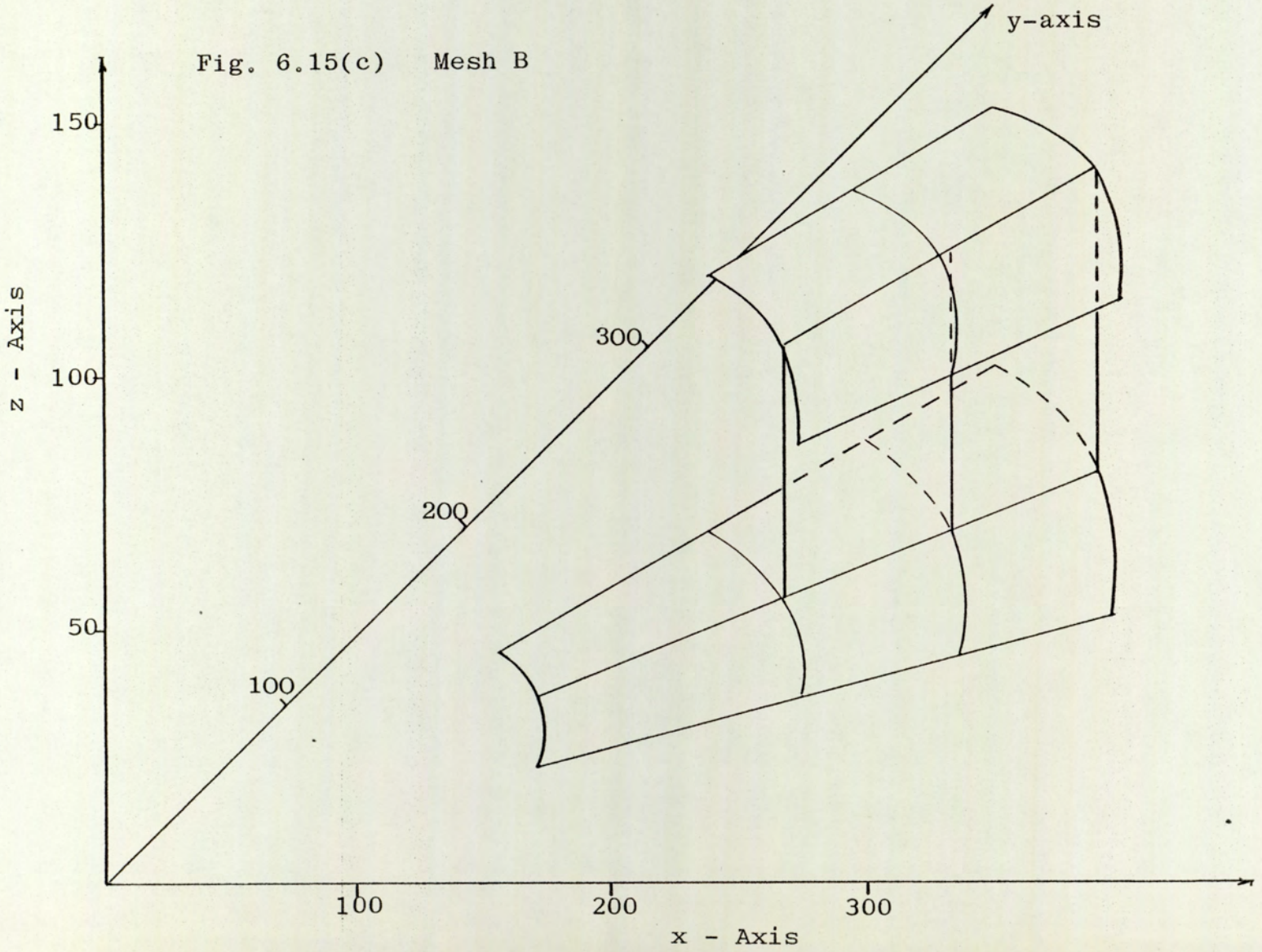
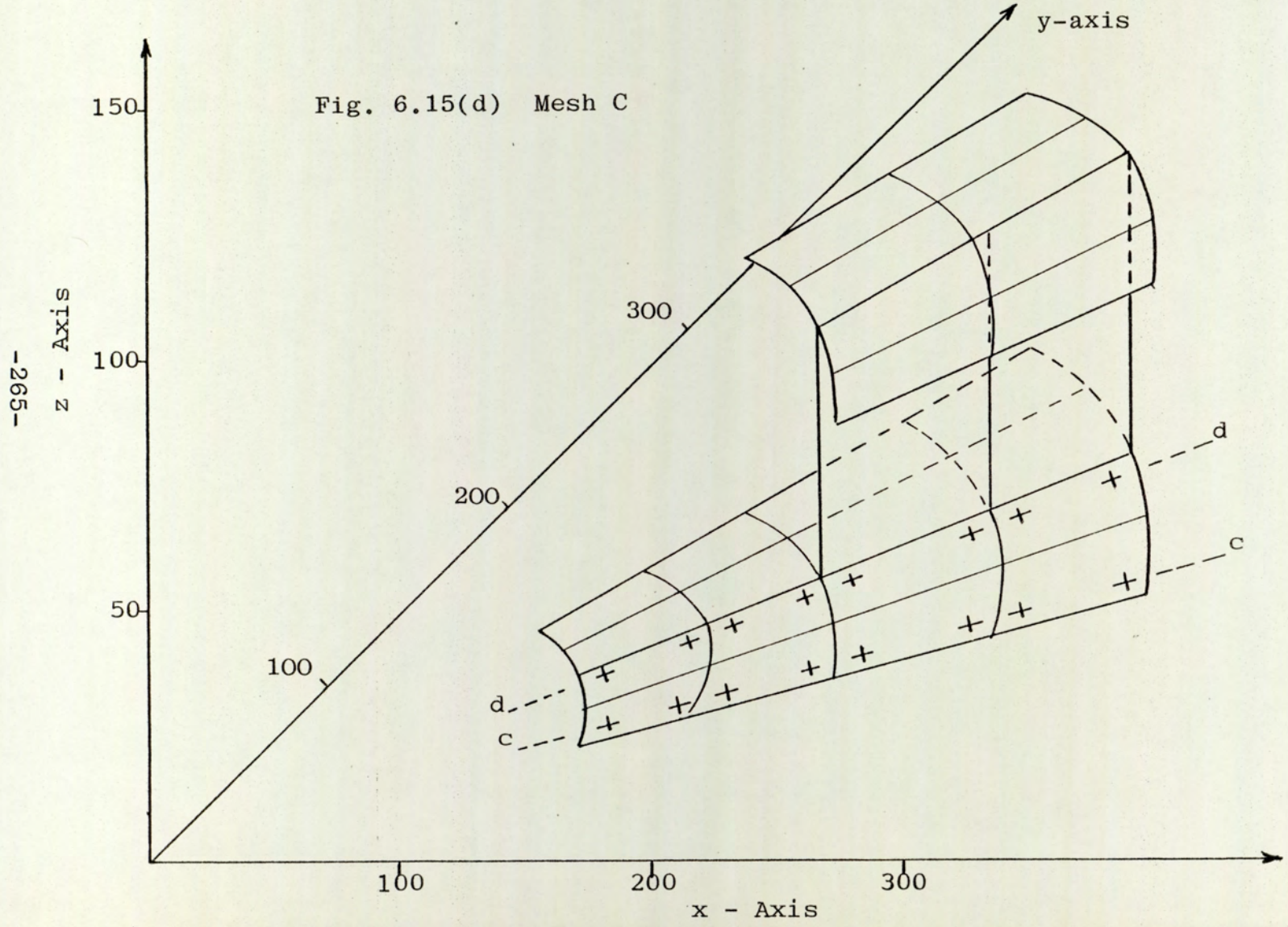


Fig. 6.15(d) Mesh C



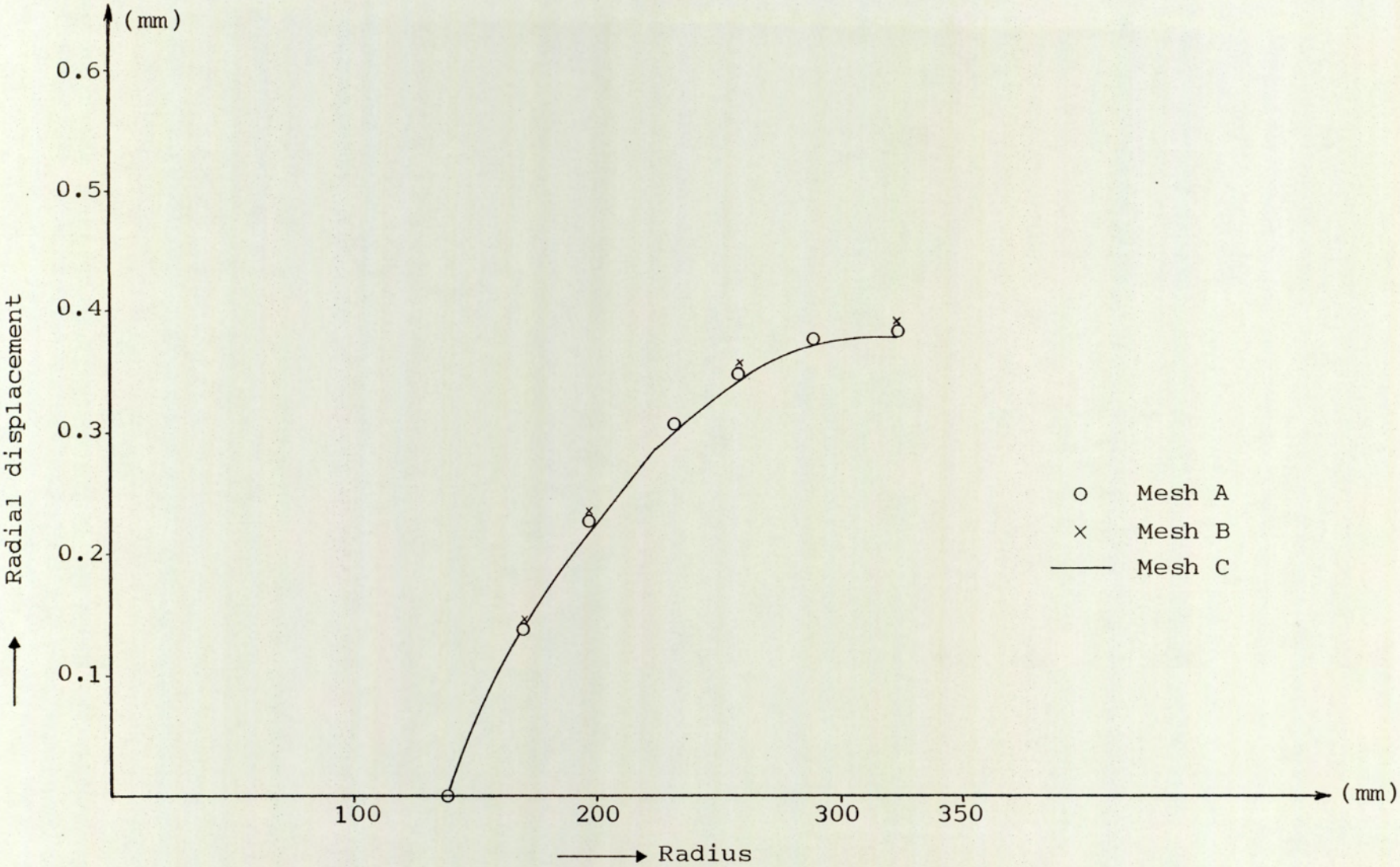


Fig.6.16 Radial displacement along the section $\theta = 27^{o}$ for impeller (i)

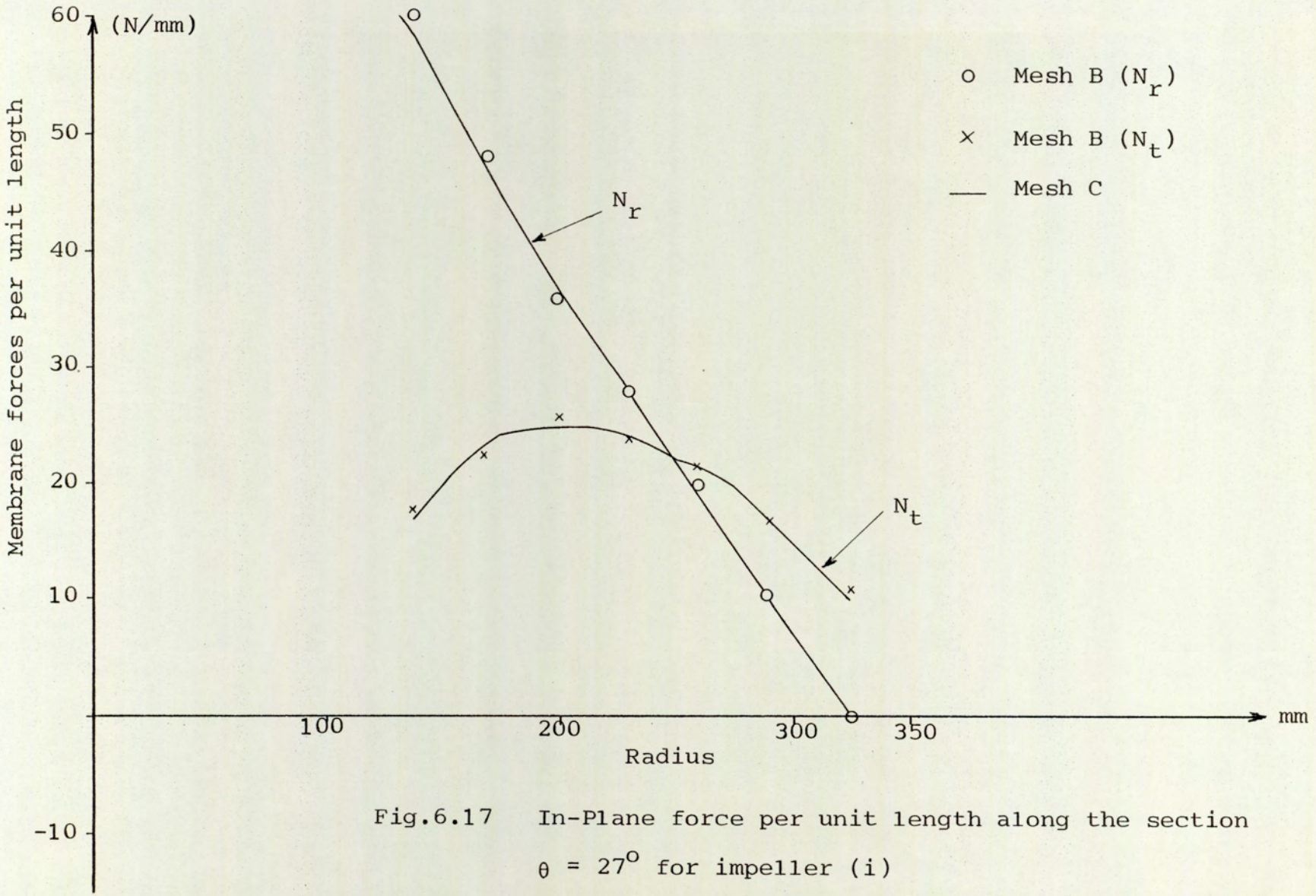


Fig.6.17 In-Plane force per unit length along the section $\theta = 27^{o}$ for impeller (i)

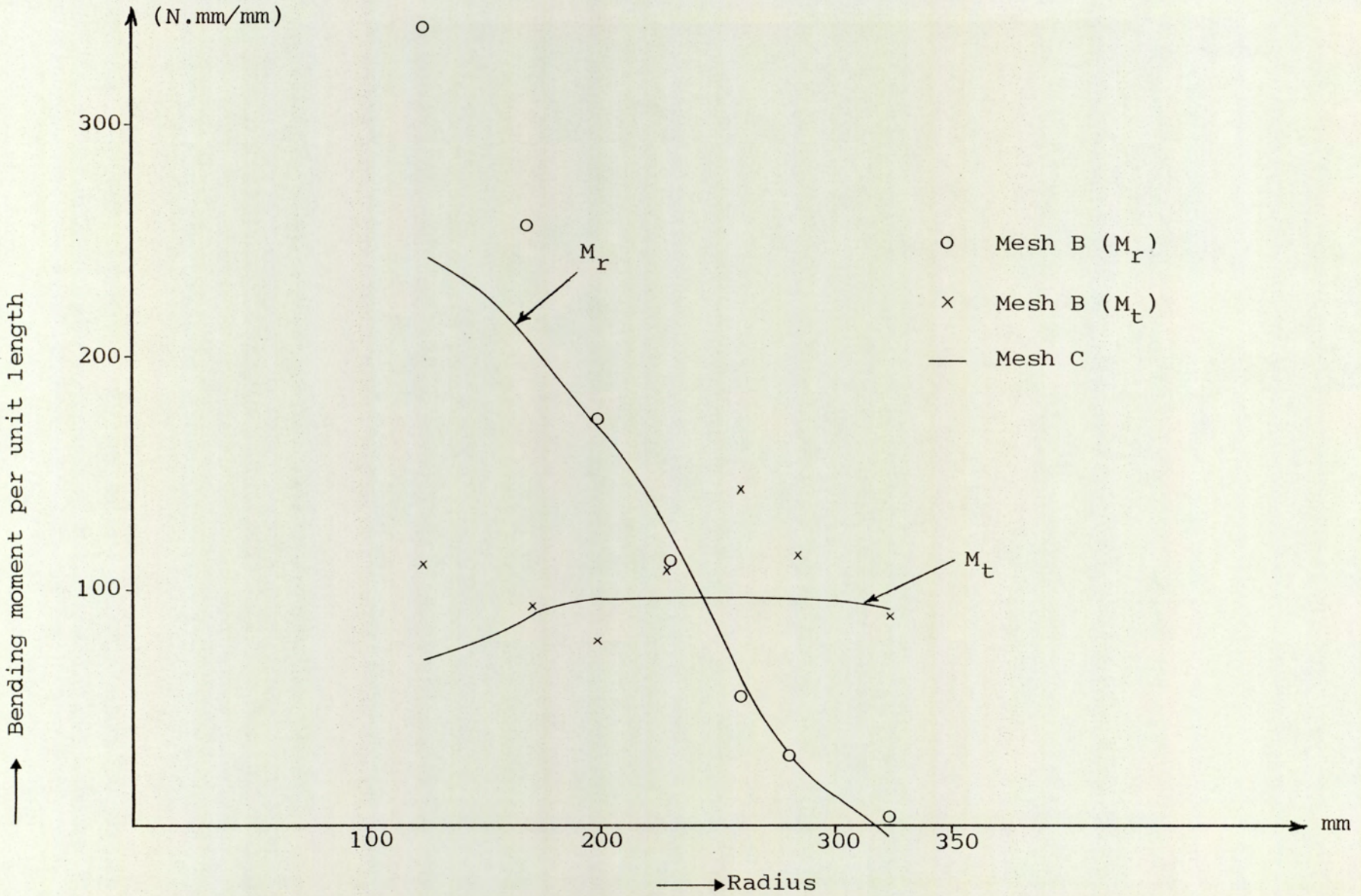


Fig.6.18 Bending moment along the section $\theta = 27^{o}$ for impeller (i)

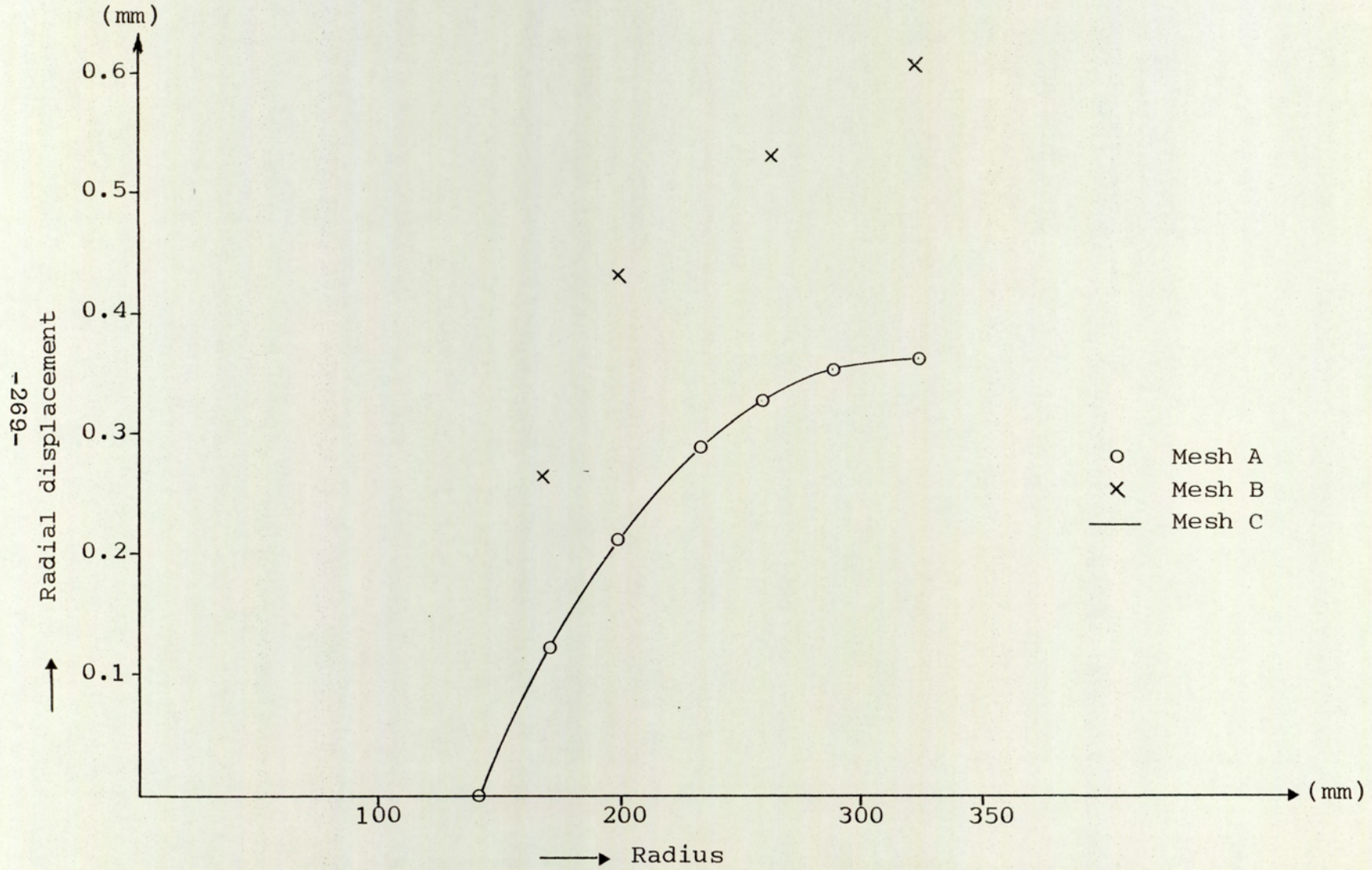


Fig.6.19 Radial displacement along the section $\theta = 27^{o}$ for impeller(ii)

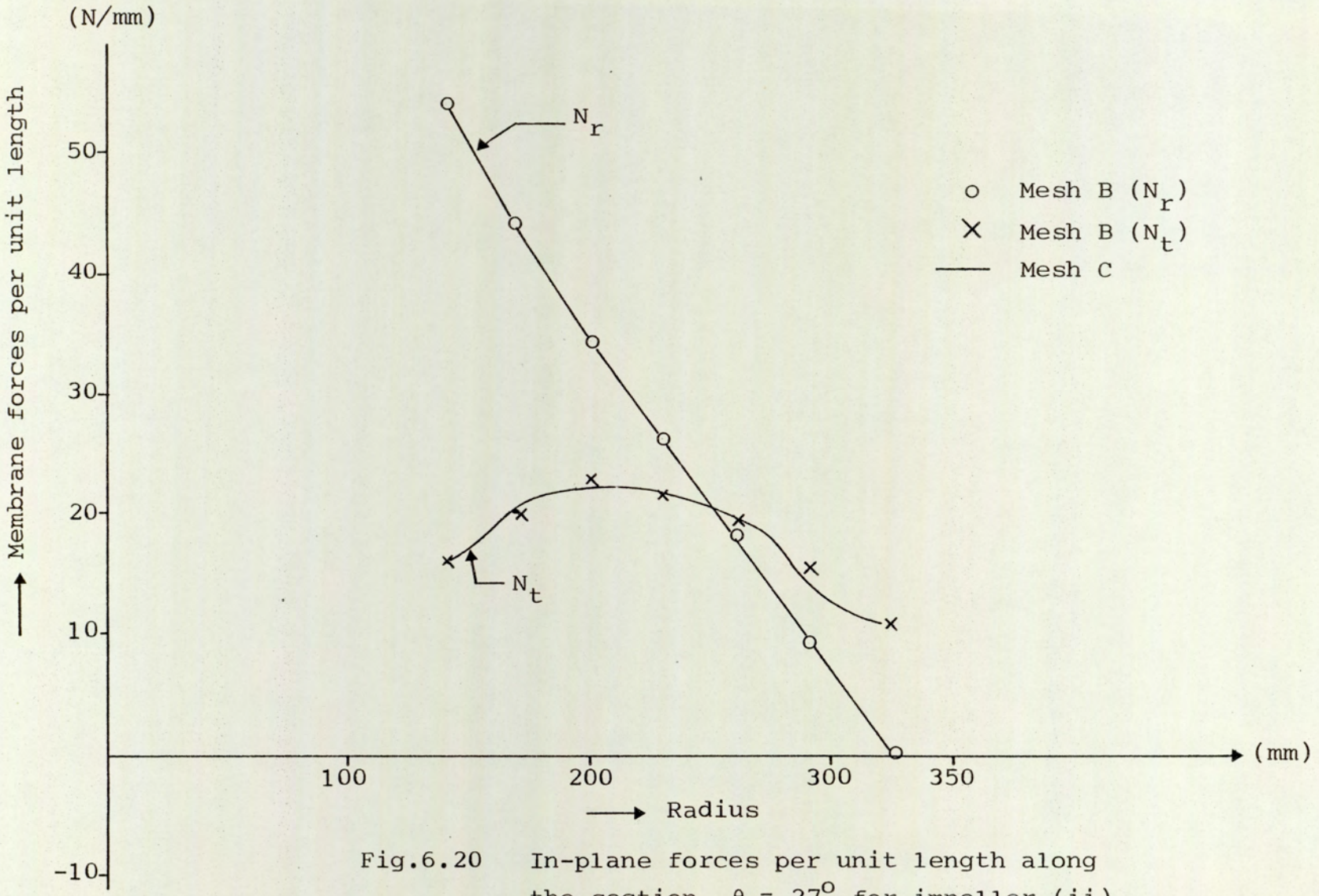


Fig.6.20 In-plane forces per unit length along the section $\theta = 27^{o}$ for impeller (ii)

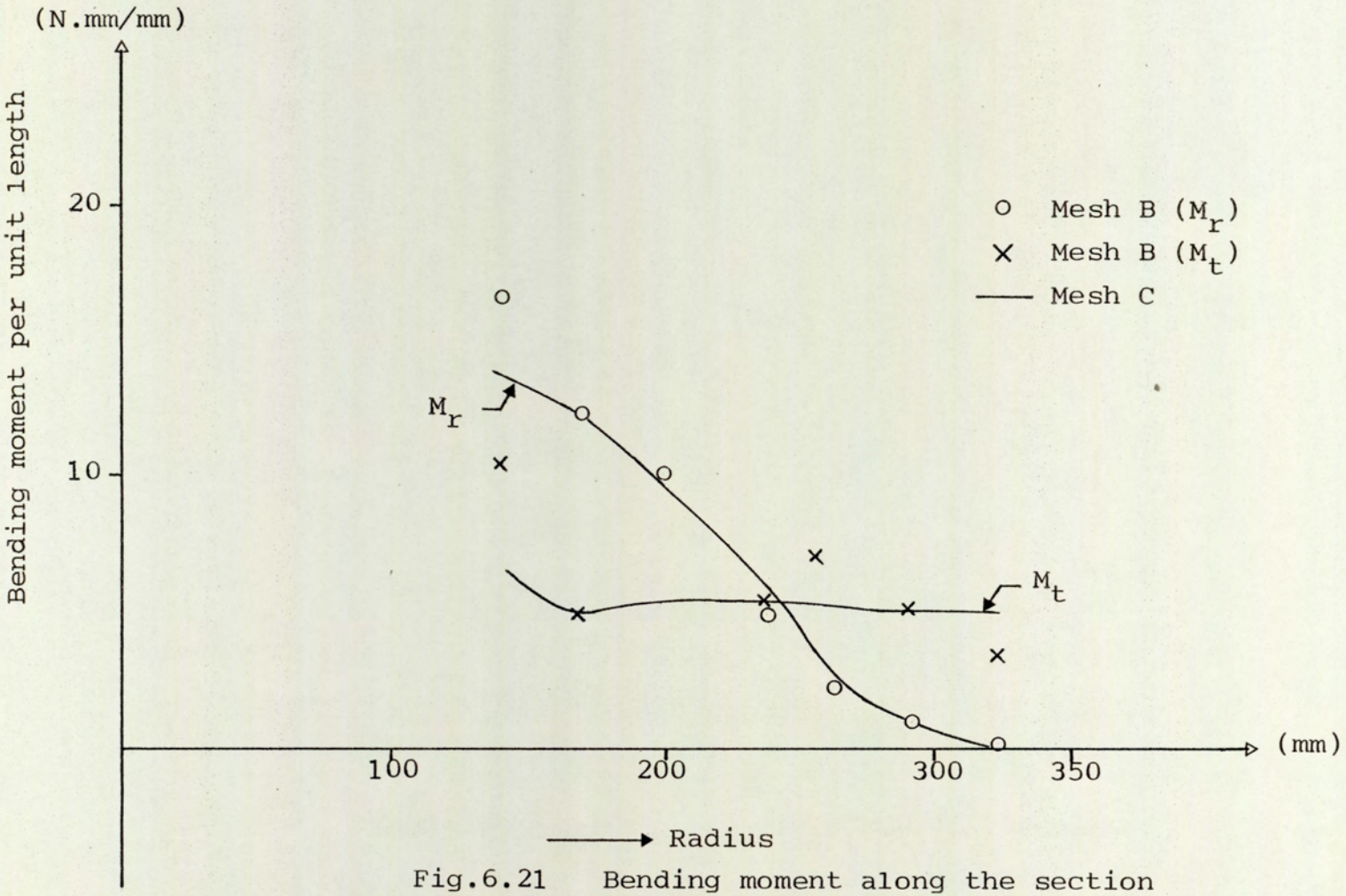


Fig.6.21 Bending moment along the section $\theta = 27^{o}$ for impeller (ii)

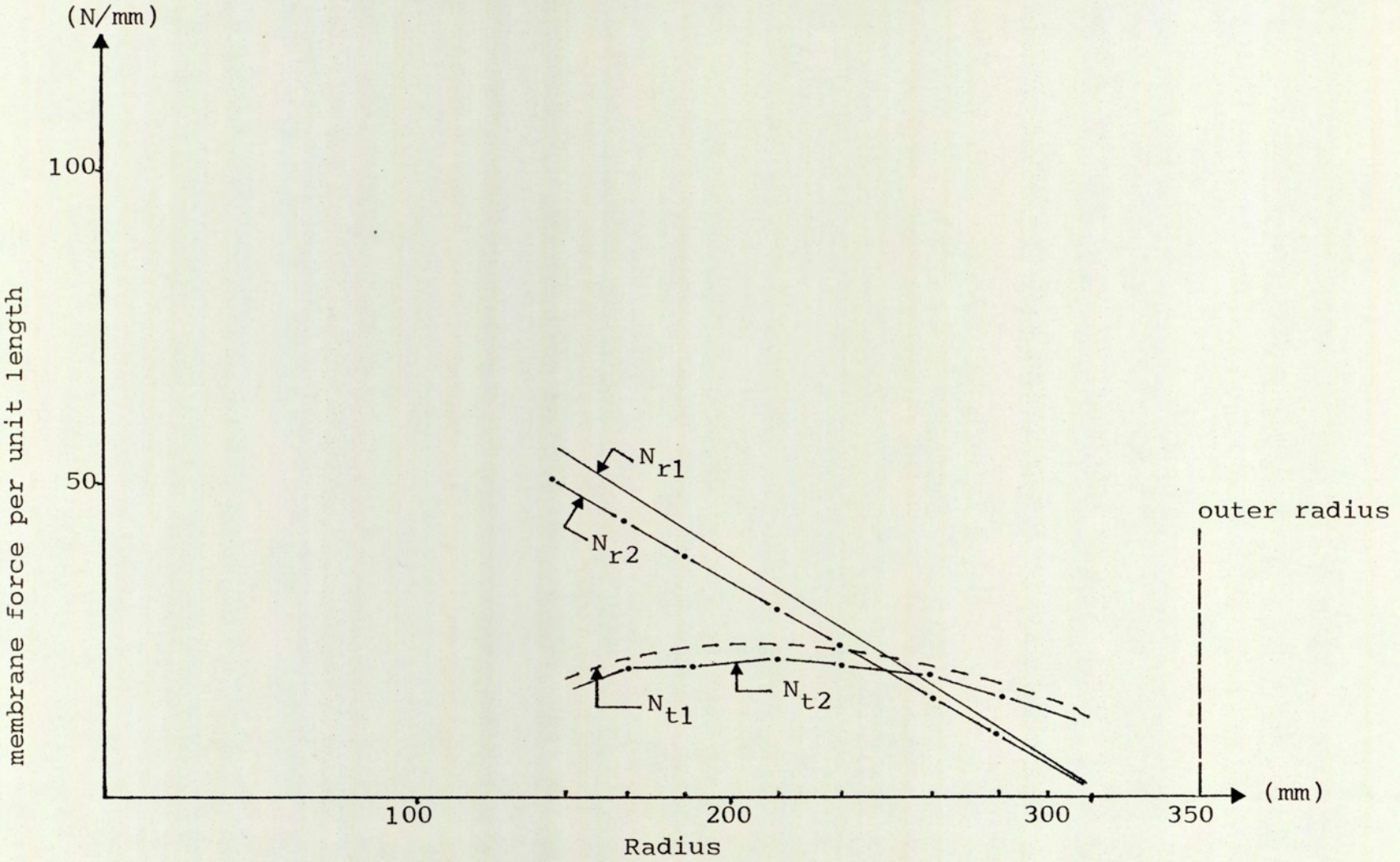


Fig.6.22 In-plane tensile forces per unit length at the backsheet for the impellers (i) and (ii) along the section $\theta = 27^{o}$ (Mesh C)

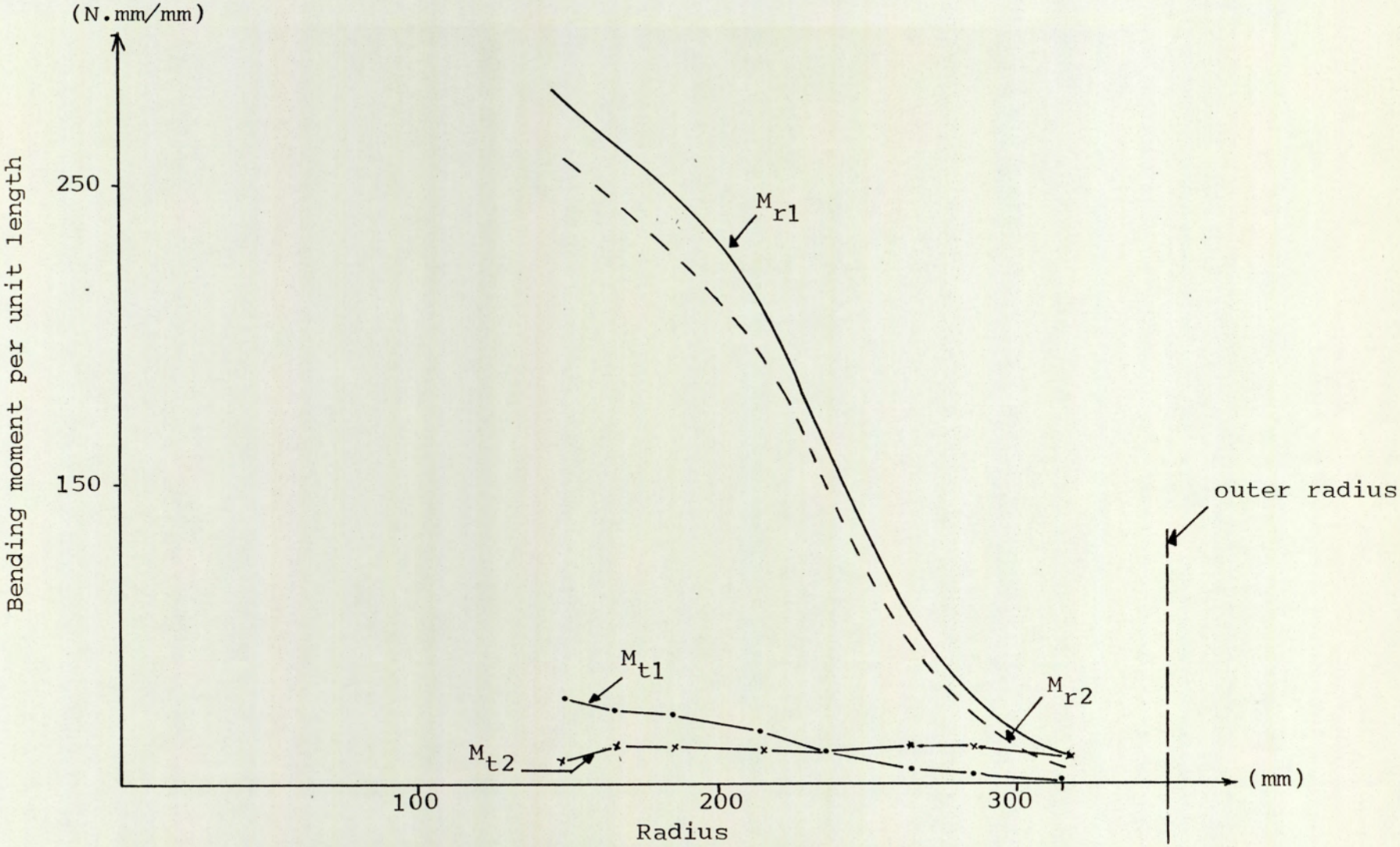


Fig.6.23 Bending moment at the backsheet for the impellers (i) and (ii) along the section $\theta = 27^{o}$ (Mesh C)

6.5 CLOSING REMARKS

The technique of analysing one sector of a structure in which the loading pattern, geometry and material properties are repeated is incorporated in the developed computer program discussed in Chapter 4. The constraints of equal node displacement should be imposed by specifying that nodes at equivalent positions on the sector boundaries. The comparison of the finite element results with the exact solution for the given examples has shown excellent agreement using meshes containing few elements. This also proves the correct representation of element centrifugal force as equivalent nodal loads using the element shape function which has been coded into the program.

The convergence study of the simplified radial fan impeller with and without conesheet, is promising in as far as the specification of the cyclic symmetry technique is concerned.

An actual fan impeller will be solved by the finite element method using the technique discussed in this chapter. The results will be compared with those obtained experimentally and described in the next chapter.

CHAPTER SEVEN

EXPERIMENTAL INVESTIGATION AND CORRELATION WITH NUMERICAL PREDICTIONS

7.1 INTRODUCTION

Generally, if a given problem is solved by a theoretical approach, the results must be verified by experimental means. In the stress analysis field, three basic conventional experimental methods are available. These are:

(i) Photoelastic method.

(ii) Brittle lacquer coating method.

(iii) Strain gauges and associated instrumentation.

The photoelastic method would not be suitable for the present task in view of the difficulties of obtaining a reasonable size model of fan impellers fabricated from thin plate.

The brittle lacquer coating methods are not suitable to achieve the objectives of this part of the work, because only qualitative results are obtained. The magnitude and direction of the actual stresses are required in order to be able to obtain a comparison between the experimental and the numerical results to assess the ability of the finite element model adequately to represent the rotating fan impeller.

The strain gauge technique has been found to be a versatile tool in this field, due to its relative

simplicity, and the fact that it is applicable directly to component under service loading conditions: it gives more accurate results than other experimental techniques.

This chapter is mainly devoted to discussing the experimental stress analysis of the thin walled centrifugal fan impellers (simplified and actual). In addition to these examples, two complicated plate structures, which have been examined experimentally at the early stages of developing the finite element package, are presented to provide a comparative solution for the problems of intersecting plates which characterise fabricated fan impellers with their blade/backsheet and blade/conesheet junctions. A comparison between the finite element process using the sectorial symmetry technique is discussed in Chapter 6, and the experimental results are given in order to assess the adequacy of this approach to treating thin walled centrifugal fan impellers.

7.2 BACKGROUND THEORY OF STRAIN GAUGES

Having decided that the strain gauge technique should be used, more background information on electrical strain gauges and information was sought and a brief summary is presented here.

7.2.1 Electrical Resistance Strain Gauges

Strain gauges technique has been applied to stress analysis problem in almost every branch of engineering. The electrical resistance strain gauge measurement, is the most popular method, in which a change in length (strain) produces a change in the electrical characteristic of resistance. The important advantage common to all electrical gauges is the relative ease with which the output can be displayed and recorded. There is a wide variety of electrical resistance strain gauges. The bonded foil resistance gauge is the most widely used, because of its small size, lightness, high sensitivity and ease of attachment and their great stability during long time and elevated temperature tests. When these gauges are strained, their resistance changes. This change of resistance is related to the change of strain by the following equation:

$$\varepsilon = \frac{(\frac{\Delta R}{R})}{F} \qquad \ldots \quad (7.1)$$

where

ε = strain.

ΔR = change in gauge resistance

R = gauge resistance

F = gauge factor or strain sensitivity.

The gauge factor is usually about 2 with the actual value for a given batch being supplied by the manufacturers. The foil strain gauge is mounted onto a specimen with an adhesive. This adhesive serves a very vital function in the strain measuring system in that it must transmit the strain from the specimen to the sensing element without distortion. The wires leading to each gauge should be firmly fixed to prevent them from flexing. The types of gauges used in the experimental investigation are either linear gauges or rectangular 45^{o} rosettes (foil type).

The linear gauges of Fig. 7.1(a) are designed to record the strain along one paritcular axis.

In general, normal and shearing stresses act in arbitrarily chosen directions at a given point so that, in such cases, the directions of the principal stresses

stresses are not always known: A single linear strain gauge cannot be used for analysis and three (or more) gauges at known orientations with respect to each other, together forming a "rosette" must be used. In the strain-rosette method the stress at a point is completely determined by measuring the unit strains at the point in three independent directions. The state of stress at a point will be obtained in terms of the principal stresses and their directions. The strain-gauge rosette equations can be obtained using the Mohr's circle for strain (see Ref. (78)). The principal strains are first determined, and then the principal stresses are evaluated using the stress-strain relations. For the rectangular 45^{o} rosette of Fig. 7.1(b), the equations for the principal strains and stresses are as follows:

The principal strains are:

$$\varepsilon_{max} = \frac{\varepsilon_A+\varepsilon_C}{2} + \frac{1}{2}\sqrt{(\varepsilon_A-\varepsilon_C)^2+(2\varepsilon_B-\varepsilon_A-\varepsilon_C)^2} \quad \ldots \quad (7.2)$$

$$\varepsilon_{min} = \frac{\varepsilon_A+\varepsilon_C}{2} - \frac{1}{2}\sqrt{(\varepsilon_A-\varepsilon_C)^2+(2\varepsilon_B-\varepsilon_A-\varepsilon_C)^2} \quad \ldots \quad (7.3)$$

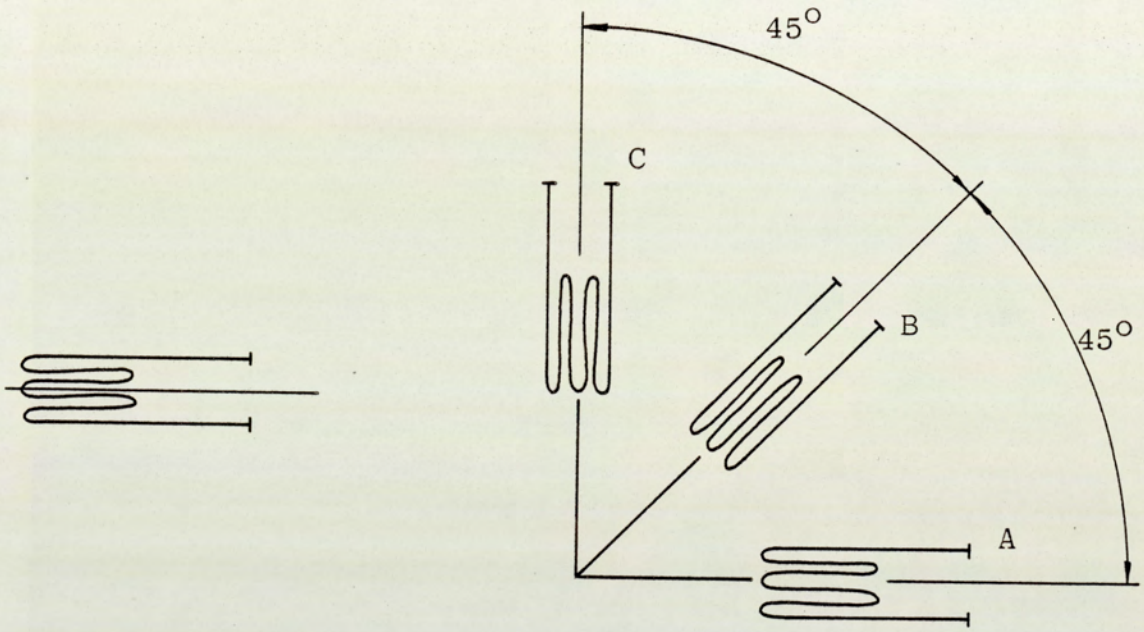
and the principal stress are:

$$\sigma_{max} = \frac{E}{1-\nu^2}(\varepsilon_{max}+\nu\varepsilon_{min}) \quad \ldots \quad (7.4)$$

$$\sigma_{min} = \frac{E}{1-\nu^2}(\varepsilon_{min}+\nu\varepsilon_{max}) \quad \ldots \quad (7.5)$$

The direction of the principal stresses is given by:

$$\varphi_p = \frac{1}{2} \tan^{-1}\left(\frac{2\varepsilon_B - \varepsilon_A - \varepsilon_C}{\varepsilon_A - \varepsilon_C}\right)$$



(a)
Linear Gauge

(b)
Three-gauge 45°
Rectangular Rosette

Fig. 7.1 Foil Strain Gauge Types

7.2.2 The Wheatstone Bridge

The Wheatstone Bridge is an electric circuit which can be employed to determine the very small change in resistance which a gauge undergoes when it is subjected to a strain. It may be employed to determine both static and dynamic strain-gauge readings. The bridge may be used either as a direct read out device, where the output voltage V_o is measured and related to strain, or the bridge may be used as a null-balance system where the output voltage V_o is adjusted to a zero value by adjusting the resistive balance of the bridge. In either of the modes of operation the bridge may be effectively be employed in a wide variety of strain-gauge applications. If the circuit shown in Fig. 7.2 is considered, a DC voltage V_i is applied across the terminals ac. It can be shown that the resulting voltage across the terminals bd is given by:

$$V_o = \frac{R_1R_3 - R_2R_4}{(R_1+R_2)(R_3+R_4)} \quad V_i \quad \ldots \quad \ldots \quad (7.6)$$

when the bridge is balanced:

$$V_o = 0$$

and

$$R_1R_3 = R_2R_4 \quad \ldots \quad (7.7)$$

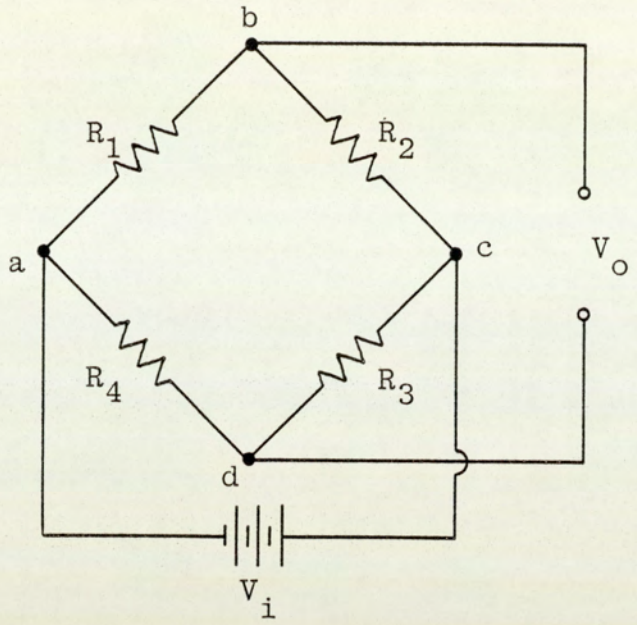


Fig. 7.2 The Wheatstone Bridge Circuit

If each value of resistance R_1, R_2, R_3 and R_4 are increased by an incremental amount ΔR_1, ΔR_2, ΔR_3 and ΔR_4 respectively, then the out-of-balance potential across bd is:

$$V_o = V_i \left| \frac{(R_1+\Delta R_1)(R_3+\Delta R_3)-(R_2+\Delta R_2)(R_4+\Delta R_4)}{(R_1+\Delta R_1)(R_2+\Delta R_2)(R_3+\Delta R_3)(R_4+\Delta R_4)} \right| \quad \ldots \quad (7.8)$$

By expanding the above expression, neglecting second order terms, and noting that

$$R_1R_3 = R_2R_4,$$

for balance it is possible to show:

$$\Delta V_o = V_i \frac{R_1 R_2}{(R_1+R_2)^2}\left(\frac{\Delta R_1}{R_1} - \frac{\Delta R_2}{R_2} + \frac{\Delta R_3}{R_3} - \frac{\Delta R_4}{R_4}\right) \quad \ldots \quad (7.9)$$

This equation represents the basic equation which governs the behaviour of the Wheatstone bridge in strain measurement.

7.3 PRELIMINARY TESTS

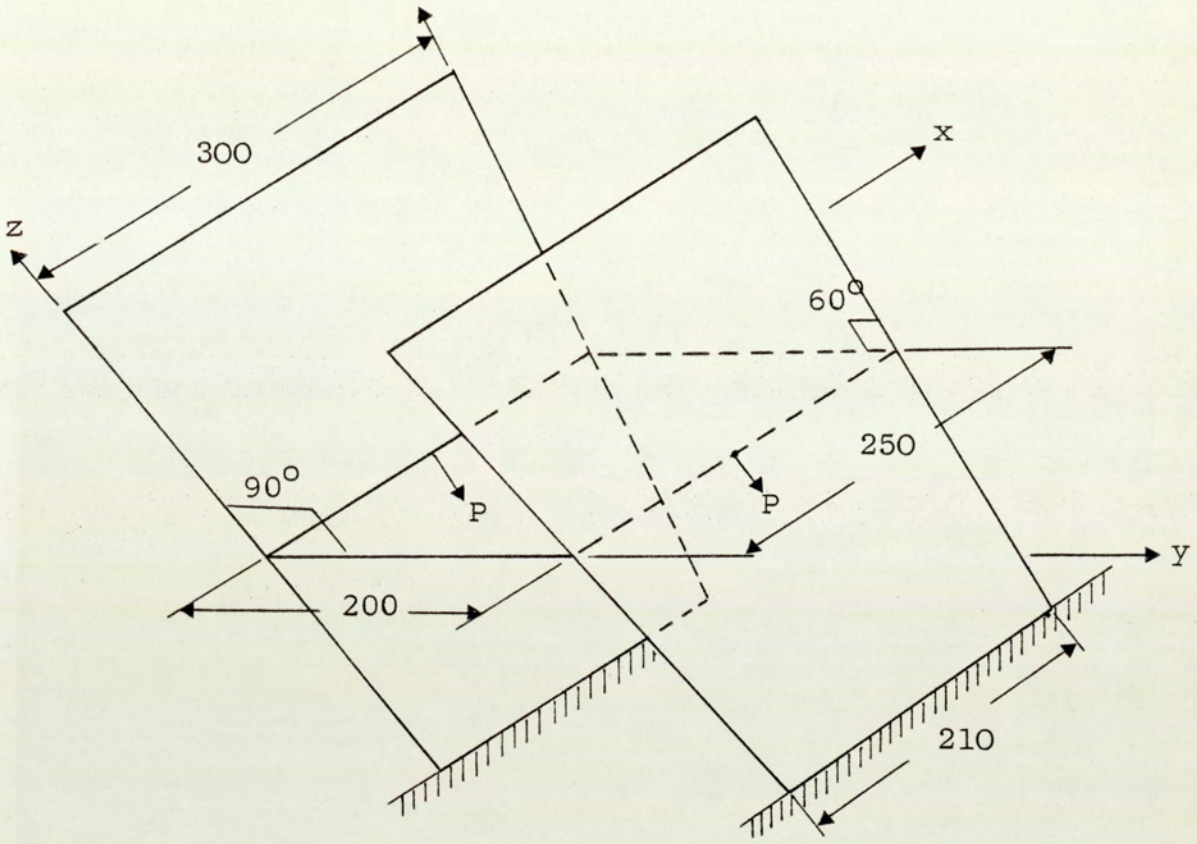
These tests were carried out at the early stages of developing the finite element program in order to confirm the ability of the semiloof shell element to model the sharp corners and multiple junctions which are found in the fan impeller. For this reason two examples were chosen:

(i) Three intersecting plates.

(ii) "L" shaped plate.

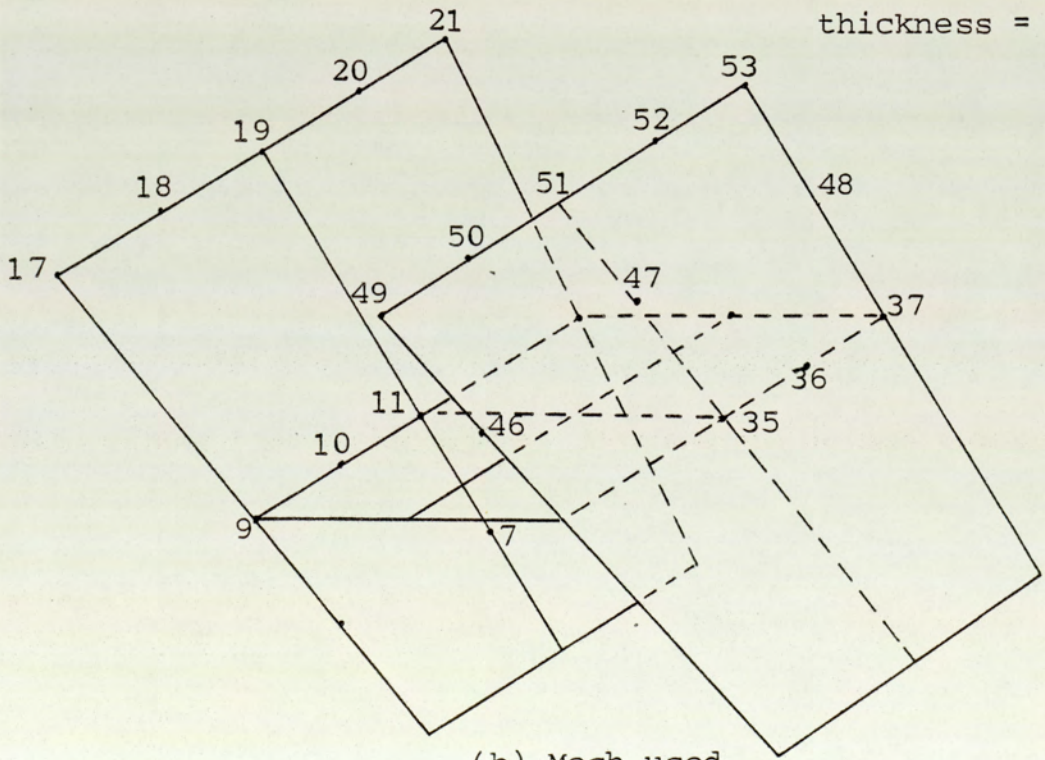
7.3.1 Three Intersecting Plates

This example was solved by the finite element method using a coarse mesh of 12 elements. The details of the geometry and the meshes used are shown in Fig. 7.3. Concentrated loads were applied in the z-direction (the positions of the applied loads are shown in Fig. 7.3(a)) and deflection was measured in y-direction using a dial gauge. This was repeated for the nodes numbered in Fig. 7.3(b). The measured deflections were compared with those obtained using the finite element method as shown in Table7.1



(a) Geometry

Dimensions in mm's
$E = 2.1\ E5\ N/mm^2$
$\nu = 0.29$
thickness = 1.01092



(b) Mesh used

Fig.7.3 Geometry and the finite element mesh used for the three intersecting plates

Table 7.1 Comparison of measured deflections with finite element results for three intersecting plates.

Node Number	Direction of measured deflection	Deflection × P (mm)		Error %
		F.E.M.	Exp.	
6	y-direction	-0.0195	-0.0225	13.33
7	y-direction	-0.0167	-0.0175	4.57
9	y-direction	-0.0226	-0.0225	0.40
10	y-direction	-0.0226	-0.0225	0.40
11	y-direction	-0.0226	-0.0225	0.40
17	y-direction	0.02702	0.02375	14.0
18	y-direction	0.0271	0.02375	14.0
19	y-direction	0.02609	0.0300	13.3
35	y-direction	-0.02269	-0.0262	13.7
36	y-direction	-0.02269	-0.0262	13.7
37	y-direction	-0.02269	-0.0262	13.7
46	y-direction	-0.0481	-0.050	3.8
47	y-direction	-0.0477	-0.045	6.0
48	y-direction	-0.048	-0.050	4.0
49	y-direction	-0.0736	-0.065	13.0
50	y-direction	-0.073	-0.08	8.75
51	y-direction	-0.072	-0.0725	0.68

7.3.2 "L" Shaped Plate

For this example, the geometry and the meshes used are shown in Figs. 7.4 to 7.7. A concentrated load P was applied in z-direction as shown in Fig. 7.4 and deflection measured in this direction using a dial gauge. This procedure was carried on for the nodes numbered in Fig. 7.6. The measured deflections were compared with those obtained from the numerical technique, as shown in Table 7.2.

7.3.2.1 Strain Gauge Tests for "L" Plate

Strain gauge tests were carried out on the "L" plate. The gauges were placed in x-direction at points corresponding to six selected integrating points for mesh B and on both directions x and y at the centroids of four chosen elements, as shown in Fig. 7.9.

7.3.2.2 Strain Gauge Circuit for "L" Shaped Plate

The strain gauges shown in Fig. 7.9 are bonded to the inside of the plate. The strain gauges are connected in the form of a Wheatstone Bridge using the Peekel "Data Strain" instrument as shown in the following diagram.

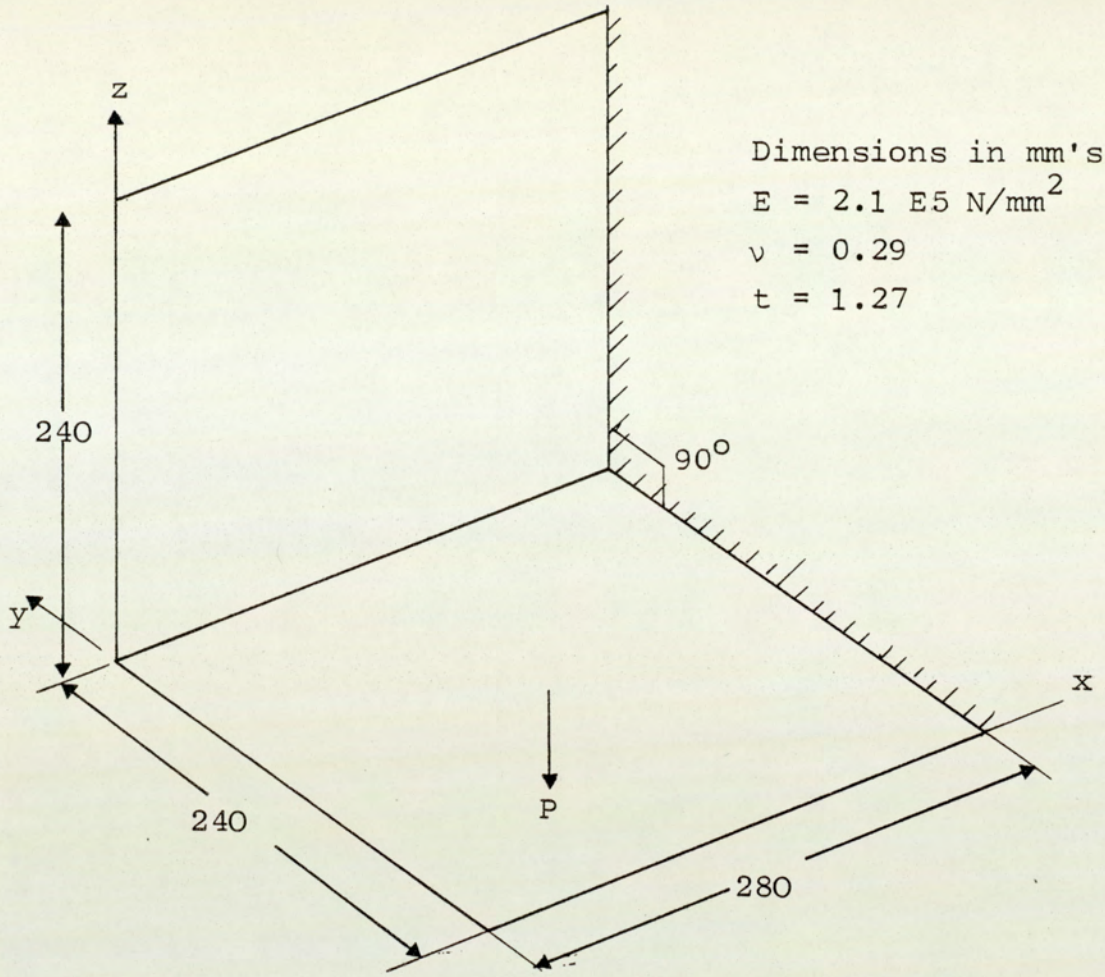


Fig.7.4 Geometry for the "L" shaped plate

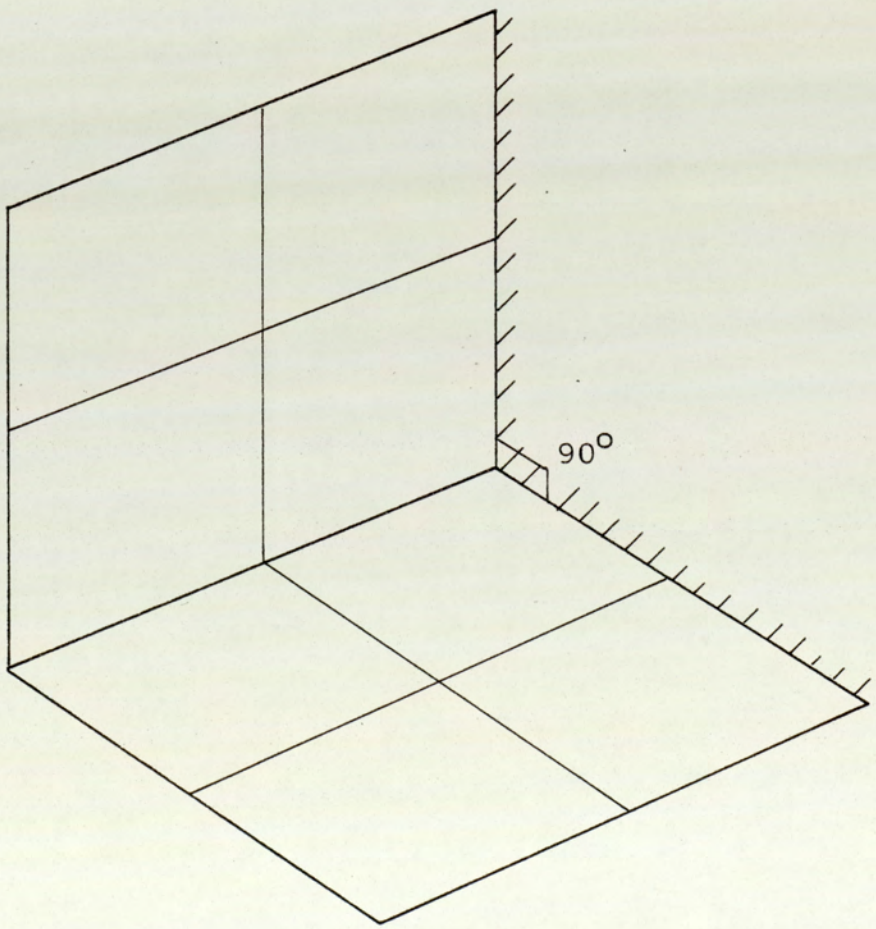


Fig.7.5 Finite element mesh A for the "L" shaped plate

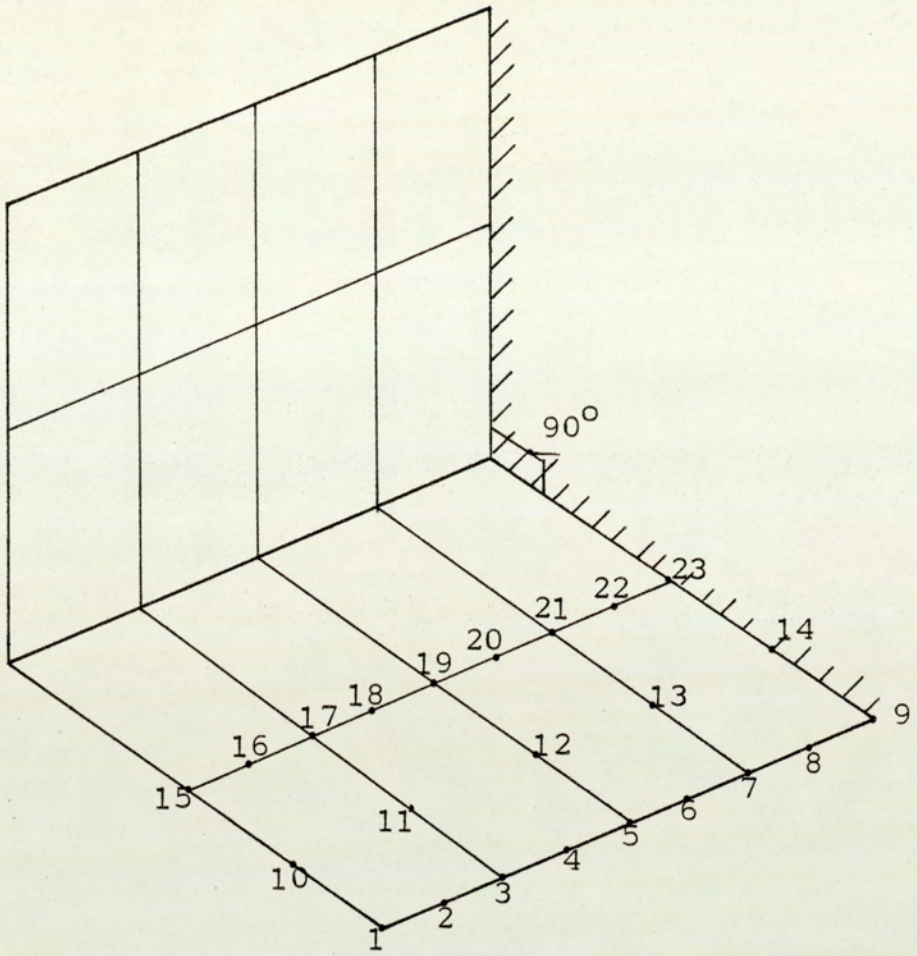


Fig.7.6 Finite element mesh B for the "L" shaped plate

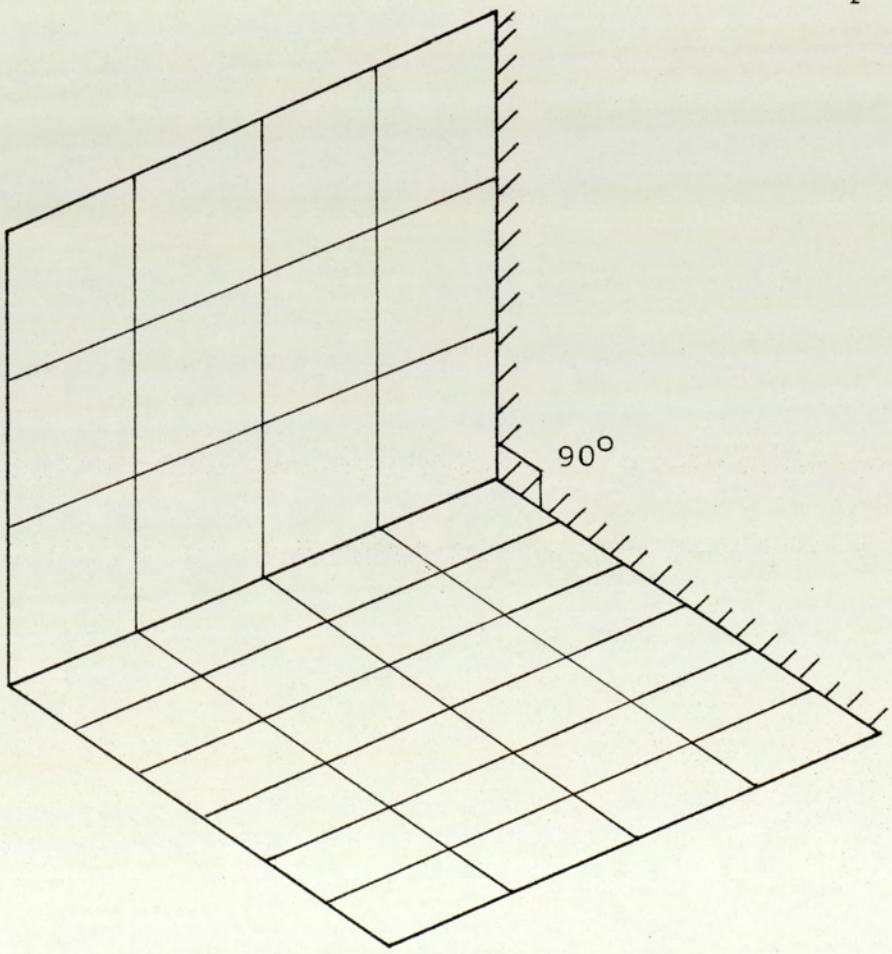
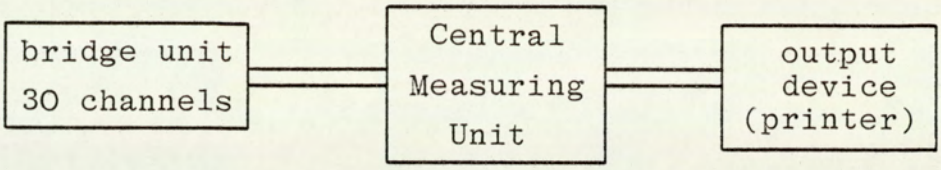


Fig.7.7 Finite element mesh C for the "L" shaped plate

Table 7.2 Comparison of Measured Deflections with Finite Element Results for the "L" Shaped Plate

Node Number	Deflection in z-direction (mm)			Error %
	Mesh A	Mesh B	Exp.	
1	0.0927	0.0885	0.082	7.0
2	-	0.079	0.080	1.25
3	0.078	0.0708	0.0698	1.4
4	-	0.0624	0.0693	10.0
5	0.048	0.0466	0.04246	9.0
6	-	0.0352	0.0397	10.2
7	0.0233	0.0163	0.0166	1.8
8	-	0.00647	0.0058	10.0
9	0	0	0	0.0
10	0.0745	0.073	0.0733	0.0
11	-	0.059	0.054	9.2
12	0.0388	0.0393	0.039	0.0
13	-	0.014	0.016	12.5
14	0	0	0	0.0
15	0.0526	0.0504	0.058	13.7
16	-	0.0455	0.043	4.6
17	0.045	0.0436	0.041	4.8
18	-	0.0406	0.0468	13.0
19	0.0394	0.0363	0.0343	5.8
20	-	0.026	0.0237	13.0
21	0.0168	0.0133	0.0129	3.0
22	-	0.0056	0.0063	12.6
23	0	0	0	0.0



Strains produced are measured in the Data Strain system by means of an AC carrier operated measuring unit. The DC output voltage from the carrier amplifier is measured by means of a high speed digital voltmeter. Strain gauge energization voltage, and input configuration mode can be selected on the rear panel according to the requirement of the test. Calibration and zero check can be carried out using a front panel switch. The measured value is displayed directly on the 5-decade tableau at the front panel of the central measuring unit. The system is completed with a printer as output device. To each measuring point a full, half or quarter bridge circuit of strain gauges can be selected.

7.3.2.3 Test Procedure

In this test, each active gauge was connected to form a quarter bridge circuit of strain gauges. A three wire system was used as shown in Fig. 7.9.

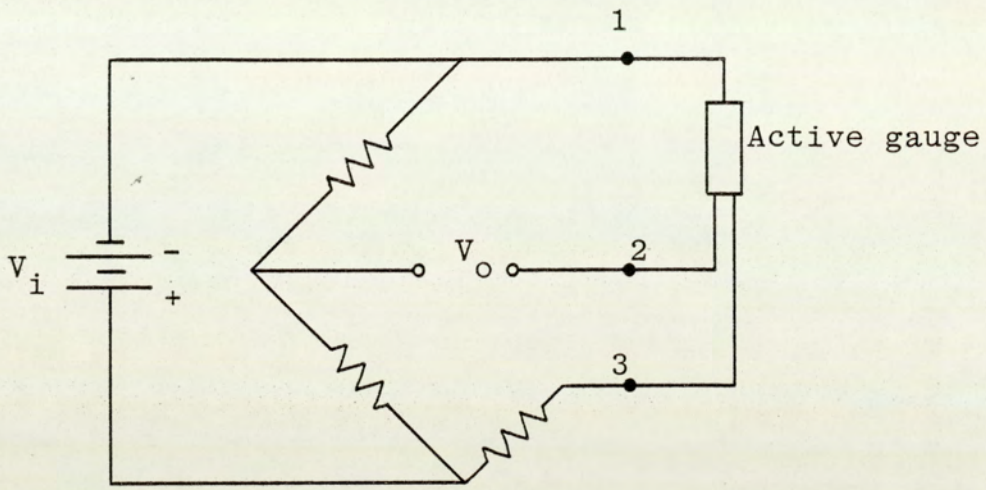


Fig. 7.8 Quarter Bridge ; 3 Wire System

Each active gauge used is manufactured in such a way that there is no resistance variation when its geometry changes solely due to temperature. However, the three wires connected to each active gauge will cancel out the effect of any temperature change in the lead wires, because the leads 1 and 3 of Fig. 7.8 are in opposite arms of the bridge. Therefore, any change in the resistance caused by the change of temperature in one lead will be compensated by the change of the resistance of the other lead. A series of tests for each strain gauge were carried out. The readings were taken for loads 8, 16, 24, 32 and 40 Newton. In order

to check the behaviour of the strain gauges, two typical gauges were selected (their positions are shown in Fig. 7.9), and the measured strain was plotted against the applied load giving the linear relationship shown in Fig. 7.10. This was expected, since the load is proportional to the strain produced, and thus gave confidence in the results obtained. This process has been applied for all the gauges used, and the measured strains for 24 N were compared with those obtained from the finite element solution and shown in Table 7.3.

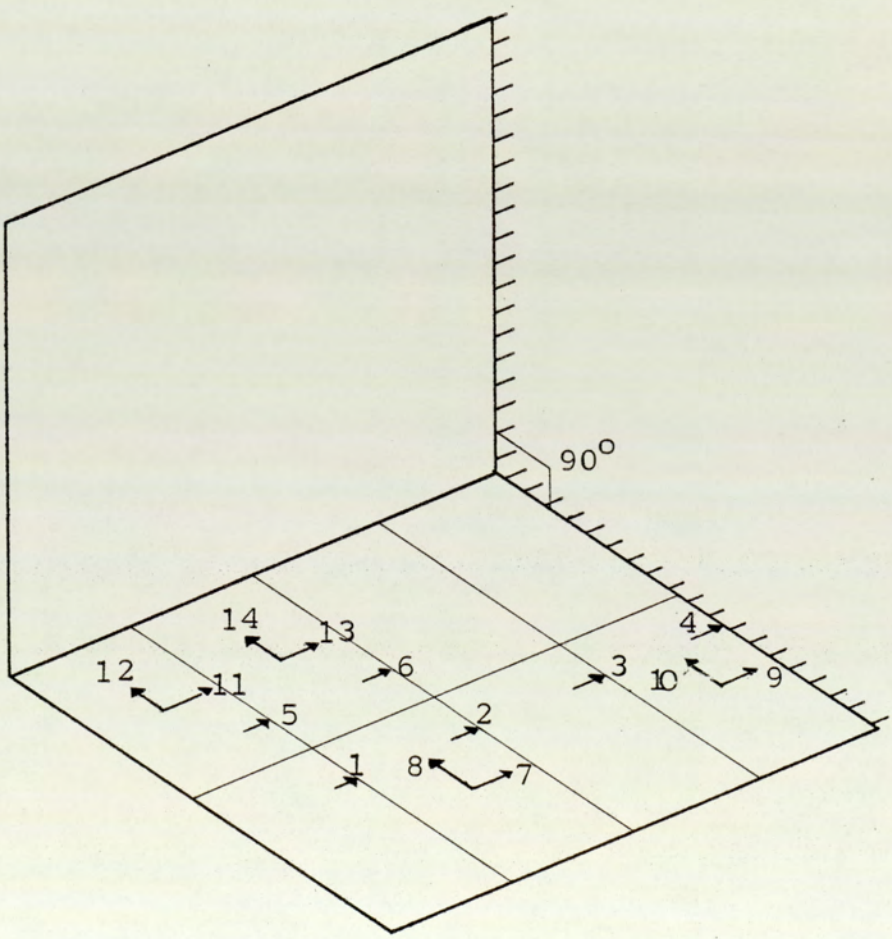


Fig. 7.9 Strain gauge distribution on the "L" shaped plate.

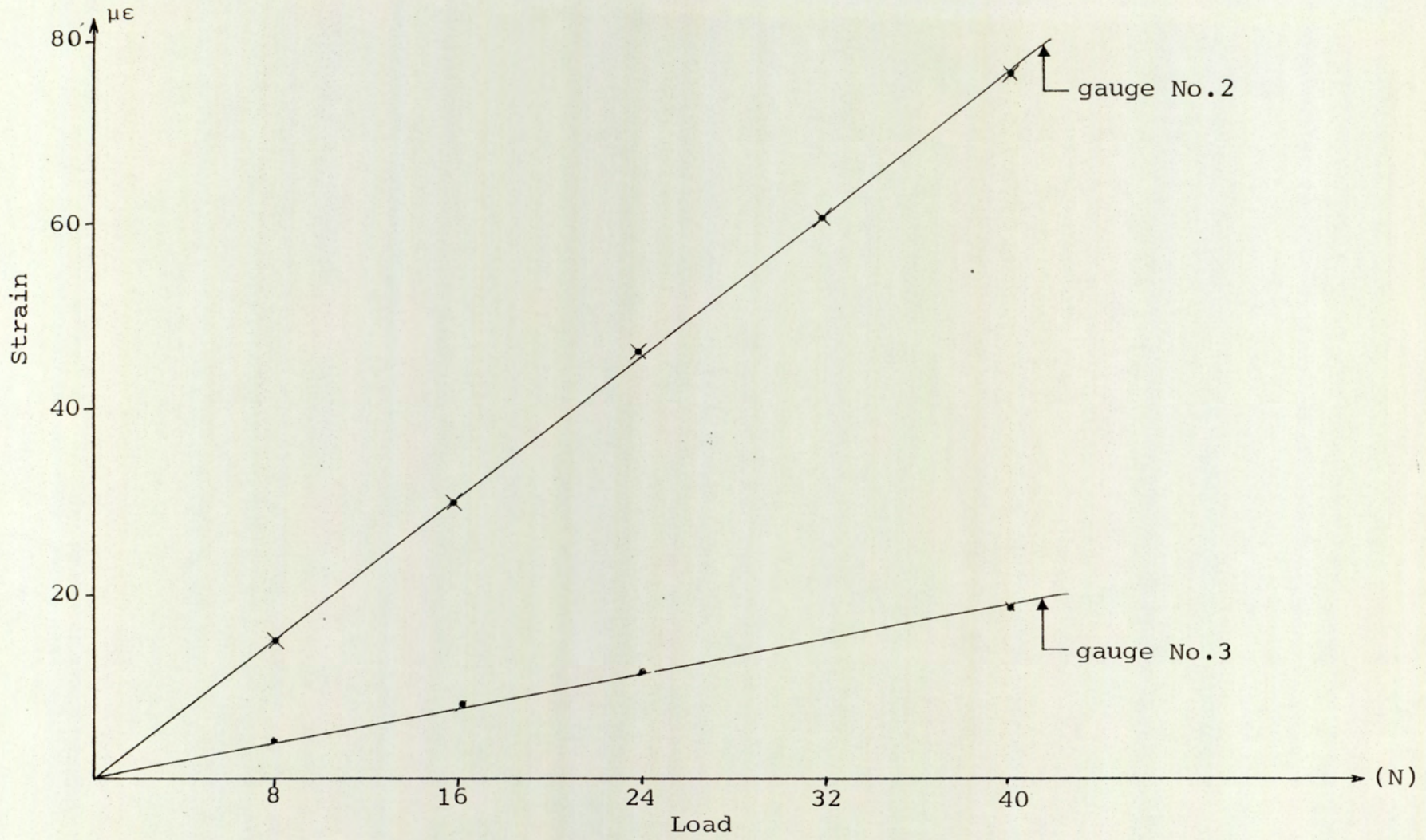


Fig.7.10 Graph of strain (με) against Load (N) for strain gauges numbered 2 and 3 on the "L" shaped plate

Table 7.3 Comparison of the Measured Strains with the Finite Element Results for the "L" Shaped Plate

Strain Gauge No.	Exp. με Results	Finite Element Results		Error %
		Mesh B	Mesh C	
1	7	5	-	28.5
2	46	46	-	0
3	11	14	-	27.2
4	81	94	-	16
5	47	43	-	8.5
6	damaged	2	-	-
7	26	13	21	19.2
8	27	25	28	3.7
9	80	70	79	1.2
10	12	10	12	0
11	5	12	5	0
12	12	9	13	8.3
13	14	3	15	7.1
14	4	1	4	0

7.3.3 Discussion on the Results for the Preliminary Tests

Tables 7.1 and 7.2 have shown very good correlation between the measured deflections and those obtained from the finite element solution using coarse meshes.

The measured strains for the "L" shaped plate have also shown reasonable agreement compared to those obtained from the finite element method as shown in Table 7.3. The results of strains for mesh B indicated some points with an error more than 20% compared to the measured strains. This is because the tested problem has sharp corners with concentrated force and such situations in the finite element analysis need finer meshes to idealise the problem. However, the results in general gave confidence in using the semiloof shell element to idealise the fan impellers having the junctions of blade/backsheet and blade/conesheet.

7.4 SIMPLIFIED FAN IMPELLER

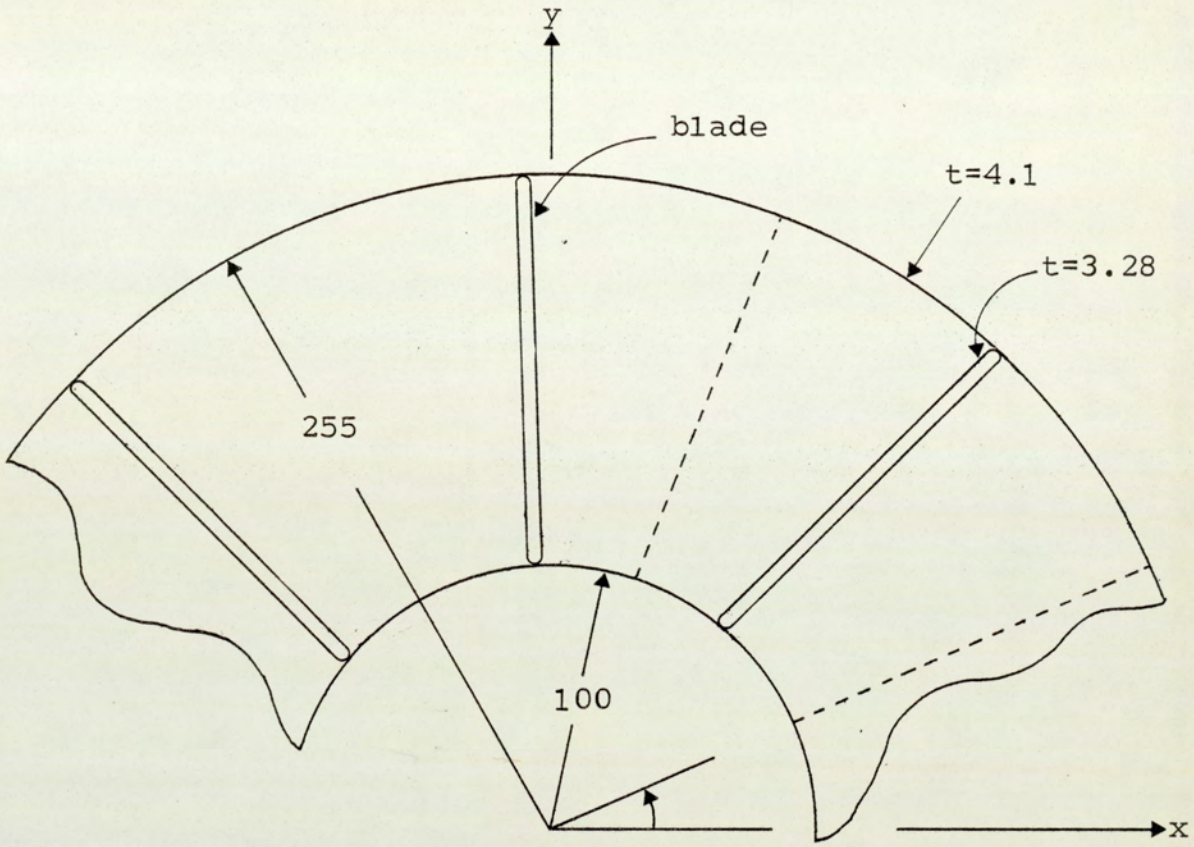
In order to confirm the technique of exploiting the sectorial symmetry, described in Chapter 6, in solving the rotating fan impellers, it was decided to analyse a simplified radial fan impeller numerically, using the finite element method and experimentally using the strain gauges technique. A photograph of this model is shown in Fig. 7.11. The impeller was constructed from mild steel. The backsheet was formed from 4.1 mm thickness steel sheet. Eight equispaced radial blades of thickness 3.38 mm were welded to the backsheet. Dimensions and positions of the blades on the backsheet are shown in Fig. 7.12.

7.4.1 Experimental Investigation

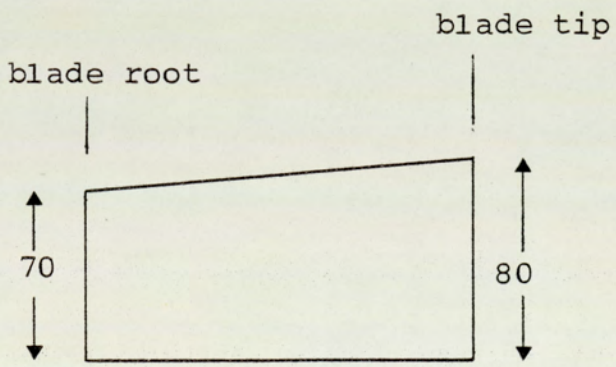
7.4.1.1 Strain Gauges Distribution

The simplified radial fan impeller was strain gauaged using linear and 45^{o} rosette strain gauges. These gauges were installed on the inside and outside of the backsheet and on the inside of the blade. The position of the gauges on the inside of the backsheet and the blades are shown in Fig. 7.13.

Fig 7.11 Simplified fan impeller



(a) Backsheet dimensions



(b) Blade dimensions

Dimensions in mm's

Fig.7.12 Dimensions and position of the blades on the backsheet for the simplified fan impeller

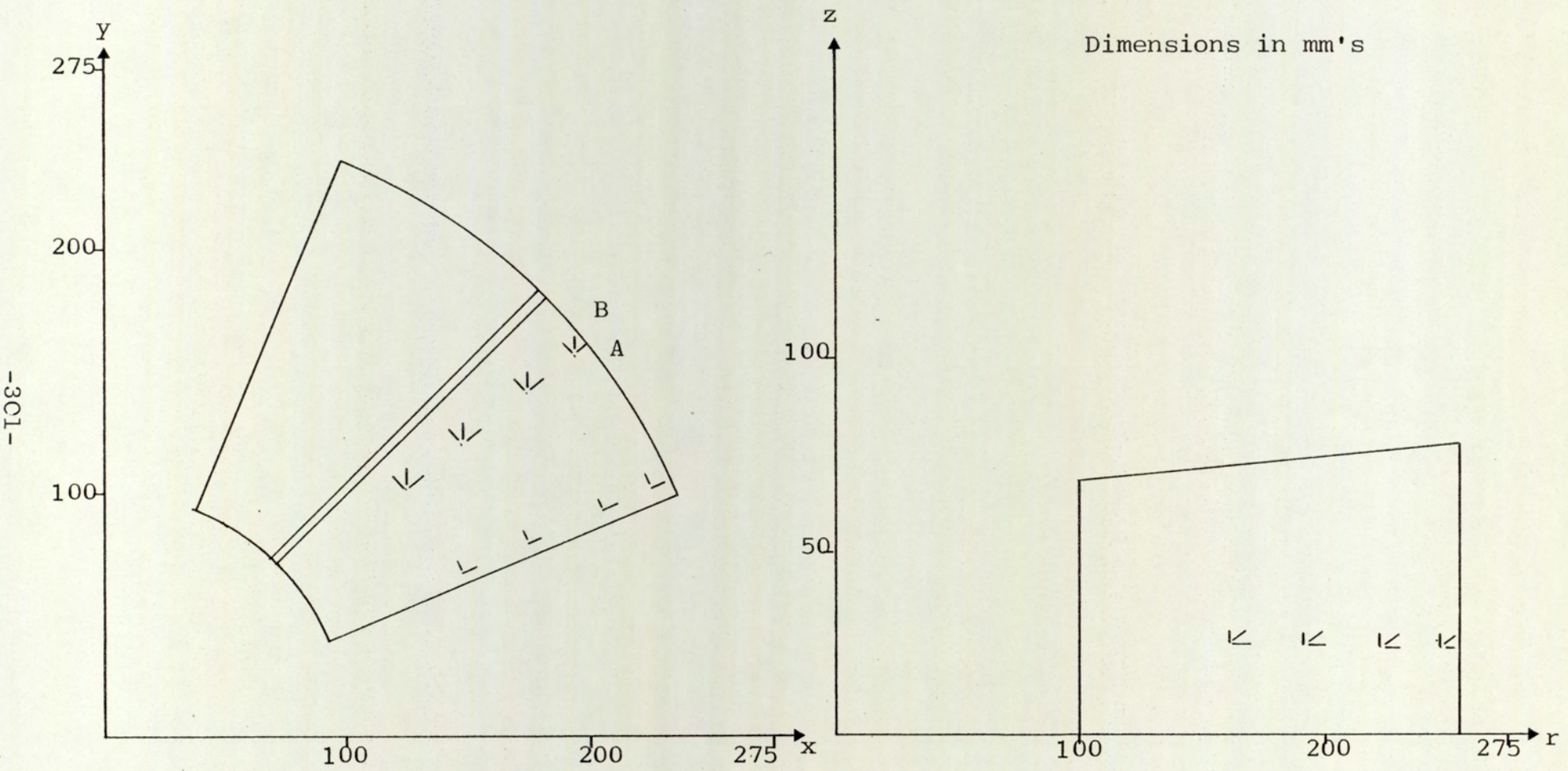


Fig.7.13 Strain gauges distribution on the radial simplified fan impeller

7.4.1.2 Experimental Apparatus

A photograph of the rig is shown in Fig. 7.14. It consists of a 50 mm diameter shaft mounted on ball bearings positioned on a rigid structure which is bolted and securely fixed to the ground. The shaft was driven by V belt and pulley system with speed ratio 1:1 from a 220V, 5 HP variable speed motor. The signals from the strain gauges mounted on the rotating impeller were obtained via slip rings.

The equipment used throughout the experiments was as follows:

(1) Simplified fan impeller.
(2) Variable speed AC motor type KNX-C184, 5 HP.
(3) Five digits digital voltmeter Solatron model A200.
(4) Michigan Scientific 10 way slip ring.
(5) Farnell Instrument Ltd. DC power supply (0-30 volts, 0-1 amp).
(6) Micro measurements linear and 45° rosette temperature compensated strain gauges type FRA-3-11 with gauge factor 2.11.

7.4.1.3 Strain Gauge Circuits

The circuit is essentially a Wheatstone bridge which is fed with four gauges. The quarter bridge three system was used in this test. In this system of wiring only one

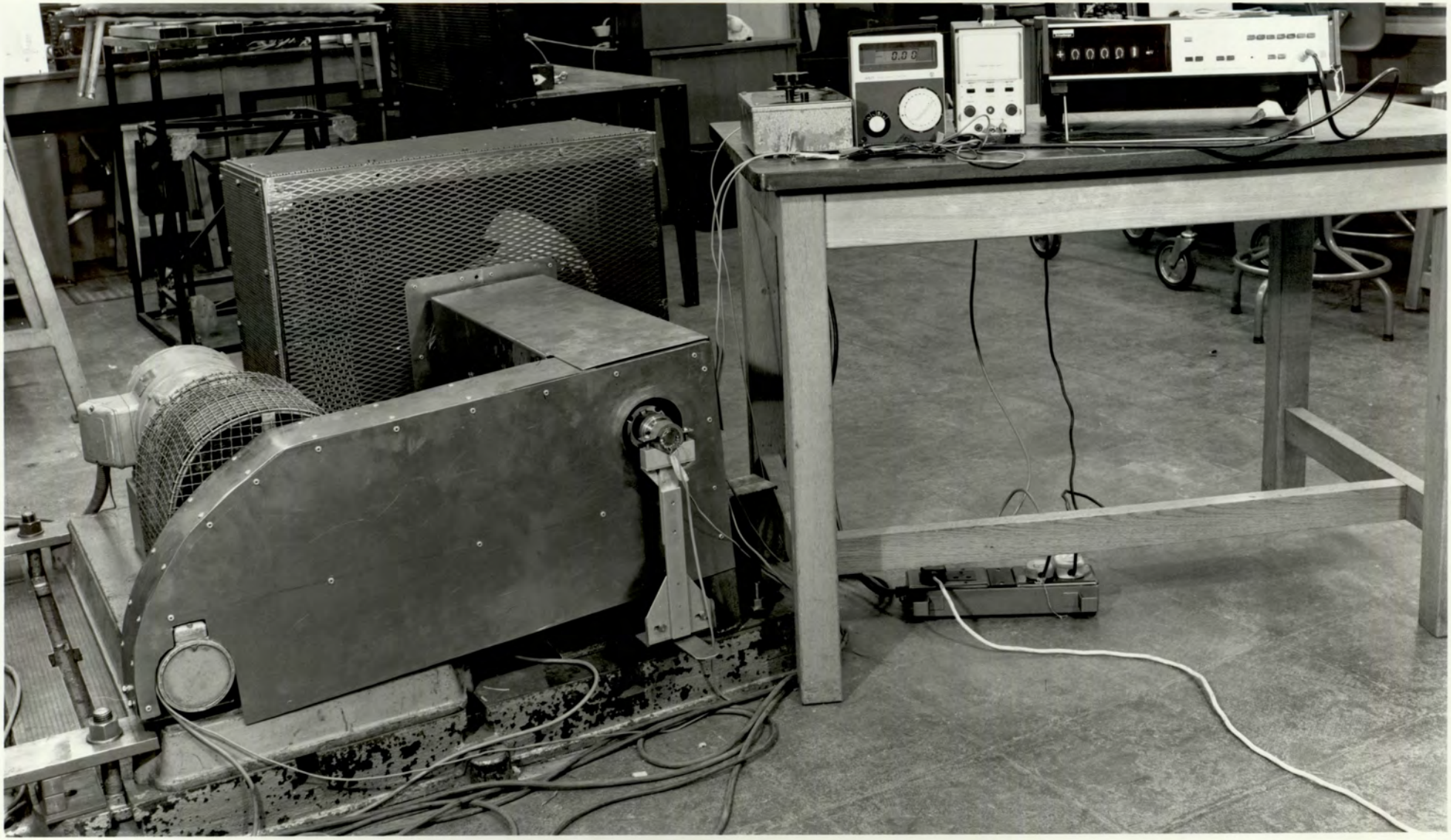


Fig 7.14 Rig used in testing the simplified fan impeller

single active gauge needs to be installed and three dummy gauges were used in the other arms of the bridge, thus the total number of strain gauges required is reduced. The dummy gauges were arranged in the adaptor fixed to the shaft as near as possible to the centre of rotation, in order to give insignificant strain on the three dummy gauges in each circuit, and thus they contribute no effect on the signal measured due to the active gauge. The signal is taken from the bridge by using a 10 channel slip ring assembly, four channels of data and two channels for the dc power supply as shown in the circuit diagram of Fig. 7.15. This circuit was recommended by Gall, Ref(79), who described the practical uses of slip rings for strain gauge measurements on the rotating objects. The slip ring noise which is caused by the slight variation in the slip connection resistance and the rubbing contacts, is the main problem when using slip ring assemblies. In this circuit, the slip ring and its rubbing contacts are outside the bridge circuit, therefore, they cannot affect the bridge balance. In well designed slip ring assemblies the peak noise attributed to the slip ring contacts is less than 0.1% of the signal.

A high accuracy five digits digital voltmeter was used to measure the output signal. The signal was constant with the fan running at a constant speed which gave confidence in the circuit used with insignificant noise of the slip ring during the test.

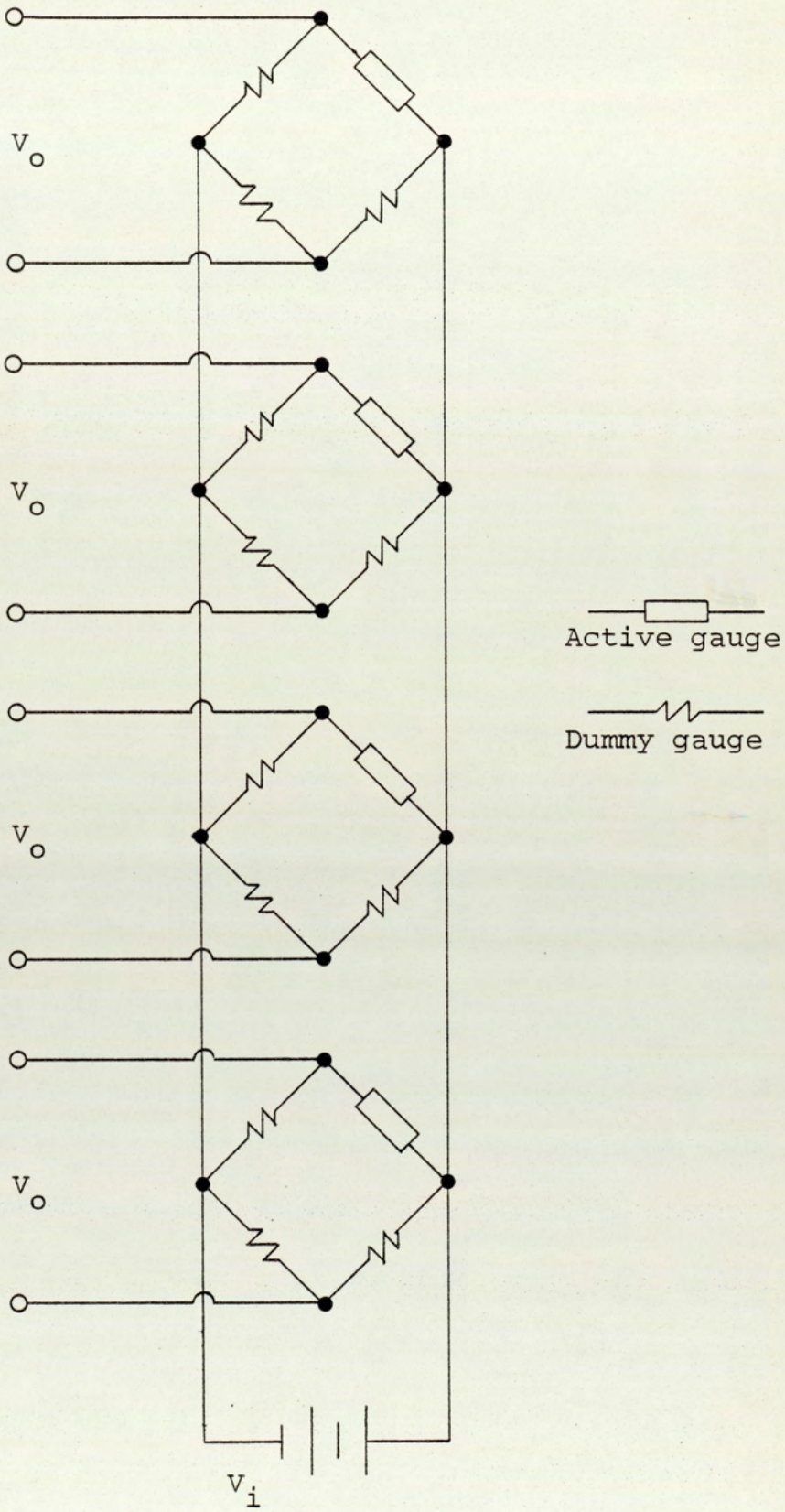


Fig.7.15 Strain gauges circuit using 10 channel slip ring

The unbalance output signal (mv) can be converted to strain as follows:

For a quarter bridge circuit, it is given by:

$$\varepsilon = \frac{4\,V_o}{V_i\,F} \qquad \ldots \quad (7.10)$$

where ε is the strain, V_o is the measured unbalance voltage (The difference between the unbalance signal with the fan stationary and with the fan rotating) due to change in resistance of the active gauges as a result of the rotating forces acting on it.

The tests were carried out at different speeds between 800 - 1300 rpm. . For each gauge, as for the two selected typical gauges shown in Fig. 7.16, the measured unbalance V_o was plotted against the speed squared giving the linear relationships expected. The position of these gauges are shown in Fig. 7.12. The behaviour of the strain gauge in this manner gave confidence in the installation of the strain gauges and the circuit used.

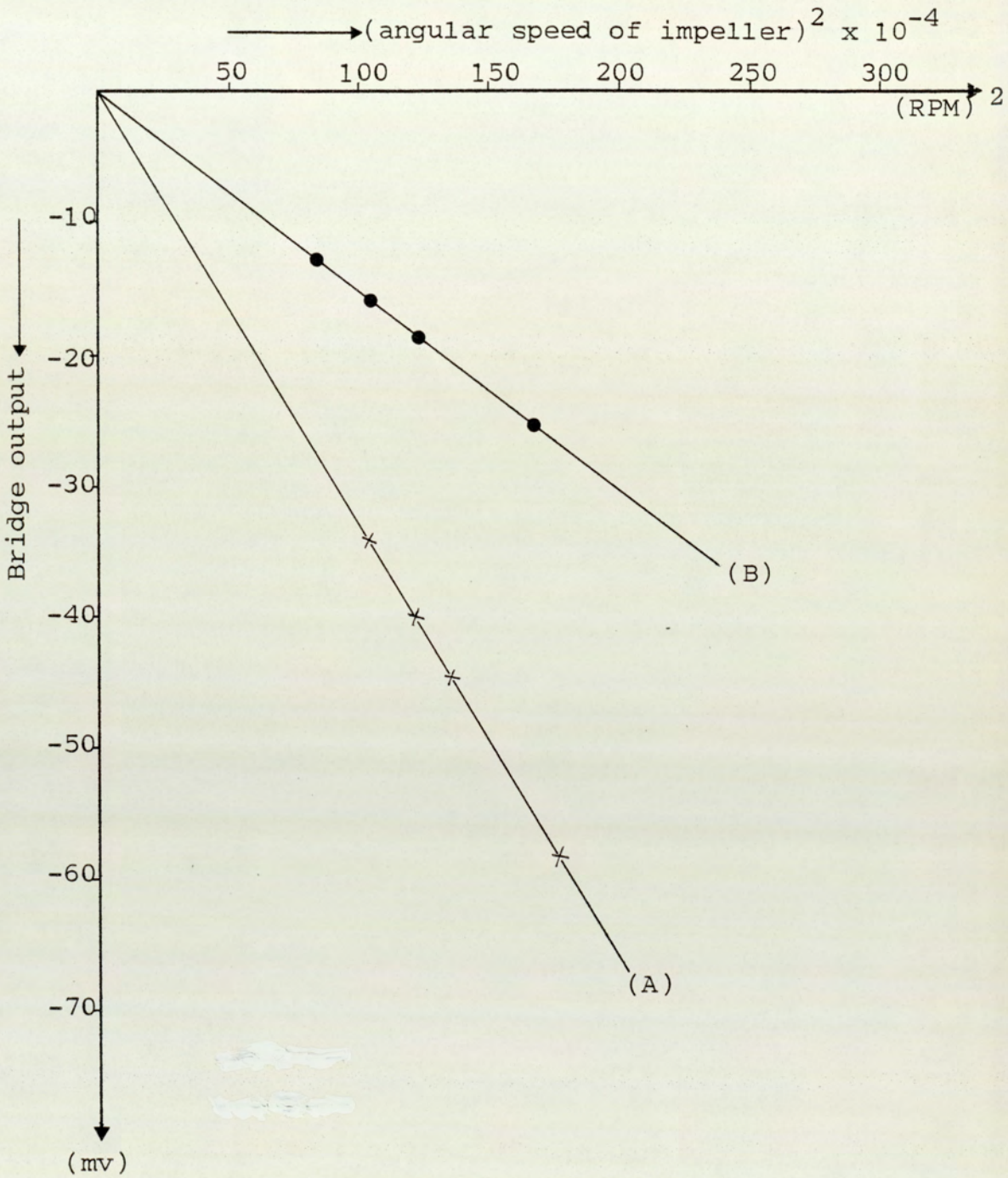


Fig.7.16 Speed squared against the output voltage for two typical strain gauges on the outside of the backsheet of the simplified model

7.4.2 Finite Element Analysis

A finite element analysis for this model has been carried out using the program developed in Chapter 6. Since the impeller is sectorially symmetric, only one-eighth of the impeller need to be divided into elements and imposing the equality of the displacements for the geometrically corresponding nodes on the sector boundaries.

7.4.2.1 Boundary Conditions

As mentioned before, each node on the radial bounding plane (a-a) shown in Fig. 7.17 needs to be specified and due to symmetry a prescribed displacement of zero in the tangential direction should be specified for these nodes. The rotation of each loof node on the bounding plane (a-a) is also imposed as zero. The structure of the fan is assumed to be built inside.

Specifications of the nodes are assigned according to the instructions given in Appendix A.

7.4.2.2 Finite Element Meshes

The finite element meshes used are shown in Figs. 7.18 to 7.20. A quadrilateral semiloof shell element was chosen as a discretised element. These meshes are as follows:

(1) Mesh I, shown in Fig. 7.18, in which the impeller was discretised into 12 and 6 elements in the backsheet and the blade respectively.

(2) Mesh II, shown in Fig. 7.19, in which the impeller was idealised by 24 and 8 elements in the backsheet and the blade respectively.

(3) Mesh III, shown in Fig. 7.20, in which the impeller was represented by 30 and 10 elements in the backsheet and the blade respectively.

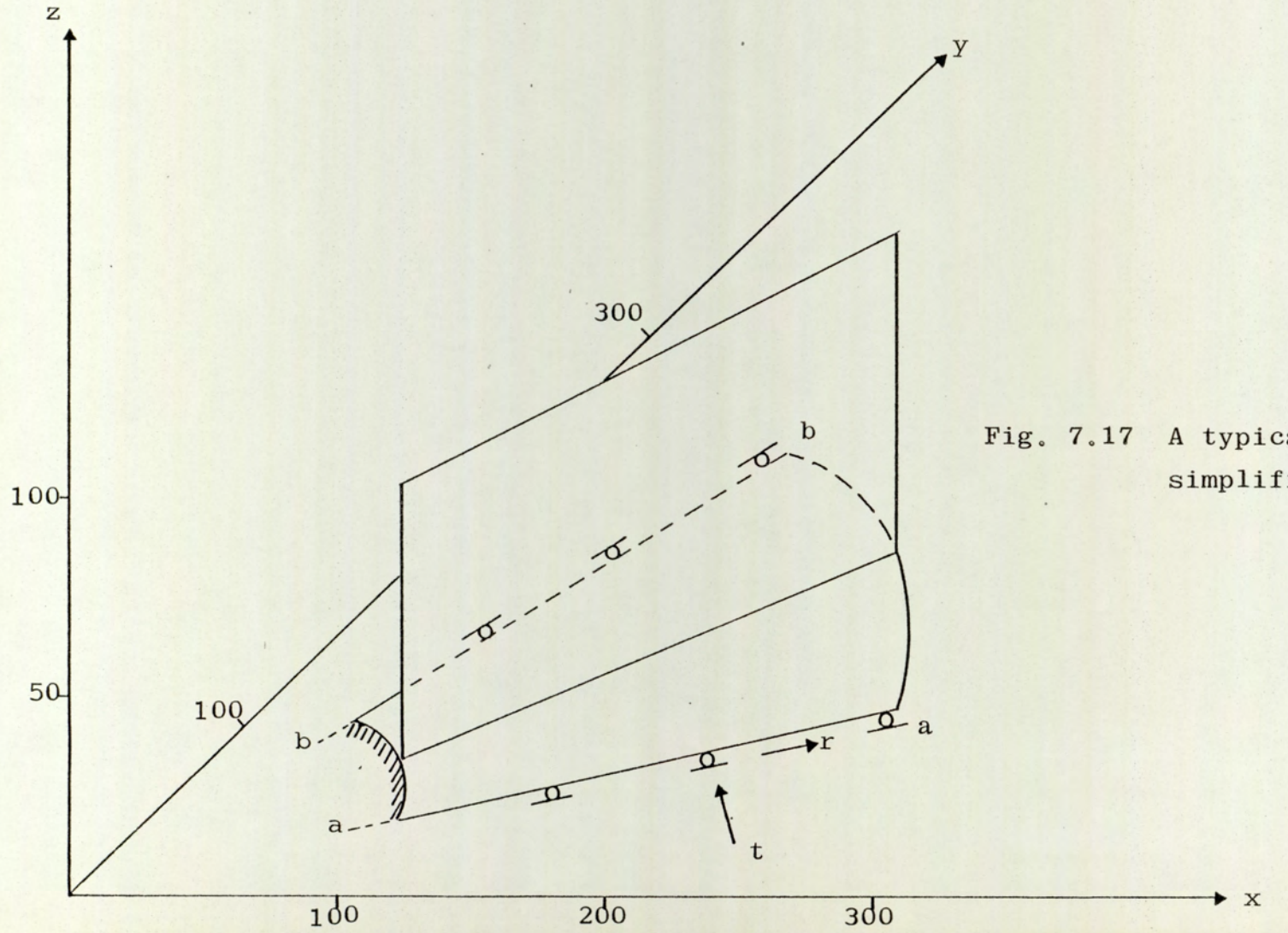


Fig. 7.17 A typical sector for the simplified fan impeller

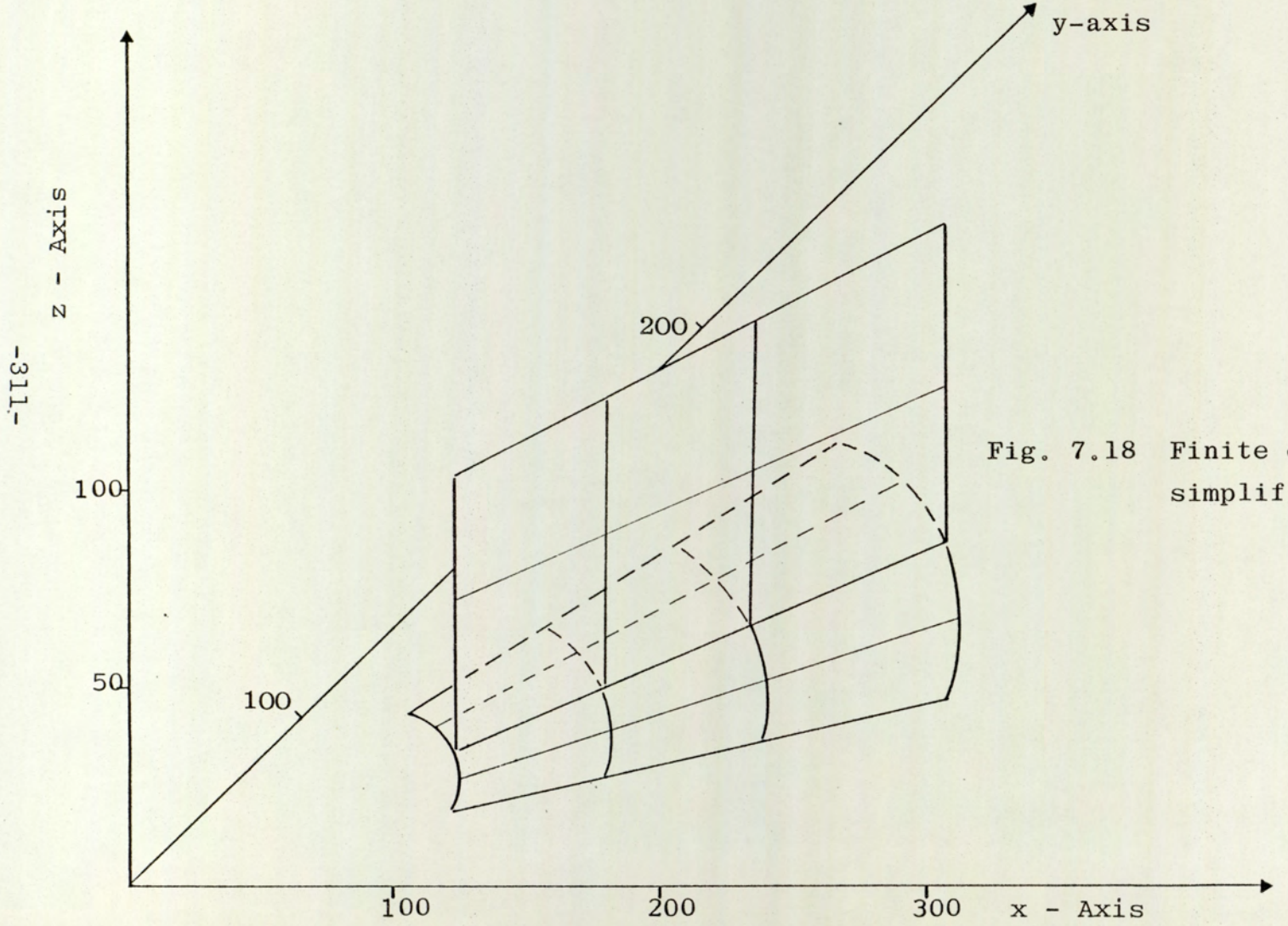


Fig. 7.18 Finite element mesh I for the simplified model

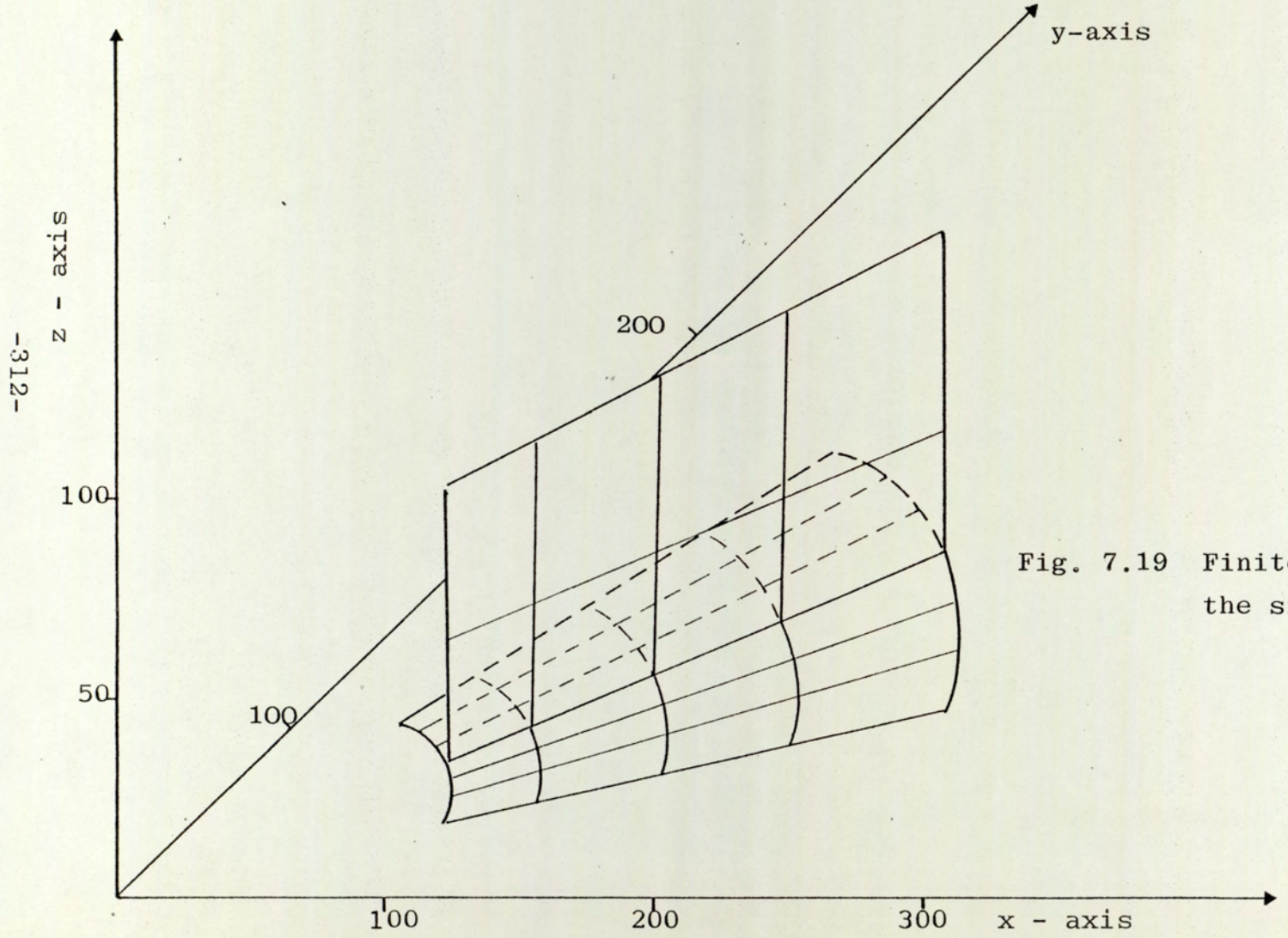


Fig. 7.19 Finite element mesh II for the simplified model

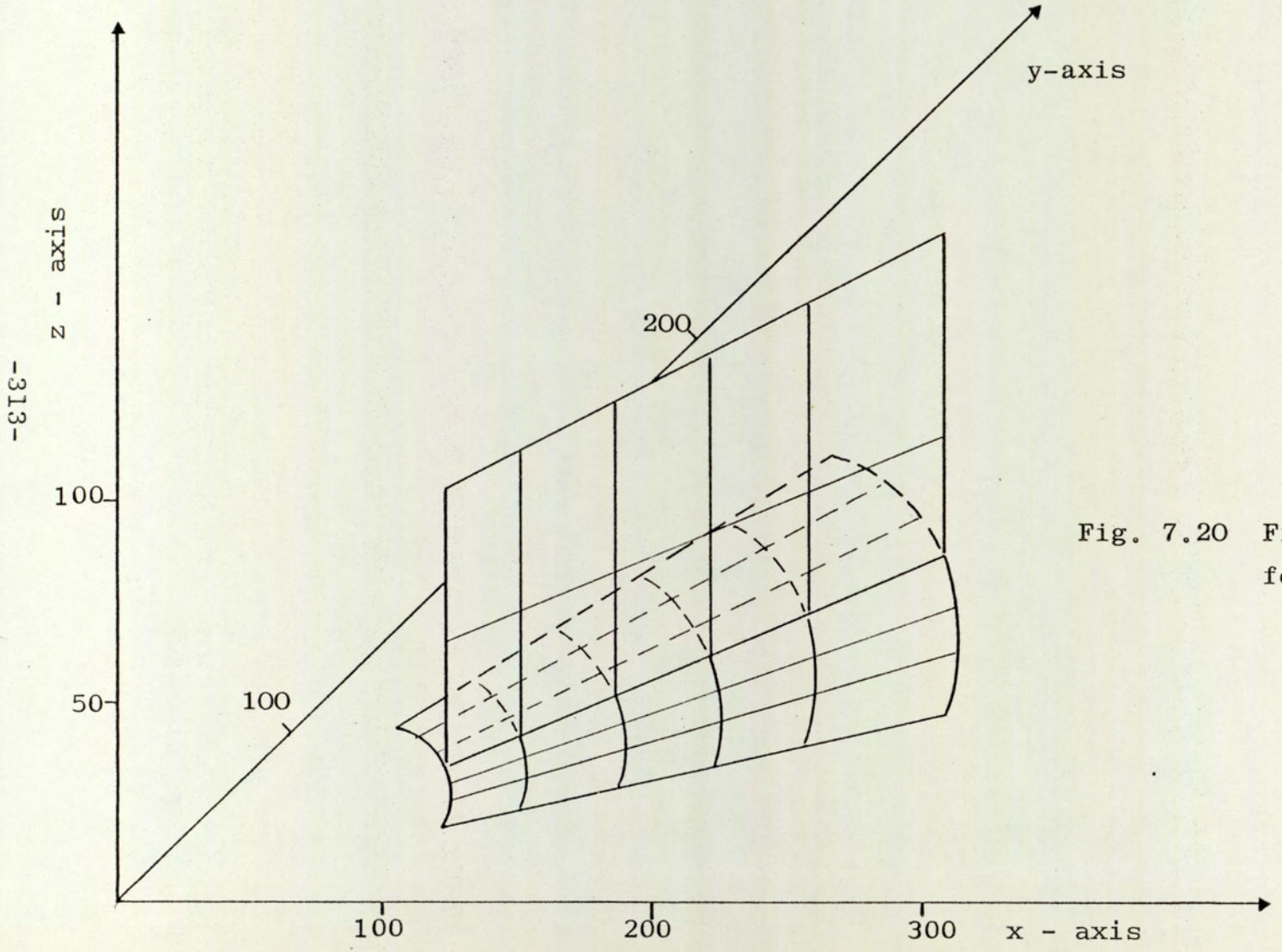


Fig. 7.20 Finite element mesh III for the simplified model

7.4.3 Discussion of the Results for the Simplified Fan Impeller

As expected, the finite element results showed that the principal stresses for the Gauss points on the line $\theta=24^{\circ}$ (between the blades) are radial and tangential. For this reason, two linear gauges were installed on this line to measure the principal strains from which at each Gauss point the radial and tangential (principal) stresses were calculated. Good agreement has been obtained for the principal stresses between the finite element results and those calculated from the strain gauges readings on the inside and outside of the backsheet as shown in Fig. 7.21 and 7.22. Tables 7.4 and 7.5 contain the experimental and the numerical values of the radial and tangential stresses for the Gauss points on the line $\theta=24^{\circ}$. As expected, both bending and membrane stress resultants exist, but the bending stresses are dominant due to the effect of the blade which becomes significant at the inner edge, because the angular distance between the welds of the adjacent blades is smaller than the outer edge. The radial stresses shown in Figs. 7.21 and 7.22 rapidly decrease to zero, as expected, for both the inside and outside the backsheet, and the high stress occurs near the edge of the backsheet.

For the numerical and experimental results of the Gauss points on line $\theta=39$, it was decided to represent the stresses in the form shown in Figs. 7.23 and 7.24, for both

the inside and outside the backsheet, to get an insight into the directions of the principal stresses. The principal directions of the stresses in the gauss points here are no longer radial and tangential as those obtained for gauss points on line $\theta=24^{o}$. This is due to the effect of the blade which becomes significant at this section. Good agreement for the magnitude and direction of the principal stresses was obtained, between the experimental and the finite element results. Both techniques indicate high stress at the inner edge as expected.

The magnitude and direction of the principal stresses calculated from the measured strains on the inside of the blade compare resonably with those obtained from the finite element solution using Mesh II as shown in Fig. 7.25. The bending stress is zero as there is no out-of-plane force acting on the blade. Excellent agreement was obtained for the radial membrane stress, as shown in Fig. 7.26, between the experimental and the numerical results for the line of the gauss points shown in Fig. 7.25. The membrane stress, as expected, decreases rapidly to zero at the free end.

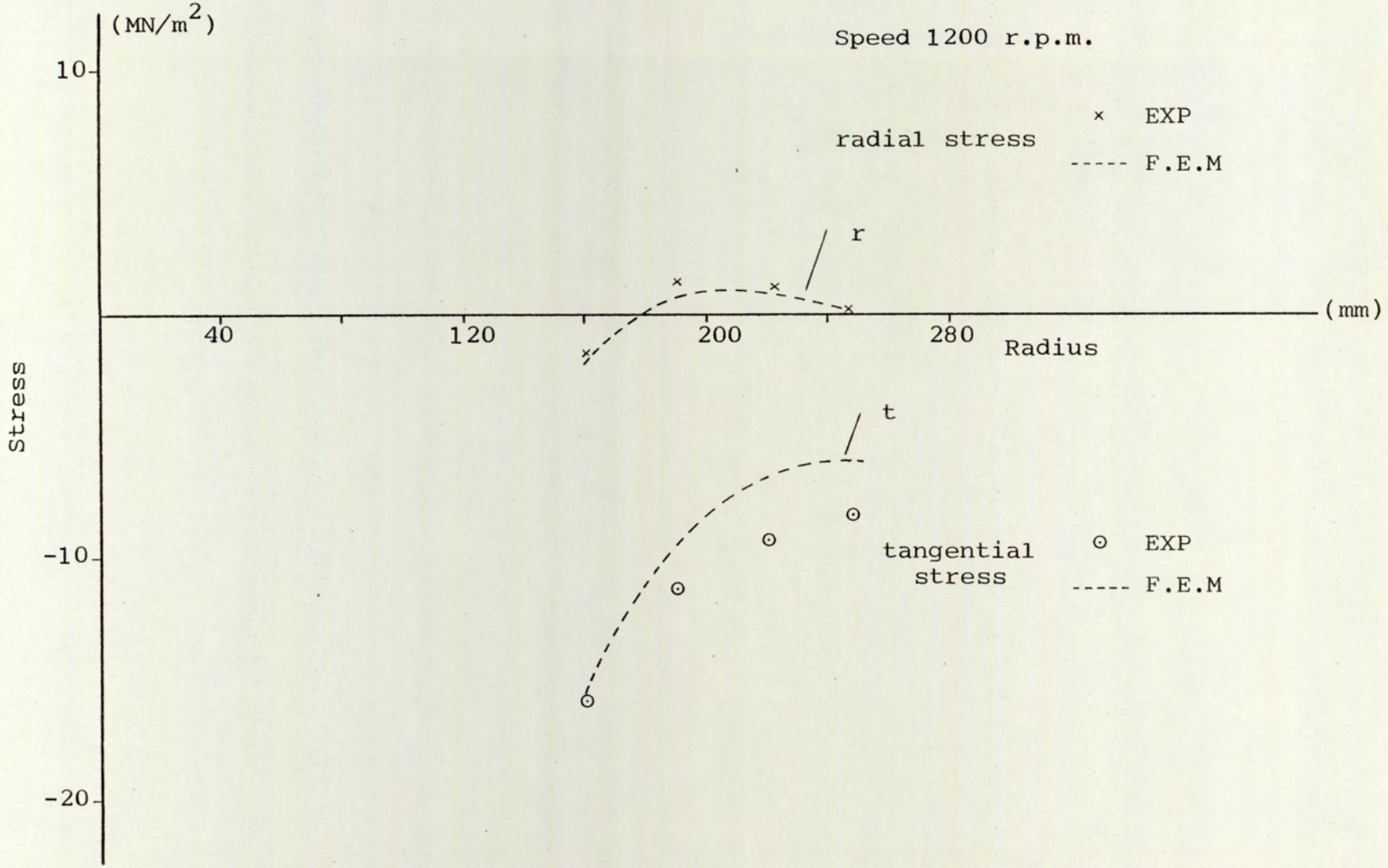


Fig.7.21 Radial and tangential stresses on outside of Backsheet along line $\theta = 24^{o}$: Simplified model

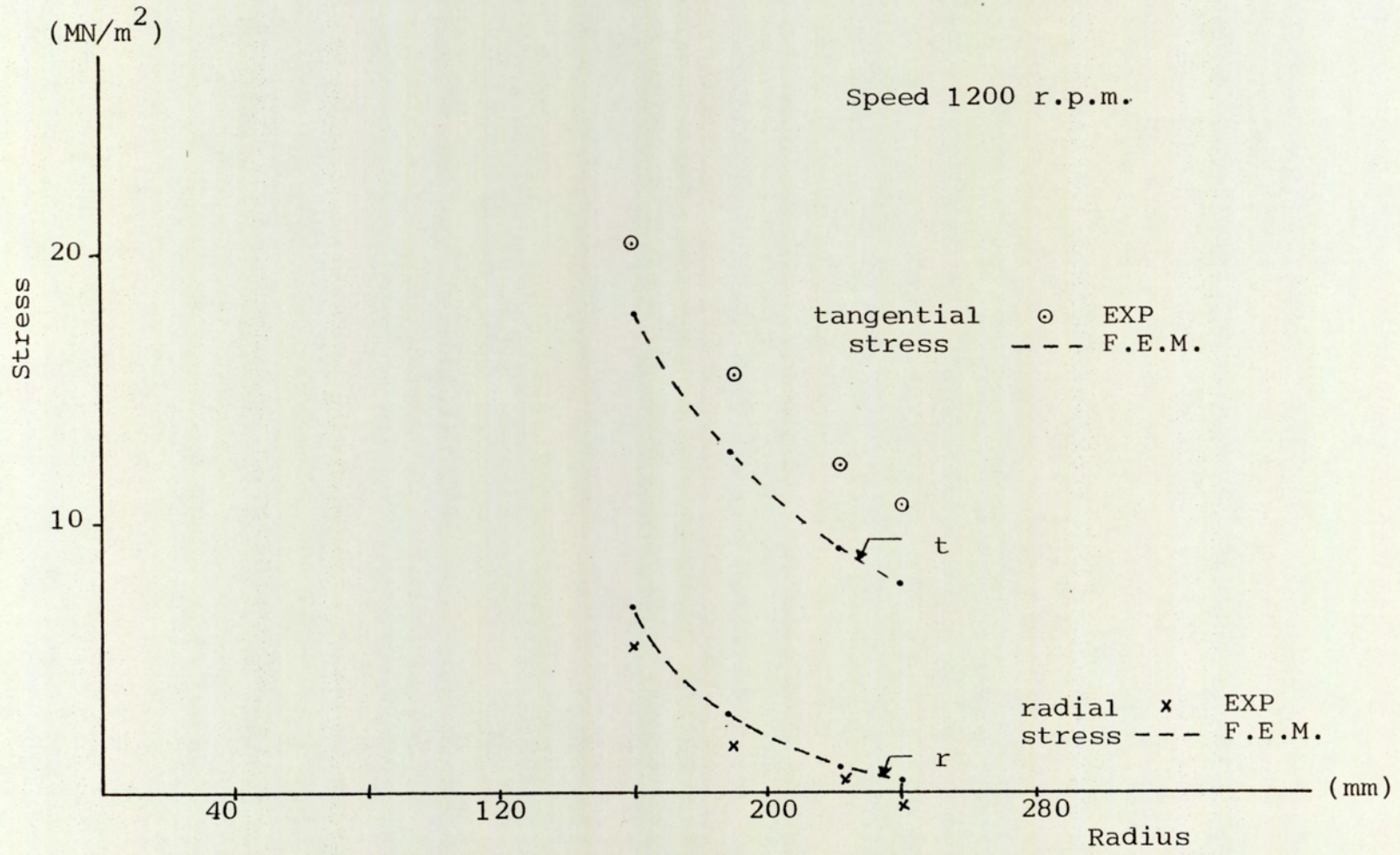


Fig.7.22 Radial and tangential stresses on inside of Backsheet along line $\theta = 24^{o}$: Simplified model

Table 7.4 Approximate values of Radial and Tangential Stresses on Outside Backsheet Along $\theta \simeq 24^{o}$; simplified model.

Radius mm	Radial Stress (MN/m^2)		Error	Tangential Stress (MN/m^2)		Error
	F.E.M.	Exp.		F.E.M.	Exp.	
260	-1.92	-1.6	-0.20	-15.36	-15.9	0.033
190	0.915	1.172	0.219	- 9.2416	-11.59	0.20
222	0.843	1.2	0.297	- 6.712	- 9.5	0.29
246	0.1318	0.417	0.683	- 6.048	- 8.7	0.30

Table 7.5 Approximate Values of Radial and Tangential Stresses on Inside Backsheet along $\theta=24^{o}$: simplified model.

Radius mm	Radial Stress (MN/m^2)		Error	Tangential Stress (MN/m^2)		Error
	F.E.M.	Exp.		F.E.M.	Exp.	
160	6.643	5.26	0.26	18.816	20.9	0.099
190	2.368	1.74	0.36	12.448	16.4	0.24
222	0.876	0.615	-0.42	9.35	12.26	0.237
246	0.347	-0.55	1.6	8.134	10.9	0.253

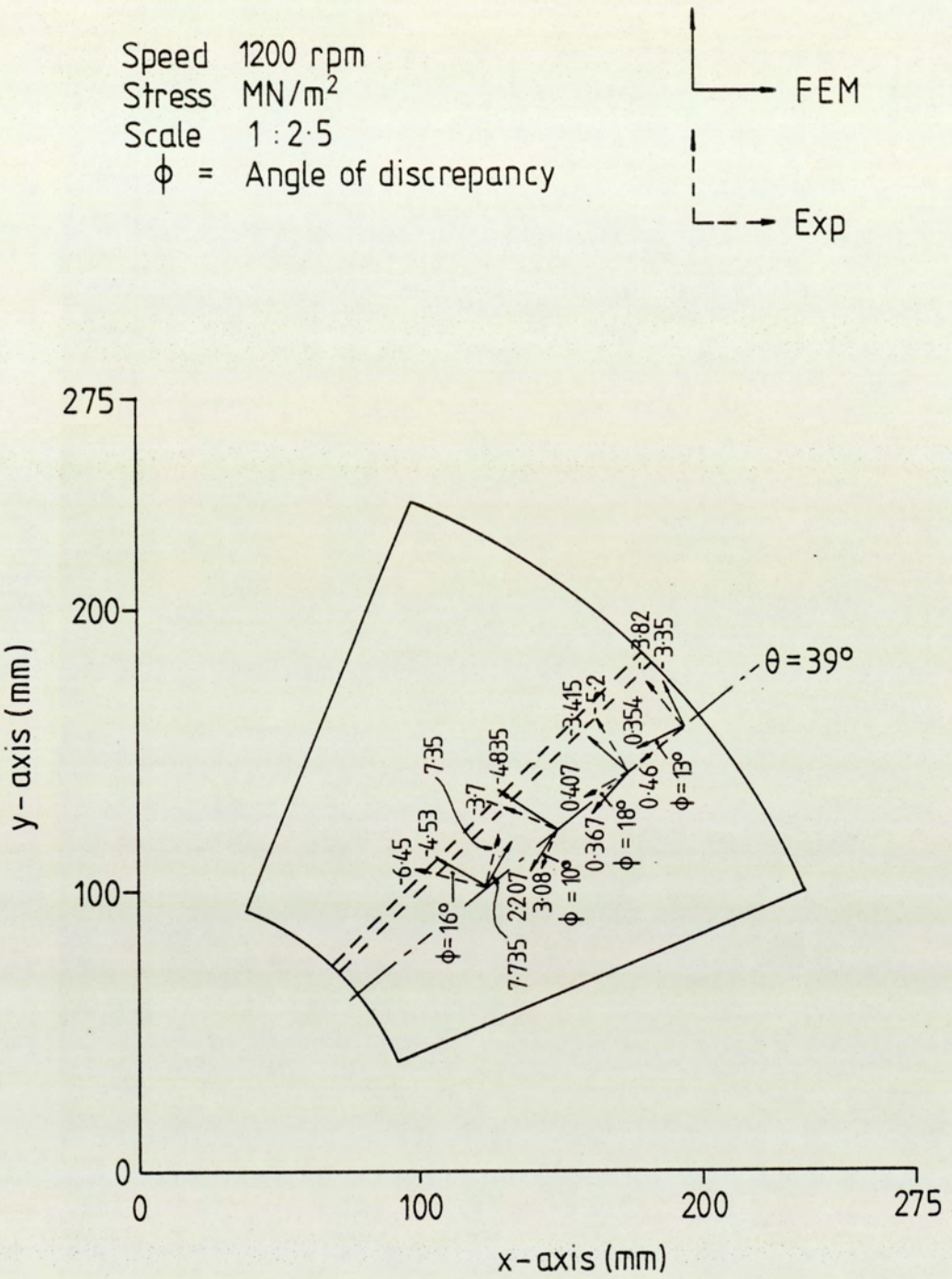


Fig 7.23 Principal stresses and their directions on outside of backsheet along line $\theta = 39°$ for the simplified model.

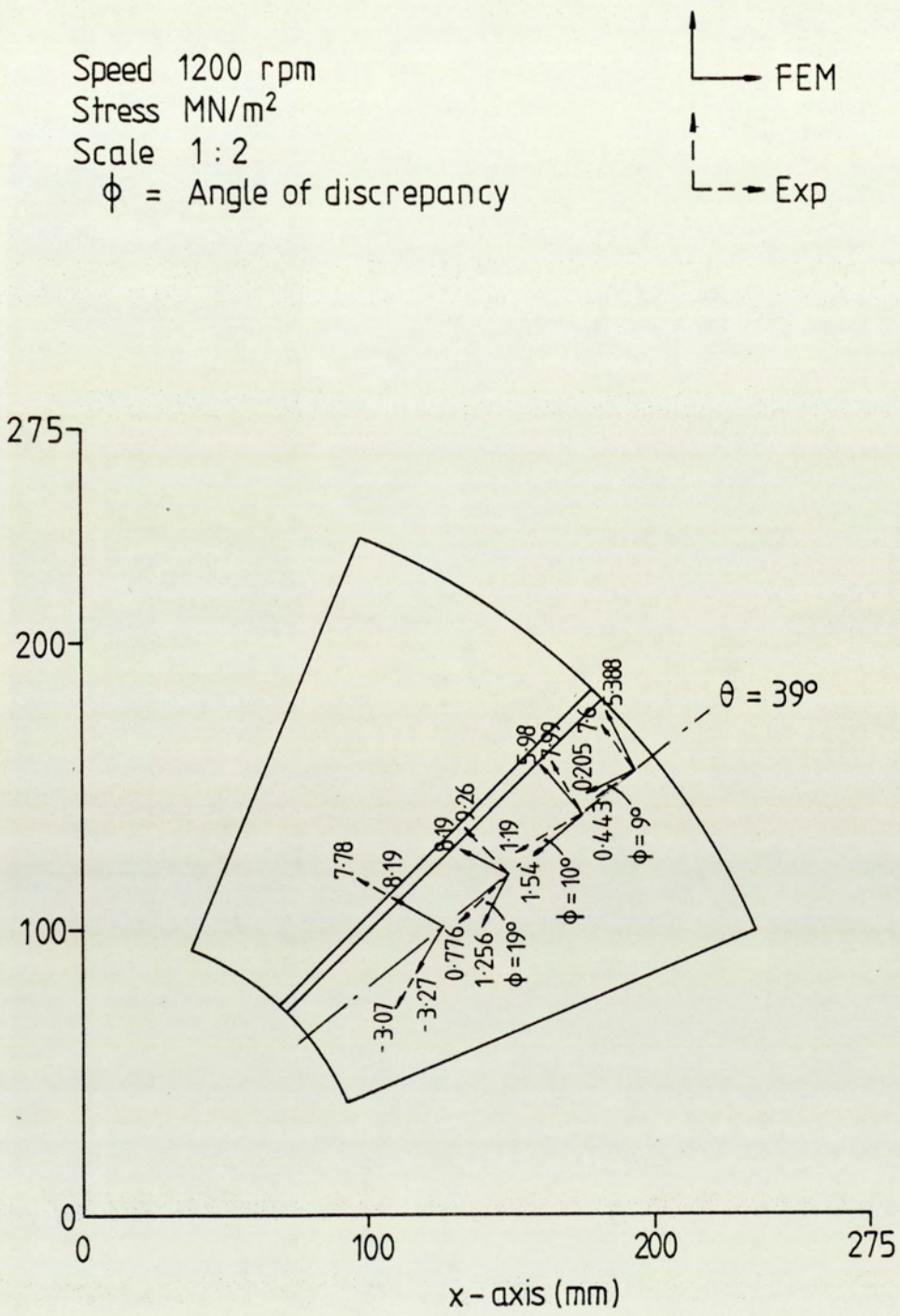


Fig 7.24 Principal stresses and their directions on inside of backsheet along θ=39°

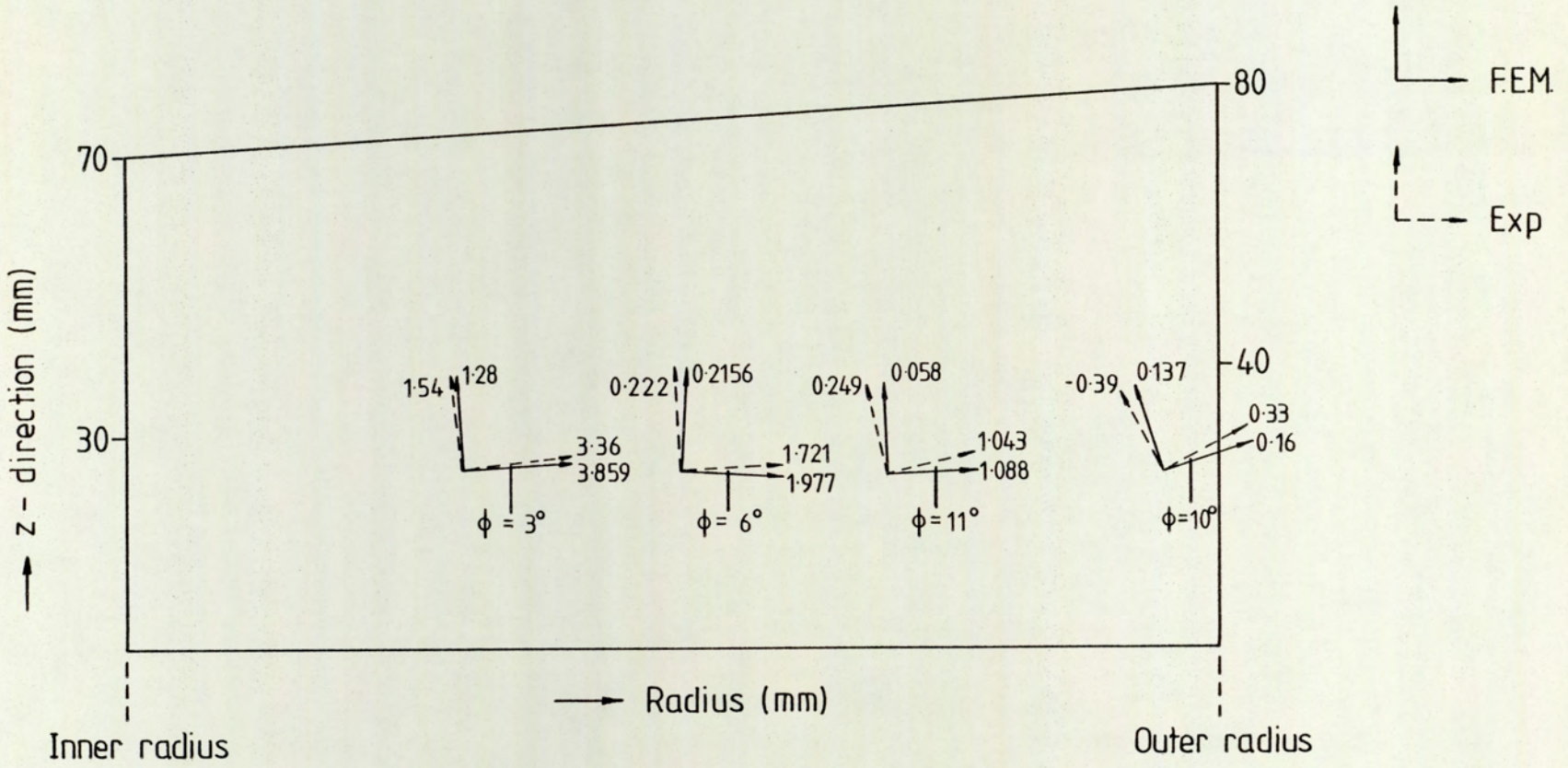


Fig 7.25 Comparison of finite element and experimental results of the principal stresses on inside surface of the blade : simplified model

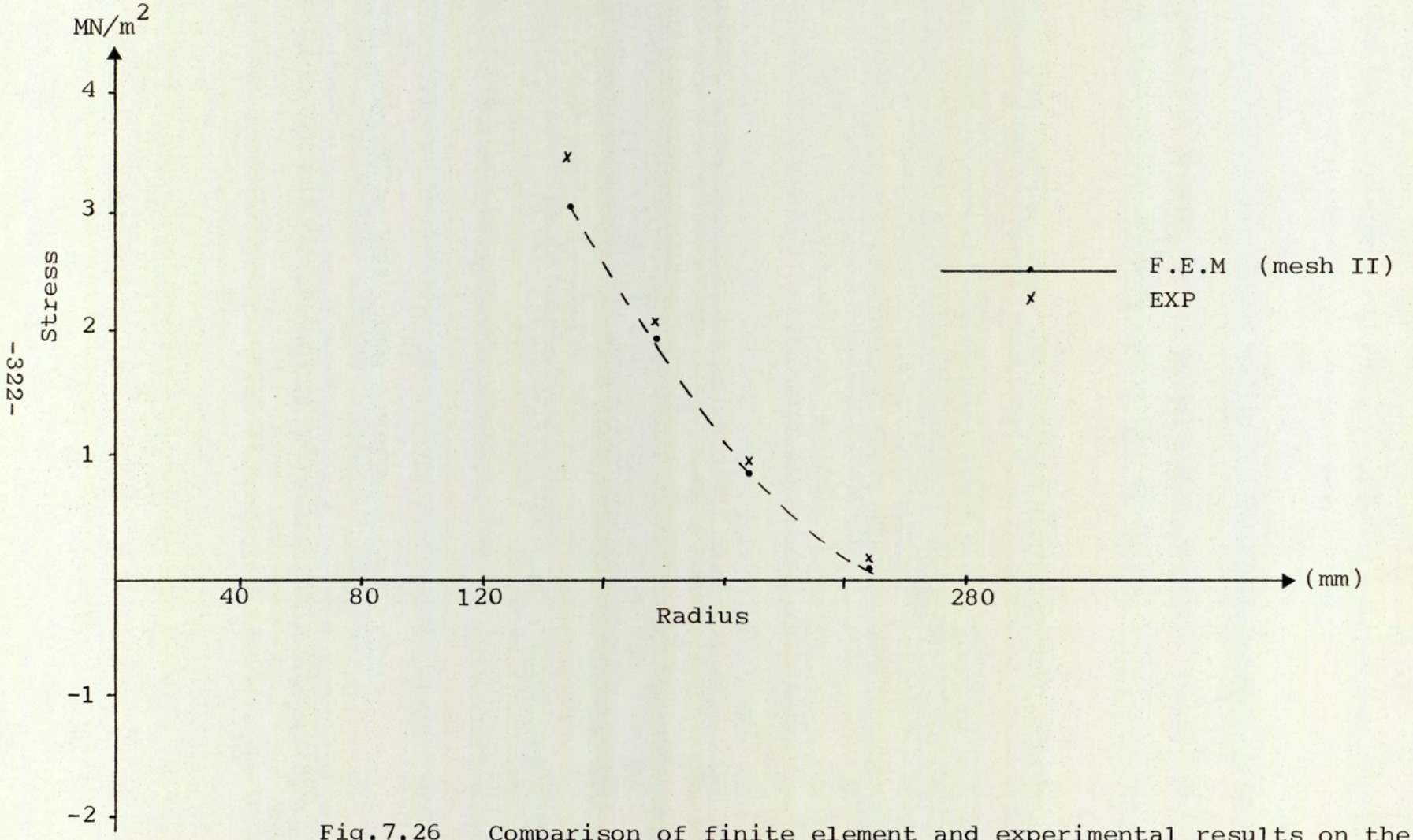


Fig.7.26 Comparison of finite element and experimental results on the inside surface of the blade : simplified model

7.4.4 Comments on the Finite Element Analysis of the Simplified Fan Impeller

Having discussed the correlation between the experimental and the numerical solutions for this impeller, it can be concluded that the modelling of such complicated structures by means of finite element by exploiting the sectorial symmetry technique is a good approach, giving an important step forward to the better stress analysis design of the rotating fan impellers.

The results of the stresses obtained using Mesh III was drawn as a reference solution in order to compare the stress resultants when using Mesh I and II. The results drawn on Figs. 7.27 to 7.30 have shown excellent convergence for the two selected sections on the backsheet i.e. at $\theta=22.5^{o}$ and 37.5^{o}.

As expected, the stress resultants (bending moment and in-plane force/unit length) drawn on Figs. 7.27 to 7.30, decrease as the radius increases and approach zero at the free end. Also, the maximum bending moment occurs at the inner edge near the intersection of the blade with the backsheet. This is due to the action of the blade and the blade welds. To estimate the maximum principal stress in the backsheet, two sections were chosen using the results of mesh II at the gauss points at $\theta=28.5^{o}$ near midway between the blades and at $\theta=43.47^{o}$ near the

blade/backsheet junction.

The principal stresses at the gauss points on line $\theta = 28.5^{o}$ are reasonably radial and tangential for both the inside and outside the backsheet as shown in Figs. 7.31 and 7.32 (except at the inner edge since this edge was assumed to be built in), while the direction and magnitude of the principal stresses at the gauss points on line $\theta = 43.47^{o}$ are altered greatly by the action of the blade as shown in Figs. 7.31 and 7.32. From the results of the principal stresses, shown in Figs. 7.31 and 7.32, on the inside and outside the backsheet, it can be concluded that the bending stresses are dominant, as expected, and on section $\theta = 43.47$ the bending component is many times greater than the membrane one. This is again due to the effect of the blade on the backsheet. The maximum stress occurs near the blade/backsheet junction at the inner edge where the angular distance between the welds of adjacent blades is smaller as mentioned before, and the bending stress becomes significant compared with the membrane stress (for example $\sigma_1^m = 3$ MN/m^2 compared to $\sigma_1^b = \pm 115$ MN/m^2).

The membrane stress resultants obtained from the finite element method using mesh II are in the directions parallel to the blade tip and root as shown on Figs 7.33 and 7.34 respectively. The stresses decrease to zero at the free end, as expected, and the maximum stress in the blade occurs at the blade/backsheet junction.

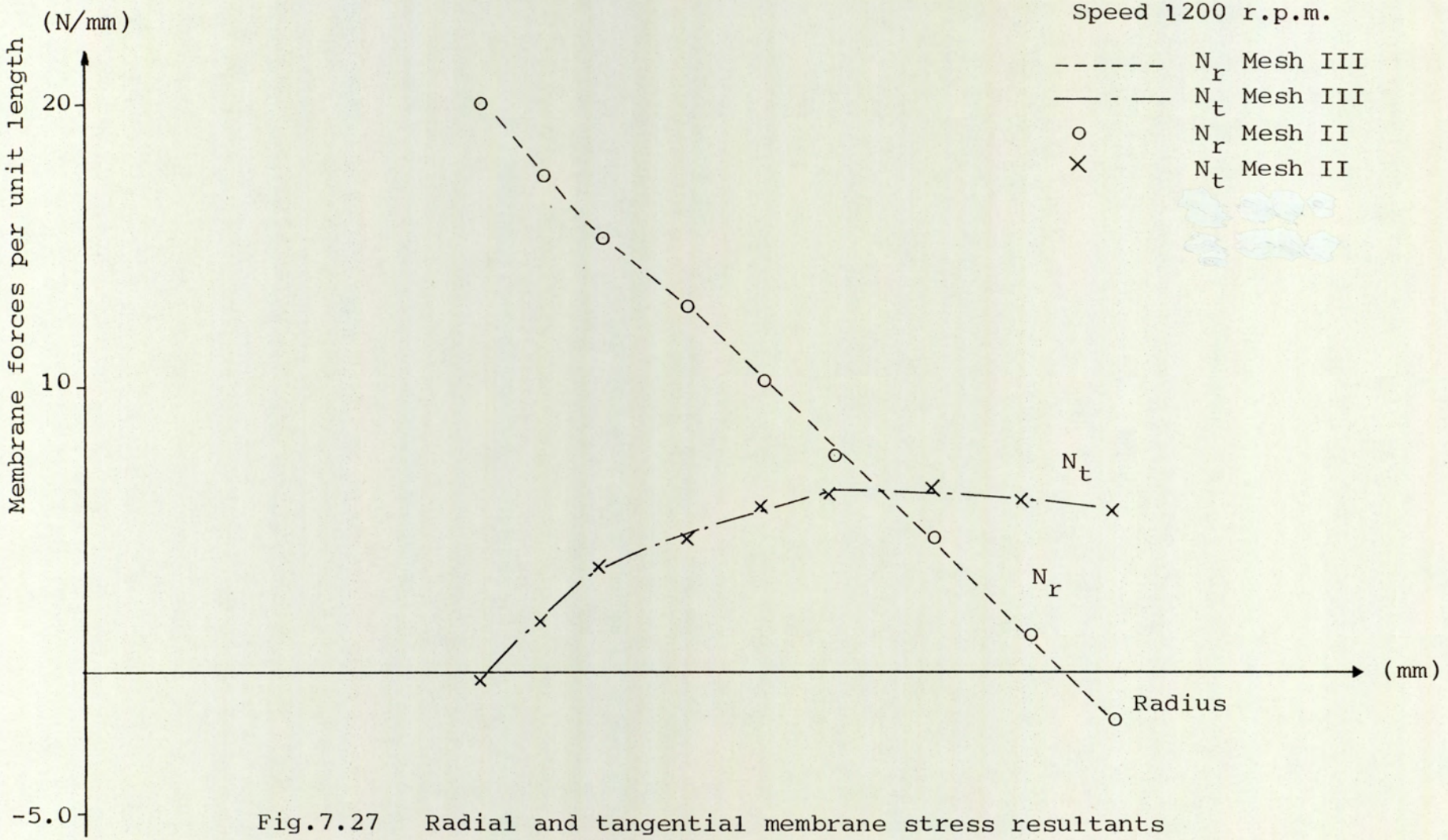


Fig.7.27 Radial and tangential membrane stress resultants for the line = 22.5^{o} : simplified model

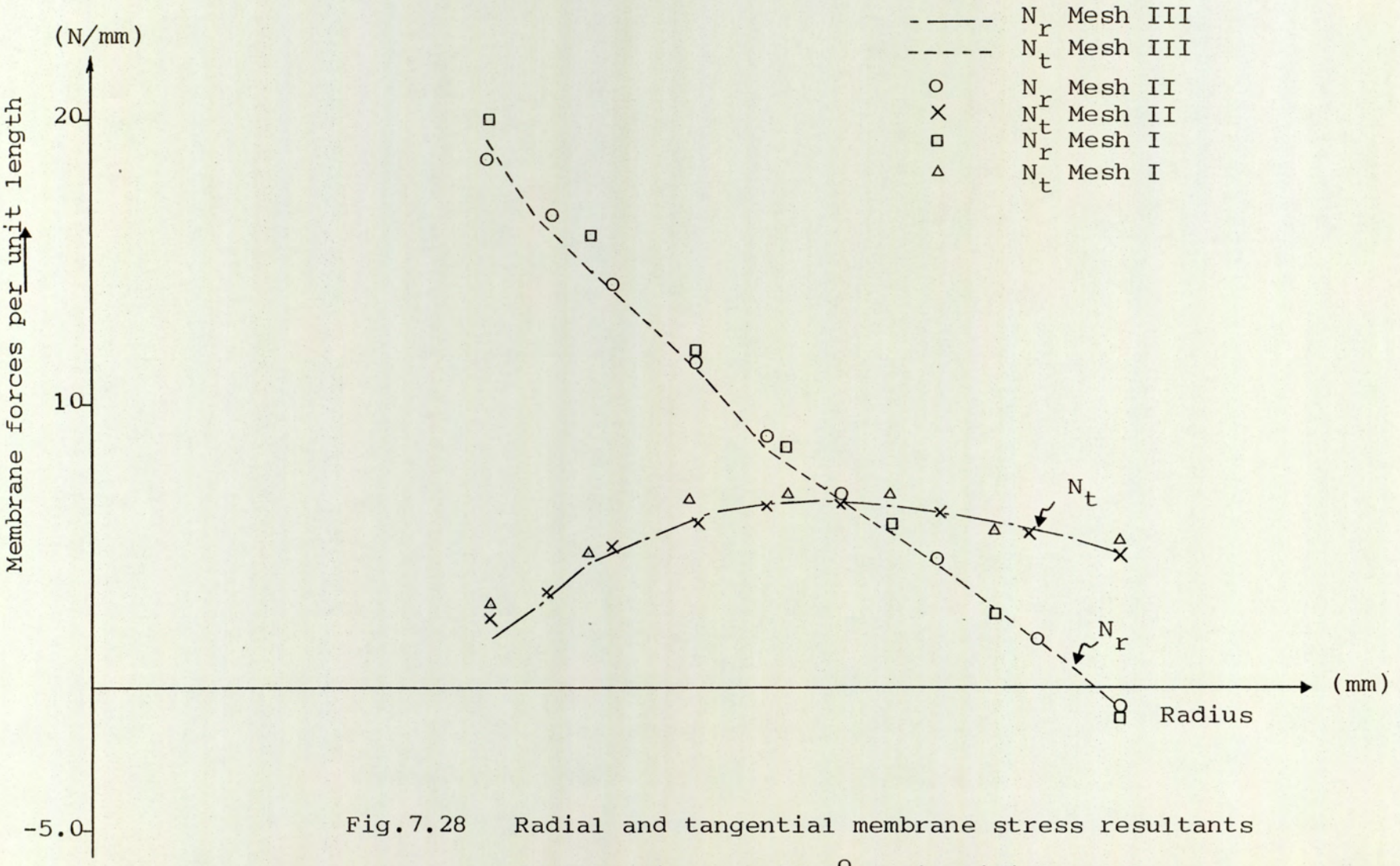


Fig.7.28 Radial and tangential membrane stress resultants for the line = 37.5^{o} : simplified model

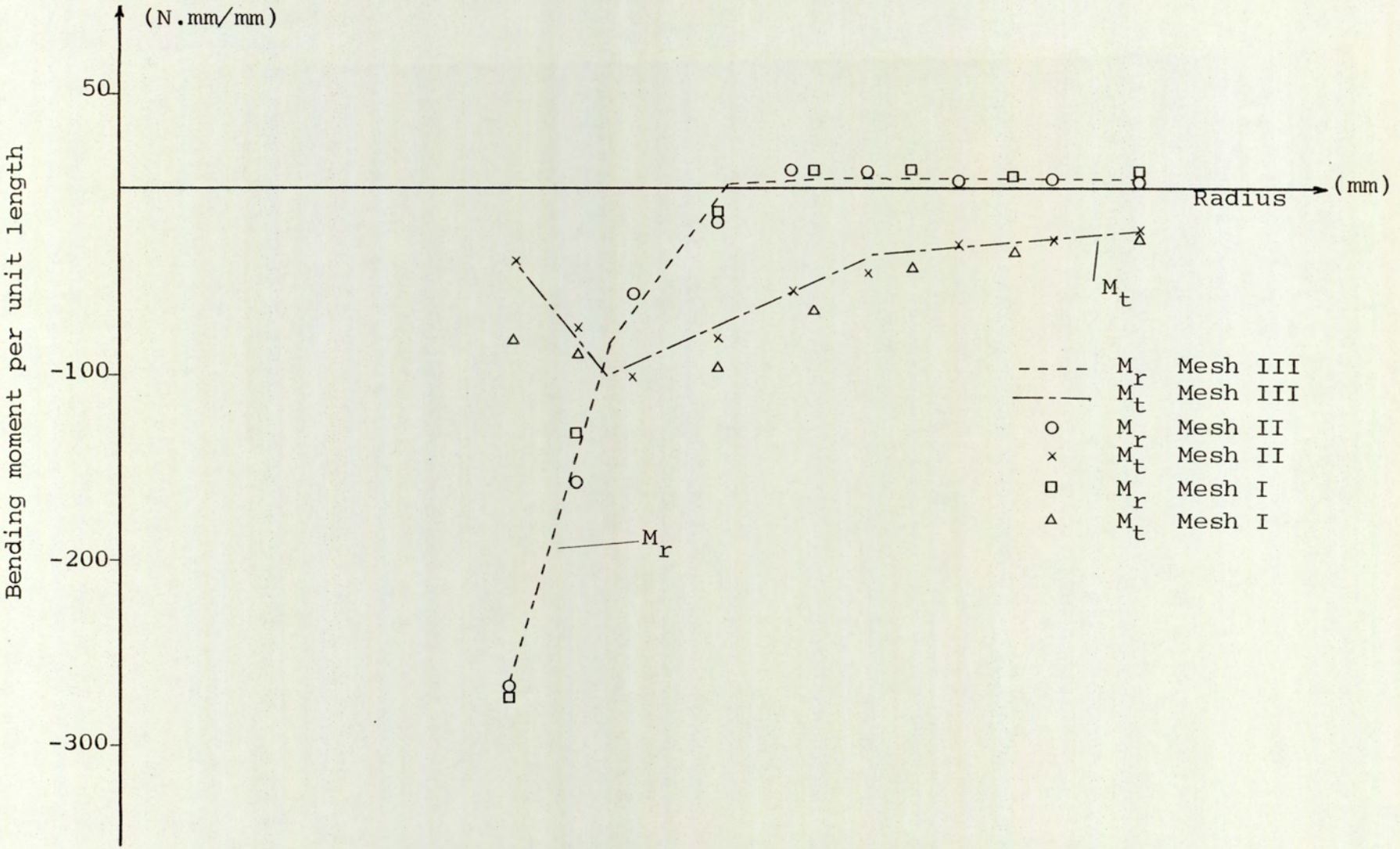


Fig.7.29 Radial and tangential bending moments along the line = 22.5^{o} : simplified model

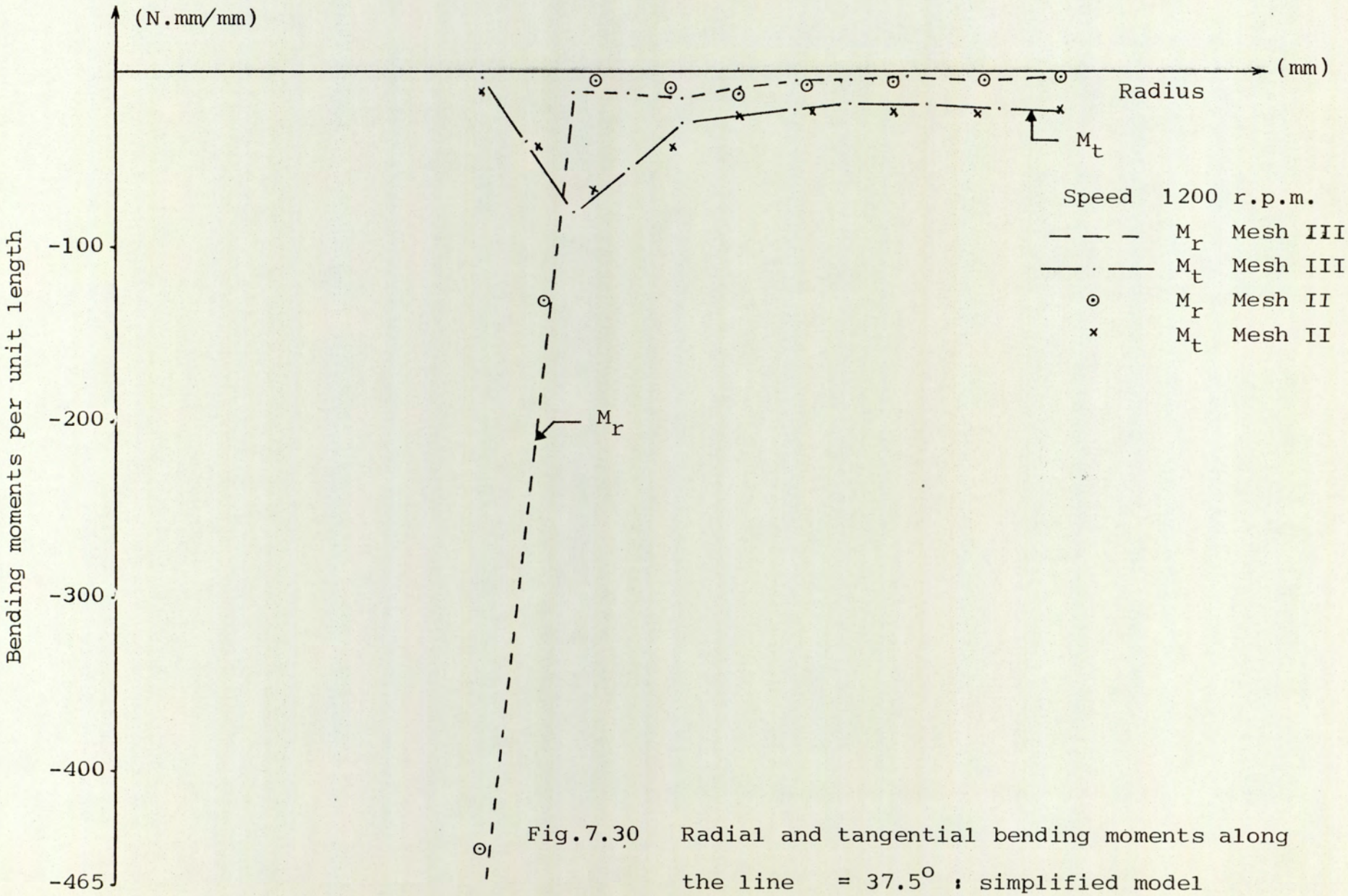


Fig.7.30 Radial and tangential bending moments along the line = 37.5° : simplified model

Finite element mesh II
Speed 1200 rpm
Scale 1 : 2

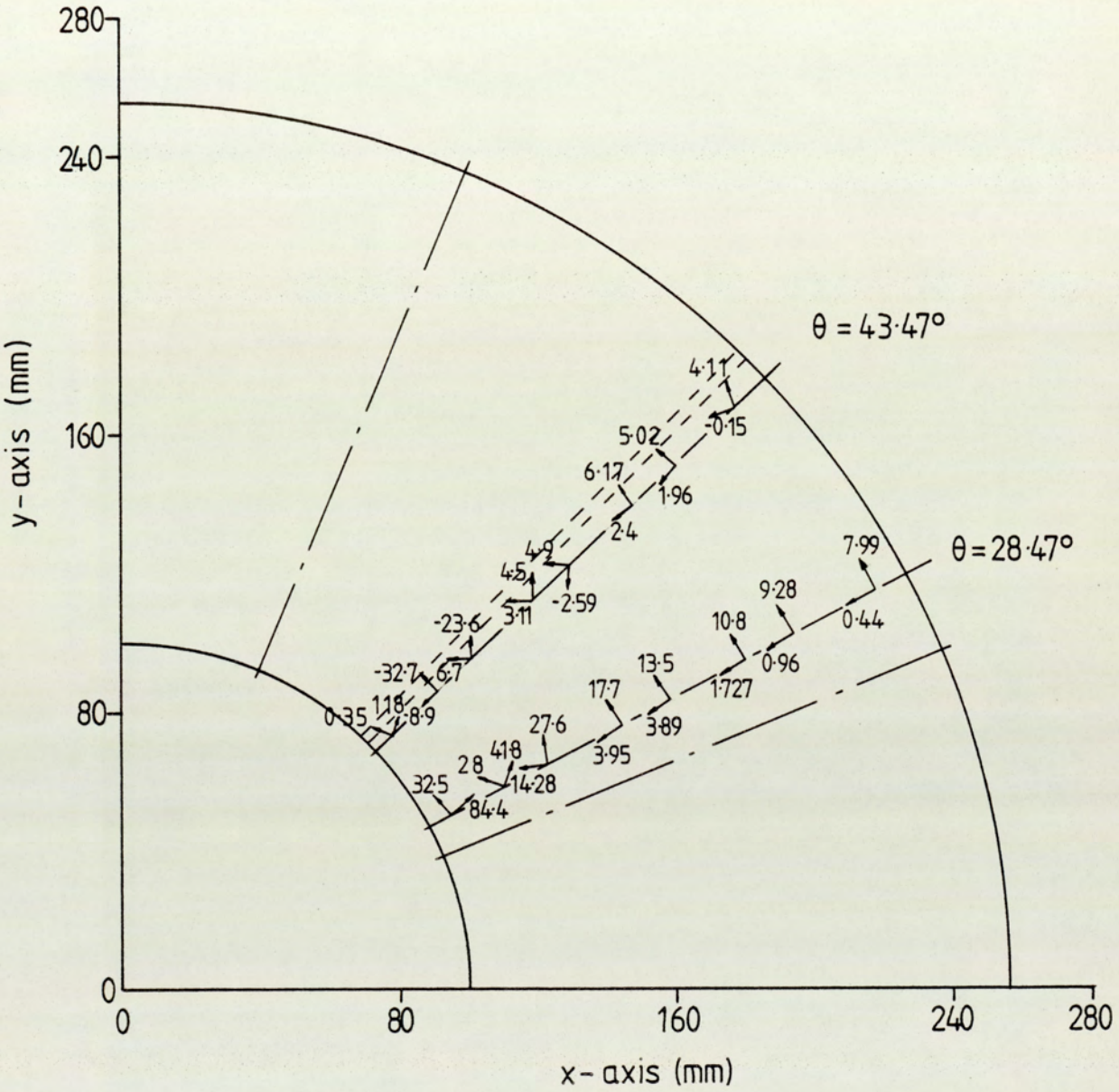


Fig 7.31 Principal stresses and their directions on outside surface of the backsheet : simplified model

Finite element mesh II
Speed 1200 rpm
Scale 1 : 2

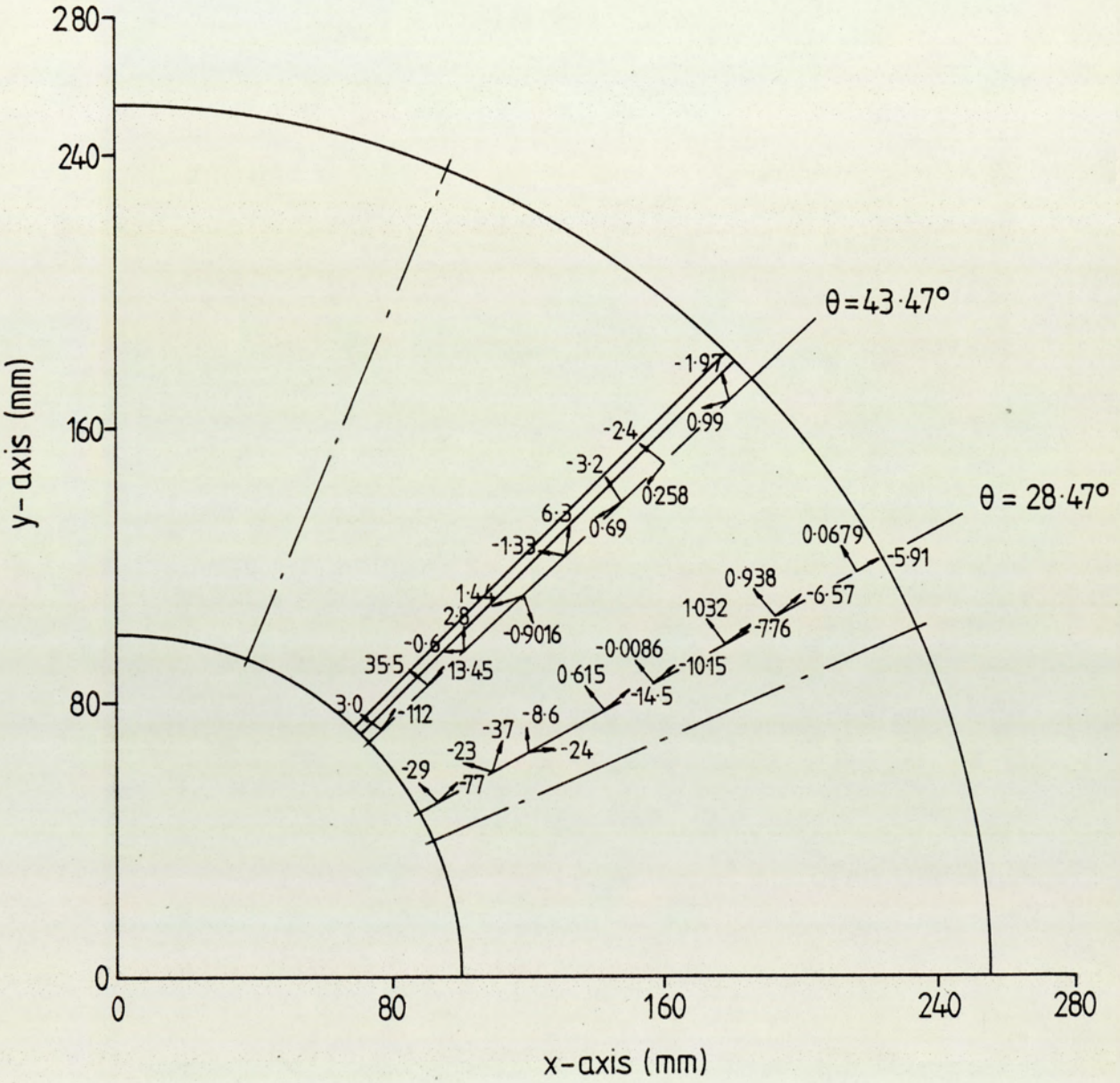


Fig 7.32 Principal stresses and their directions on inside surface of the backsheet : simplified model

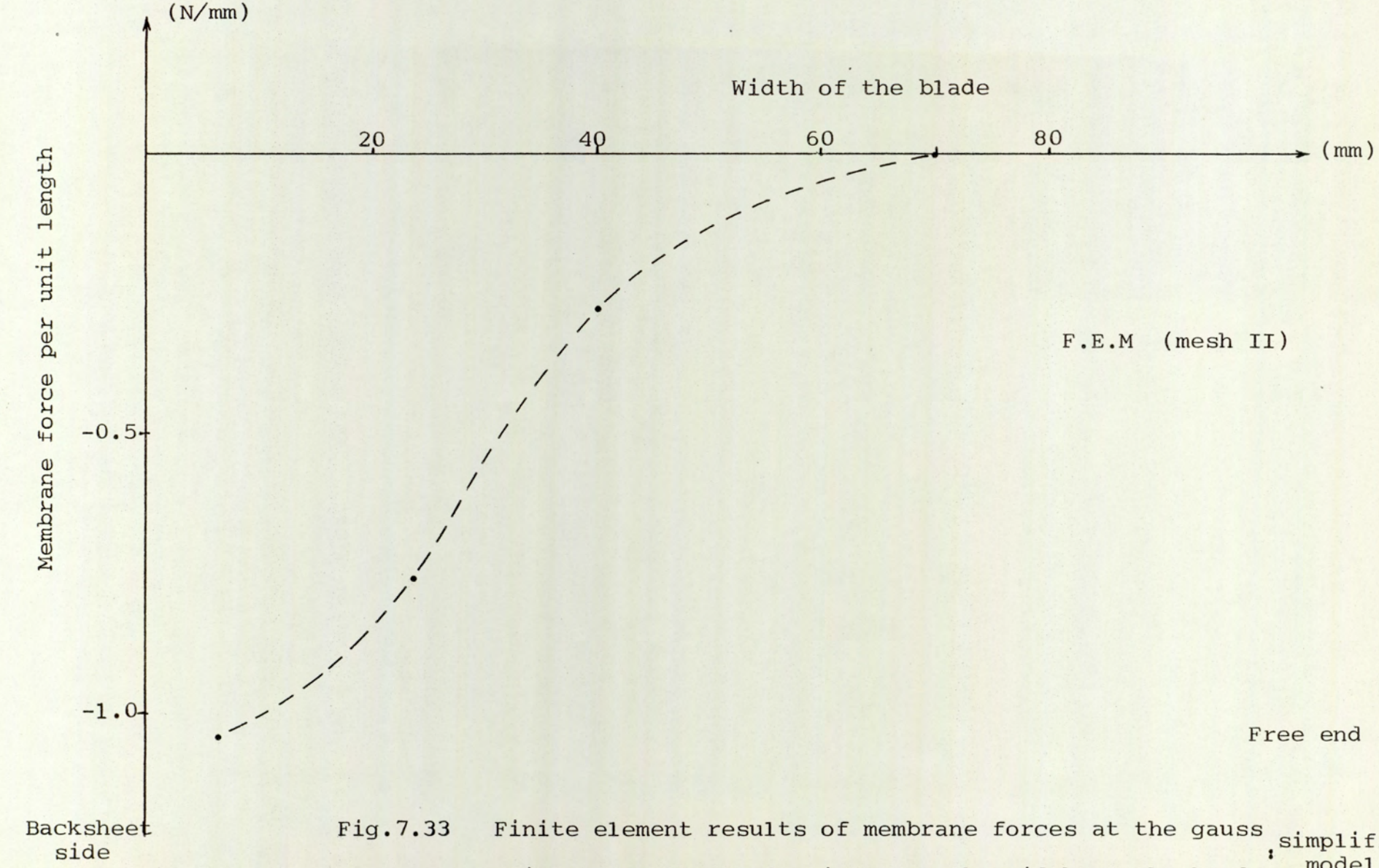


Fig.7.33 Finite element results of membrane forces at the gauss points near the blade tip along the width of the blade ; simplified model

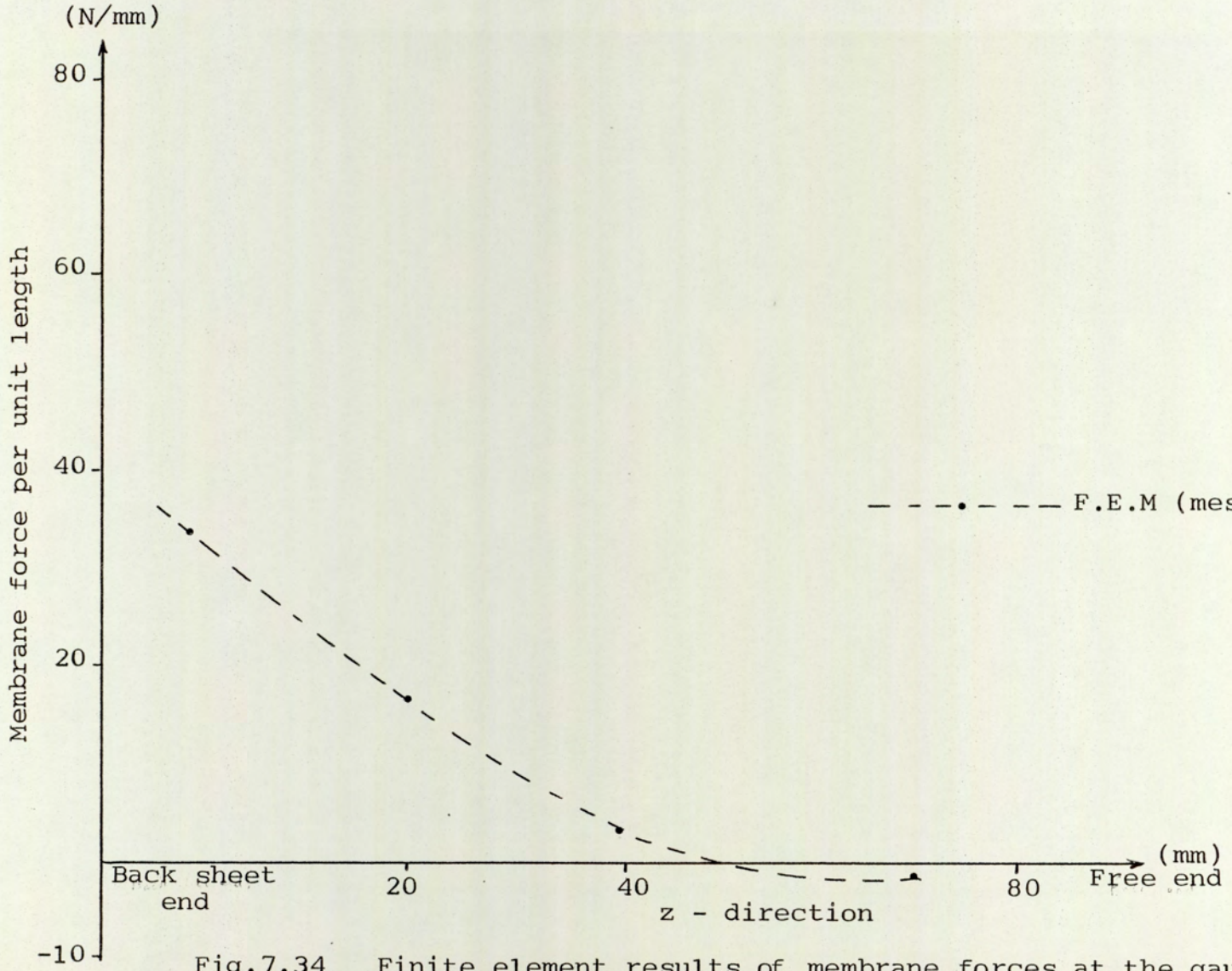


Fig.7.34 Finite element results of membrane forces at the gauss points near the tip along the width of the blade : simplified model

7.5 ACTUAL FAN IMPELLER

The actual fan impeller is the final problem to be analysed numerically and experimentally in this work. The correlation between the results obtained from these techniques will be investigated.

The model chosen was typical of one in commercial production. It was a 650 mm diameter laminar, backward bladed impeller having ten equispaced blades and was balanced statically and dynamically to reduce the vibration: a photograph is shown in Fig. 7.35. The model was mounted on the same rig used for the previous model, except that the shaft used was modified to 100 mm diameter and was supported in bearings on rigid plate structure as shown in Fig. 7.36. The dimensions of the backsheet, blades and the conesheet are shown in Figs. 7.37, 7.38 and 7.39 respectively.

7.5.1 Experimental Investigation

The fan was strain gauged using 45^{o} rosette gauges. The rosette gauges were installed on both the inside and outside surfaces of the backsheet, blade and conesheet. The position of these gauges on the outside surfaces of the backsheet, blade and conesheet are shown in Figs. 7.37, 7.38 and 7.39 respectively, and those on the inside surfaces were in similar positions.

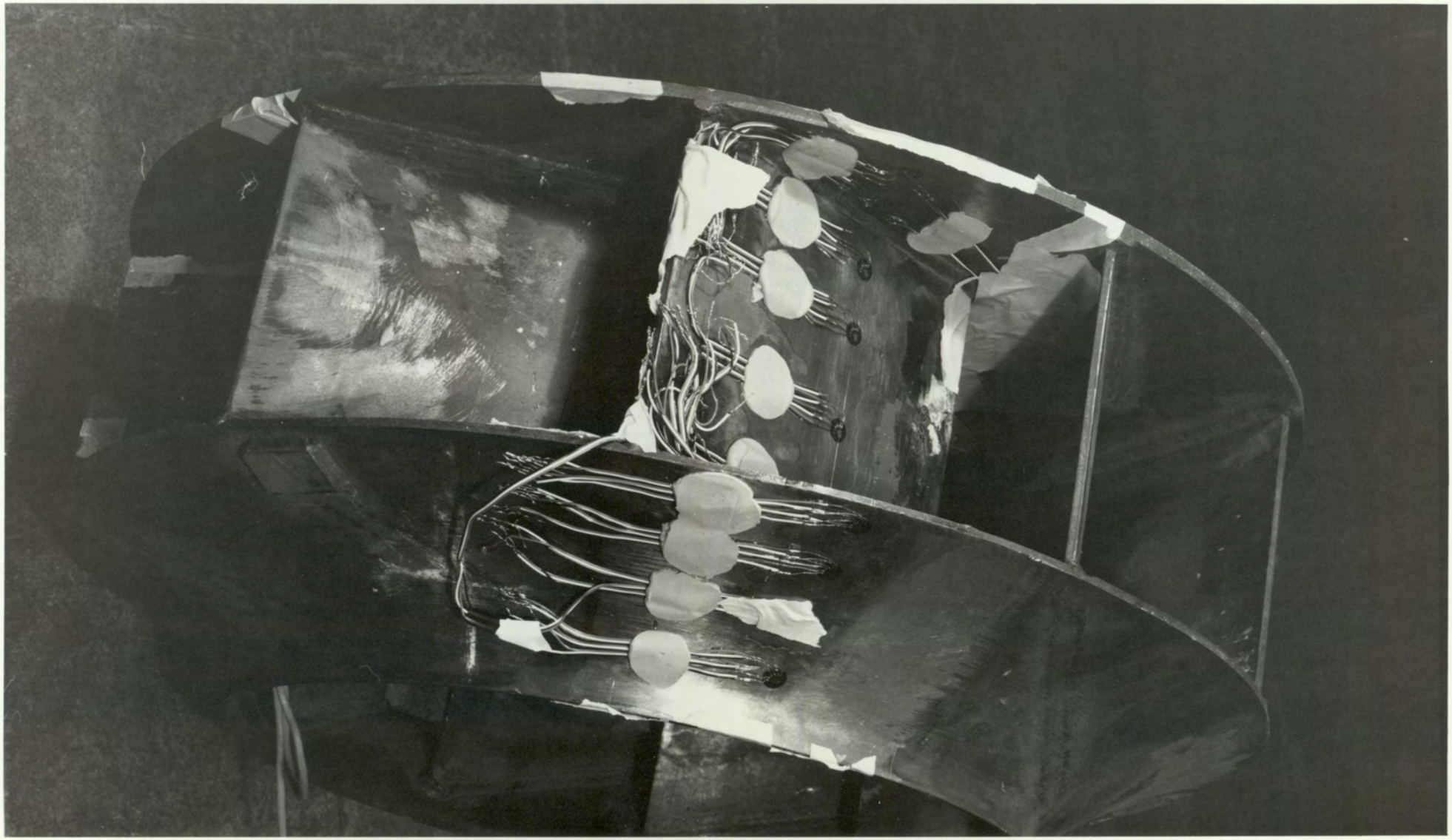


Fig.7.35 Actual fan impeller

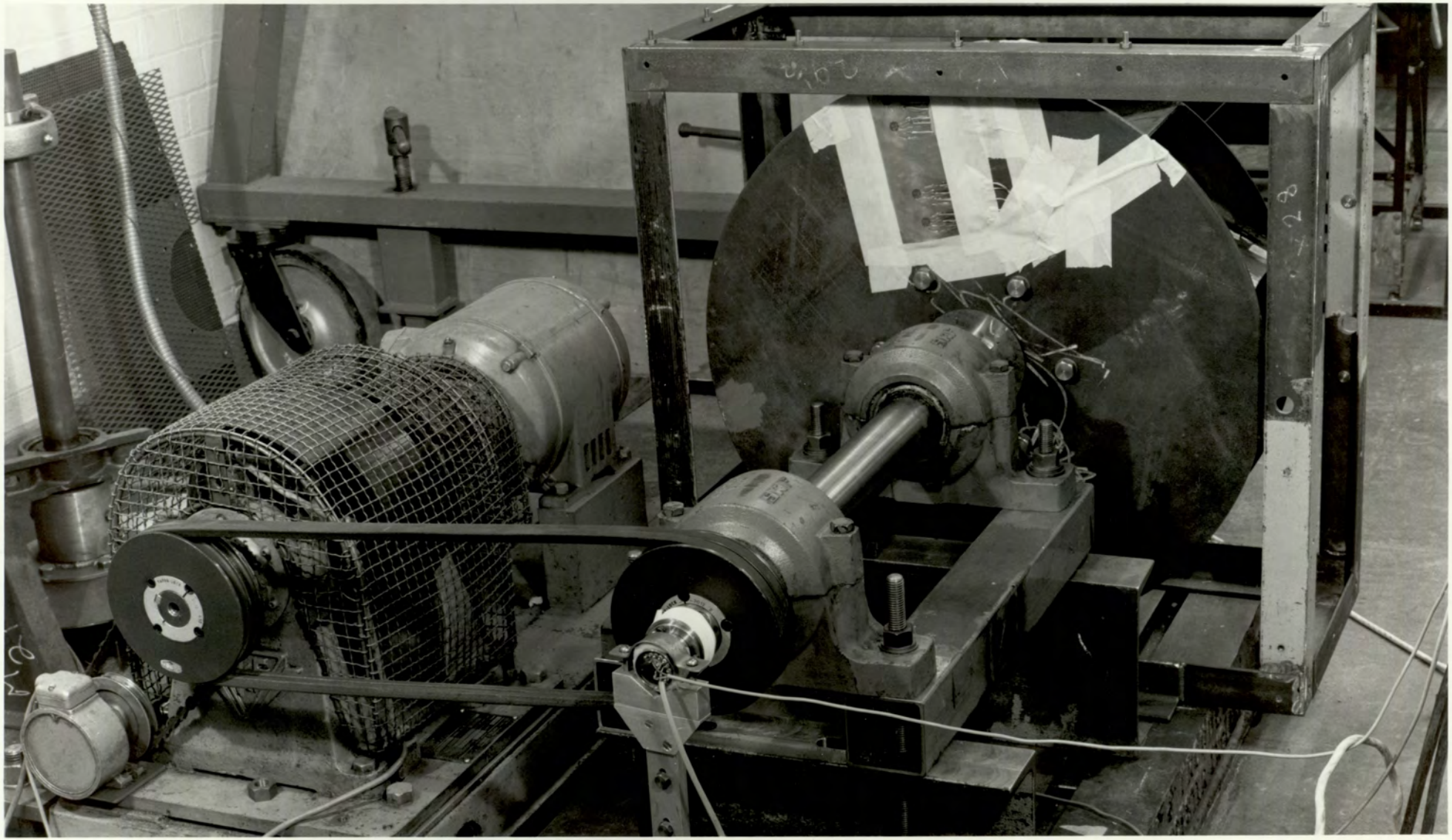


Fig 7.36 Rig used in testing the actual fan impeller

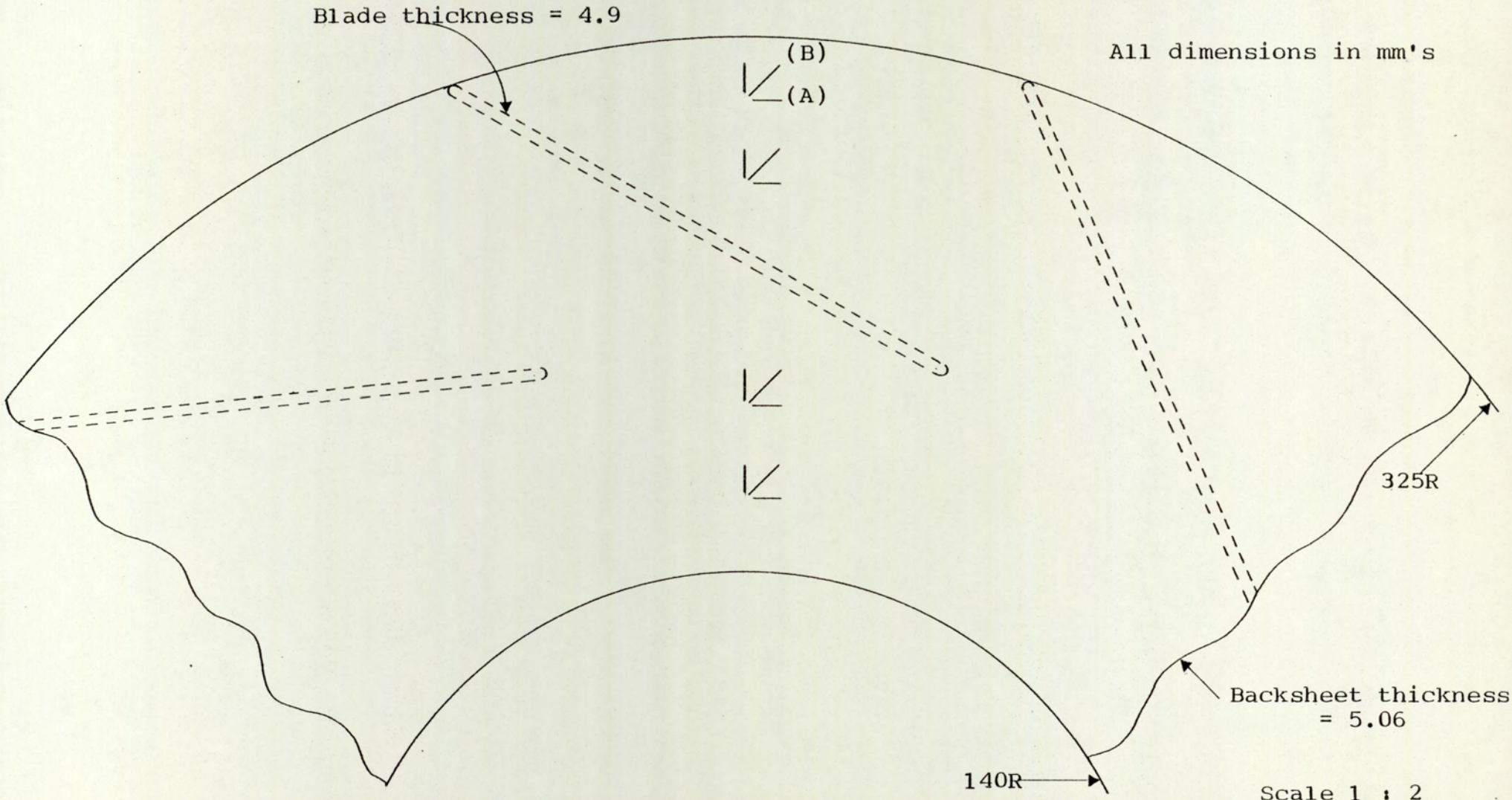


Fig.7.37 Dimensions of the backsheet and strain gauges distribution on the outside surface : actual model

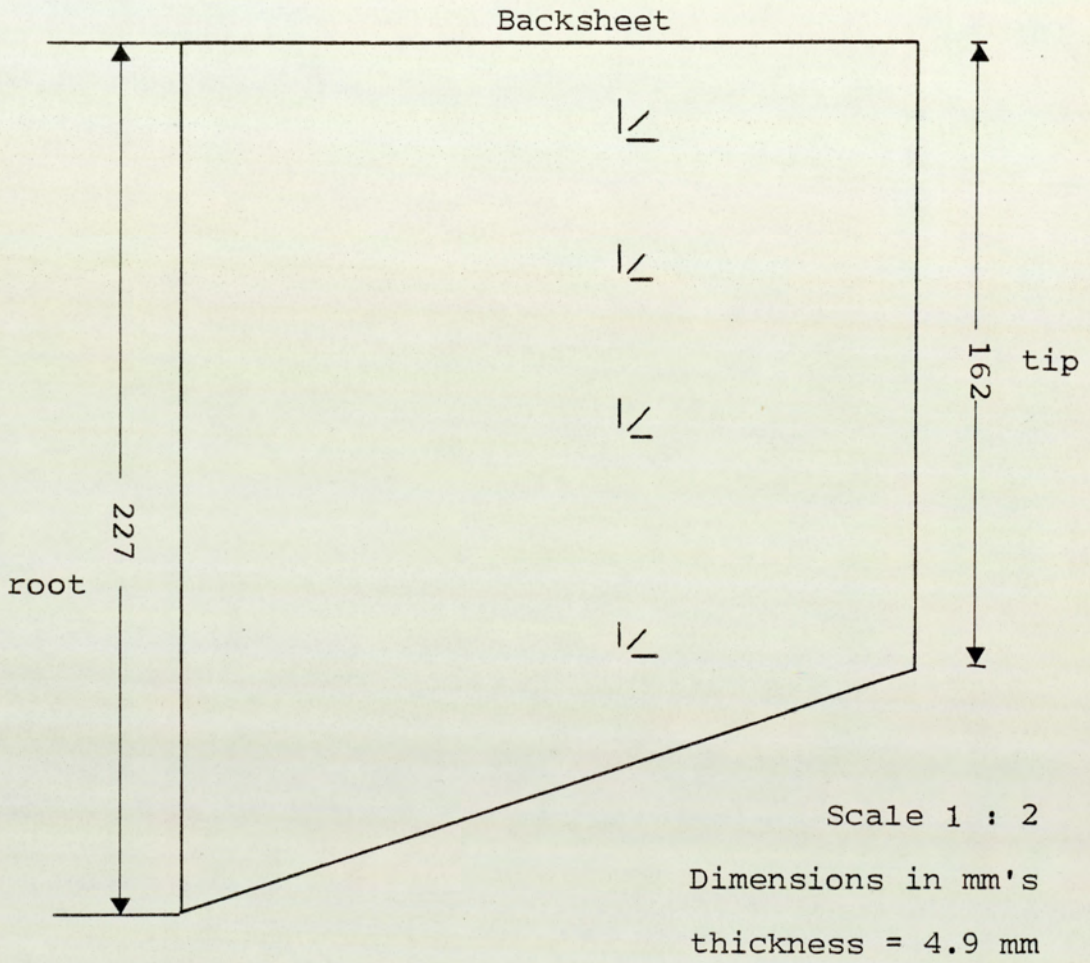


Fig.7.38 Dimensions of the blade and strain gauge distribution on the top surface : actual model

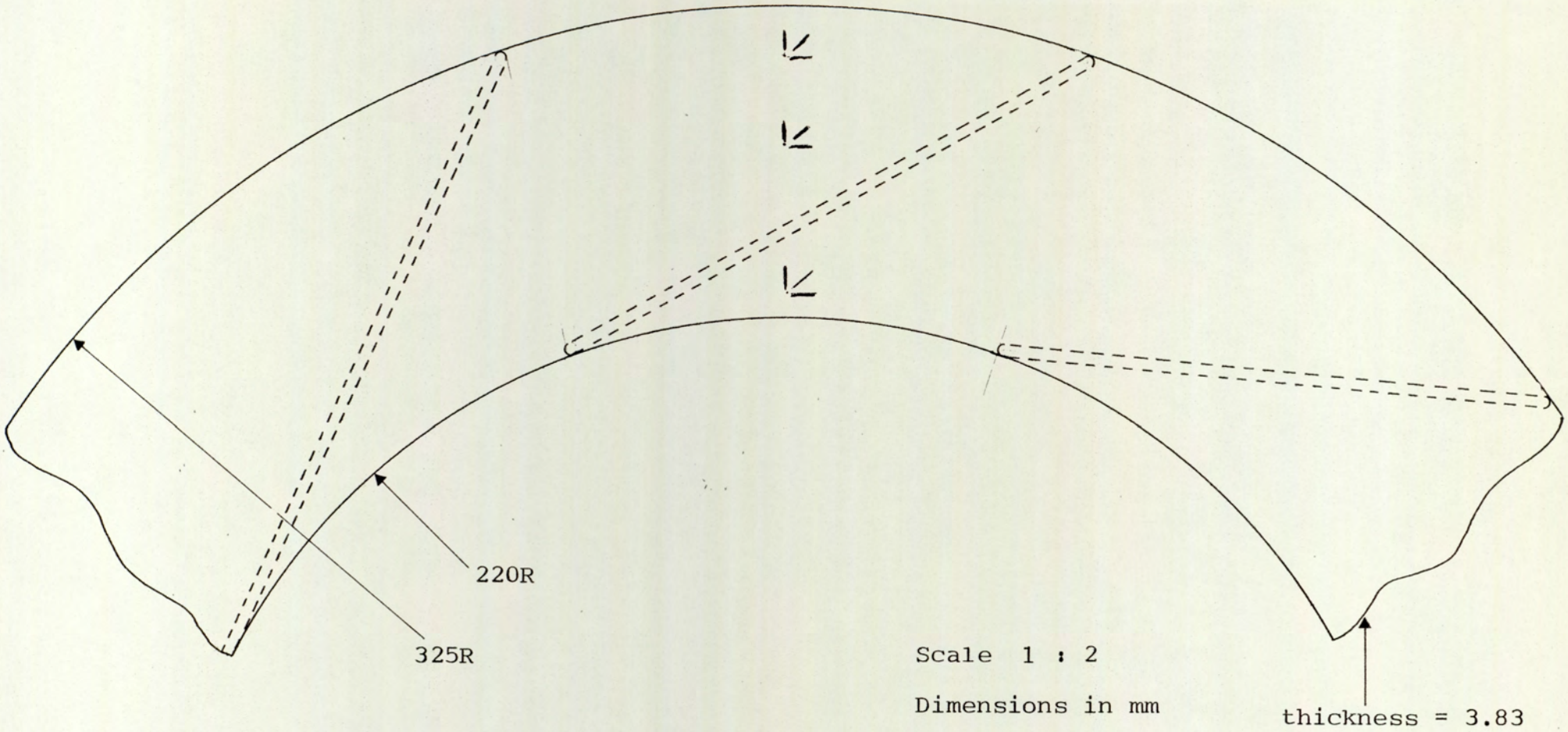


Fig.7.39 Dimensions of the conesheet and strain gauges distribution on the outside surface : actual model

The experimental apparatus described in section 7.4.1 was used to carry out strain gauge tests on this model, but it was not capable of producing enough power to drive this fan at its normally rated speed of 2000 rpm. As the motor speed was increased more than about 1200 rpm it was overloaded, i.e. it took more than the specified current caused the motor to run hot. Therefore, it was decided to carry out the strain gauge tests at speeds ranging from 800-1200 rpm even though it was required that some strain gauge signals could be rather low. For each strain gauge, as shown in Fig. 7.40 for the two typical strain gauges, whose positions are shown in Fig. 7.37, the bridge output voltage was plotted against the speed squared. A straight line was obtained in each case which confirms the expected behaviour of the strain gauges, and thus gave confidence in the measured results. The corresponding strains were calculated from the measured unbalance output voltage at speed of 1000 rpm. by using equation (7.10). The stresses were then calculated using the stress-strain relations.

7.5.2 Finite Element Analysis

Since the actual fan impeller was sectorially symmetric only one tenth of the impeller needed to be analysed. This sector comprises part of the backsheet, one complete blade and a part of the conesheet. A triangular semiloof shell element was used. Once the division was ready,

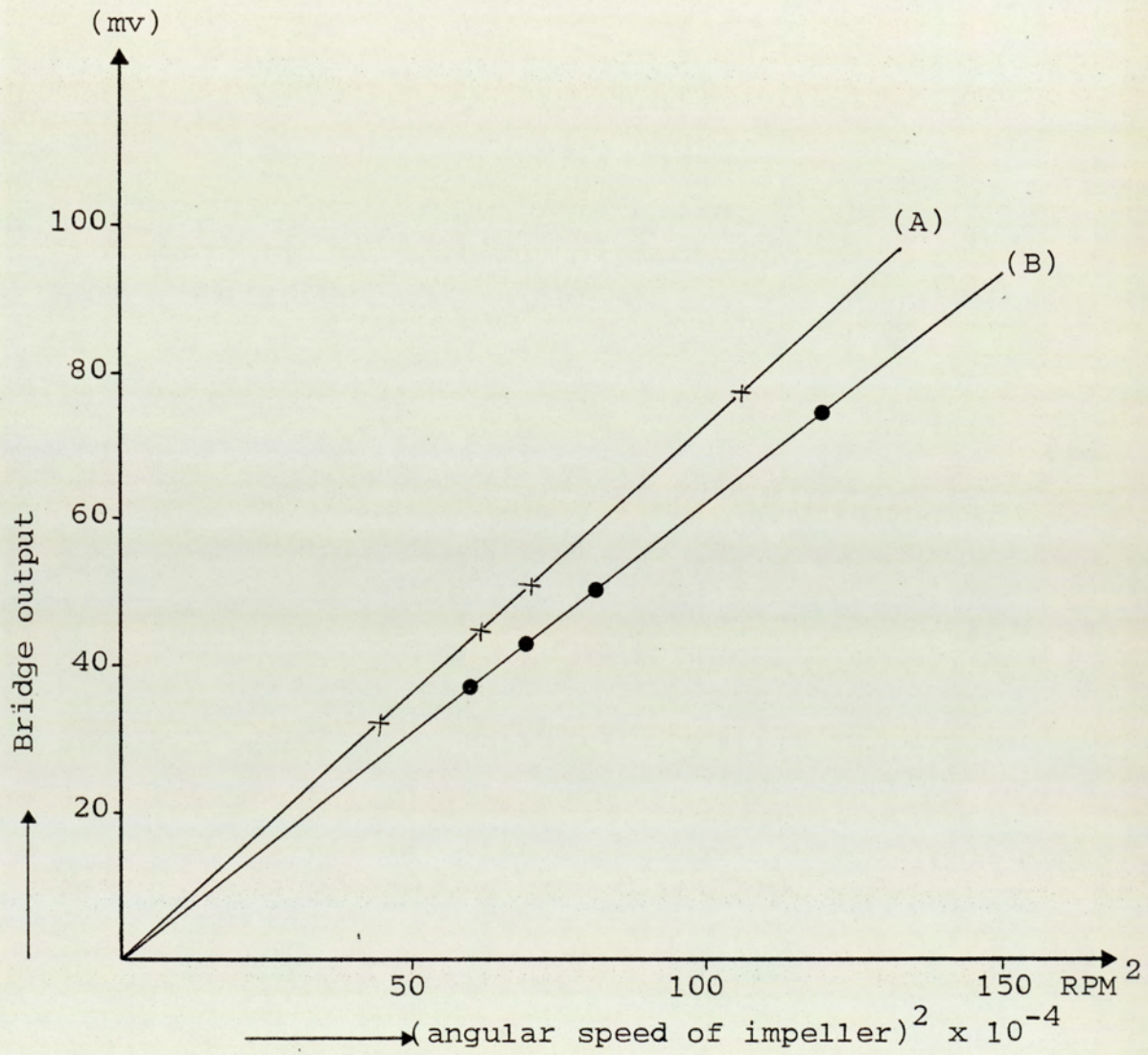


Fig.7.40 Speed squared against the output voltage for two typical strain gauges on the outside of the backsheet of the actual model

the system topology was recorded by numbering the elements consecutively starting from those in the backsheet, then proceeding to the blade and the conesheet.

7.5.2.1 Boundary Conditions

In this analysis, the inner edges were assumed to be built in. The displacements at geometrically corresponding nodes on the sector boundaries (a-a) and (b-b) of Fig. 7.41 were prescribed to be the same. As before, the codes for these nodes were assigned according to the instructions given in Appendix A.

7.5.2.2 Finite Element Meshes

The finite element meshes used to represent the sector of the actual fan impeller were as follows:

(1) Mesh I, shown in Fig. 7.42, in which the fan impeller was discretised into 16, 12 and 8 elements in the backsheet, blade and the conesheet respectively.

(2) Mesh II, shown in Fig. 7.43, in which the fan impeller was discretised into 24, 18 and 18 elements in the backsheet, blade and the conesheet respectively.

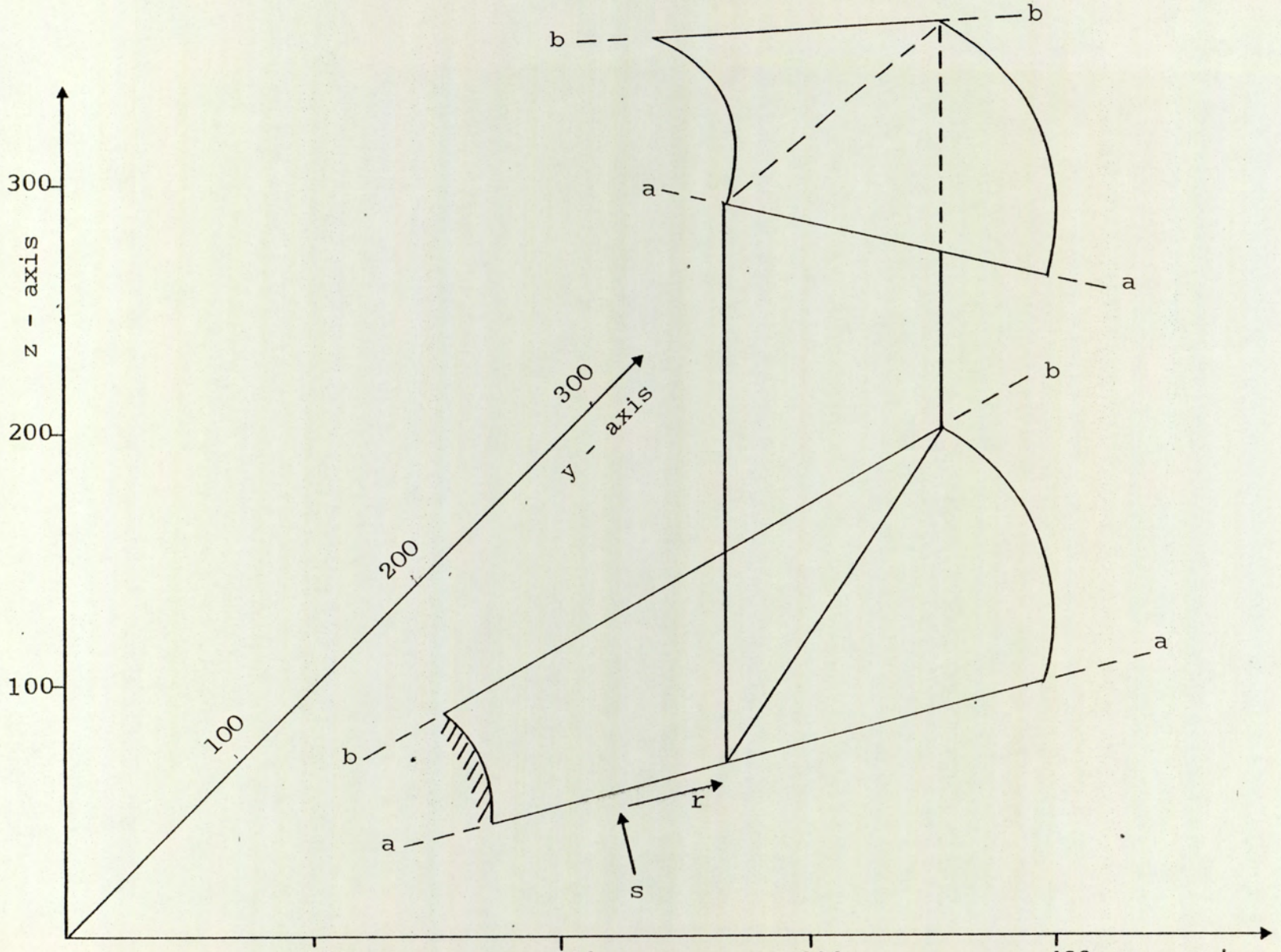
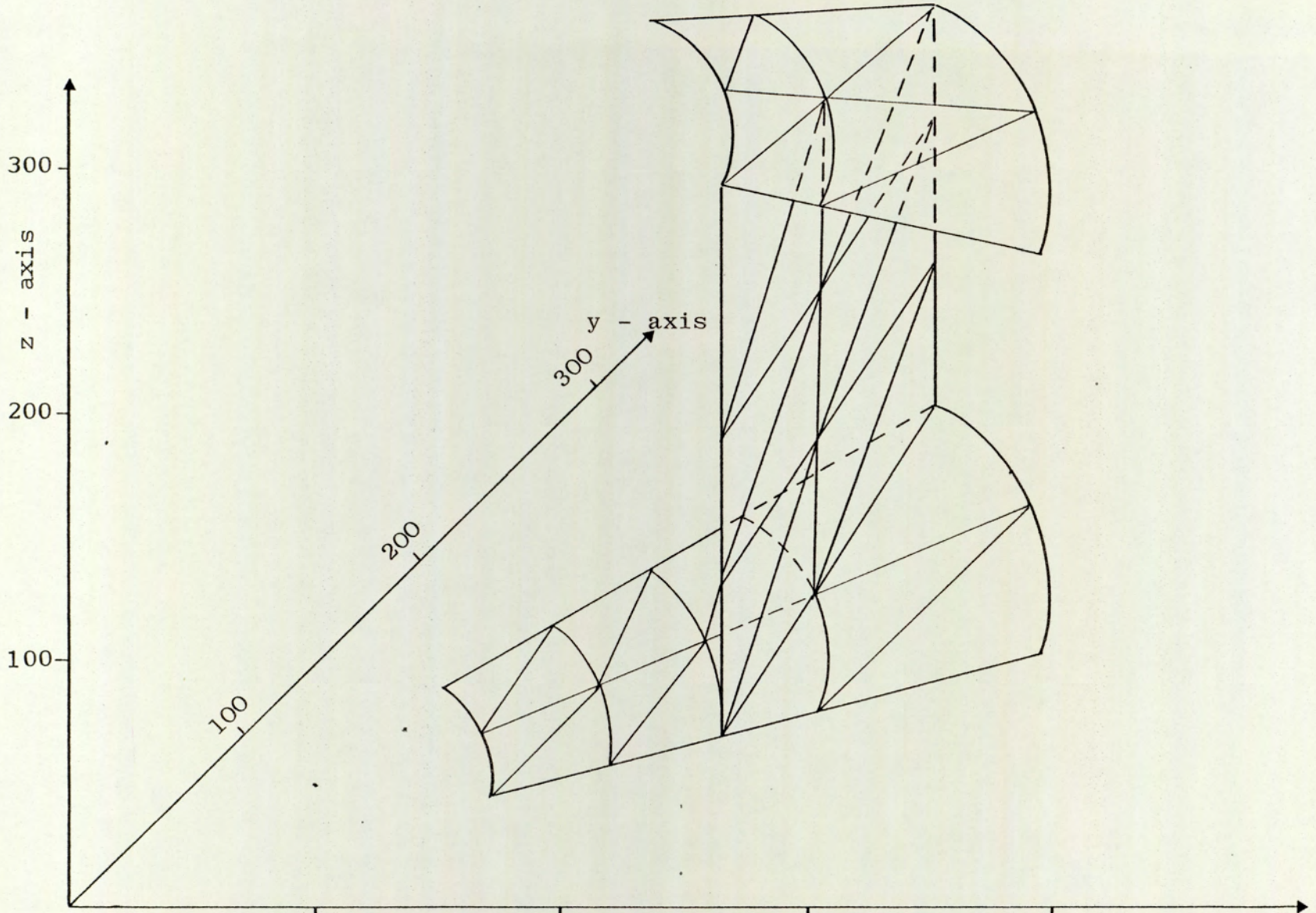


Fig.7.41 A typical sector of the actual fan impeller



Fig,7.42 Finite element mesh I for the actual model

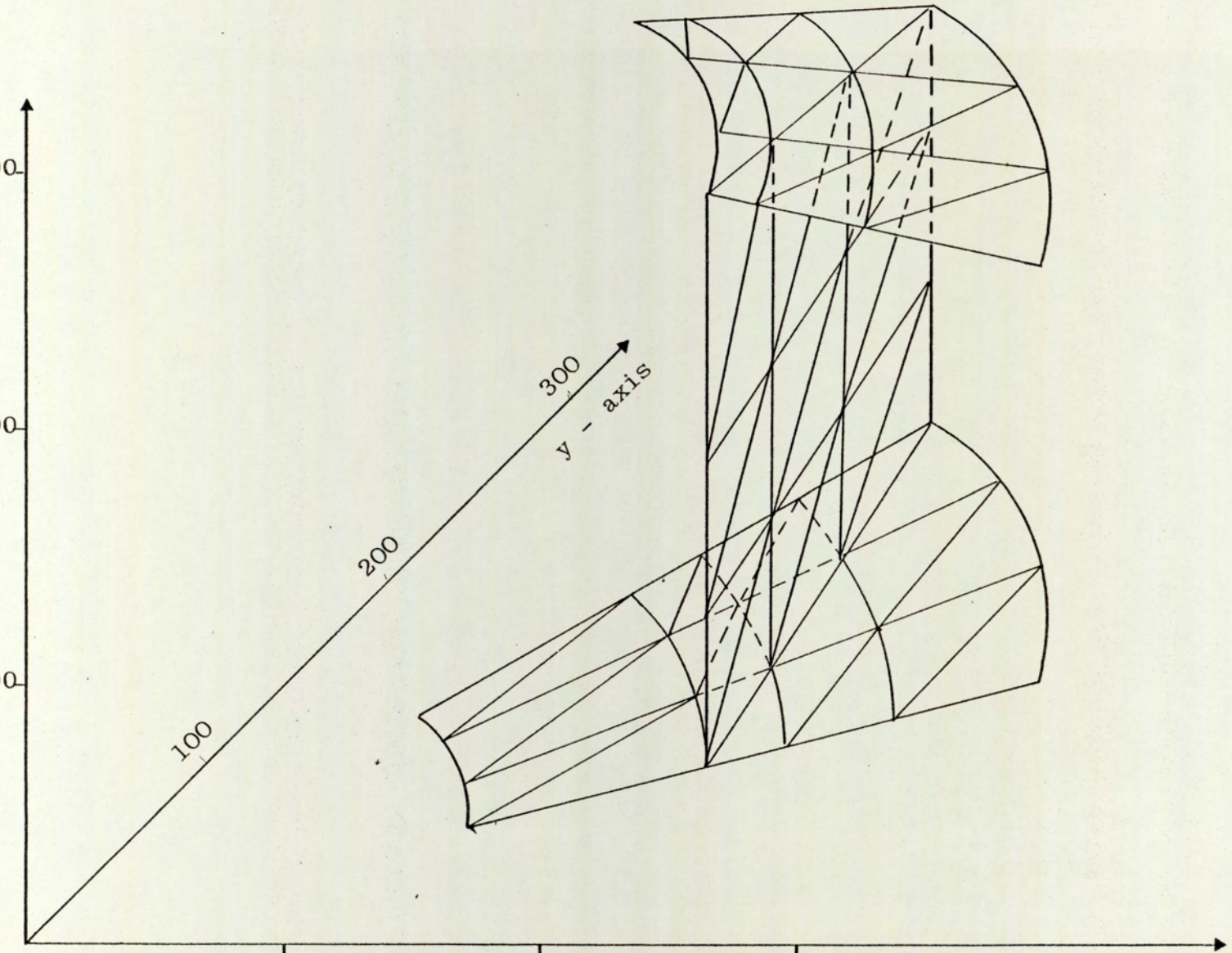


Fig.7.43 Finite element mesh II for the actual model

7.5.3 Discussion of the Results for the Actual Fan Impeller

(a) Comparison of the Results for the Backsheet

It was decided to present the results on the line of strain gauges on the backsheet in the form of radial and tangential stresses on the inside and outside surfaces as shown in Fig. 7.44 and 7.45. Generally, reasonable agreement was obtained between the experimental and the numerical results.

If the strain gauge positioned at R = 267 mm is considered for direct comparison, a good agreement between the calculated stresses from the measured strains and those obtained from the finite element solution is noticed. These results are as follows:

	Exp Tangential	Exp Radial
Outside the backsheet →	10.97	13.67
F.E.M.	13.315	18.9
Inside the backsheet →	-8.75	-10.7
F.E.M.	-9.1	-14.3

In general, the investigation of the radial and tangential stresses on the inner and outer surfaces of the backsheet for the line of strain gauges indicates that the stress field is very complex and the bending stresses are dominant.

The finite element results for the inside and outside of the backsheet at the element centroids, using meshes I and II, are shown in Figs. 7.46 to 7.49. The calculated principal stresses from the measured strains are also shown on these figures. As expected, the direction of the principal stresses on the areas towards the outer radius (i.e. in between the blades) are reasonably radial and tangential while those near the blade/backsheet junction are altered because of the very complex interaction of the blade with the backsheet. The results also indicate, that the bending stresses are dominant and the highest stresses occur just below the blades. This was expected due to the blade effect and blade welds on the backsheet.

(b) Comparison of the Results for the Blade

The calculated stresses from the measured strains on the top and bottom surfaces of the blade are presented in the form of the longitudinal and transverse stresses. They compare well with those obtained from the finite element meshes of Figs. 7.42 and 7.43. The comparisons areshown in Figs. 7.50 to 7.53. On examining the experimental and numerical stress distribution on the top and bottom surfaces of the blade, it can be seen the bending stresses are dominant, and the maximum bending stress occurs at the centre of the blade. Also, the stress

distribution at the blade/backsheet is higher than that at the blade/conesheet junction indicating that the backsheet applies a greater restraining moment than the flexible conesheet.

(c) Comparison of the Results for the Conesheet

The strain gauge rosettes were installed on the inside and outside conesheet surfaces. It was not possible to install more than three gauges on the considered line particularly on the inside of the conesheet, because of the cramped space making the soldering of the leads impossible. As for the backsheet, the results of the principal stresses on the inside and outside the conesheet were presented in the form shown in Figs 7.54 and 7.55.

From both techniques, the direction of the principal stresses away from the blade are reasonably radial and tangential. Qualitative comparison, shown in Figs. 7.54 and 7.55, between the experimental and the numerical results has shown discrepancies, especially for the strain gauge positioned below the blade.

This was expected due to the presence of weld material which becomes significant at the blade/thin conesheet junction. Examining the finite element results of Figs. 7.54 and 7.55, it can be seen that the bending stress is higher than the membrane stress near the blade/conesheet

junction (for example, the estimated stresses at point A, the σ_1^m is 11.15 MN/m^2 , while the σ_1^b is 23.35 MN/m^2 and σ_2^m is 0.835 MN/m^2 while σ_2^b is 1.525 MN/m^2). This was expected again, because of the action of the blade on the thin sheet from which the conesheet was constructed. Although, the bending stresses are dominant, membrane stresses are significant and in some areas (towards the outer radius) are of the same order or higher than the bending stresses.

It is interesting to note the radial stresses (reasonably radial) at the free end and away from the blade/conesheet junction tend to be zero which gave confidence in the numerical technique in the analysis of such complicated components.

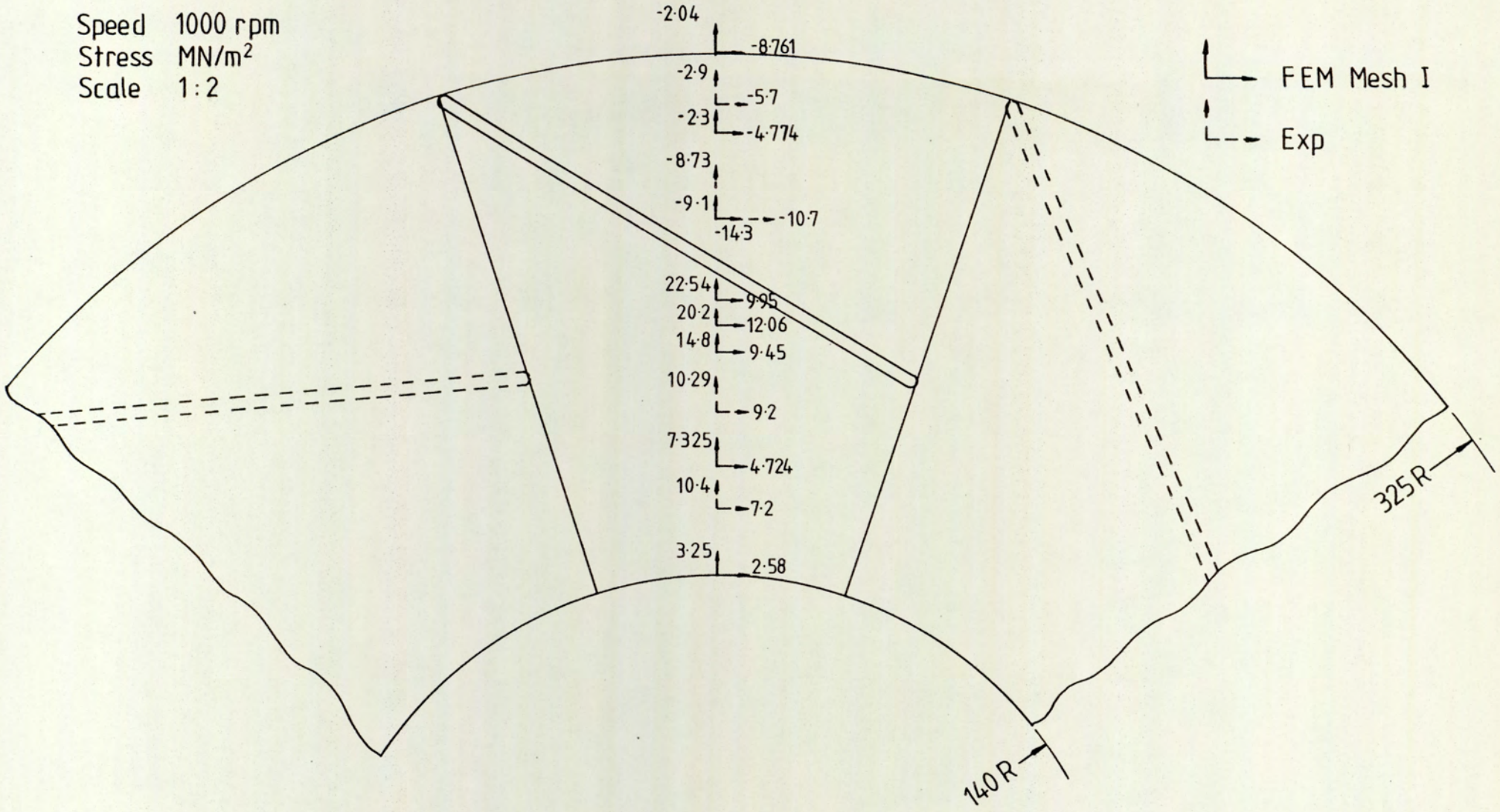


Fig 7.44 Radial and tangential stresses on inside surface of the backsheet : actual model

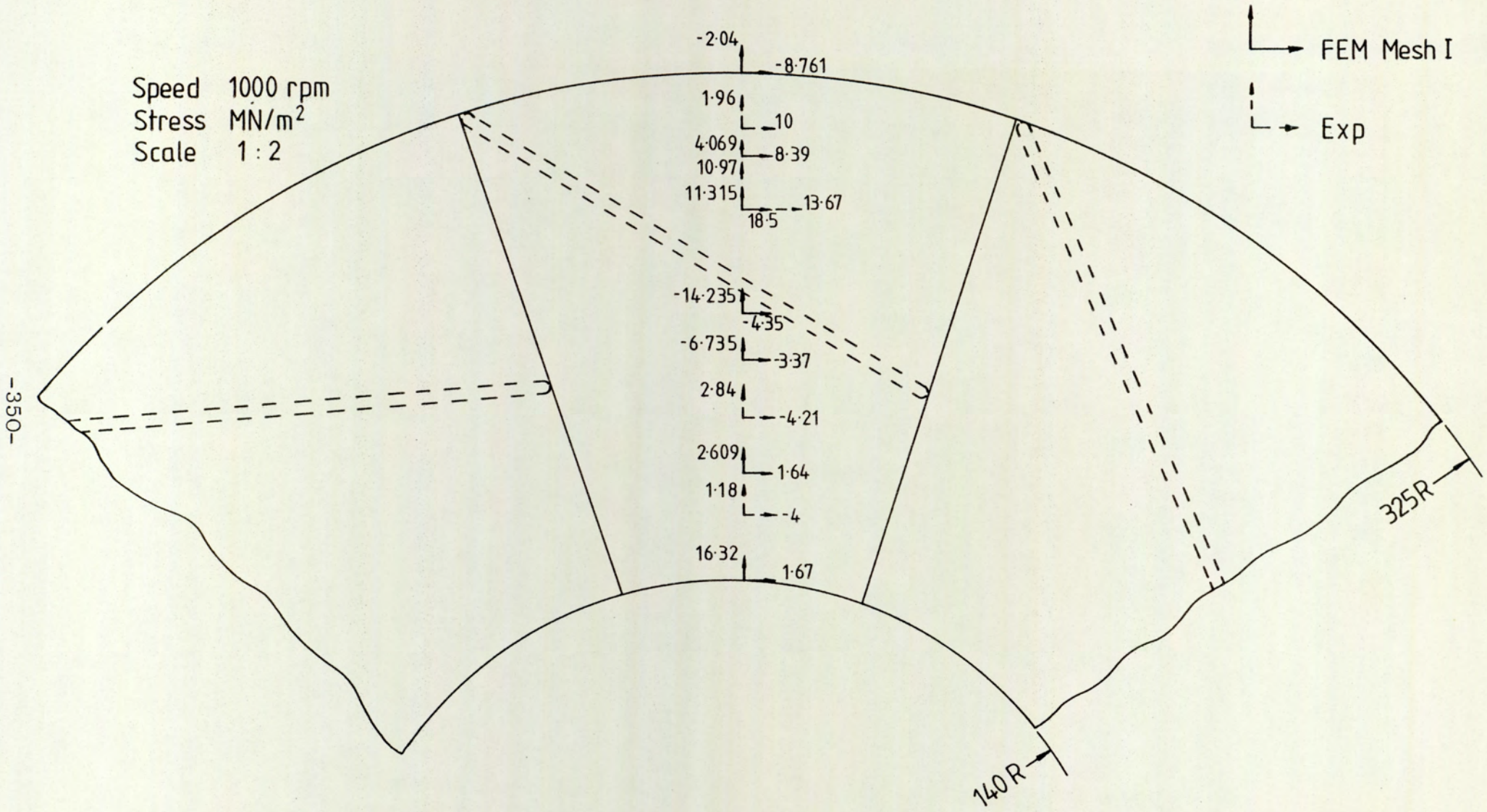


Fig 7.4.5 Radial and tangential stresses on outside surface of the backsheet : actual model

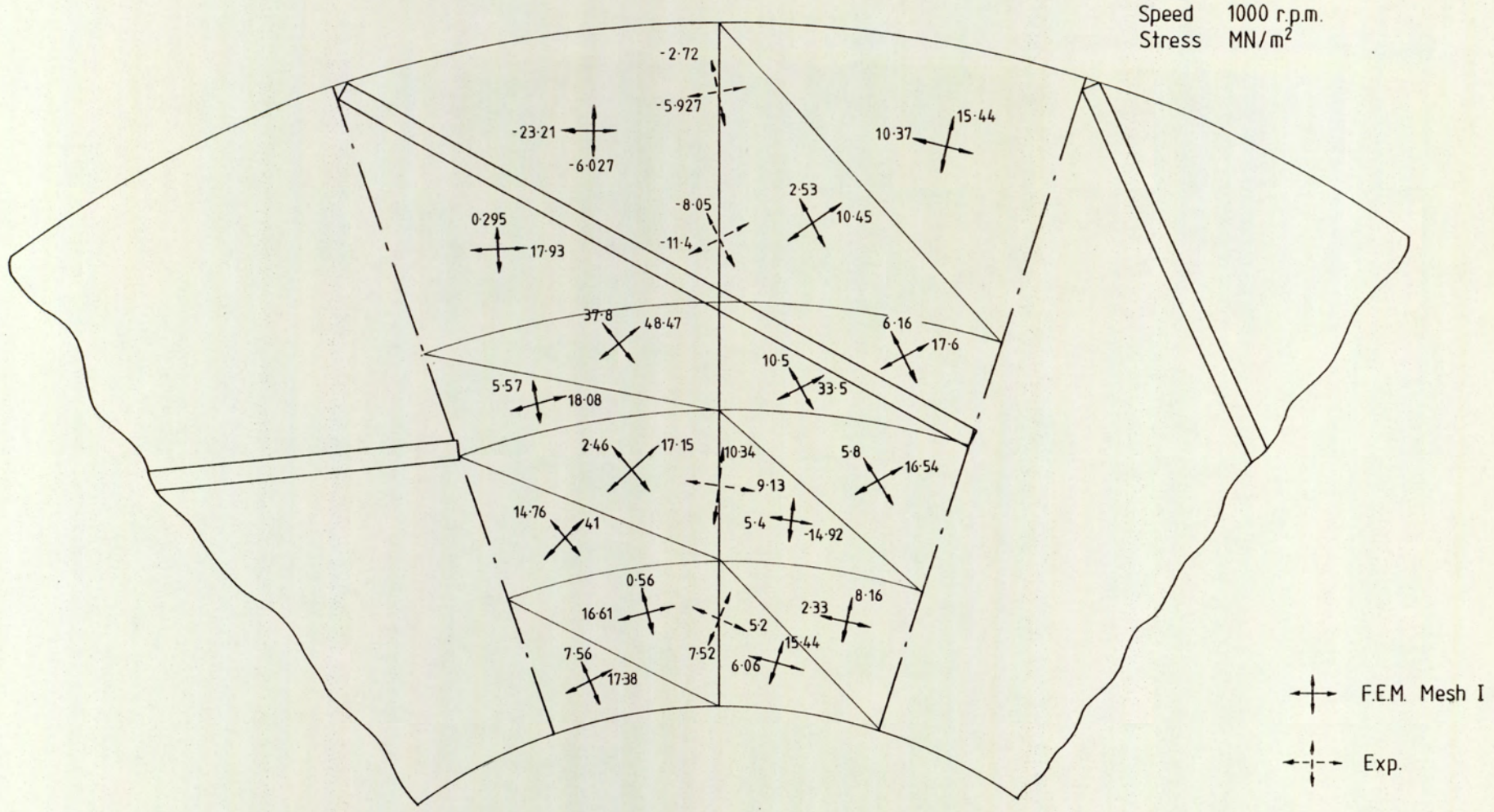


Fig 7.46 Principal stresses and directions on inside the backsheet of the actual model

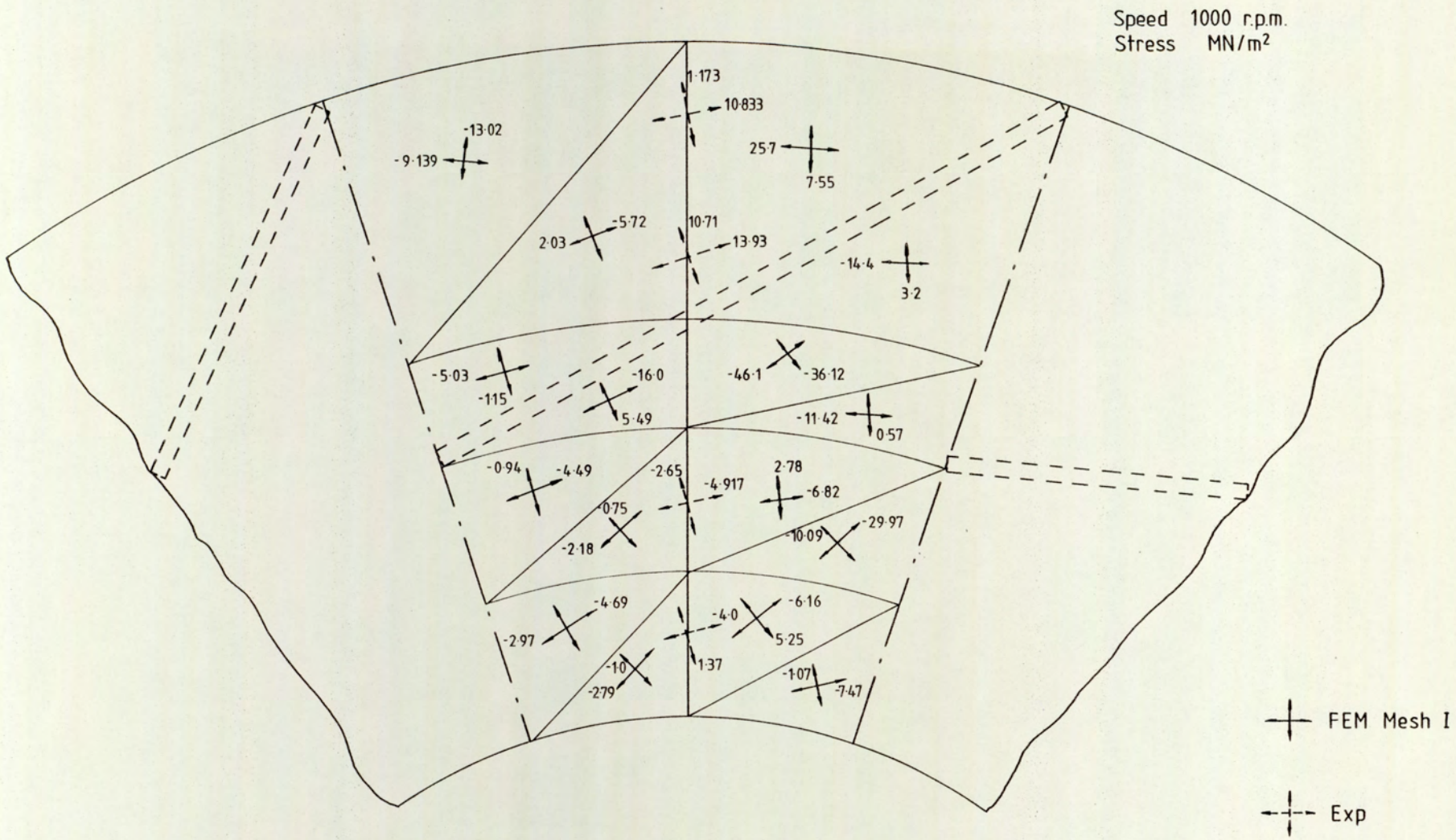


Fig 7.47 Principal stresses and directions on outside the backsheet of the actual model

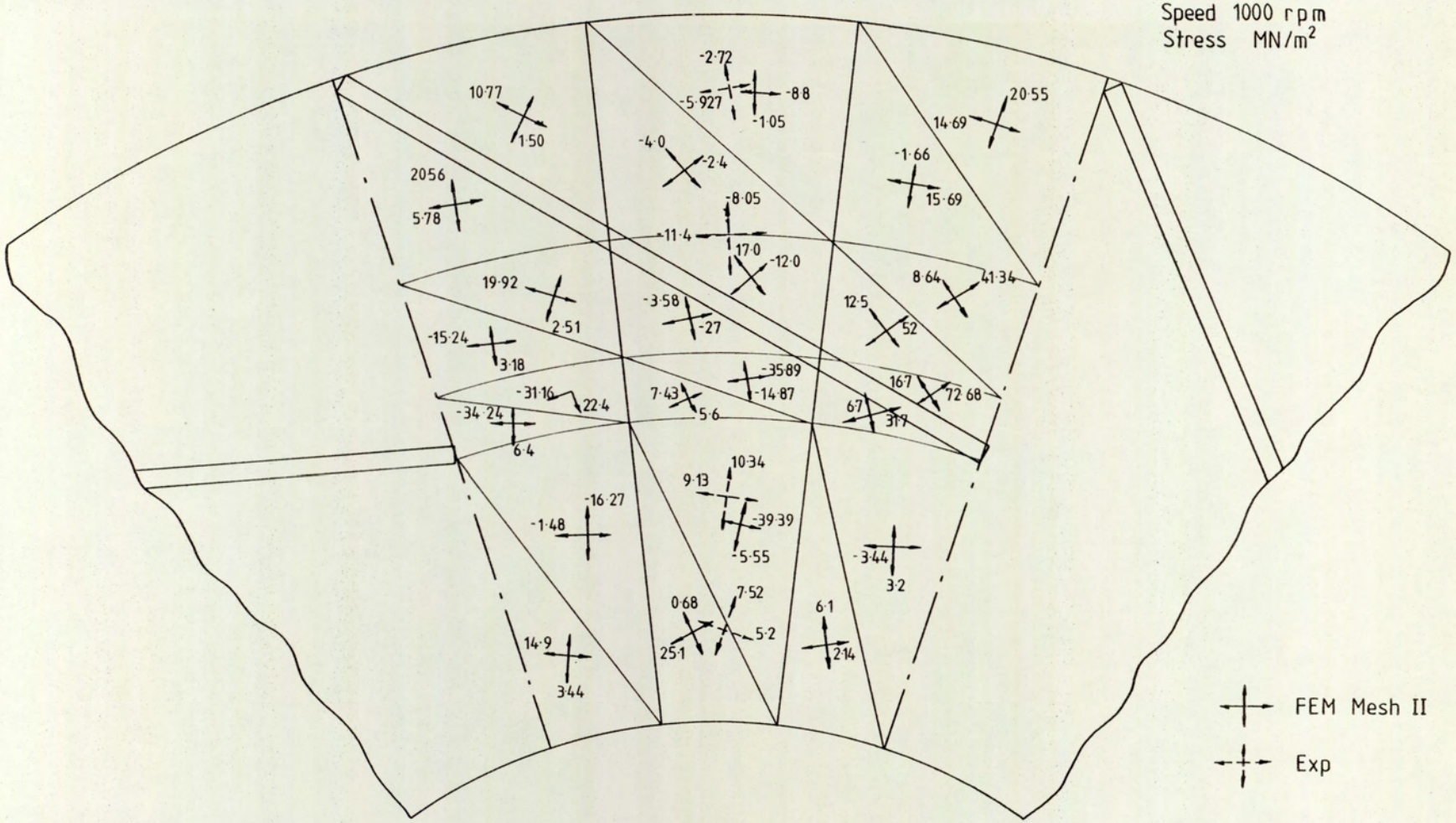


Fig 7.48 Principal stresses and directions on inside the backsheet of the actual model

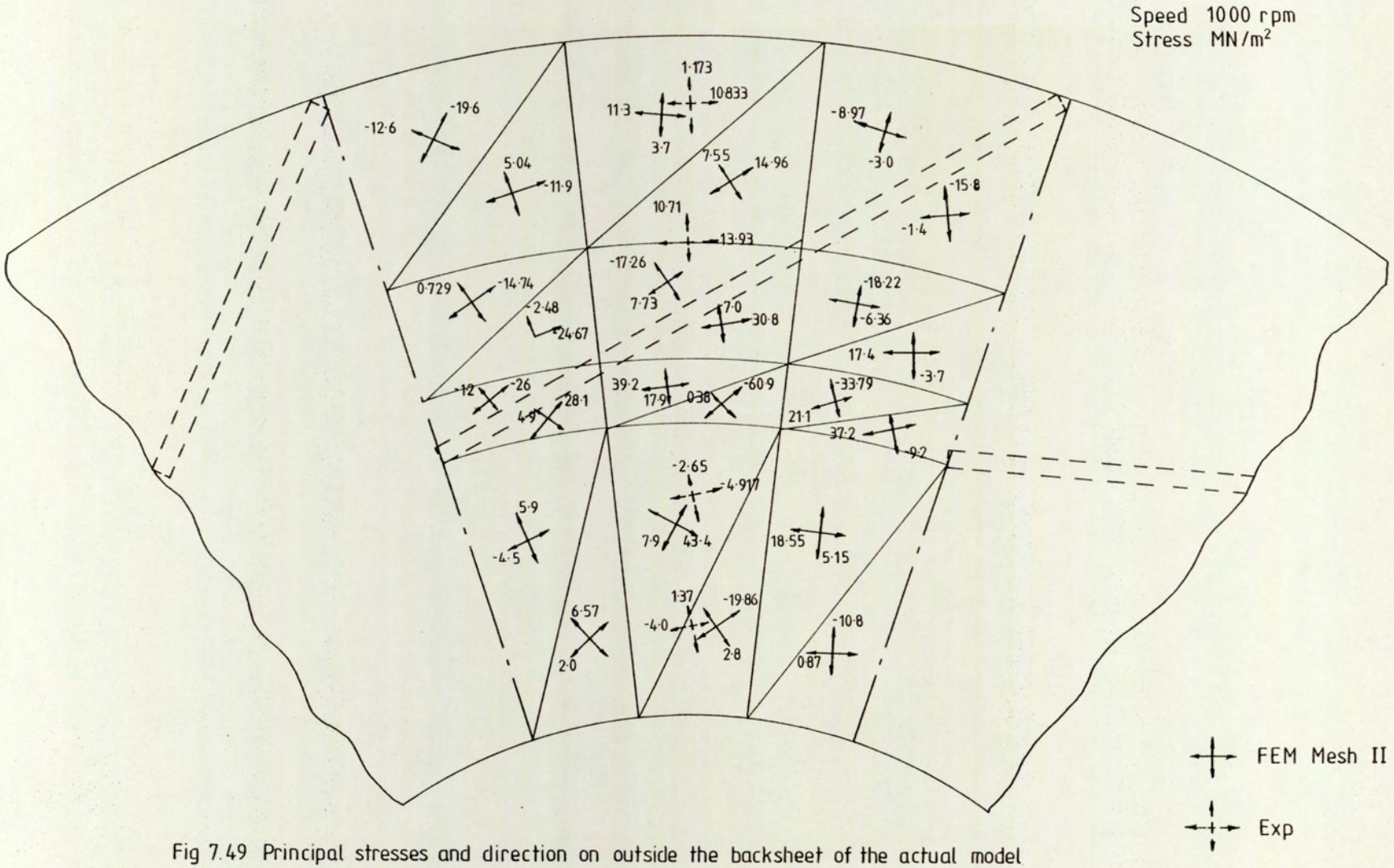


Fig 7.49 Principal stresses and direction on outside the backsheet of the actual model

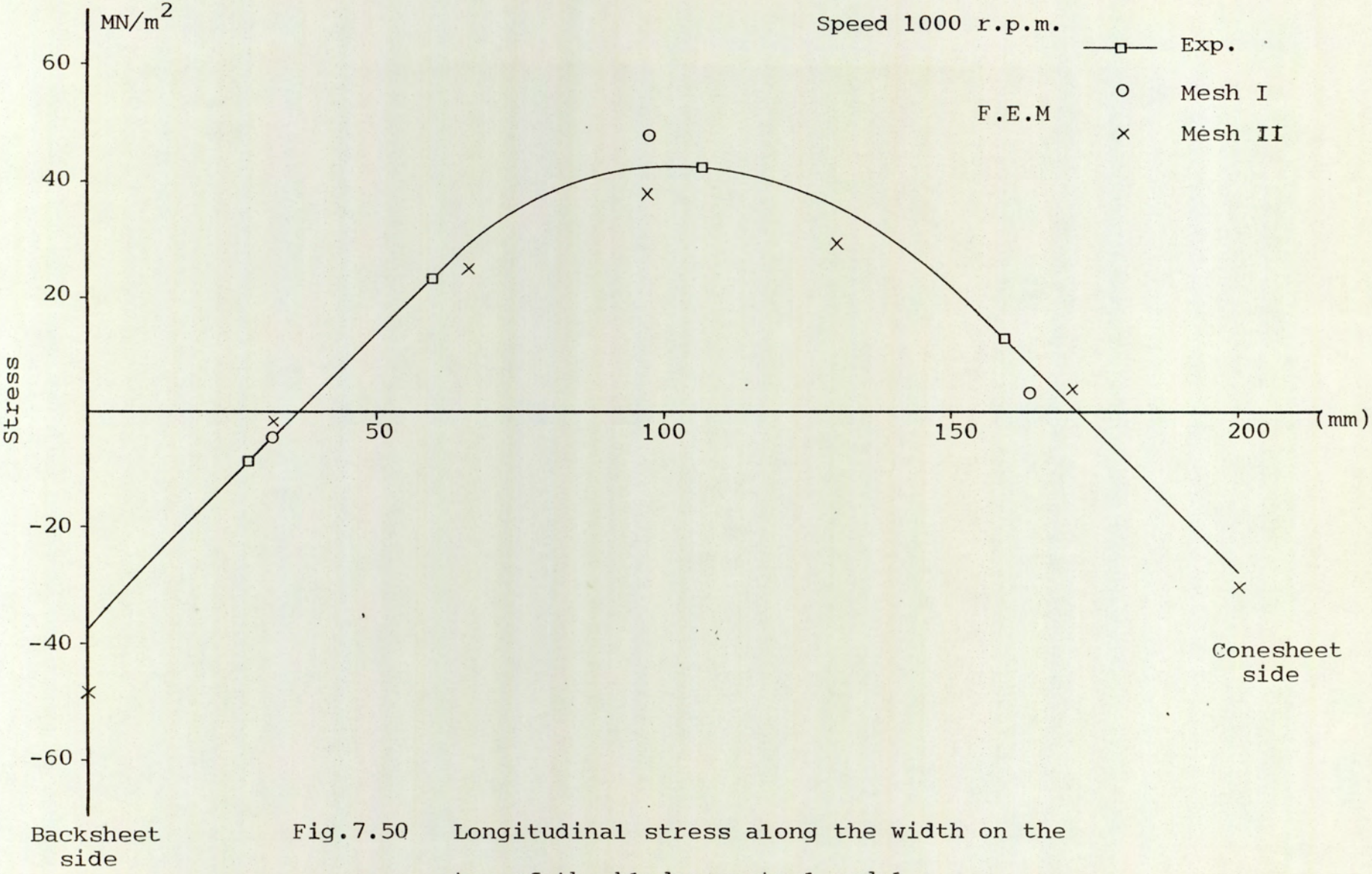


Fig.7.50 Longitudinal stress along the width on the top of the blade : actual model

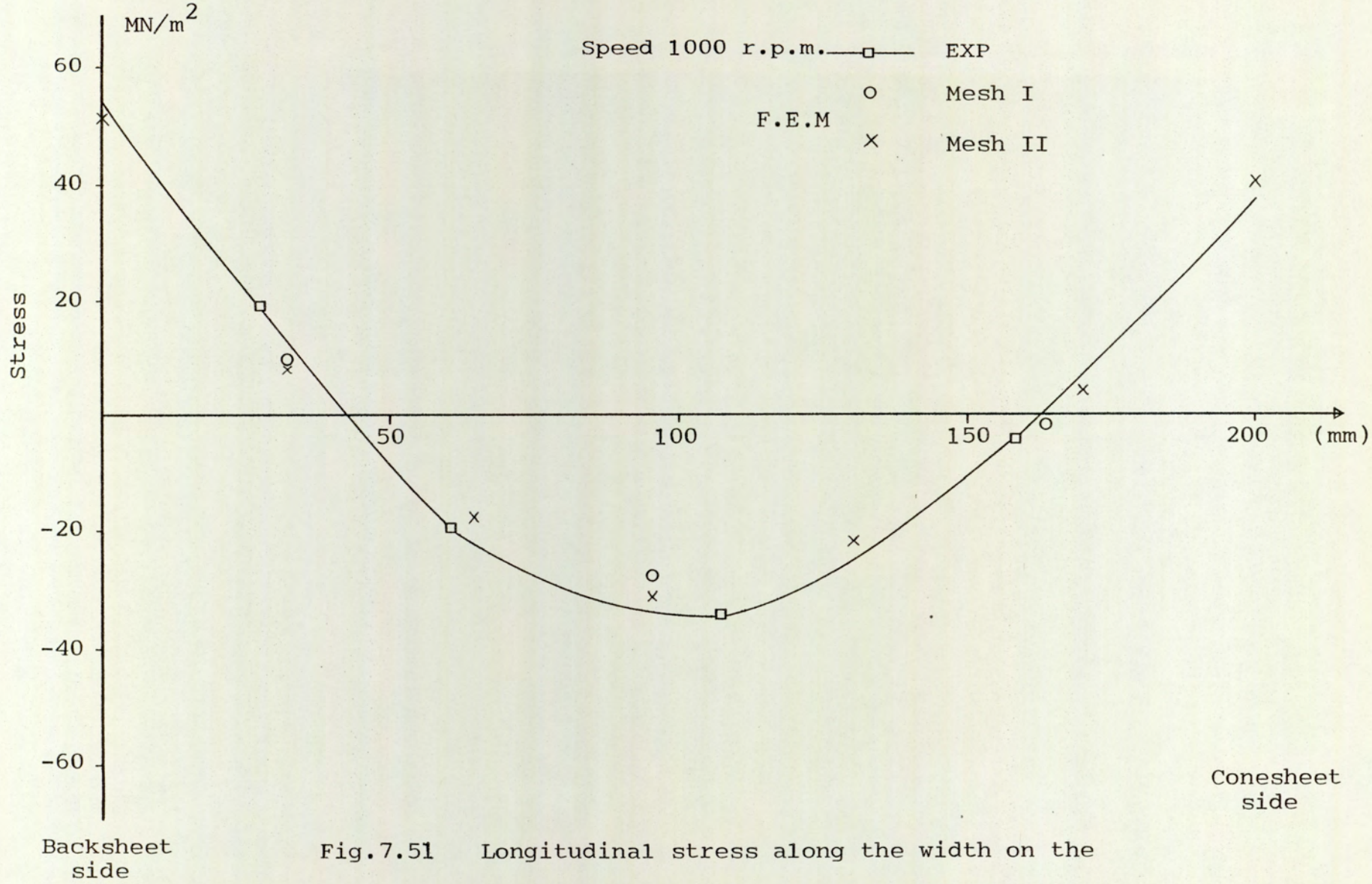


Fig.7.51 Longitudinal stress along the width on the bottom of the blade : actual model

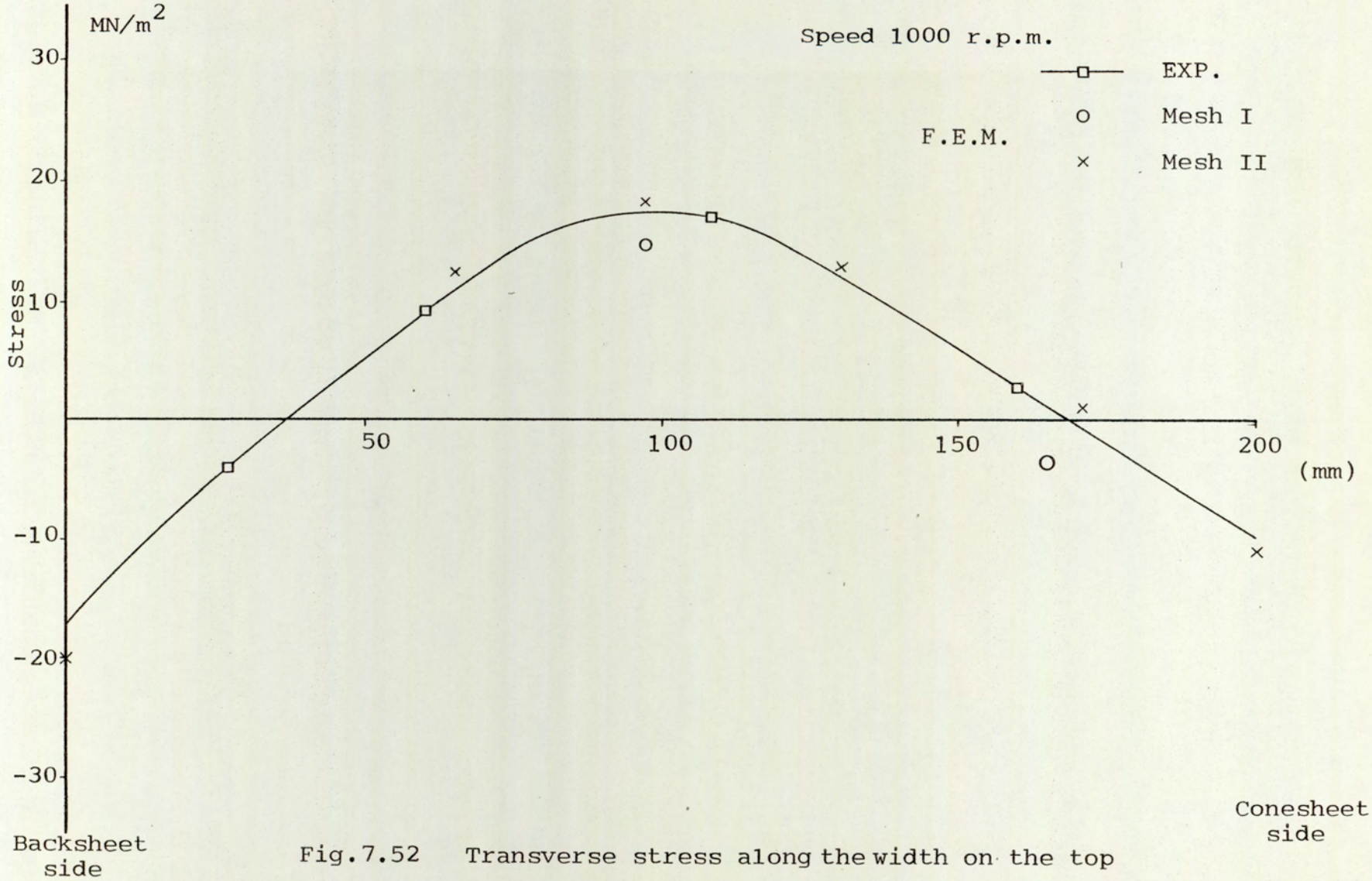


Fig.7.52 Transverse stress along the width on the top of the blade : actual model

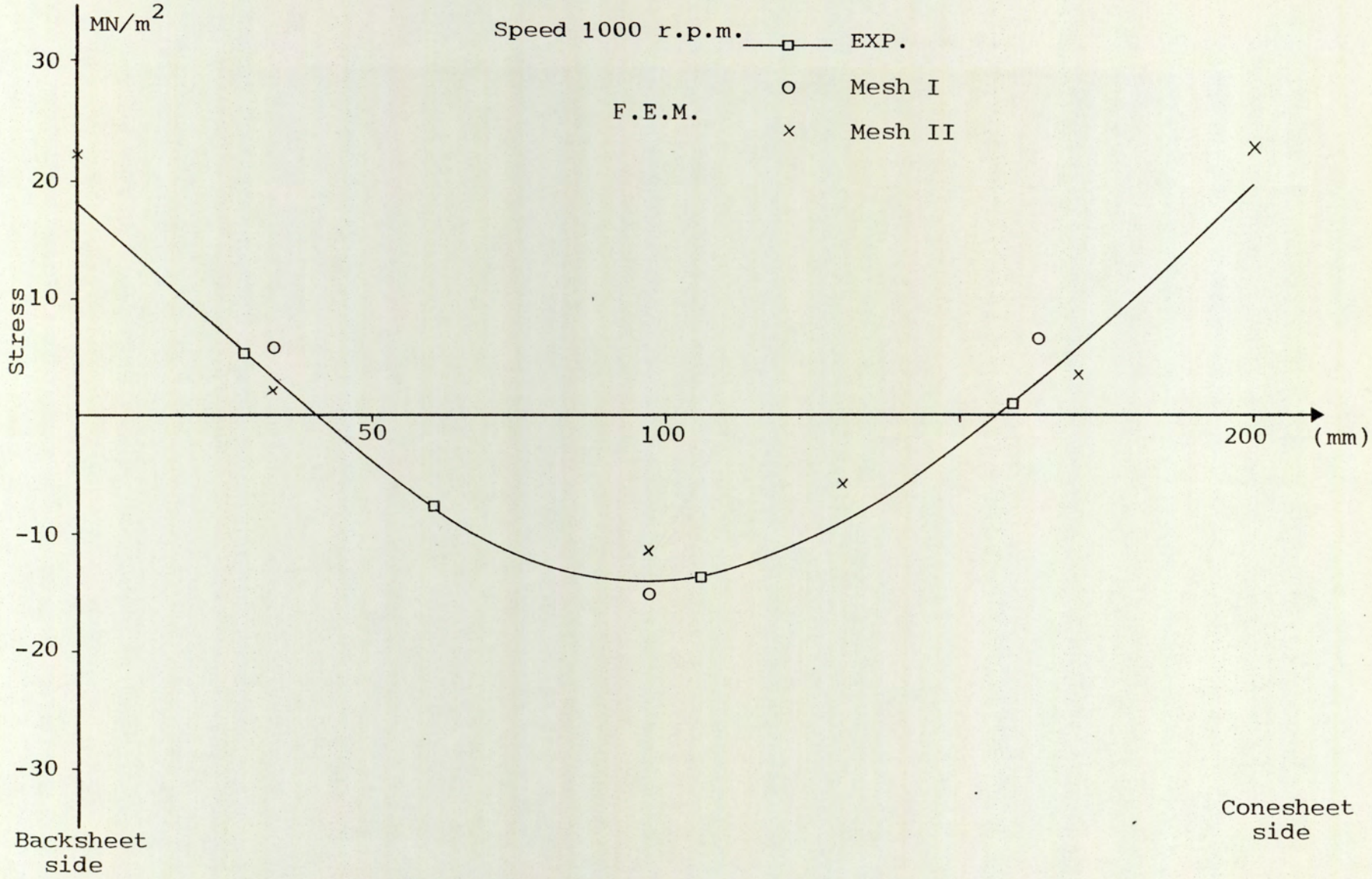


Fig.7.53 Transverse stress along the width on the bottom of the blade : actual model

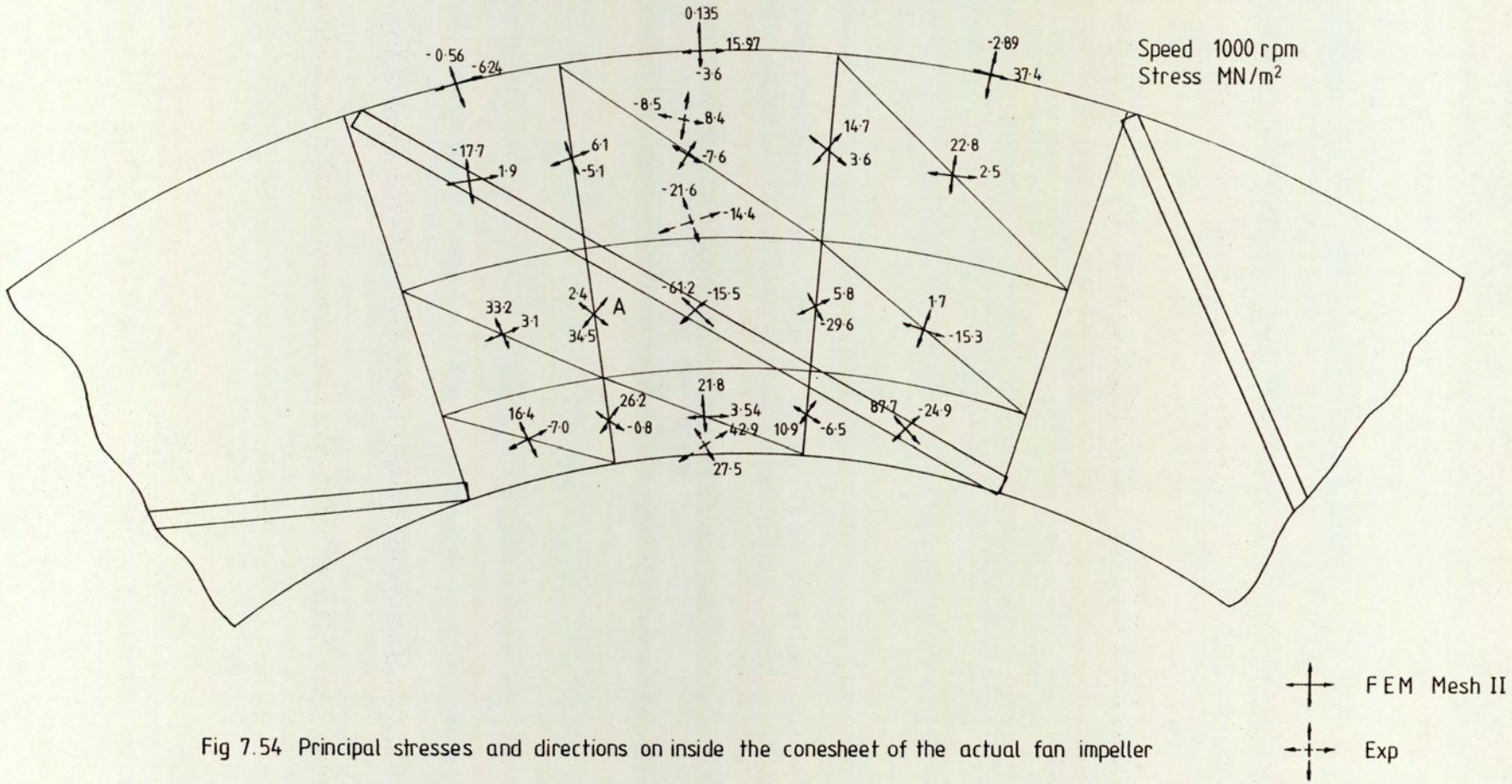


Fig 7.54 Principal stresses and directions on inside the conesheet of the actual fan impeller

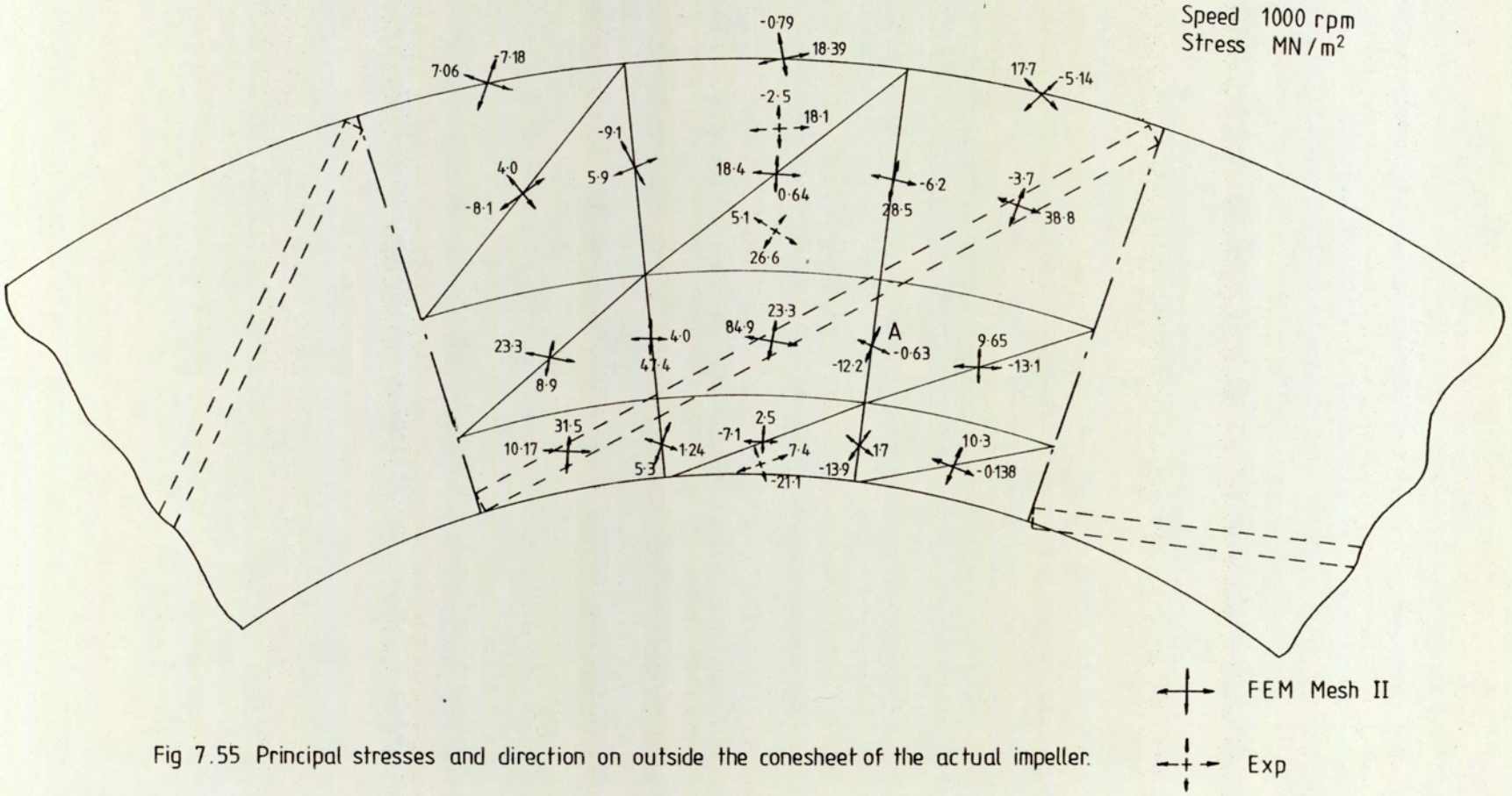


Fig 7.55 Principal stresses and direction on outside the conesheet of the actual impeller.

7.6 CONCLUSIONS ON THE NUMERIAL ANALYSIS OF THE ROTATING FAN IMPELLERS

The objective of this research work, to develop a finite element package for rotating fan impellers, has been achieved and confirmed by an experimental programme.

The reasonable correlation between the numerical and experimental techniques has proved the adequacy of:

(1) Representation of the centrifugal force for each integrating point in the element as equivalent nodal forces using the element shape function which has automated in the finite element program.

(2) The facility of exploiting sectorial symmetry in the rotating fan impellers and using the appropriate boundary conditions (as well as the equal movements of goemetrically corresponding node pairs) which has been incorporated in the finite element program.

(3) The suitable design of the finite element meshes which produced acceptable results. Finite element solutions converged to the experimental results as the mesh was made finer (as explained for the simplified model), but at the expense of heavy data preparation and computer cost.

Since the size of the memory of the Hewlett Packard model 9845B computer available is limited to 187 K bytes, therefore it was not possible to use a finer than Mesh II in the actual model analysis.

In conclusion, the developed finite element package, using the high performance semiloof shell element, has been seen to provide accurate information on the stress distribution in rotating fan impellers.

7.7 CLOSING REMARKS

In this chapter, the evaluation of the numerical results for the fan impellers has been carried out by a comparison of the finite element predictions with those obtained experimentally, using the strain gauge technique. An acceptable conclusion was obtained in the comparison of the results. These comparisons have been carried out for a simplified model and a commercial fan impeller. Some relevant points about the discrepancies were discussed.

In the following chapter, an overall summary and conclusions of the whole research, as well as some recommendations for further work are presented.

CHAPTER EIGHT

SUMMARY, CONCLUSIONS AND RECOMMENDATIONS FOR FURTHER WORK

Having achieved all the aims set forward in Chapter One, this chapter presents a summary, conclusions and recommendations for futher work.

8.1 SUMMARY

The work carried out in this thesis could be summarised as follows:

1. The finite element method was applied to complex thin plate and shell structures. A computer program SMILOF based on the semiloof shell element was developed and tested against several plate and shell structures. The results were compared with numerical or analytical results reported by other researchers.

2. The computer program was extended to accept the segmental solving routine to solve more complicated plate and shell structures.

3. The computer package IMPSMF was further developed to take advantage of the sectorial symmetry of rotating fan impellers. In particular, this was tested on rotating cones and the rotating discs, for the displacements and stresses, prior to application to an actual impeller.

4. The convergence of the displacements and stresses was studied for a simplified radial fan impeller with and

without a conesheet.

5. An experimental program was carried out for a simplified and an actual fan impellers.

6. The experimental results were compared with the numerical predictions.

8.2 CONCLUSIONS

From the results of this work, the following conclusions and observations appear to be valid.

1. The computer program SMILOF developed to solve thin plate and shell structures for displacements and stresses is well tested and reliable.

2. The rate of convergence of the semiloof shell element is higher than for the other elements with which it was compared.

3. The numerical modelling technique used (finite element method) for the fan impellers proved to be satisfactory for design work.

4. It is possible to analyse only one sector of the fan impellers successfully using the computer package IMPSMF. Exploiting sectorial symmetry yields significant computational benefits. This package may be used for a detailed study of the complex interaction between the backsheet, blades and the conesheet before the production

stage in fan industries. This significantly reduces the need for experimental investigations which are costly in money, manpower and time.

5. Good agreement was obtained between the experimental and numerical predictions for the simplified radial fan impeller and a satisfactory agreement for the actual one.

6. The position, magnitude and direction of the principal stresses on the backsheet, blade and conesheet, obtained from the finite element analysis agreed fairly well with those obtained using the experimental programme.

7. The "user friendly" desk top computer with suitable memory size may be used for handling complicated engineering problems.

8. The accuracy of the results depends on the number of elements into which the structure is divided in the course of the finite element analysis. The greater the number of elements the more accurate the results would be. This is limited by the computer facility available for the work, since the use of finer finite element meshes needs more memory size.

9. Finally, it is possible to introduce the computer program "IMPSMF" within a C.A.D. package to provide a useful design tool for rotating fan impellers, and for this reason the programs have been designed to be used by the stress analysis engineers without much computer experience.

8.3 SUGGESTIONS FOR FURTHER WORK

The following suggestions for further study are made to improve on the present work and to carry this research forward:

1. Development of a finite element mesh generation program based on the semiloof shell element (triangular and quadrilateral type) for the general thin plate and shell structures. The orientation of this programme should make it possible to take into account the sectorial symmetry of rotating fan impellers. This facility will reduce the tedious preparation of the input data when changing from one mesh to another. References (80, 81, 82, 83) are offered as a starting point in developing this package.

2. The execution time for the segmental solving routine could be improved by using a hard disc instead of the floppy disc used in this work. FPRINT≠ and FREAD≠ are used in this disc which improves the speed (as much as 25:1) over the ordinary PRINT≠ and READ≠ statements. The new version of the desk top computer HP9845C together with the hard disc provide an excellent means for implementating the finite element package IMPSMF for the rotating fan impellers and it can be an efficient recognised package in the industrial fields.

3. The developed finite element programs may be extended to analyse other problems related to the fan impellers such as vibration and fatigue analysis.

APPENDICES

APPENDIX A

USER INSTRUCTIONS FOR THE COMPUTER PROGRAMS

A.1 A Guide to data preparation and sample problem for the computer program SMILOF

The primary purpose of this document is to set out the rules which must be followed in using the program SMILOF. Data instructions and a sample problem will be presented.

A.1.1 Data input instructions

(A) Number of jobs to be solved.

(B) Type of element (1)*

(C) For the first job.

(1) Number of elements.

(2) Number of nodes.

(3) Number of sets of forces.

(4) Principal stresses and strains(2)

(5) Number of nodes where skewed boundary conditions are applied.

(6) Number of materails.

(7) Type of output(3).

(8) Gravity effect (weight/unit area).

* Superscript refer to notes which follow the data input instructions.

(9) Normal distributed pressure.

(10) Rotational speed rpm.

(D) For the number of nodes in B(2), input

(1) x-coordinate

(2) y-coordinate

(3) z-coordinate

(E) Nodal connections and material number of each element[4].

(F) For the number of elements in C(1), input

(1) Number of like elements (similar thickness)

(2) Thickness

(3) String of like nodes.

(repeat F(1) and F(2) upto the number of elements in the discretised structure).

(G) Number of specified nodes.

(H) For the number of specified nodes in G, input

(1) Number of like nodes.

(2) Code[5] (prescribed load or displacement)

(3) Value of prescribed load or displacement in x-direction.

(4) Value of prescribed load or displacement in y-direction.

(5) Value of prescribed load or displacement in z-direction.

(6) String of like nodes.

(repeat H(1) to H(6) up to the number of specified nodes).

(I) Number of specified midside nodes.

(J) For the number of specified midside nodes in I, input:

(1) Number of like midside nodes
(2) Code[6] (prescribed rotation or normal moment)
(3) Prescribed loof nodes rotations or normal moments for the two loof nodes positioned on the right and left of the current midside node.
(4) String of like midside nodes.

(repeat J(1) to J(4) up to the number of specified midside nodes).

(K) For elements with material number 1, input:

(1) Elastic constants[7]
(2) Density.

(repeat K(1) and K(2) for material numbers 2,3,....etc.)
If C(5) = 0, pass section L

(L) For the number of skewed nodes in C(5), input:

(a) Node number
(b) Angle of skew[8].

(M) For the next job repeat from (B).

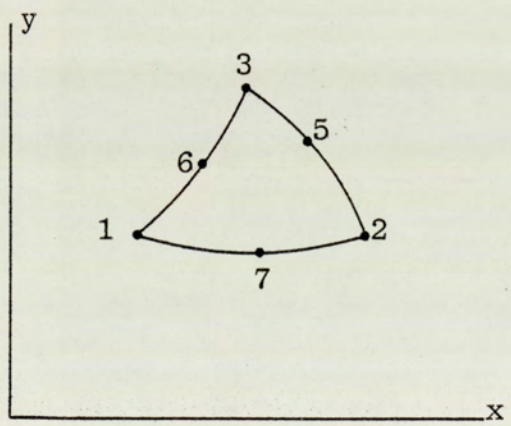
Notes

1. Put, 0 for triangular mesh
   1 for quadrilateral mesh.

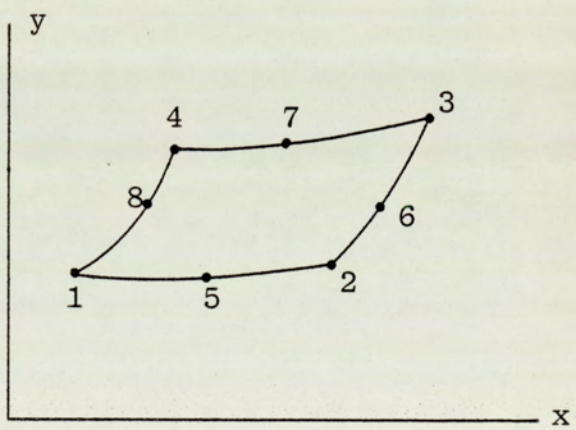
2. Put, 1 if the principal stresses are required
   0 if not.

3. Put, 1 if nodal strains and stresses are required
   2 if element strains and stresses are required
   3 if both nodal and element strains and stresses are required.

4. The nodal connections and material number for each element must be specified using the following sequence:



Triangular Type



Quadrilateral Type

If M is the material number, then input

1 2 3 4 5 6 M for the triangular type, and

1 2 3 4 5 6 7 8 M for the quadrilateral type.

5. If Kode=

O, This represents prescribed loads in the x, y and z direction.

1, this represents prescribed displacements in x and y and load in z direction.

2, this represents prescribed load in y and displacements in x and z direction.

3, this represents prescribed load in z and displacements in x and z direction.

4, this represents prescribed load in z and displacements in x and y direction.

5, this represents prescribed displacement in x and loads in y and z direction.

6, this represents prescribed displacement in y and loads in x and z direction.

7, this represents prescribed displacement in z and loads in x and y direction.

6. If Kode 1 =

O, this represents prescribed normal moments on both loof nodes (to the right and to the left of the current midside node).

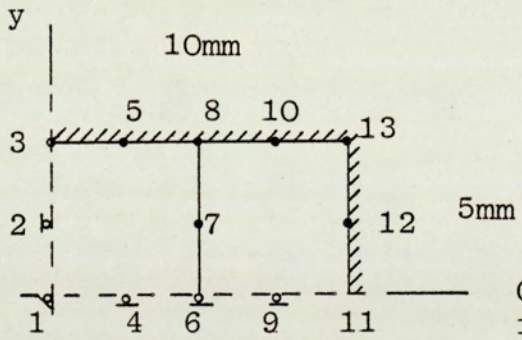
3, this represents prescribed loof rotations on both loof nodes.

7. Elastic constants are E and ν.

8. Specified in degrees and taken in anti-clockwise direction from the global x-axis.

A.1.2 Sample Problem

If the problem shown in Fig. A.1 is considered, the input data instructions necessary for running this case



Central concentrated force = 1N

Thickness = 0.5

ν = 0.29

E = 2.1 E5 N/m^2

ρ = 7.8 E-9 N/mm^3

Fig. A.1

(A) : 1

(B) : 1

(C) : Control variables

1, 13, 1, 1, 0, 1, 3, 0, 0, 0

(D) Nodal coordinates

```
 0,   0,   0
 0, 2.5,   0
 0,   5,   0
2.5,  0,   0
2.5,  5,   0
 5,   0,   0
 5,  2.5,  0
 5,   5,   0
7.5,  0,   0
7.5,  5,   0
10,   0,   0
10,  2.5,  0
10,   5,   0
```

(E) Nodal connections and material number.

```
 1   6   8   3   4   7   5   2   1
 6  11  13   8   9  12  10   7   1
```

(F) 2, 0.5, 1, 2

(G) +(H) : Specified nodes

```
12,  7,  1,  0,  0,  0,  3,  5,  8,  10,  11,  12,  13
     3,  6,  0,  0,  0,  4,  6,  9
     1,  5,  0,  0,  0,  2
     1,  4,  0,  0, -1
```

(I) + (J) Specified midside nodes

6, 6, 3, 0, 2, 4, 5, 9, 10, 12

(K) Elastic constants:

2.1 E5, 0.29, 7.8 E-9.

A.2 A Guide to data preparation and sample problem for the package IMPSMF

Input data sequence is similar to that explained in section A.1.1 with the addition of new item after C, say C1. This contains the new following input data:

C1 (1) Nk - Number of nodes having similar behaviour. If Nk = 0 pass C1(2) and C1(3).

C1 (2) Alp - The semi cone angle.

C1 (3) The array St(*) containing the numbers of nodes having similar behaviour.

The following points should be noticed:

1. There is no need to specify the nodes on the sector boundary b-b.

2. The element and nodal connections matrix contain 10 columns. The element type (backsheet, blade or conesheet) is stored in the tenth column.

3. For non-periodical structures Nk = 0.

4. Alp = 0 for radial bladed impellers.

A.2.1 Sample Problem

If the sector of radial fan impeller of Fig. A.2 is considered, the sequence of the input data are as follows:

(A) : 1

(B) : 1

(C) : 3, 18, 3, 1, 6, 1, 3, 0, 0, 2 0 0 0

(C1) :

St(*)
3, 0
1, 16
2, 17
3, 18

(D) :
92.3 38.2 0
163.8 67.8 0
235.3 97.4 0
83 55.5 0
211.6 141.5 0
70.7 70.7 0
125.5 125.5 0

```
      180.3   180.3    0
       70.7    70.7   35
      180.3   180.3   40
       70.7    70.7   70
      125.5   125.5   75
      180.3   180.3   80
       55.5    83      0
      141.5   211.6    0
       38.2    92.3    0
       67.8   163.8    0
       97.4   235.3    0

(E)   1   3   8   6   2   5   7   4   1   1
      6   8  18  16   7  15  17  14   1   1
      6   8  13  11   7  10  12   9   1   2

(F)   2,  4.1,  1,  2,  1,  3.28,  3

(G)   6

(H)   4,  1,  0,  0,  0,  1,  4,  6,  14,
      2,  6,  0,  0,  0,  2,  3

(I)   3

(J)   3,  3,  0,  2,  4,  14

(K)   2.1 E5,   0.29,  7.8 E-9
```

(L)	1	22.5
	2	22.5
	3	22.5
	16	67.5
	17	67.5
	18	67.5

A.3 User instructions for the programs SMILOF and IMPSMF

Having the required input data file prepared according to the instructions given in Appendix A.1.1, the program SMILOF operating instructions are as follows:

1. LOAD "SMILOF:F8", press CONT
2. Press RUN button
3. Read the instructions appearing on the CRT
4. Choose the printing device (paper or CRT)
5. Input the name of the data file
6. Input the job name
7. If the strains and stresses are required, you are asked to CREATE the appropriate size file(s) as required by the control variables. This operation is fully discussed in the HP manual.

For the program IMPSMF, the steps of operating instructions are similar to those used in running the program SMILOF, except that the user is asked to CREATE a number of files equal to the number of segments. These depend on the size of the system stiffness matrix. As the machine is interactive, therefore some instructions are appeared in running the program.

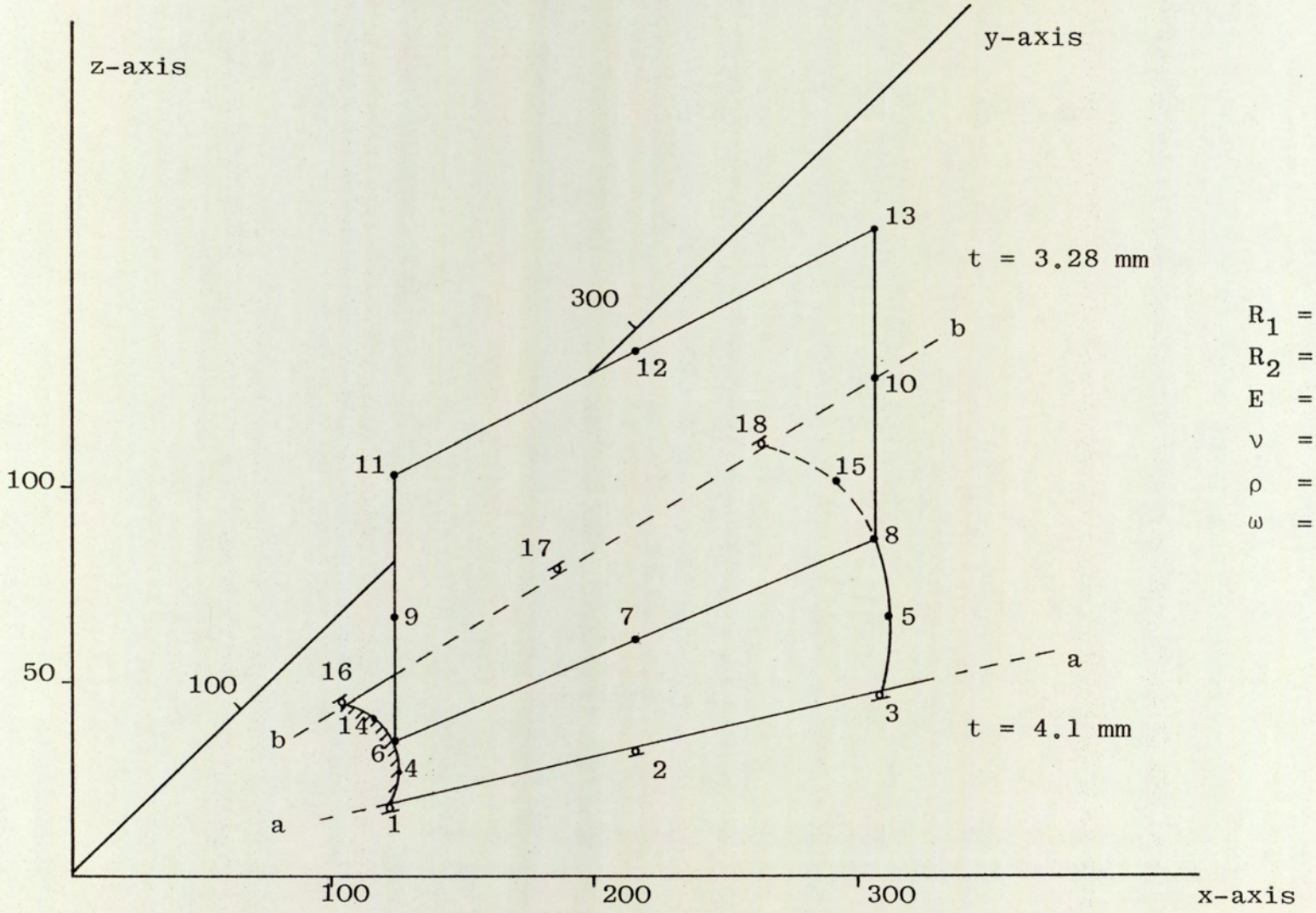


Fig. A.2 Typical finite element discretisation of impeller sector.

APPENDIX A.3

STRESSES AND STRAINS

This appendix presents the expressions for the stresses and strains of a surface layer of shell at a distance z from the middle surface.

(i) The normal and shearing strains are given by

$$\varepsilon_X = \overset{m}{\varepsilon}_X + z\,\overset{b}{\varepsilon}_X$$

$$\varepsilon_Y = \overset{m}{\varepsilon}_Y + z\,\overset{b}{\varepsilon}_Y$$

$$\varepsilon_{XY} = \overset{m}{\varepsilon}_{XY} + z\,\overset{b}{\varepsilon}_{XY}$$

(ii) The normal and shearing stresses are given by

$$\sigma_X = \frac{N_X}{t} + 12\,\frac{M_X}{t^3}\,.z$$

$$\sigma_Y = \frac{N_Y}{t} + 12\,\frac{M_Y}{t^3}\,.z$$

$$\sigma_{XY} = \frac{N_{XY}}{t} + 12\,\frac{M_{XY}}{t^3}\,.z$$

APPENDIX B

B.1 Listing of the Computer Package SMILOF

The main program SMILOF and the following subprograms are listed:

(1) SMINPT

(2) CONSTR

(3) SMASBL

(4) SMLDAP

(5) SMSKLD

(6) SMYVBS

(7) SMELST

(8) SMNDST

B.2 Listing of the Computer Package IMPSMF

The Master Program IMPSMF and the following subroutines are listed:

(1) "Maddarray"

(2) "Massembly"

(3) "Mskewdcon"

(4) "Skewiml"

(5) "Segsol"

(6) "Skewim2"

```
10    ! RE-STORE"SMILOF"
20    OPTION BASE 1
30    DIM Xita(2),K(13470),Xx(70),Yy(70),Ulx(70,1),Vly(70,1),W
      lz(70,1),B(6,32),C(3,21),Zz(70),Q(300,1),Thtxz(70,1),A$(
      20)
40    DIM Thikn(24),Th(10),Angsk(32)
50    INTEGER Node(24,9),Lnodz,Lvabz,Nelemt,Qort,Kode(70,1),Ko
      de1(70,1),Add(300),N,Nfree,Matno,Case,Nspec,Nsetfs,K,I,J
      ,Nnode,Count,Prnt,Princ,Nskw,Op
60    INTEGER Test(70),Nosk(32)
70    PRINTER IS 16
80    PRINT PAGE,SPA(25),"****STRESS ANALYSIS OF THIN SHELL ST
      RUCTURES****";
90    PRINT SPA(32),"Stress Analysis Group:Dept. of Mech. Eng.
      ",LIN(1)
100   PRINT SPA(32),"University of Aston",LIN(2)
110   PRINT "Press CLEAR then CONT";
120   PRINT PAGE,SPA(19),"*****Object Of The Program*****"
130   PRINT
140   PRINT "In this program finite element formulation of the
      Semiloof Shell";
150   PRINT "Element Is Presented";
160   PRINT "The problem of thin shell and plate structures may
      be solved ";
170   PRINT "For displ. ,Strains And Stresses";
180   PRINT "Both versions are implemnted here,Quadrilateral and
      triangular ";
190   PRINT "Semiloof shell element";
200   PRINT "Press CLEAR then CONT";
210   PAUSE
220   DISP "Choose the printing devive 0 for paper 16 for
      screen";
230   INPUT T
240   PRINTER IS T
250   DISP "What is the Name of your input data file";
260   INPUT Name$
270   ASSIGN #2 TO Name$,C
280   IF NOT C THEN GOTO 330
290   BEEP
300   DISP "File not found!Try again";
310   WAIT 2000
320   GOTO 260
330   READ #2;Qort,Njob
340   FOR Count=1 TO Njob
350   LINPUT "Type in name of the job.......not more than 20
      charecters",A$
360   PRINT "Job name........";A$,LIN(1)
370   READ #2;Nelemt,Nnode,Nsetfs,Princ,Nskw,Nmat,Prnt,Ch1,Ch2,Omg
380   PRINT "No of elements.........";Nelemt
390   PRINT
400   PRINT "No of nodes.............";Nnode
410   PRINT
420   IF (Prnt<>1) AND (Prnt<>3) THEN GOTO 470
430   DISP "Input the name of the file(Es$)to be  created to s
      tore B[*] to find Nod. Strs.";
440   INPUT Es$
450   DISP "CREATE Es$ in a suitble size to store data for B(*
      ),Ro(*),Thik,poin(*)for the integrating points for the e
      lements then Press CONT";
460   PAUSE
470   IF (Prnt<>2) AND (Prnt<>3) THEN GOTO 520
480   DISP "Input the name of the file(Ns$)to be created to store
      B[*] to find Ele. Strs.";
490   INPUT Ns$
500   DISP "CREATE Es$ in a suitble size to store data for B(*
      ),Ro(*),Thik,poin(*)for the element centroids THEN Press
       CONT";
```

```
510   PAUSE
520   PRINT LIN(2)
530   REDIM Xx(Nnode),Yy(Nnode),Zz(Nnode),Kode(Nnode,Nsetfs),K
      ode1(Nnode,Nsetfs),Node(Nelemt,7+2*Qort),Ulx(Nnode,Nsetf
      s),Vly(Nnode,Nsetfs),Wlz(Nnode,Nsetfs)
540   REDIM Thtxz(Nnode,Nsetfs),C(Nmat,21),Test(Nnode)
550   REDIM B(6,24+8*Qort),Thikn(Nelemt),Th(Nmat),Angsk(Nskw+1
      ),Nosk(Nskw+1)
560   IF Qort=0 THEN PRINT SPA(12),"Element selected is a tria
      ngular type";
570   IF Qort=1 THEN PRINT SPA(12),"Element selected is a quad
      rilateral type";
580   Omg=Omg*2*PI/60
590   LINK "SMINPT",1600
600   CALL Feinput(Xx(*),Yy(*),Zz(*),Ulx(*),Vly(*),Wlz(*),Thtx
      z(*),Thikn(*),#2,Kode(*),Kode1(*),Test(*),Node(*),Nfree,
      Nnode,Nelemt,Qort)
610   REDIM Q(Nfree,Nsetfs),Add(Nfree)
620   CALL Addarray(Test(*),Nelemt,Nnode,Add(*),Node(*),Qort,N
      free)
630   LINK "SMCNTR",1600
640   FOR Matno=1 TO Nmat
650   CALL Constrel(C(*),Th(*),#2,Matno)
660   NEXT Matno
670   IF Nskw=0 THEN GOTO 710
680   FOR I=1 TO Nskw
690   READ #2;Nosk(I),Angsk(I)
700   NEXT I
710   REDIM K(Add(Nfree))
720   LINK "SMASBL",1600
730  CALL Assembly(Q(*),K(*),Xx(*),Yy(*),Zz(*),C(*),B(*),Th(*)
     ,Omg,Ch1,Ch2,Thikn(*),Angsk(*),Ns$,Es$,Node(*),Test(*),N
      elemt,Add(*),Nfree,Nskw,Nosk(*),Qort,Prnt)
740   LINK "SMLDAP",1600
750   K=0
760   FOR I=1 TO Nnode
770   J=4-2*Test(I)
780   K=K+5-2*Test(I)
790   CALL Loadapp(Ulx(I,1),Vly(I,1),Wlz(I,1),Thtxz(I,1),Q(*),
      Kode(I,1),Kode1(I,1),K,1,J)
800   NEXT I
810   IF Nskw=0 THEN GOTO 840
820   LINK "SMSKLD",1600
830   CALL Skewload(Q(*),Angsk(*),Test(*),Nsetfs,Nosk(*),Nskw,1)
840   LINK "SMYVBS",1600
850   REM ---INTROUDUCTION TO KINEMATIC CONSTRAINTS
860   K=0
870   FOR I=1 TO Nnode
880   J=4-2*Test(I)
890   K=K+5-2*Test(I)
900   IF Kode(I,1)=0 THEN GOTO Test
910   IF (Kode(I,1)=2) OR (Kode(I,1)=6) THEN GOTO Kc2
920   IF Kode(I,1)=7 THEN GOTO Kc3
930   CALL Bounconst(Ulx(I,1),Q(*),K(*),K-J,Nfree,1,Add(*))
940   IF Kode(I,1)=3 THEN GOTO Kc3
950   IF Kode(I,1)=5 THEN GOTO Test
960 Kc2:  CALL Bounconst(Vly(I,1),Q(*),K(*),K-J+1,Nfree,1,Add(*))
970   IF (Kode(I,1)=4) OR (Kode(I,1)=6) THEN GOTO Test
980 Kc3:  CALL Bounconst(Wlz(I,1),Q(*),K(*),K-J+2,Nfree,1,Add(*))
990 Test:  IF (J<>4) OR (Kode1(I,1)=0) THEN GOTO Next
1000  CALL Bounconst(Thtxz(I,1),Q(*),K(*),K-J+3,Nfree,1,Add(*))
1010  CALL Bounconst(Thtxz(I,1),Q(*),K(*),K-J+4,Nfree,1,Add(*))
1020 Next: NEXT I
1030  PRINT LIN(3)
1040  PRINT SPA(10);"------Symvbsol begins------"
1050  CALL Symvbsol(K(*),K(*),Q(*),Add(*),Nfree,1)
```

```
1060  IF Nskw=0 THEN GOTO 1090
1070  LINK "SMSKLD",1600
1080  CALL Skewload(Q(*),Angsk(*),Test(*),Nsetfs,Nosk(*),Nskw,-1)
1090  PRINT ,SPA(4);"Nodal displacements"
1100  PRINT ,SPA(3);"--------------------"
1110  PRINT LIN(2)
1120  PRINT "Node";"Disp in X-dir",SPA(5),"Disp in Y-dir",SPA(
      5),"Disp in Z-dir"
1130  K=0
1140  FOR I=1 TO Nnode
1150  J=4-2*Test(I)
1160  K=K+5-2*Test(I)
1170  PRINT USING "3D,3X,3(MD.8DE,3X)";I,Q(K-J,1),Q(K-J+1,1),Q
      (K-J+2,1)
1180  PRINT
1190  NEXT I
1200  PRINT
1210  PRINT SPA(4);"Rotations at loof nodes"
1220  PRINT SPA(4);"-------------------------"
1230  K=0
1240  PRINT
1250  FOR I=1 TO Nnode
1260  K=K+5-2*Test(I)
1270  J=4-2*Test(I)
1280  IF Test(I)<>0 THEN GOTO N
1290  PRINT USING "3D,3X,2(MD.8DE,3X)";I,Q(K-J+3,1),Q(K-J+4,1)
1300  PRINT
1310 N: NEXT I
1320  PRINT
1330  IF (Prnt<>1) AND (Prnt<>3) THEN GOTO 1360
1340  LINK "SMELST",1600
1350  CALL Elstrs(Es$,Q(*),B(*),C(*),Node(*),Test(*),Nelemt,Nn
      ode,Qort,Princ)
1360  LINK "SMNDST",1600
1370  IF (Prnt<>2) AND (Prnt<>3) THEN GOTO 1390
1380  CALL Nodstr(Ns$,Q(*),B(*),C(*),Node(*),Test(*),Nelemt,Nn
      ode,Qort,Princ)
1390  END
```

```
10     ! RE-SAVE"SMINPT"
20    SUB Feinput(Xx(*),Yy(*),Zz(*),Ulx(*),Vly(*),Wlz(*),Tht(*)
      ,Thikn(*),#2,INTEGER Kode(*),Kode1(*),Test(*),Node(*),Nf
       ree,Nnode,Nelemt,Qort)
30     OPTION BASE 1
40     DIM J(100)
50     INTEGER I,J,K,L,Nspec,Nspecm
60     MAT Test=ZER
70     MAT Kode=ZER
80     MAT Kode1=ZER
90     MAT Ulx=ZER
100    MAT Vly=ZER
110    MAT Wlz=ZER
120    MAT Tht=ZER
130    FOR I=1 TO Nnode
140    READ #2;Xx(I),Yy(I),Zz(I)
150    NEXT I
160    READ #2;Node(*)
170    READ #2;K,Thik
180    FOR I=1 TO K
190    READ #2;J(I)
200    Thikn(J(I))=Thik
210    NEXT I
220    L=L+K
230    IF L>=Nelemt THEN GOTO 250
240    GOTO 170
250    FOR I=1 TO Nnode
260    FOR J=1 TO Nelemt
270    FOR K=1 TO 6+2*Qort
280    IF I<>Node(J,K) THEN GOTO 310
290    IF K<=3+Qort THEN Test(I)=1
300    GOTO A
310    NEXT K
320    NEXT J
330 A: NEXT I
340    FOR I=1 TO Nnode
350    Nfree=Nfree+5-2*Test(I)
360    NEXT I
370    MAT J=ZER
380    READ #2;Nspec
390    L=0
400    READ #2;K,Kod,Ul,Vl,Wl
410    FOR I=1 TO K
420    READ #2;J(I)
430    Kode(J(I),1)=Kod
440    Ulx(J(I),1)=Ul
450    Vly(J(I),1)=Vl
460    Wlz(J(I),1)=Wl
470    NEXT I
480    L=L+K
490    IF L>=Nspec THEN GOTO 510
500    GOTO 400
510    READ #2;Nspecm
520    L=0
530    IF Nspecm=0 THEN GOTO 630
540    READ #2;K,Kod1,Thet
550    FOR I=1 TO K
560    READ #2;J(I)
570    Kode1(J(I),1)=Kod1
580    Tht(J(I),1)=Thet
590    NEXT I
600    L=L+K
610    IF L>=K THEN GOTO 630
620    GOTO 540
630   REM THE NODAL POINT DATA
```

```
640  PRINT LIN(4)
650  PRINT SPA(15);"Nodal point data"
660  PRINT SPA(12);"------------------",LIN(2)
670   FOR I=1 TO Nnode
680   PRINT "Node";SPA(3);"X Coord";SPA(5);"Y Coord";SPA(4);"Z
     Coord";SPA(5);"Type";SPA(3);"X Disp";SPA(5);"Y DISP";SP
     A(5);"Z Disp"
690   PRINT SPA(48);"or load";SPA(5);"or load";SPA(5);"or load"
700   PRINT "----";SPA(3);"-------";SPA(5);"-------";SPA(4);"-
     ------";SPA(5);"----";SPA(3);"------";SPA(5);"------";SP
     A(5);"------"
710   FOR I=1 TO Nnode
720   PRINT USING "3D,3X,3(MD.3DE,2X),2D,2X,3(MD.3DE,2X)";I,Xx
     (I),Yy(I),Zz(I),Kode(I,1),Ulx(I,1),Vly(I,1),Wlz(I,1)
730   NEXT I
740  PRINT
750  PRINT LIN(4)
760  PRINT SPA(15),"Element data"
770  PRINT SPA(12),"------------",LIN(2)
780  PRINT "Element";SPA(12+6*Qort);"Nodal connections";SPA(15
     +4*Qort);"Material"
790  PRINT "-------";SPA(12+6*Qort);"----------------";SPA(15
     +4*Qort);"--------"
800  PRINT
810  FOR W=1 TO Nelemt
820  IF Qort=0 THEN GOTO 850
830   PRINT USING "3D,7X,8(3D,2X),13X,2D";W,Node(W,1),Node(W,2
     ),Node(W,3),Node(W,4),Node(W,5),Node(W,6),Node(W,7),Node
     (W,8),Node(W,9)
840   GOTO 860
850  PRINT USING "3D,6X,6(3D,2X),13X,2D";W,Node(W,1),Node(W,2)
     ,Node(W,3),Node(W,4),Node(W,5),Node(W,6),Node(W,7)
860  NEXT W
870  PRINT
880  SUBEND !END OF FEINPUT
890  SUB Addarray(INTEGER Test(*),Nelemt,Nnode,Add(*),Node(*),
     Qort,Nfree)
900   INTEGER Itemp,Addtemp,L,I,K,M
910   Add(1)=1
920   Add(2)=3
930   Add(3)=6
940   Addtemp=Add(3)
950   Itemp=4
960   FOR I=2 TO Nnode
970   K=I
980   N=0
990   FOR Z=1 TO Nelemt
1000  FOR L=1 TO 6+2*Qort
1010  IF Node(Z,L)<>I THEN GOTO 1040
1020  N=N+1
1030  A(N)=Z
1040  NEXT L
1050  NEXT Z
1060  FOR Z=1 TO N
1070  M=A(Z)
1080  FOR L=1 TO 6+2*Qort
1090  IF Node(M,L)<K THEN K=Node(M,L)
1100  NEXT L
1110  NEXT Z
1120  M=0
1130  IF K=1 THEN GOTO 1170
1140  FOR L=1 TO K-1
1150  M=M+5-2*Test(L)
1160  NEXT L
1170 Add(Itemp)=Addtemp+Itemp-M
1180  Add(Itemp+1)=Add(Itemp)+(Itemp+1)-M
```

```
1190  Add(Itemp+2)=Add(Itemp+1)+(Itemp+2)-M
1200  IF Test(I)=1 THEN GOTO Aud
1210  Add(Itemp+3)=Add(Itemp+2)+(Itemp+3)-M
1220  Add(Itemp+4)=Add(Itemp+3)+(Itemp+4)-M
1230 Aud: Addtemp=Add(Itemp+4-2*Test(I))
1240  Itemp=Itemp+5-2*Test(I)
1250 Next: NEXT I
1260  DISP "Note:the one dimensional stiffness matrix has";Add
      (Nfree);"elements"
1270 SUBEND !END OF ADDARRAY
```



```
2800  ! RE-SAVE"SMCNTR"
2810  SUB Constrel(Z(*),Th(*),#2,INTEGER Matno)
2820  REM ---Calculation of elastic properties for several ele
      ments.
2821  OPTION BASE 1
2822  DIM A(2)
2840  READ #2;A(*),Th(Matno)
2850  IF Matno<>1 THEN GOTO 2900
2860  PRINT LIN(3)
2870  PRINT "Material &elastic properties"
2880  PRINT "*****************************",LIN(1)
2881  PRINT
2882  PRINT "Modulous of elasticity=";A(1)
2883  PRINT
2884  PRINT "Poissons ratio=";A(2)
2885  PRINT
2921  PRINT "Density of this material Number";Matno;"=";Th(Matno)
2950  C13=C14=C15=C16=C23=C24=C25=C26=C31=C32=C34=C35=C36=C41=
      C42=C43=C46=C51=C52=C53=C56=C61=C62=C63=C64=C65=0
2960  Plane=A(1)/(1-A(2)^2)
2970  C11=Plane
2980  C12=C21=Plane*A(2)
3000  C22=Plane
3010  C33=Plane*.5*(1-A(2))
3020  Bend=Plane/12
3030  C44=C55=Bend
3040  C54=C45=Bend*A(2)
3070  C66=Bend*.5*(1-A(2))
3080  Z(Matno,1)=C11
3090  Z(Matno,2)=C12
3100  Z(Matno,3)=C13
3110  Z(Matno,4)=C14
3120  Z(Matno,5)=C15
3130  Z(Matno,6)=C16
3140  Z(Matno,7)=C22
3150  Z(Matno,8)=C23
3160  Z(Matno,9)=C24
3170  Z(Matno,10)=C25
3180  Z(Matno,11)=C26
3190  Z(Matno,12)=C33
3200  Z(Matno,13)=C34
3210  Z(Matno,14)=C35
3220  Z(Matno,15)=C36
3230  Z(Matno,16)=C44
3240  Z(Matno,17)=C45
3250  Z(Matno,18)=C46
3260  Z(Matno,19)=C55
3270  Z(Matno,20)=C56
3280  Z(Matno,21)=C66
3290  SUBEND !End of Constrel subroutine
```

```
10    ! RE-SAVE"SMASBL"
20    SUB Assembly(Q(*),K(*),Xx(*),Yy(*),Zz(*),C(*),B(*),Th(*)
      ,Omg,Ch1,Ch2,Thikn(*),Angsk(*),Ns$,Es$,INTEGER Node(*),T
      est(*),N,Add(*),I,Nskew,Nosk(*),Qort,Pr)
30    OPTION BASE 1
40    DIM Elxyzt(9,4),Wshel(13,45),Xita(2),Ro(3,3),Felvec(24+8
      *Qort),Shear(11,43),Xyzpre(8,4),Vloof(3,36),A(5),Poin(3)
      ,Bod(3),Ke(24+8*Qort,24+8*Qort)
50    INTEGER J,M,Z,Lvabz,Nozpre,Swop(6),Lnodz,Itest,Icoun1,Ic
      oun2,Jf1,Nelemt,Sub1,Sub2,Sub3,Sub4,Sub5,Sub6,Nfree,Mod(
      8),A1(8)
60    Lnodz=6+2*Qort
70    Lvabz=4*Lnodz
80    Nelemt=N
90    Nfree=I
100   IF (Pr=1) OR (Pr=3) THEN ASSIGN #3 TO Ns$
110   IF (Pr=2) OR (Pr=3) THEN ASSIGN #4 TO Es$
120   IF Qort=0 THEN GOTO 180
130   A(1)=A(2)=A(3)=A(4)=.95
140   A(5)=.2
150   W(1,1)=W(1,2)=W(2,2)=W(4,1)=-.592348878
160   W(2,1)=W(4,2)=W(3,1)=W(3,2)=.592348878
170   GOTO 220
180   A(1)=A(2)=A(3)=A(4)=1
190   W(1,1)=W(2,2)=0
200   W(1,2)=W(2,1)=W(3,1)=W(3,2)=.5
210   W(4,1)=W(4,2)=.333333
220   MAT K=ZER
230   MAT Q=ZER
240   Mod(1)=1
250   Mod(2)=4+Qort
260   Mod(3)=2
270   Mod(4)=5+Qort
280   Mod(5)=3
290   Mod(6)=6+Qort
300   Mod(7)=4
310   Mod(8)=8
320   FOR Z=1 TO Nelemt
330   FOR I=1 TO 6+2*Qort
340   A1(I)=Node(Z,I)
350   NEXT I
360   FOR I=1 TO 6+2*Qort
370   Node(Z,I)=A1(Mod(I))
380   NEXT I
390   MAT Ke=ZER
400   MAT Xyzpre=ZER
410   MAT Elxyzt=ZER
420   Nozpre=0
430   MAT Shear=ZER
440   Thick=Thikn(Z)
450  FOR I=1 TO Lnodz
460   K=INT(ABS(Node(Z,I)))
470   Elxyzt(I,4)=Thick
480   Elxyzt(I,1)=Xx(K)
490   Elxyzt(I,2)=Yy(K)
500   Elxyzt(I,3)=Zz(K)
510   NEXT I
520   FOR M=1 TO 4+Qort
530   IF M<>5 THEN GOTO 560
540   Xita(1)=Xita(2)=0
550   GOTO 580
560   Xita(1)=W(M,1)
570   Xita(2)=W(M,2)
580   CALL Haloof(Xita(*),Area,Thik,Ro(*),Poin(*),Elxyzt(*),Ws
      hel(*),Thick,Shear(*),Xyzpre(*),Vloof(*),Swop(*),Node(*)
      ,Lnodz,Nozpre,Z,Nelemt,Qort)
```

```
590    Jn=Node(Z,7+2*Qort)
600    FOR I=1 TO Lvabz
610    B(1,I)=Wshel(4,I)
620    B(2,I)=Wshel(7,I)
630    B(3,I)=Wshel(5,I)+Wshel(6,I)
640    B(4,I)=Wshel(10,I)
650    B(5,I)=Wshel(12,I)
660    B(6,I)=2*Wshel(11,I)
670    NEXT I
680    Detj=Area
690    IF (M=4) AND (Qort=0) THEN GOTO 1240
700    FOR K=1 TO 4
710    FOR I=K TO 21+8*Qort+(K-1) STEP 4
720    FOR J=I TO 21+8*Qort+(K-1) STEP 4
730    A=C(Jn,1)*B(1,I)+C(Jn,2)*B(2,I)+C(Jn,3)*B(3,I)
740    B=C(Jn,2)*B(1,I)+C(Jn,7)*B(2,I)+C(Jn,8)*B(3,I)
750    C=C(Jn,3)*B(1,I)+C(Jn,8)*B(2,I)+C(Jn,12)*B(3,I)
760    D=(C(Jn,16)*B(4,I)+C(Jn,17)*B(5,I)+C(Jn,18)*B(6,I))*Thik
       *Thik
770    E=(C(Jn,17)*B(4,I)+C(Jn,19)*B(5,I)+C(Jn,20)*B(6,I))*Thik
       *Thik
780    F=(C(Jn,18)*B(4,I)+C(Jn,20)*B(5,I)+C(Jn,21)*B(6,I))*Thik
       *Thik
790    Ke(J,I)=Ke(I,J)=Ke(J,I)+A(M)*(B(1,J)*A+B(2,J)*B+B(3,J)*C
       +B(4,J)*D+B(5,J)*E+B(6,J)*F)*Detj*Thik
800    IF J=24+8*Qort THEN GOTO L6
810    Ke(I,J+1)=Ke(J+1,I)=Ke(J+1,I)+A(M)*(B(1,J+1)*A+B(2,J+1)*
       B+B(3,J+1)*C+B(4,J+1)*D+B(5,J+1)*E+B(6,J+1)*F)*Detj*Thik
820    IF J=23+8*Qort THEN GOTO L6
830    Ke(I,J+2)=Ke(J+2,I)=Ke(J+2,I)+A(M)*(B(1,J+2)*A+B(2,J+2)*
       B+B(3,J+2)*C+B(4,J+2)*D+B(5,J+2)*E+B(6,J+2)*F)*Detj*Thik
840    IF J=22+8*Qort THEN GOTO L6
850    Ke(I,J+3)=Ke(J+3,I)=Ke(J+3,I)+A(M)*(B(1,J+3)*A+B(2,J+3)*
       B+B(3,J+3)*C+B(4,J+3)*D+B(5,J+3)*E+B(6,J+3)*F)*Detj*Thik
860 L6:  NEXT J
870    NEXT I
880    NEXT K
890    IF (Ch1=0) AND (Ch2=0) THEN GOTO 1050
900    IF Ch1=0 THEN GOTO Press
910    Weight=Area*Ch1*A(M)
920    FOR K=1 TO 24+8*Qort
930    Gash=Felvec(K)
940    FOR I=1 TO 3
950    Gash=Gash+Weight*Ro(3,I)*Wshel(I,K)
960    NEXT I
970    IF Ch2<>0 THEN GOTO 1020
980    Felvec(K)=Gash
990    GOTO 1040
1000 Press: FOR K=1 TO Lvabz
1010 Press=Ch2
1020   Gash=Felvec(K)
1030   Felvec(K)=Gash+Press*Area*A(M)*Wshel(3,K)
1040 NEXT K
1050   IF Omg=0 THEN GOTO 1180
1060   Ro=Th(Node(Z,7+2*Qort))
1070   FOR I=1 TO 3
1080   Bod(I)=Ro*Omg*Omg*Poin(I)*Thik*Area*A(M)
1090   Bod(3)=0
1100   NEXT I
1110   FOR K=1 TO Lvabz
1120   Gash=Felvec(K)
1130   FOR I=1 TO 3
1140   Gash=Gash+(Bod(1)*Ro(1,I)+Bod(2)*Ro(2,I)+Bod(3)*Ro(3,I))
       *Wshel(I,K)
1150   NEXT I
1160   Felvec(K)=Gash
```

```
1170  NEXT K
1180  IF Pr=0 THEN GOTO 1290
1190 IF (M=4+Qort) OR (Pr=2) THEN GOTO 1240
1200  MAT PRINT #3;B
1210  PRINT #3;Thik
1220  MAT PRINT #3;Ro
1230  MAT PRINT #3;Poin
1240  IF (M<>4+Qort) OR (Pr=1) OR (Pr=0) THEN GOTO 1290
1250  MAT PRINT #4;B
1260  PRINT #4;Thik
1270  MAT PRINT #4;Ro
1280  MAT PRINT #4;Poin
1290  NEXT M
1300  IF Nskew=0 THEN GOTO 1320
1310  CALL Skewedcon(Angsk(*),Ke(*),Nskew,Nosk(*),Node(*),Test
      (*),Z,Qort)
1320  Icoun1=1
1330  Icoun2=0
1340  Itest=1
1350  FOR I=1 TO 6+2*Qort
1360  N=0
1370  FOR K=1 TO Node(Z,I)
1380  N=N+5-2*Test(K)
1390  NEXT K
1400  FOR J=1 TO 6+2*Qort
1410  Jf1=0
1420  L=0
1430  FOR K=1 TO Node(Z,J)
1440  L=L+5-2*Test(K)
1450  NEXT K
1460  FOR K=1 TO J
1470  Jf1=Jf1+5-2*Test(Node(Z,K))
1480  NEXT K
1490  FOR M=4-2*Test(Node(Z,J)) TO 0 STEP -1
1500  Sub1=N-4+2*Test(Node(Z,I))
1510  Sub2=N-3+2*Test(Node(Z,I))
1520  Sub3=L-M
1530  Sub4=N-2+2*Test(Node(Z,I))
1540  IF Test(Node(Z,I))<>0 THEN GOTO 1570
1550  Sub5=N-1
1560  Sub6=N
1570  IF Sub1<Sub3 THEN GOTO Lab1
1580  K(Add(Sub1)-Sub1+Sub3)=K(Add(Sub1)-Sub1+Sub3)+Ke(Itest,J
      f1-M)
1590 Lab1: IF Sub2<Sub3 THEN GOTO Lab2
1600  K(Add(Sub2)-Sub2+Sub3)=K(Add(Sub2)-Sub2+Sub3)+Ke(Itest+1
      ,Jf1-M)
1610 Lab2: IF Sub4<Sub3 THEN GOTO Lab3
1620  K(Add(Sub4)-Sub4+Sub3)=K(Add(Sub4)-Sub4+Sub3)+Ke(Itest+2
      ,Jf1-M)
1630 Lab3: IF Test(Node(Z,I))<>0 THEN GOTO Lab
1640  IF Sub5<Sub3 THEN GOTO Lab4
1650  K(Add(Sub5)-Sub5+Sub3)=K(Add(Sub5)-Sub5+Sub3)+Ke(Itest+3
      ,Jf1-M)
1660 Lab4: IF Sub6<Sub3 THEN GOTO Lab
1670  K(Add(Sub6)-Sub6+Sub3)=K(Add(Sub6)-Sub6+Sub3)+Ke(Itest+4
      ,Jf1-M)
1680 Lab: NEXT M
1690  NEXT J
1700  IF Icoun2>Icoun1 THEN GOTO 1750
1710  Icoun1=Icoun1-1
1720  Icoun2=Icoun2+1
1730  Itest=Itest+3
1740  GOTO Nexti
1750  Icoun1=Icoun1+1
1760  Icoun2=Icoun2-1
```

```
1770  Itest=Itest+5
1780 Nexti: NEXT I
1790  IF (Ch1=0) AND (Ch2=0) AND (Omg=0) THEN GOTO 2020
1800  J=1
1810  FOR I=1 TO 6+2*Qort STEP 2
1820  M=0
1830  N=0
1840  IF Node(Z,I)=1 THEN GOTO Az1
1850  FOR K=1 TO Node(Z,I)-1
1860 M=M+5-2*Test(K)
1870  NEXT K
1880 Az1: FOR K=1 TO Node(Z,I+1)-1
1890  N=N+5-2*Test(K)
1900 NEXT K
1910  Q(M+1,1)=Q(M+1,1)+Felvec(J)
1920  Q(M+2,1)=Q(M+2,1)+Felvec(J+1)
1930  Q(M+3,1)=Q(M+3,1)+Felvec(J+2)
1940  Q(N+1,1)=Q(N+1,1)+Felvec(J+3)
1950  Q(N+2,1)=Q(N+2,1)+Felvec(J+4)
1960  Q(N+3,1)=Q(N+3,1)+Felvec(J+5)
1970  Q(N+4,1)=Q(N+4,1)+Felvec(J+6)
1980  Q(N+5,1)=Q(N+5,1)+Felvec(J+7)
1990  J=J+8
2000  NEXT I
2010  MAT Felvec=ZER
2020  FOR I=1 TO 6+2*Qort
2030  Node(Z,I)=A1(I)
2040  NEXT I
2050  NEXT Z
2060  IF (Pr=3) OR (Pr=1) THEN ASSIGN #3 TO *
2070  IF (Pr=3) OR (Pr=2) THEN ASSIGN #4 TO *
2080  N=Nelemt
2090  I=Nfree
2100  SUBEND !End of Assembly
2110  SUB Haloof(Xita(*),Area,Thik,Fram(*),Poin(*),Elxyzt(*),W
      shel(*),Thick,Shear(*),Xyzpre(*),Vloof(*),INTEGER Swop(*
      ),Lnods(*),Lnodz,Nozpre,Nel,Nelz,Qort)
2120  OPTION BASE 1
2130  DIM Areav(3),Frame(3,3),Gensid(6,4),Sigt(3),Thikdd(3,3),
      Trans(2,2),Xgaus(4,4),Xiloof(9,4),Xlocal(2),Xyzdd(3,3),W
      corn(10,3),Wloof(10,3),Point(3)
2140  INTEGER Lnodza,Lvabz,Limz,Lvabza,Lvabzz,Lnomax,Lnodzh
2150  INTEGER Nstage
2160  Lnodza=Lnodz+1
2170  Lvabz=Lnodz*4
2180  Limz=3*Lnodz/2-1
2190  Lvabza=Lvabz+1
2200  Lvabzz=Lvabz+Limz
2210  Lnomax=Lnodz
2220  MAT READ Gensid
2230  DATA 1,0,1,0,-1,1,0,1,0,-1,-1,0,-.5,1,0,-1,-.5,1,0,1,-.5
      ,1,0,1
2240  MAT READ Xiloof
2250  DATA .211324866,0,-.577350269,-1,.788675134,0,.577350269
      ,-1,.788,.21132486651,1,-.577350269,.211324866,.788,1,.5
      77350269,0,.788,.577350269,1,0,.211324866
2260  DATA -.577350269,1,.3333333,.3333333,-1,.577350269,0,0,-
      1,-.577350269,0,0,0,0
2270  MAT READ Xgaus
2280  DATA 0,.5,-.577350269,-.577350269,.5,0,.577350269,-.5773
      50269,.5,.5,.577350269,.577350269,.5,.5,-.577350269,.577
      350269
2290  MAT Wshel=ZER
2300  REM ---Generate Nstage to define the path through  Haloof
2310  IF (Lnodz<>6) AND (Lnodz<>8) AND (Lvabz<>Lnodz*4) THEN GOTO
      Message
```

```
2320  Nstage=4
2330  IF Lnodz<>Nozpre THEN Nstage=2
2340  Nozpre=Lnodz
2350  FOR Lnod=1 TO Lnodz
2360  FOR Nx=1 TO 4
2370  IF Elxyzt(Lnod,Nx)<>Xyzpre(Lnod,Nx) THEN Nstage=2
2380  NEXT Nx
2390  NEXT Lnod
2400  IF Nstage=4 THEN GOTO N18
2410  REM ---Initialization for new element ,Nstage=1 find the
      center coordinate.
2420  FOR Nx=1 TO 4
2430  Gash=0
2440  Lnodzh=Lnodz/2
2450  FOR Korn=1 TO Lnodzh
2460  FOR K=1 TO 2
2470  Gash=Gash+8*Elxyzt(2*Korn+K-2,Nx)/(216*K-408-Lnodz*(21*K
      -41))
2480  NEXT K
2490  NEXT Korn
2500  Elxyzt(9,Nx)=Gash
2510  NEXT Nx
2520  REM ---Diagnostic for a new element.relate coordinate centre
2530  FOR I=1 TO Lnodz
2540  IF Elxyzt(I,4)<=0 THEN GOTO Message
2550  IF I=Lnodz THEN GOTO N9
2560  Ja=I+1
2570  FOR J=Ja TO Lnodz
2580  IF ABS(Lnods(Nel,I)=ABS(Lnods(Nel,J))) THEN GOTO Message
2590  FOR K=1 TO 3
2600  IF Elxyzt(I,K)<>Elxyzt(J,K) THEN GOTO N8
2610  NEXT K
2620  GOTO Message
2630 N8: NEXT J
2640 N9: FOR Nx=1 TO 4
2650  IF Nx<>4 THEN Elxyzt(I,Nx)=Elxyzt(I,Nx)-Elxyzt(9,Nx)
2660  Xyzpre(I,Nx)=Elxyzt(I,Nx)!recognise the old element
2670  NEXT Nx
2680  NEXT I
2690  REM -- interpolate to estimate normal thickness at loof
      nodes.
2700  Vloof(1,Lvabza)=Elxyzt(9,4)
2710  FOR Nside=1 TO 6
2720  Swop(Nside)=1
2730  NEXT Nside
2740  Last=Lnodz-1
2750  FOR Next=1 TO Lnodz STEP 2
2760  Mid=Last+1
2770  IF ABS(Lnods(Nel,Next))<ABS(INT(Lnods(Nel,Last))) THEN S
      wop(Mid/2)=-1
2780  Vloof(1,4*Last-3)=.455341801*Elxyzt(Last,4)+.666666667*E
      lxyzt(Mid,4)-.122008468*Elxyzt(Next,4)
2790  Vloof(1,4*Mid-3)=-.122008468*Elxyzt(Last,4)+.666666667*E
      lxyzt(Mid,4)+.455341801*Elxyzt(Next,4)
2800  ! ---Check that the midside nodes are reasonably central
2810  Gash=0
2820  Gish=0
2830  Gush=0
2840  FOR I=1 TO 3
2850  Elmid=Elxyzt(Mid,I)
2860  Gash=Gash+(Elxyzt(Next,I)-Elmid)^2
2870  Gish=Gish+(Elxyzt(Last,I)-Elmid)^2
2880  Gush=Gush+(Elxyzt(Last,I)+Elxyzt(Next,I)-Elmid-Elmid)^2
2890  NEXT I
2900  Last=Next
2910  NEXT Next
```

```
2920  REM ---Organise loop around loof nodes for Nstage=2
2930  REM ---FOR Nstage=2 TO 4
2940 N15: Nloof=0
2950 N16: Nloof=Nloof+1
2960  FOR I=1 TO 2
2970  IF (Nstage=2) OR (Nloof<=Lnodz) THEN Xlocal(I)=Xiloof(Nl
      oof,Lnodz+I-6)
2980  REM ---also around the integrating points if nstage=3
2990  IF (Nstage=3) AND (Nloof>Lnodz) THEN Xlocal(I)=Xgaus(Nlo
      of-Lnodz,Lnodz+I-6)
3000  NEXT I
3010  GOTO N23
3020  REM ---Test whether input point is a loof node plus or m
      inus .0001
3030 N18: FOR I=1 TO 2
3040  Xlocal(I)=Xita(I)
3050  NEXT I
3060  Nloof=Lnodza
3070  FOR Maybe=1 TO Lnodz
3080  FOR I=1 TO 2
3090  IF ABS(Xlocal(I)-Xiloof(Maybe,Lnodz+I-6))>.0001 THEN GOTO
      N22
3100  NEXT I
3110  Nloof=Maybe
3120 N22: NEXT Maybe
3130 N23:!
3140  CALL Sfr(Xlocal(*),Wcorn(*),Wloof(*),Lnodz,Nstage)
3150  K=0
3160  FOR I=1 TO 3
3170  FOR J=1 TO 3
3180  Gash=0
3190  FOR L=1 TO Lnodz
3200  Gash=Gash+Wcorn(L+K,I)*Elxyzt(L,J)
3210  NEXT L
3220  Xyzdd(J,I)=Gash
3230  IF Nstage=2 THEN GOTO N26
3240  Gash=0
3250  FOR L=1 TO Lnodza
3260  Gash=Gash+Wloof(L+K,I)*Vloof(J,4*L-1)
3270  NEXT L
3280  Thikdd(J,I)=Gash
3290 N26: NEXT J
3300  K=1
3310  NEXT I
3320  REM ---Create a vector area=Varea,at given point Xi,Eta
3330  Areav(1)=Xyzdd(2,2)*Xyzdd(3,3)-Xyzdd(3,2)*Xyzdd(2,3)
3340  Areav(2)=Xyzdd(3,2)*Xyzdd(1,3)-Xyzdd(1,2)*Xyzdd(3,3)
3350  Areav(3)=Xyzdd(1,2)*Xyzdd(2,3)-Xyzdd(2,2)*Xyzdd(1,3)
3360  Areasq=Areav(1)*Areav(1)+Areav(2)*Areav(2)+Areav(3)*Areav(3)
3370  IF Areasq<=0 THEN GOTO Message
3380  Area=SQR(Areasq)
3390  REM ---Normalize vector area into Frame,Col 3 as loca unit
      normal Z
3400  REM ---Col 2 of Frame Becomes unit y around edge
3410  FOR I=1 TO 3
3420  Frame(I,3)=Areav(I)/Area
3430  Gash=0
3440  FOR J=1 TO 2
3450  Gash=Gash+Gensid(INT((Nloof+1)/2),Lnodz+J-6)*Xyzdd(I,J+1)
3460  NEXT J
3470  Frame(I,2)=Gash
3480  NEXT I
3490  REM ---Normalize Y,and implement swop by reversing sign of
      y.
3500  REM ---Put approximation vector thickness into vloof for
      nstage=2
```

```
3510  Sidesq=Frame(1,2)*Frame(1,2)+Frame(2,2)*Frame(2,2)+Frame
      (3,2)*Frame(3,2)
3520  IF Sidesq<=0 THEN GOTO Message
3530  Side=SQR(Sidesq)
3540  FOR I=1 TO 3
3550  Frame(I,2)=Frame(I,2)*Swop(INT((Nloof+1)/2))/Side
3560  IF Nstage<>2 THEN GOTO N31
3570  Vloof(I,4*Nloof-2)=Frame(I,2)
3580  Vloof(I,4*Nloof-1)=Frame(I,3)*Vloof(1,4*Nloof-3)
3590  Vloof(I,4*Nloof)=Frame(I,3)
3600 N31: NEXT I
3610  REM --- Clo 1 is unit X,the outward pointing in plane normal
3620  Frame(1,1)=Frame(2,2)*Frame(3,3)-Frame(3,2)*Frame(2,3)
3630  Frame(2,1)=Frame(3,2)*Frame(1,3)-Frame(1,2)*Frame(3,3)
3640  Frame(3,1)=Frame(1,2)*Frame(2,3)-Frame(2,2)*Frame(1,3)
3650  REM --- Check that normals are reasonably parrallel while
      Nstage=2
3660  IF Nstage>2 THEN GOTO N35
3670  IF Nloof=1 THEN GOTO N67
3680  Kz=4*Nloof-4
3690  FOR K=4 TO Kz STEP 4
3700  Point(1)=Vloof(2,4*Nloof)*Vloof(3,K)-Vloof(3,4*Nloof)*Vl
      oof(2,K)
3710  Point(2)=Vloof(3,4*Nloof)*Vloof(1,K)-Vloof(1,4*Nloof)*Vl
      oof(3,K)
3720  Point(3)=Vloof(1,4*Nloof)*Vloof(2,K)-Vloof(2,4*Nloof)*Vl
      oof(1,K)
3730  Cossq=Point(1)*Point(1)+Point(2)*Point(2)+Point(3)*Point(3)
3740  IF Cossq>.75 THEN GOTO Message
3750  NEXT K
3760  REM ---Place contribution of cental node in Vloof(Nstafe=2)
3770  IF Nloof<=Lnodz THEN GOTO N67
3780  Thikc=Vloof(1,Lvabza)
3790  FOR I=1 TO 3
3800  FOR J=1 TO 2
3810  Vloof(I,Lvabz+J)=Frame(I,J)*Thikc
3820  NEXT J
3830  NEXT I
3840  GOTO N67
3850  REM ---Create the 2*2 Jacobian matrix and invert it.(Nst
      age=3 or 4)
3860 N35: FOR J=1 TO 2
3870  FOR I=1 TO 2
3880  Trans(J,I)=Frame(1,I)*Xyzdd(1,J+1)+Frame(2,I)*Xyzdd(2,J+
      1)+Frame(3,I)*Xyzdd(3,J+1)
3890  NEXT I
3900  NEXT J
3910  Gash=Trans(1,1)
3920  Trans(1,1)=Trans(2,2)/Area
3930  Trans(2,2)=Gash/Area
3940  Trans(1,2)=-Trans(1,2)/Area
3950  Trans(2,1)=-Trans(2,1)/Area
3960  REM ---Transform Wcorn and Wloof into local axes
3970  FOR N=1 TO Lnodza
3980  FOR I=1 TO 2
3990  Gash=0
4000  Gish=0
4010  FOR J=1 TO 2
4020  Gash=Gash+Trans(I,J)*Wcorn(N+1,J+1)
4030  Gish=Gish+Trans(I,J)*Wloof(N+1,J+1)
4040  NEXT J
4050  Wcorn(N,I+1)=Gash
4060  Wloof(N,I+1)=Gish
4070  NEXT I
4080  NEXT N
4090  REM ---put the thickness and derivatives into loca coord
      inates system
```

```
4100  FOR I=1 TO 3
4110  FOR J=1 TO 2
4120  Point(J)=0
4130  FOR K=1 TO 2
4140  Point(J)=Point(J)+Trans(J,K)*Thikdd(I,K+1)
4150  NEXT K
4160  NEXT J
4170  FOR J=1 TO 2
4180  Thikdd(I,J+1)=Point(J)
4190  NEXT J
4200  NEXT I
4210  FOR J=1 TO 3
4220  FOR I=1 TO 3
4230  Point(I)=Thikdd(1,J)*Frame(1,I)+Thikdd(2,J)*Frame(2,I)+T
      hikdd(3,J)*Frame(3,I)
4240  NEXT I
4250  FOR I=1 TO 3
4260  Thikdd(I,J)=Point(I)
4270  NEXT I
4280  NEXT J
4290  Thik=Thikdd(3,1)
4300  IF Thik<=0 THEN GOTO Message
4310  REM ---Find the change in local X,Y derivatives across t
      hickness of shell
4320  FOR Lnod=1 TO Lnodza
4330  IF Nstage<>4 THEN GOTO N51
4340  FOR I=2 TO 3
4350  Gash=0
4360  FOR J=1 TO 2
4370  Gash=Gash-Thikdd(J,I)*Wcorn(Lnod,J+1)
4380  NEXT J
4390  Point(I)=Gash
4400  NEXT I
4410  REM ---Create Wshel=shape function array displacement terms
      first
4420 N51: Korn=INT((Lnod+1)/2)
4430  FOR K=1 TO 3
4440  Kol=2*Korn+3*Lnod+K-5
4450  IF Lnod>Lnodz THEN Kol=5*Lnodz+2+K
4460  FOR N=1 TO 3
4470  Fact=Frame(K,N)
4480  Wshel(N,Kol)=Wcorn(Lnod,1)*Fact
4490  IF (Nstage=4) AND (N=3) THEN Fact=0
4500  FOR Nd=2 TO 3
4510  Wshel(N+N+Nd,Kol)=Wcorn(Lnod,Nd)*Fact
4520  NEXT Nd
4530  NEXT N
4540  FOR N=1 TO 2
4550  FOR Nd=2 TO 3
4560  Wshel(N+7,Kol)=Wshel(N+7,Kol)-Thikdd(Nd-1,1)*Wshel(N+N+N
      d,Kol)/Thik
4570  IF Nstage=4 THEN Wshel(N+N+Nd+6,Kol)=(Point(Nd)*Frame(K,
      N)+Thikdd(3,Nd)*Wcorn(Lnod,N+1)*Frame(K,3))/Thik
4580  NEXT Nd
4590  NEXT N
4600  NEXT K
4610  REM ---Introduce rotation terms with bending action Wshel
4620  FOR L=1 TO 2
4630  Kol=(L-1)*4*Lnodz+(2-L)*6*Korn+Lnod
4640  IF Lnod>Lnodz THEN Kol=5*Lnodz+3-L
4650  FOR N=1 TO 2
4660  Fact=Vloof(1,4*Lnod+L-4)*Frame(1,N)+Vloof(2,4*Lnod+L-4)*
      Frame(2,N)+Vloof(3,4*Lnod+L-4)*Frame(3,N)
4670  Wshel(N+7,Kol)=Fact*Wloof(Lnod,1)/Thik
4680  IF Nstage<>4 THEN GOTO N56
4690  FOR Nd=2 TO 3
```

```
4700  Wshel(N+N+Nd+6,Kol)=Fact*Wloof(Lnod,Nd)/Thik
4710  NEXT Nd
4720 N56: NEXT N
4730  NEXT L
4740  NEXT Lnod
4750  REM ---Combine last three Columns of Wshel to Create normal
      deflection
4760  IF Lnodz=6 THEN GOTO N61
4770  Iz=3*Nstage+1
4780  FOR I=1 TO Iz
4790  Gash=0
4800  FOR K=1 TO 3
4810  Gash=Gash+Wshel(I,42+K)*Vloof(K,4*Lnodz+4)
4820  NEXT K
4830  Wshel(I,43)=Gash
4840  NEXT I
4850 N61: IF Nstage=4 THEN GOTO N86
4860  REM ---Create array shear for introducing the constraints
4870  IF Nloof>Lnodz THEN GOTO N63
4880  FOR I=1 TO Lvabzz
4890  Shear(Nloof,I)=Wshel(9,I)
4900  Shear(11,I)=Shear(11,I)+Wshel(8,I)*Side*Thik*Swop(INT((N
      loof+1)/2))
4910  NEXT I
4920  GOTO N67
4930 N63: FOR Kol=1 TO Lvabzz
4940  FOR Nxy=1 TO 2
4950  Gash=Shear(Lnodz+Nxy,Kol)
4960  FOR Mxy=1 TO 2
4970  Fact=Frame(1,Mxy)*Vloof(1,4*Lnodz+Nxy)+Frame(2,Mxy)*Vloo
      f(2,4*Lnodz+Nxy)+Frame(3,Mxy)*Vloof(3,4*Lnodz+Nxy)
4980  Gash=Gash+Wshel(Mxy+7,Kol)*Area*Thik*Fact
4990  NEXT Mxy
5000  Shear(Lnodz+Nxy,Kol)=Gash
5010  NEXT Nxy
5020  NEXT Kol
5030  REM ---Complete loop around loof nodes to create Vloof or
      shear
5040 N67: IF Nloof<=Lnodz THEN GOTO N16
5050 IF (Nstage=3) AND (Nloof<3*Lnodz/2) THEN GOTO N16
5060  IF Nstage<>2 THEN GOTO N76
5070  REM ---Create plus or minus sum of thickness vector at loof
      nodes Nstage=2
5080  FOR I=1 TO 3
5090  Gash=0
5100  FOR N=3 TO Lvabz STEP 4
5110  Gash=-Gash+Vloof(I,N)
5120  NEXT N
5130  Sigt(I)=Gash
5140  REM ---And 3*3 matrix associated with it stored in Xyzdd
5150  FOR J=1 TO 3
5160  Gash=0
5170  IF I=J THEN Gash=Lnodz
5180  FOR N=2 TO Lvabz STEP 4
5190  Gash=Gash-Vloof(I,N)*Vloof(J,N)
5200  NEXT N
5210  Xyzdd(I,J)=Gash
5220  NEXT J
5230  NEXT I
5240  REM ---Get the adjucate of the 3*3 symmetric positive def
5250  K=3
5260  FOR I=1 TO 3
5270  Frame(1,K)=Xyzdd(2,I)*Xyzdd(3,6-I-K)-Frame(3,I)*Xyzdd(2,
      6-I-K)
5280  Frame(2,K)=Xyzdd(3,I)*Xyzdd(1,6-I-K)-Frame(1,I)*Xyzdd(3,
      6-I-K)
```

```
5290  Frame(3,K)=Xyzdd(1,I)*Xyzdd(2,6-I-K)-Frame(2,I)*Xyzdd(1,
      6-I-K)
5300  K=I
5310  NEXT I
5320  Determ=Xyzdd(1,1)*Frame(1,1)+Xyzdd(2,1)*Frame(2,1)+Xyzdd
      (3,1)*Frame(3,1)
5330  FOR I=1 TO 3
5340  Prod=Frame(1,I)*Sigt(1)+Frame(2,I)*Sigt(2)+Frame(3,I)*Si
      gt(3)
5350  Point(I)=Prod/Determ
5360  NEXT I
5370  REM ---Correct Vector thickness in Vloof
5380  Fact=1
5390  FOR N=2 TO Lvabz STEP 4
5400  Fact=-Fact
5410  Prod=Point(1)*Vloof(1,N)+Point(2)*Vloof(2,N)+Point(3)*Vl
      oof(3,N)
5420  FOR I=1 TO 3
5430  Vloof(I,N+1)=Vloof(I,N+1)-Fact*(Point(I)-Prod*Vloof(I,N))
5440  NEXT I
5450  REM ---Create Differential vectors to define rotations
5460  ! this complete work for Nstage=2
5470  Tfirst=Vloof(1,N-1)
5480   Vloof(1,N-1)=Vloof(2,N)*Vloof(3,N+1)-Vloof(3,N)*Vloof(2
       ,N+1)
5490   Vloof(2,N-1)=Vloof(3,N)*Vloof(1,N+1)-Vloof(1,N)*Vloof(3
       ,N+1)
5500   Vloof(3,N-1)=Vloof(1,N)*Vloof(2,N+1)-Vloof(2,N)*Vloof(1
       ,N+1)
5510   FOR I=1 TO 3
5520  Vloof(I,N)=Vloof(I,N)*Tfirst
5530  NEXT I
5540  NEXT N
5550  Nz=4*Lnodza
5560  Nstage=3
5570  GOTO N15
5580 N76: !
5590  REM ---Shear has been created in Nloof loop for Nstage=3
      ,Choose pivot for reducing array shear and do row changi
      ng
5600  FOR Lim=1 TO Limz
5610  Kp=Lvabz+Lim
5620  Pivot=0
5630  FOR L=Lim TO Limz
5640  IF ABS(Pivot)>ABS(Shear(L,Kp)) THEN GOTO N79
5650  Lbig=L
5660  Pivot=Shear(Lbig,Kp)
5670 N79: NEXT L
5680 FOR K=1 TO Lvabzz
5690  Change=Shear(Lbig,K)
5700  Shear(Lbig,K)=Shear(Lim,K)
5710  Shear(Lim,K)=Change/Pivot
5720  NEXT K
5730  REM --Reduce array shear to create constraints matrix this
      complete work for Nstage=3
5740  FOR Nrow=1 TO Limz
5750  Fact=Shear(Nrow,Kp)
5760  IF (Nrow=Lim) OR (Fact=0) THEN GOTO N82
5770  FOR Kol=1 TO Lvabzz
5780  Shear(Nrow,Kol)=Shear(Nrow,Kol)-Fact*Shear(Lim,Kol)
5790  NEXT Kol
5800 N82: NEXT Nrow
5810  NEXT Lim
5820  Nstage=4
5830  GOTO N18
5840  REM ---Use array Shear to constrain Wshel at given Point
      ,Xi,Eta
```

```
5850 N86: FOR I=1 TO Lvabz
5860  FOR J=1 TO 13
5870  Gash=Wshel(J,I)
5880  FOR K=1 TO Limz
5890  Gash=Gash-Wshel(J,K+Lvabz)*Shear(K,I)
5900  NEXT K
5910  Wshel(J,I)=Gash
5920  NEXT J
5930  NEXT I
5940  REM ---Implement Swop exchange two normal slopes.
5950  FOR N=8 TO Lvabz STEP 8
5960  IF Swop(N/8)=1 THEN GOTO N92
5970  FOR J=1 TO 13
5980  Change=Wshel(J,N)
5990  Wshel(J,N)=Wshel(J,N-1)
6000  Wshel(J,N-1)=Change
6010  NEXT J
6020 N92: NEXT N
6030  REM ---Assemble Uxz,Uyz,Vxz,Vyz to create Wxx,Wxy,Wyy
6040  FOR N=1 TO Lvabz
6050  Wshel(10,N)=-Wshel(10,N)
6060  Wshel(11,N)=-.5*(Wshel(11,N)+Wshel(12,N))
6070  Wshel(12,N)=-Wshel(13,N)
6080  NEXT N
6090  REM ---Put point fram in common also area side with inte
      grating factors
6100  Area=Area*(Lnodz-5.6)/2.4
6110  Side=Side*(Lnodz-4)/4
6120  FOR I=1 TO 3
6130 Poin(I)=Xyzdd(I,1)+Elxyzt(9,I)
6140  FOR J=1 TO 3
6150  Fram(I,J)=Frame(I,J)
6160  NEXT J
6170  NEXT I
6180  GOTO 6210
6190 Message: PRINT "Error ----Please check the input data";
6200  STOP
6210  SUBEND!To subroutine shell
6220  SUB Sfr(Xlocal(*),Wcorn(*),Wloof(*),INTEGER Lnodz,Nstage)
6230  OPTION BASE 1
6240  MAT Wcorn=ZER
6250  MAT Wloof=ZER
6260  DIM Md(4),Termv(46),Coef(247)
6270  MAT READ Md
6280  DATA 8,43,90,171
6290  MAT Termv=ZER
6300  Termv(2)=1
6310  Xi=Xlocal(1)
6320  Eta=Xlocal(2)
6330  MAT READ Coef
6340  DATA 1,-3,-3,2,4,2,0,4,0,-4,-4,0,0!1
6350  DATA -1,0,2,0,0,0,0,0,0,4,0,0,0,-1,0,0!2
6360  DATA 2,0,0,4,0,-4,-4,.910683603,1.577350269!3
6370  DATA -6.041451884,-6.196152423,2.464101615,8.92820323,1.
      732050808!4
6380  DATA -.244016936,.422649731,2.041451884,4.196152423,-4.4
      64101615!5
6390  DATA -4.92820323,-1.732050808,.333333333,-1.422649731,-2
      .577350269!6
6400  DATA -1.464101615,5,5.464101615,1.732050808,.333333333!7
6410  DATA -2.577350269,-1.422649731,5.464101615,5,-1.464101615!8
6420  DATA -1.732050808,-.244016936,2.041451884,.422649731,-4.
      9282032319
6430  DATA -4.464101615,4.196152423,1.732050807,.910683602,-6.
      041451884!10
6440  DATA 1.577350269,8.92820323,2.464101615,-6.196152422,-1.
      732050807!11
```

```
6450  DATA -1,6,6,-6,-6,-6,0,-.25,0,0,.25,.25,.25,-.25,-.25,0!12
6460  DATA .5,0,-.5,-.5,0,0,0,.5,0,-.25,0,0,.25,-.25,.25,.25,-
      .25,0!13
6470  DATA .5,.5,0,0,0,-.5,-.5,0,0,-.25,0,0,.25,.25,.25,.25,.2
      5,0!14
6480  DATA .5,0,.5,-.5,0,0,0,-.5,0,-.25,0,0,.25,-.25,.25,-.25,
      .25,0!15
6490  DATA .5,-.5,0,0,0,-.5,.5,0,0,1,0,0,-1,0,-1,0,0,1!16
6500  DATA 0,.216506351,-.375,-.09375!0
6510  DATA .216506351,.28125,-.649519053,.375,-.324759526!1
6520  DATA 0,-.216506351,-.375,-.09375,-.216506351!2
6530  DATA .28125,.649519053,.375,.324759526,0!3
6540  DATA .375,.216506351,.28125,-.216506351,-.09375!4
6550  DATA -.375,-.649519053,-.324759526,0,.375!5
6560  DATA -.216506351,.28125,.216506351,-.09375,-.375!6
6570  DATA .649519053,.324759526,0,-.216506351,.375!7
6580  DATA -.09375,.216506351,.28125,.649519053,-.375!8
6590  DATA -.324759526,0,.216506351,.375,-.09375!9
6600  DATA -.216506351,.28125,-.649519053,-.375,.324759526!1
6610  DATA 0,-.375,-.216506351,.28125,-.216506351!2
6620  DATA -.09375,.375,.649519053,-.324759526,0!3
6630  DATA -.375,.216506351,.28125,.216506351,-.09375!4
6640  DATA .375,-.649519053,.324759526,1,0,0,-.75,0,-.75,0,0,0!5
6650  Ia=2
6660  Nz=INT((Lnodz+Nstage-3)/2)
6670  FOR N=1 TO Nz
6680  Ian=Ia+N
6690  N2=N+15
6700  N3=N+30
6710  FOR J=Ia TO Ian
6720  Termv(J+N)=Termv(J)*Xi
6730  Termv(J+N2)=Termv(J)*(Ian-J)
6740  Termv(J+N3)=Termv(J-1)*(J-Ia)
6750  NEXT J
6760  Ia=Ian
6770  Termv(Ia+N)=Termv(Ia-1)*Eta
6780  NEXT N
6790  FOR I=8 TO 38 STEP 15
6800  IF Lnodz=6 THEN Termv(I)=2*(Termv(I)-Termv(I+3))+3*(Term
      v(I+1)-Termv(I+2))
6810  IF Lnodz=8 THEN Termv(I)=Termv(I+2)
6820  IF Lnodz=8 THEN Termv(I+2)=Termv(I+6)
6830  NEXT I
6840  Nfoisz=INT((Nstage+1)/2)
6850  FOR Nfois=1 TO Nfoisz
6860  Nz=INT(3*Lnodz/2)+Nfois-4
6870  IF Nz<>10 THEN GOTO N12
6880  Nz=9
6890  FOR I=10 TO 40 STEP 15
6900  Termv(I)=Termv(I+3)-Termv(I+5)
6910  NEXT I
6920 N12: K=0
6930  FOR I=1 TO 3
6940  FOR N=1 TO Nz
6950  Gash=0
6960  Mdel=Md(Lnodz+Nfois-6)+N*Nz-15*I
6970  Ma=16*I-14
6980  Mz=15*I+Nz-14
6990  FOR M=Ma TO Mz
7000  Gash=Gash+Termv(M)*Coef(M+Mdel)
7010  NEXT M
7020  IF Nfois=1 THEN Wcorn(N+K,I)=Gash
7030  IF Nfois=2 THEN Wloof(N+K,I)=Gash
7040  NEXT N
7050  K=1
7060  NEXT I
```

```
7070  NEXT Nfois
7080  SUBEND
7090  SUB Skewedcon(Angsk(*),Ke(*),INTEGER Nskew,Nosk(*),Node(
      *),Test(*),Z,Qort)
7100  OPTION BASE 1
7110  DIM S(24+8*Qort,24+8*Qort),S1(24+8*Qort)
7120  MAT S=IDN
7130  K=0
7140  FOR I=1 TO 6+2*Qort
7150  K=K+5-2*Test(Node(Z,I))
7160  FOR J=1 TO Nskew
7170  IF Node(Z,I)<>Nosk(J) THEN GOTO Skb
7180  A=K-4+2*Test(Node(Z,I))
7190  S(A+1,A+1)=S(A,A)=COS(Angsk(J)*.0174533)
7200  S(A+1,A)=SIN(Angsk(J)*.0174533)
7210  S(A,A+1)=-S(A+1,A)
7220 Skb: NEXT J
7230  NEXT I
7240  FOR I=1 TO 24+8*Qort
7250  FOR J=1 TO 24+8*Qort
7260  Gash=0
7270  FOR K=1 TO 24+8*Qort
7280  Gash=Gash+S(K,J)*Ke(K,I)
7290  NEXT K
7300  S1(J)=Gash
7310  NEXT J
7320  FOR K=1 TO 24+8*Qort
7330  Ke(K,I)=S1(K)
7340  NEXT K
7350  NEXT I
7360  FOR I=1 TO 24+8*Qort
7370  FOR J=1 TO 24+8*Qort
7380  Gash=0
7390  FOR K=1 TO 24+8*Qort
7400  Gash=Gash+Ke(I,K)*S(K,J)
7410  NEXT K
7420  S1(J)=Gash
7430  NEXT J
7440  FOR K=1 TO 24+8*Qort
7450  Ke(I,K)=S1(K)
7460  NEXT K
7470  NEXT I
7480  SUBEND!
```

PROGRAMME STORED IN FILE : SMLDAP Page 1
LISTED ON : 6/7/83

```
10     ! RE-SAVE"SMLDAP"
20     SUB Loadapp(B,C,L,E,D(*),INTEGER A,G,K,F,Irot)
30     IF A=1 THEN GOTO Aud1
40     IF (A=3) OR (A=5) THEN GOTO Aud2
50     IF A=4 THEN GOTO Aud3
60     D(K-Irot,F)=D(K-Irot,F)+B
70     IF A=2 THEN GOTO Aud1
80     IF A=6 THEN GOTO Aud3
90 Aud2:   D(K-Irot+1,F)=D(K-Irot+1,F)+C
100    IF (A=3) OR (A=7) THEN GOTO Aud1
110 Aud3:  D(K-Irot+2,F)=D(K-Irot+2,F)+L
120        IF (G=0) OR (Irot<>4) THEN GOTO Aud1
130    D(K-Irot+3,F)=D(K-Irot+3,F)+E
140    D(K-Irot+4,F)=D(K-Irot+4,F)+E
150 Aud1:  SUBEND !end of loading
```

PROGRAMME STORED IN FILE : SMSKLD Page 1
LISTED ON : 6/7/83

```
10     ! RE-SAVE"SMSKLD"
20     SUB Skewload(Q(*),Angsk(*),INTEGER Test(*),Nsetf,Nosk(*)
       ,Nskew,Op)
30     INTEGER A,I,J
40     FOR I=1 TO Nskew
50     A=0
60     Sn=SIN(Angsk(I)*.0174533)
70     Cn=COS(Angsk(I)*.0174533)
80     FOR J=1 TO Nsetf
90     FOR K=1 TO Nosk(I)
100    A=A+5-2*Test(K)
110    NEXT K
120    M=Q(A-4+2*Test(Nosk(I)),J)
130    N=Q(A-3+2*Test(Nosk(I)),J)
140    O=Q(A-2+2*Test(Nosk(I)),J)
150    IF Test(Nosk(I))<>0 THEN GOTO 180
160    P=Q(A-1,J)
170    R=Q(A,J)
180    Q(A-4+2*Test(Nosk(I)),J)=Cn*M+Sn*Op*N
190    Q(A-3+2*Test(Nosk(I)),J)=-Op*Sn*M+Cn*N
200    NEXT J
210    NEXT I
220    SUBEND! End of Skewload
```

PROGRAMME STORED IN FILE : SMYVBS Page 1
LISTED ON : 6/7/83

```
10     ! RE-SAVE"SMYVBS"
20     SUB Bounconst(U,R(*),Ak(*),INTEGER N,Neq,F,A(*))
30     INTEGER M,K,Cj
40     IF N<>1 THEN GOTO 70
50     Cj=1
60     GOTO 80
70     Cj=N-(A(N)-A(N-1))+1
80     FOR K=Cj TO N
90     R(K,F)=R(K,F)-Ak(A(N)-N+K)*U
100    Ak(A(N)-N+K)=0
110    NEXT K
120    IF N+1>Neq THEN GOTO Lz
130    FOR K=N+1 TO Neq
140    Cj=K-(A(K)-A(K-1))+1
150    IF Cj>N THEN GOTO 180
```

```
160  R(K,F)=R(K,F)-Ak(A(K)-K+N)*U
170  Ak(A(K)-K+N)=0
180  NEXT K
190 Lz: Ak(A(N))=1
200  R(N,F)=U
210  SUBEND !END OF BOUNCONST
220  SUB Symvbsol(A(*),L(*),B(*),INTEGER S(*),N,R)
230  INTEGER G,H,I,J,K,M,P,Q,T,U,V
240  H=0
250  FOR I=1 TO N
260  T=I+H-S(I)+1
270  G=H+1
280  P=S(T)-1
290  FOR J=T TO I-1
300  Q=P+1
310  H=H+1
320  P=S(J)
330  K=J+Q-P
340  V=H-P
350  U=G
360  Y=A(H)
370  IF K>T THEN U=U+K-T
380  FOR U=U TO H-1
390  Y=Y-L(U)*L(U-V)
400  NEXT U
410  Y=Y/L(H-V)
420  L(H)=Y
430  FOR M=1 TO R
440  B(I,M)=B(I,M)-B(J,M)*Y
450  NEXT M
460  NEXT J
470  Y=A(H+1)
480  FOR U=G TO H
490  Y=Y-L(U)^2
500  NEXT U
510  IF Y>0 THEN GOTO 540
520  PRINT "****PROGRAM FAILED IN SYMVBOS SUBROUTINE*****"
530  STOP
540  H=H+1
550  Y=SQR(Y)
560  L(H)=Y
570  FOR M=1 TO R
580  B(I,M)=B(I,M)/Y
590  NEXT M
600  NEXT I
610  REM REDUCTION COMPLETE
620  FOR I=N TO 1 STEP -1
630  Y=L(H)
640  FOR M=1 TO R
650  B(I,M)=B(I,M)/Y
660  NEXT M
670  IF I=1 THEN GOTO 780
680  J=I
690  P=S(I-1)
700  FOR H=H-1 TO P+1 STEP -1
710  J=J-1
720  Y=L(H)
730  FOR M=1 TO R
740  B(J,M)=B(J,M)-B(I,M)*Y
750  NEXT M
760  NEXT H
770  H=P
780  NEXT I
790  SUBEND !END OF Symvbsol
```

```
10    ! RE-SAVE"SMELST"
20    SUB Elstrs(Q(*),B(*),C(*),Es$,INTEGER Node(*),Test(*),Ne
      lemt,Nnode,Qort,Princ)
30    OPTION BASE 1
40    DIM Eldspl(24+8*Qort,1),Point(3),Estrs(6),Estrn(6),Ro(3,
      3),Sig(3),Princ(2),Dmy(12)
50    INTEGER I,J,Z
60    PRINT SPA(15);"Element strains and stresses"
70    PRINT SPA(13);"------------------------------";
80    ASSIGN #4 TO Es$
90    FOR Z=1 TO Nelemt
100   Jn=Node(Z,7+2*Qort)
110   MAT READ #4;B
120   READ #4;Thik
130   MAT READ #4;Ro
140   MAT READ #4;Point
150   CALL Eldispl(Eldspl(*),Q(*),Node(*),Test(*),Z,Qort,Nelemt)
160   FOR J=1 TO 6
170   Gash=0
180   FOR I=1 TO (6+2*Qort)*4
190   Gash=Gash+B(J,I)*Eldspl(I,1)
200   NEXT I
210   Estrn(J)=Gash
220   NEXT J
230   Estrs(1)=(Estrn(1)*C(Jn,1)+Estrn(2)*C(Jn,2))*Thik
240   Estrs(2)=(Estrn(1)*C(Jn,2)+Estrn(2)*C(Jn,7))*Thik
250   Estrs(3)=Estrn(3)*C(Jn,12)*Thik
260   Estrs(4)=(Estrn(4)*C(Jn,16)+Estrn(5)*C(Jn,17))*Thik*Thik
      *Thik
270   Estrs(5)=(Estrn(4)*C(Jn,17)+Estrn(5)*C(Jn,19))*Thik*Thik
      *Thik
280   Estrs(6)=Estrn(6)*C(Jn,21)*Thik*Thik*Thik
290   PRINT LIN(3)
300   PRINT USING 310;Z,Estrn(1),Estrn(2),Estrn(3),Estrn(4),Es
      trn(5),Estrn(6)
310    IMAGE 2D,2X,"Exin=",MD.4DE,2X,"Eyin=",MD.4DE,2X,"Exyin=
       ",MD.4DE//2X,"Exb=",MD.4DE,3X,"Eyb=",MD.4DE,2X,"Exyb=",M
      D.4DE
320    PRINT
330   PRINT USING 340;Z,Estrs(1),Estrs(2),Estrs(3),Estrs(4),Es
      trs(5),Estrs(6)
340    IMAGE 2D,2X,"Nxin=",MD.4DE,2X,"Nyin=",MD.4DE,2X,"Nxyin=
       ",MD.4DE//5X,"Mxb=",MD.4DE,2X"Myb=",MD.4DE,2X,"Mxyb=",MD
      .4DE
350    PRINT SPA(10),"These stresses at point "
360    PRINT SPA(2),"X=";Point(1),"Y=";Point(2),"Z=";Point(3)
370    PRINT SPA(10),"The local axes are given by",LIN(1)
380    MAT PRINT Ro
390    FOR I=1 TO 4 STEP 3
400    Dmy(1)=Estrn(I)*Ro(1,1)*Ro(1,1)+Estrn(I+1)*Ro(1,2)*Ro(1
       ,2)+Estrn(I+2)*Ro(1,1)*Ro(1,2)*2
410    Dmy(2)=Estrn(I)*Ro(2,1)*Ro(2,1)+Estrn(I+1)*Ro(2,2)*Ro(2
       ,2)+Estrn(I+2)*Ro(2,1)*Ro(2,2)*2
420    Dmy(3)=Estrn(I)*Ro(3,1)*Ro(3,1)+Estrn(I+1)*Ro(3,2)*Ro(3
       ,2)+Estrn(I+2)*Ro(3,1)*Ro(3,2)*2
430    Dmy(4)=Estrn(I)*Ro(1,1)*Ro(2,1)+Estrn(I+1)*Ro(1,2)*Ro(2
       ,2)+Estrn(I+2)*(Ro(1,1)*Ro(2,2)+Ro(1,2)*Ro(2,1))
440    Dmy(5)=Estrn(I)*Ro(2,1)*Ro(3,1)+Estrn(I+1)*Ro(2,2)*Ro(3
       ,2)+Estrn(I+2)*(Ro(2,1)*Ro(3,2)+Ro(2,2)*Ro(3,1))
450    Dmy(6)=Estrn(1)*Ro(3,1)*Ro(1,1)+Estrn(2)*Ro(3,2)*Ro(1,2
       )+Estrn(3)*(Ro(3,1)*Ro(1,2)+Ro(3,2)*Ro(1,1))
460    Dmy(7)=Estrs(I)*Ro(1,1)*Ro(1,1)+Estrs(I+1)*Ro(1,2)*Ro(1
       ,2)+Estrs(I+2)*Ro(1,1)*Ro(1,2)
470    Dmy(8)=Estrs(I)*Ro(2,1)*Ro(2,1)+Estrs(I+1)*Ro(2,2)*Ro(2
       ,2)+Estrs(I+2)*Ro(2,1)*Ro(2,2)
```

```
480     Dmy(9)=Estrs(I)*Ro(3,1)*Ro(3,1)+Estrs(I+1)*Ro(3,2)*Ro(3
        ,2)+Estrs(I+2)*(Ro(3,1)*Ro(3,2))
490     Dmy(10)=Estrs(I)*Ro(1,1)*Ro(2,1)*2+Estrs(I+1)*Ro(1,2)*R
        o(2,2)*2+Estrs(I+2)*(Ro(1,1)*Ro(2,2)+Ro(1,2)*Ro(2,1))
500     Dmy(11)=Estrs(I)*Ro(2,1)*Ro(3,1)*2+Estrs(I+1)*Ro(2,2)*R
        o(3,2)*2+Estrs(I+2)*(Ro(2,1)*Ro(3,2)+Ro(2,2)*Ro(3,1))
510     Dmy(12)=Estrs(I)*Ro(3,1)*Ro(1,1)*2+Estrs(I+1)*Ro(3,2)*R
        o(1,2)*2+Estrs(I+2)*(Ro(3,1)*Ro(1,2)+Ro(3,2)*Ro(1,1))
520     IF I=1 THEN PRINT SPA(4),"Memb. Strns. & Strs. in glob.
        axes ",LIN(1)
530     IF I=4 THEN PRINT SPA(4),"Bend. Strns. & Strs. in glob.
        axes ",LIN(1)
540    PRINT USING 560;Z,Dmy(1),Dmy(2),Dmy(3),Dmy(4),Dmy(5),Dmy(6)
541    PRINT ,LIN(2)
550    PRINT USING 570;Z,Dmy(7),Dmy(8),Dmy(9),Dmy(10),Dmy(11),D
       my(12)
551    PRINT ,LIN(2)
560     IMAGE 2D,2X,"Exg=",MD.4DE,2X,"Eyg=",MD.4DE,2X,"Ezg=",MD
        .4DE//2X,"Exyg",MD.4DE,3X,"Exzg=",MD.4DE,2X,"Eyzg=",MD.4
       DE
570     IMAGE 2D,2X,"Sxg=",MD.4DE,2X,"Syg=",MD.4DE,2X,"Szg=",MD
        .4DE//2X,"Sxyg",MD.4DE,3X,"Sxzg=",MD.4DE,2X,"Syzg=",MD.4
       DE
580     NEXT I
590     IF Princ<>1 THEN GOTO 790
600     FOR I=1 TO 4 STEP 3
610     IF I=1 THEN PRINT "Memb. Prcpl. Strs. at Cntr.of Ele.";Z
620     IF I=4 THEN PRINT "Bend. Prcpl. Strs. at Cntr.of Ele.";Z
630     Theta=.5*ATN(2*Estrs(I+2)/(Estrs(I)-Estrs(I+1)+1E-30))
640 N2:    IF Theta<0 THEN Theta=Theta+3.1415926536
650     Sn=SIN(Theta)
660     Cn=COS(Theta)
670     Princ(1)=Estrs(I)+Sn*(2*Cn*Estrs(I+2)+Sn*(Estrs(I+1)-Es
        trs(I)))
680     Princ(2)=Estrs(I)+Estrs(I+1)-Princ(1)
690     IF ABS(Princ(1))>=.9999999*ABS(Princ(2)) THEN GOTO N4
700     Theta=Theta-1.570796327
710     GOTO N2
720 N4:    Degree=Theta*57.29577951
730     PRINT
740     PRINT "First principle strss=";Princ(1),LIN(1)
750     PRINT "At an angle=";Degree;"Degree to OX",LIN(1)
760     PRINT "Secon principal stress=";Princ(2),LIN(1)
770     PRINT SPA(6),"Direction of this princpal Stress",LIN(1)
780     NEXT I
790    NEXT Z
800    ASSIGN * TO #4
810    SUBEND !End of Elstrs
```

```
10    ! RE-SAVE"SMNDST"
20     SUB Nodstr(Q(*),B(*),C(*),Ns$,INTEGER Node(*),Test(*),Ne
       lemt,Nnode,Qort,Princ)
30     OPTION BASE 1
40     DIM Eldspl(24+8*Qort,1),Point(3),A(4,4),Strs(6,4),Strn(6
       ,4),Ndstm(Nnode,6),Ndsnm(Nnode,6),Ndstb(Nnode,6),Ndsnb(N
       node,6),Ro(3,3),Sig(3),Princ(2),Dmy(12)
50     DIM Stngm(6,4),Stsgb(6,4)
60     MAT READ A
70     DATA 1.80660700845,-.462509857989,.118412707526,-.462509
       857989,-.462509857988,1.80660700845,-.462509857988,.1184
       12707525,.118412707524,-.462509857987
80     DATA 1.80660700845,-.462509857987,-.462509857988,.118412
       707525,-.462509857988,1.80660700845
90     PRINT SPA(15),"Nodal strains & stresses"
100    PRINT SPA(12),"-------------------------"
110    ASSIGN #3 TO Ns$
120    INTEGER J2,I,J,Ic,I1,I2,Z
130    FOR Z=1 TO Nelemt
140    CALL Eldispl(Eldspl(*),Q(*),Node(*),Test(*),Z,Qort,Nelemt)
150    FOR I=1 TO 3+Qort
160    MAT READ #3;B
170    READ #3;Thik
180    MAT READ #3;Ro
190    MAT READ #3;Point
200    FOR K=1 TO 6
210    Gash=0
220    FOR J=1 TO (6+2*Qort)*4
230    Gash=Gash+B(K,J)*Eldspl(J,1)
240    NEXT J
250    Strn(K,I)=Gash
260    NEXT K
270    Jn=Node(Z,7+2*Qort)
280    Strs(1,I)=(Strn(1,I)*C(Jn,1)+Strn(2,I)*C(Jn,2))*Thik
290    Strs(2,I)=(Strn(1,I)*C(Jn,2)+Strn(2,I)*C(Jn,7))*Thik
300    Strs(3,I)=Strn(3,I)*C(Jn,12)*Thik
310    Strs(4,I)=(Strn(4,I)*C(Jn,16)+Strn(5,I)*C(Jn,17))*Thik*T
       hik*Thik
320    Strs(5,I)=(Strn(4,I)*C(Jn,17)+Strn(5,I)*C(Jn,19))*Thik*T
       hik*Thik
330    Strs(6,I)=Strn(6,I)*C(Jn,21)*Thik*Thik*Thik
340    PRINT LIN(2)
350    PRINT SPA(3),"Strns. & Strs. at The Integ. point ";I;"
       Elemt. No";Z
360    PRINT LIN(1)
370    PRINT USING 380;I,Strn(1,I),Strn(2,I),Strn(3,I),Strn(4,I
       ),Strn(5,I),Strn(6,I)
380    IMAGE 2D,3X,"Exin=",MD.4DE,3X,"Eyin=",MD.4DE,3X,"Exyin="
       ,MD.4DE//5X,"Exb=",MD.4DE,3X,"Eyb=",MD.4DE,3X,"Exyb=",MD
       .4DE/
390    PRINT USING 400;I,Strs(1,I),Strs(2,I),Strs(3,I),Strs(4,I
       ),Strs(5,I),Strs(6,I)
400    IMAGE 2D,3X,"Nxin=",MD.4DE,3X,"Nyin=",MD.4DE,3X,"Nxyin="
       ,MD.4DE//5X,"Mxb=",MD.4DE,3X,"Myb=",MD.4DE,3X,"Mxyb=",MD
       .4DE
410    PRINT
420     PRINT SPA(10),"These stresses at point",LIN(1)
430    PRINT SPA(2),"X=";Point(1),"Y=";Point(2),"Z=";Point(3)
440    PRINT LIN(2)
450    PRINT SPA(10),"The local axes are given by",LIN(1)
460    MAT PRINT Ro
470    FOR L=1 TO 4 STEP 3
480    Dmy(1)=Strn(L,I)*Ro(1,1)*Ro(1,1)+Strn(L+1,I)*Ro(1,2)*Ro(
       1,2)+Strn(L+2,I)*Ro(1,1)*Ro(1,2)*2
490    Dmy(2)=Strn(L,I)*Ro(2,1)*Ro(2,1)+Strn(L+1,I)*Ro(2,2)*Ro(
       2,2)+Strn(L+2,I)*Ro(2,1)*Ro(2,2)*2
```

```
500   Dmy(3)=Strn(L,I)*Ro(3,1)*Ro(3,1)+Strn(L+1,I)*Ro(3,2)*Ro(
      3,2)+Strn(L+2,I)*Ro(3,1)*Ro(3,2)*2
510   Dmy(4)=Strn(L,I)*Ro(1,1)*Ro(2,1)+Strn(L+1,I)*Ro(1,2)*Ro(
      2,2)+Strn(L+2,I)*(Ro(1,1)*Ro(2,2)+Ro(1,2)*Ro(2,1))
520   Dmy(5)=Strn(L,I)*Ro(2,1)*Ro(3,1)  +Strn(L+1,I)*Ro(2,2)*R
      o(3,2)  +Strn(L+2,I)*(Ro(2,1)*Ro(3,2)+Ro(2,2)*Ro(3,1))
530   Dmy(6)=Strn(L,I)*Ro(3,1)*Ro(1,1)+Strn(L+1,I)*Ro(1,2)*Ro(
      3,2)+Strn(L+2,I)*(Ro(3,1)*Ro(1,2)+Ro(3,2)*Ro(1,1))
540   Dmy(7)=Strs(L,I)*Ro(1,1)*Ro(1,1)+Strs(L+1,I)*Ro(1,2)*Ro(
      1,2)+Strs(L+2,I)*Ro(1,1)*Ro(1,2)
550   Dmy(8)=Strs(L,I)*Ro(2,1)*Ro(2,1)+Strs(L+1,I)*Ro(2,2)*Ro(
      2,2)+Strs(L+2,I)*Ro(2,1)*Ro(2,2)
560   Dmy(9)=Strs(L,I)*Ro(3,1)*Ro(3,1)+Strs(L+1,I)*Ro(3,2)*Ro(
      3,2)+Strs(L+2,I)*Ro(3,1)*Ro(3,2)
570   Dmy(10)=Strs(L,I)*Ro(1,1)*Ro(2,1)*2+Strs(L+1,I)*Ro(1,2)*
      Ro(2,2)*2+Strs(L+2,I)*(Ro(1,1)*Ro(2,2)+Ro(1,2)*Ro(2,1))
580   Dmy(11)=Strs(1,I)*Ro(2,1)*Ro(3,1)*2+Strs(2,I)*Ro(2,2)*Ro
      (3,2)*2+Strs(3,I)*(Ro(2,1)*Ro(3,2)+Ro(2,2)*Ro(3,1))
590   Dmy(12)=Strs(L,I)*Ro(1,1)*Ro(3,1)*2+Strs(L+1,I)*Ro(3,2)*
      Ro(1,2)*2+Strs(L+2,I)*(Ro(3,1)*Ro(1,2)+Ro(3,2)*Ro(1,1))
600   IF L<>1 THEN GOTO 660
610   FOR K=1 TO 6
620   Stngm(K,I)=Dmy(K)
630   Stsgm(K,I)=Dmy(K+6)
640   NEXT K
650   GOTO 700
660   FOR K=1 TO 6
670   Stngb(K,I)=Dmy(K)
680   Stsgb(K,I)=Dmy(K+6)
690   NEXT K
700   NEXT L
710   PRINT
720    IF Princ<>1 THEN GOTO 900
730    FOR Ip=1 TO 4 STEP 3
740    IF Ip=1 THEN PRINT "Memb.Prcpl. Tens. at Intg. Pt.";I
750    IF Ip=4 THEN PRINT "Bend.Prcpl. Mom.  at Intg. Pt.";I
760    Theta=.5*ATN(2*Strs(Ip+2,I)/(Strs(Ip,I)-Strs(Ip+1,I)+1E
       -30))
770 N2:  IF Theta<0 THEN Theta=Theta+3.1415926536
780   Sn=SIN(Theta)
790   Cn=COS(Theta)
800   Princ(1)=Strs(Ip,I)+Sn*(2*Cn*Strs(Ip+2,I)+Sn*(Strs(Ip+1,
      I)-Strs(Ip,I)))
810   Princ(2)=Strs(Ip,I)+Strs(Ip+1,I)-Princ(1)
820   IF ABS(Princ(1))>=.9999999*ABS(Princ(2)) THEN GOTO N4
830   Theta=Theta-1.570796327
840   GOTO N2
850 N4:  Degree=Theta*57.29577951
860   PRINT "First Pricipal stress=";Princ(1),LIN(1)
870   PRINT "At an angle=";Degree;"Degree to OX",LIN(1)
880   PRINT "Second principal stress=";Princ(2)
890   NEXT Ip
900   NEXT I
910   IF Qort=0 THEN GOTO 1090
920   FOR I1=1 TO 6
930   I=1
940   FOR J=1 TO 4
950   Gash=Gush=Gish=Gosh=0
960   FOR I2=1 TO 4
970   Gash=Gash+A(J,I2)*Stngm(I1,I2)
980   Gush=Gush+A(J,I2)*Stngb(I1,I2)
990   Gish=Gish+A(J,I2)*Stsgm(I1,I2)
1000  Gosh=Gosh+A(J,I2)*Stsgb(I1,I2)
1010  NEXT I2
1020  Ndstm(Node(Z,I),I1)=Ndstm(Node(Z,I),I1)+Gish
1030  Ndstb(Node(Z,I),I1)=Ndstb(Node(Z,I),I1)+Gosh
```

```
1040  Ndsnm(Node(Z,I),I1)=Ndsnm(Node(Z,I),I1)+Gash
1050  Ndsnb(Node(Z,I),I1)=Ndsnb(Node(Z,I),I1)+Gush
1060  I=I+1
1070  NEXT J
1080  NEXT I1
1090 NEXT Z
1100  ASSIGN #3 TO *
1110  IF Qort=0 THEN GOTO 1750
1120  FOR U=1 TO Nnode
1130  IF Test(U)<>1 THEN GOTO 1280
1140  Cnt=0
1150  FOR Z=1 TO Nelemt
1160  FOR Cnt1=1 TO 6+2*Qort
1170  IF Node(Z,Cnt1)<>U THEN GOTO Nexn
1180  Cnt=Cnt+1
1190  GOTO 1210
1200 Nexn: NEXT Cnt1
1210  NEXT Z
1220   FOR I1=1 TO 6
1230   Ndstm(U,I1)=Ndstm(U,I1)/Cnt
1240   Ndstb(U,I1)=Ndstb(U,I1)/Cnt
1250   Ndsnm(U,I1)=Ndsnm(U,I1)/Cnt
1260   Ndsnb(U,I1)=Ndsnb(U,I1)/Cnt
1270   NEXT I1
1280   NEXT U
1290   FOR U=1 TO Nnode
1300   IF Test(U)=1 THEN GOTO Nex
1310   FOR Z=1 TO Nelemt
1320   FOR I=4+Qort TO 6+2*Qort
1330   IF U<>Node(Z,I) THEN GOTO Nex1
1340   FOR I1=1 TO 6
1350   IF I<>6+2*Qort THEN Ndstm(U,I1)=(Ndstm(Node(Z,I-(3+Qort
       )),I1)+Ndstm(Node(Z,I-(2+Qort)),I1))/2
1360   IF I<>6+2*Qort THEN Ndstb(U,I1)=(Ndstb(Node(Z,I-(3+Qort
       )),I1)+Ndstb(Node(Z,I-(2+Qort)),I1))/2
1370   IF I=6+2*Qort THEN Ndstm(U,I1)=(Ndstm(Node(Z,1),I1)+Nds
       tm(Node(Z,3+Qort),I1))/2
1380   IF I=6+2*Qort THEN Ndstb(U,I1)=(Ndstb(Node(Z,1),I1)+Nds
       tb(Node(Z,3+Qort),I1))/2
1390   IF I<>6+2*Qort THEN Ndsnm(U,I1)=(Ndsnm(Node(Z,I-(3+Qort
       )),I1)+Ndsnm(Node(Z,I-(2+Qort)),I1))/2
1400   IF I<>6+2*Qort THEN Ndsnb(U,I1)=(Ndsnb(Node(Z,I-(3+Qort
       )),I1)+Ndsnb(Node(Z,I-(2+Qort)),I1))/2
1410   IF I=6+2*Qort THEN Ndsnm(U,I1)=(Ndsnm(Node(Z,1),I1)+Nds
       nm(Node(Z,3+Qort),I1))/2
1420   IF I=6+2*Qort THEN Ndsnb(U,I1)=(Ndsnb(Node(Z,1),I1)+Nds
       nb(Node(Z,3+Qort),I1))/2
1430   NEXT I1
1440   GOTO Nex
1450 Nex1:  NEXT I
1460  NEXT Z
1470 Nex: NEXT U
1480  PRINT
1490  PRINT SPA(20),"Smoothed Strains & Stresses ",LIN(2)
1500  PRINT LIN(4)
1510  PRINT SPA(12);"Nodal Strains";SPA(30),"Stresses";LIN(1)
1520  PRINT SPA(10);"****************";SPA(30),"**********";LIN(1)
1530  PRINT "Node";SPA(5);"Exmg";SPA(7);"Eymg";SPA(7);"Ezmg";S
      PA(10);"Exymg       Exzmg         Eyzmg"
1540  PRINT "----";SPA(5);"----";SPA(7);"----";SPA(7);"----";S
      PA(10);"-----       ----          ----"
1550  FOR I=1 TO Nnode
1560  PRINT USING "3D,2X,3(MD.4DE,X),4X,3(MD.4DE,   X)
        ";I,Ndsnm(I,1),Ndsnm(I,2),Ndsnm(I,3),Ndsnm(I,4),Ndsn
      m(I,5),Ndsnm(I,6)
1570  NEXT I
```

```
1580  PRINT "Node";SPA(6);"Nxg";SPA(7);"Nyg";SPA(10);"Nzg";SPA
      (14);"Nxyg       Nxzg          Nyzg"
1590  PRINT "----";SPA(6);"---";SPA(7);"---";SPA(10);"---";SPA
      (14);"----       --          ---"
1600  FOR I=1 TO Nnode
1610  PRINT USING "3D,2X,3(MD.4DE,X),4X,3(MD.4DE,   X)
         ";I,Ndstm(I,1),Ndstm(I,2),Ndstm(I,3),Ndstm(I,4),Ndst
      m(I,5),Ndstm(I,6)
1620  NEXT I
1630  PRINT SPA(12);"Nodal Strains";SPA(30),"Stresses";LIN(1)
1640  PRINT SPA(12);"*************";SPA(30),"********";LIN(1)
1650  PRINT "Node";SPA(7);"Exbg";SPA(7);"Eybg";SPA(7);"Ezbg";S
      PA(12);"Exybg     Exzbg       Eyzbg",LIN(1)
1660  PRINT "----";SPA(7);"----";SPA(7);"----";SPA(7);"-----";
      SPA(12);"-----     -----      ------",LIN(1)
1670  FOR I=1 TO Nnode
1680  PRINT USING "3D,2X,3(MD.4DE,X),4X,3(MD.4DE,   X)
         ";I,Ndsnb(I,1),Ndsnb(I,2),Ndsnb(I,3),Ndsnb(I,4),Ndsn
      b(I,5),Ndsnb(I,6)
1690  NEXT I
1700  PRINT "Node";SPA(7);"Mxg";SPA(7);"Myg";SPA(9);"Mzg";SPA(
      12);"Mxyg      Mxzg         Myzg ",LIN(1)
1710  PRINT "----";SPA(7);"---";SPA(7);"---";SPA(9);"---";SPA(
      12);"----      ---          ----",LIN(1)
1720  FOR I=1 TO Nnode
1730  PRINT USING "3D,2X,3(MD.4DE,X),4X,3(MD.4DE,   X)
         ";I,Ndstb(I,1),Ndstb(I,2),Ndstb(I,3),Ndstb(I,4),Ndst
      b(I,5),Ndstb(I,6)
1740  NEXT I
1750  SUBEND !End of Nodstr
1760  SUB Eldispl(Eldspl(*),Q(*),INTEGER Node(*),Test(*),Z,Qor
      t,Nelemt)
1770  Mod(1)=1
1780  Mod(2)=4+Qort
1790  Mod(3)=2
1800  Mod(4)=5+Qort
1810  Mod(5)=3
1820  Mod(6)=6+Qort
1830  Mod(7)=4
1840  Mod(8)=8
1850  J=1
1860  FOR I=1 TO 6+2*Qort
1870  J2=0
1880  I1=Mod(I)
1890  A=Test(Node(Z,I1))
1900  FOR J1=1 TO Node(Z,I1)
1910  J2=J2+5-2*Test(J1)
1920  NEXT J1
1930  Eldspl(J,1)=Q(J2-4+2*A,1)
1940  Eldspl(J+1,1)=Q(J2-3+2*A,1)
1950  Eldspl(J+2,1)=Q(J2-2+2*A,1)
1960  IF A<>0 THEN GOTO Nex
1970  Eldspl(J+3,1)=Q(J2-1,1)
1980  Eldspl(J+4,1)=Q(J2,1)
1990 Nex: J=J+5-2*A
2000   NEXT I
2010  SUBEND !End of Eldispl
```

```
10    ! RE-STORE"IMPSMF"
20     OPTION BASE 1
30     DIM K(8500)
40     DIM Xita(2),Xx(147),Yy(147),Ulx(147,1),Vly(147,1),Wlz(14
       7,1),B(6,32),C(1,21),Zz(147),Q(650,1),Thetxz(147,1)
50     DIM Thikn(60),Th(10),Angsk(33),Qb(120)
60     DIM Add(0:650),Lim(0:15)
70     DIM A$(10),B$(20)
80     INTEGER Node(60,10),Qort,Kode(147,1),Kode1(147,1),N,Matn
       o,Nspec,Nsetfs,K,I,J,Nnode,Count,Princ,Nskewd,Nskewc
90     INTEGER Test(147),Nosk(33),Nk,St(16,2),Row(0:15),Col(0:1
       5),Seg,Ck
100    COM Ck1,Ck2,Ck3,Ck4,Chta,Chtp,Size
110    COM Alp,Prnt,Nfree,Nskw,Nelemt,Ch1,Ch2,Omg
120    OVERLAP
130    PRINTER IS 16
140    PRINT PAGE,SPA(25),"*****CAD OF FAN IMPELLERS****",LIN(1)
150    PRINT SPA(32),"Exploiting of Sectorial symmetry are Impe
       lmented";
160    PRINT "Press CLEAR then CONT";
170    PAUSE
180    DISP "Choose the printing devive 0 for paper 16 for
       screen";
190    INPUT T
200    PRINTER IS T
210    DISP "input the of nodes on the disc side & on  the cone
       side Nskewd,Nskewc";
220    INPUT Nskewd,Nskewc
230    PRINT "Nskewd=";Nskewd,SPA(10),"Nskewc=",Nskewc
240    DISP "What is the Name of your input data file";
250    INPUT Name$
260    ASSIGN #2 TO Name$,C
270    IF NOT C THEN GOTO 320
280    BEEP
290    DISP "File not found!Try again";
300    WAIT 2000
310    GOTO 180
320    READ #2;Qort,Njob
330    FOR Count=1 TO Njob
340    LINPUT "Type in name of the job.......not more than 20
       charecters",A$
350    PRINT "Job name.......";A$,LIN(1)
360    READ #2;Nelemt,Nnode
370    READ #2;Nsetfs,Princ,Nskw,Nmat,Prnt
380    PRINT "No of elements.........";Nelemt
390    PRINT
400    PRINT "No of nodes.............";Nnode
410    PRINT
420    IF (Prnt<>1) AND (Prnt<>3) THEN GOTO 470
430    DISP "Input the name of the file E$ to be created to store
       B[*] to find Nod. Strs.";
440    INPUT E$
450    DISP "CREATE E$ THEN Press CONT";
460    PAUSE
470    IF (Prnt<>2) AND (Prnt<>3) THEN GOTO 520
480    DISP "Input the name of the file N$ to be  created to store
       B[*] to find Ele. Strs.";
490    INPUT N$
500    DISP "CREATE N$ THEN Press CONT";
510    PAUSE
520    PRINT LIN(2)
530    REDIM Xx(Nnode),Yy(Nnode),Zz(Nnode),Kode(Nnode,Nsetfs),K
       ode1(Nnode,Nsetfs),Node(Nelemt,8+2*Qort),Ulx(Nnode,Nsetf
       s),Vly(Nnode,Nsetfs),Wlz(Nnode,Nsetfs)
540    REDIM Thetxz(Nnode,Nsetfs),C(Nmat,21),Test(Nnode)
```

```
550   REDIM B(6,24+8*Qort),Thikn(Nelemt),Th(Nmat),Angsk(Nskw+1
      ),Nosk(Nskw+1)
560   IF Qort=0 THEN PRINT SPA(12),"Element selected is a tria
      ngular type";
570   IF Qort=1 THEN PRINT SPA(12),"Element selected is a quad
      rilateral type";
580   READ #2;Nk,Ch1,Ch2,Omg,Alp
590   IF Nk=0 THEN GOTO 650
600   REDIM St(Nk,2)
610   MAT READ #2;St
620   DISP "input No of nodes on the disc side & on  the cone
      side Nskewd,Nskewcon (a-a)Boundary";
630   INPUT Nskewd,Nskewc
640   PRINT "Nskewd=";Nskewd,SPA(10),"Nskewc=",Nskewc
650   Omg=Omg*2*PI/60
660   LINK "FINPIM",1610
670   CALL Finp(Xx(*),Yy(*),Zz(*),Ulx(*),Vly(*),Wlz(*),Thetxz(
      *),Thikn(*),Nfree,Nelemt,#2,Nnode,Kode(*),Kode1(*),Node(
      *),Test(*),Qort,Nk,St(*))
680   L=0
690   FOR I=1 TO Nk
700   L=L+5-2*Test(St(I,2))
710   NEXT I
720   Nfree0=Nfree+L
730   REDIM Q(Nfree0,Nsetfs),Add(0:Nfree),Qb(L)
740   CALL Maddarray(Add(*),Lim(*),Nfree,Nelemt,Row(*),Col(*),
      Test(*),Nnode,Node(*),Qort,Seg,Ck,Nk,St(*))
750   LINK "CONSIM",1610
760   FOR Matno=1 TO Nmat
770   CALL Constrel(C(*),Th(*),#2,Matno)
780   NEXT Matno
790   FOR I=1 TO Seg
800   DISP "What is the name of the file number";I
810   INPUT A$(I)
820   DISP "CREATE This file in the suitable sizeTHEN Press CONT";
830   PAUSE
840   NEXT I
850   IF Nskw=0 THEN GOTO 890
860   FOR I=1 TO Nskw
870   READ #2;Nosk(I),Angsk(I)
880   NEXT I
890   REDIM K(0:Size)
900   LINK "ASEMIM",1610
910 CALL Massembly(N$,E$,A$(*),Q(*),Xx(*),Yy(*),Zz(*),C(*),Th(
    *),Thikn(*),Angsk(*),Qb(*),K(*),Add(*),Lim(*),Node(*),Te
      st(*),Nosk(*),Qort,Nk,St(*),Seg)
920   LINK "LOADIM",1610
930   K=0
940   FOR I=1 TO Nnode-Nk
950   J=4-2*Test(I)
960   K=K+5-2*Test(I)
970   CALL Loadapp(Ulx(I,1),Vly(I,1),Wlz(I,1),Q(*),Kode(I,1),K
      ,1,J)
980   NEXT I
990   IF Nskw=0 THEN GOTO 1060
1000  IF Nk=0 THEN GOTO 1060
1010  LINK "SKWIM1",1610
1020  CALL Skewim1(Q(*),Angsk(*),Qb(*),Alp,Nskw,Test(*),Nsetfs
      ,Nosk(*),St(*),Nk,Nskewd,Nskewc)
1030  FOR I=1 TO L
1040  Q(I,1)=Q(I,1)+Qb(I)
1050  NEXT I
1060  LINK "GEOMIM",1610
1070  REM ---INTROUDUCTION TO KINEMATIC CONSTRAINTS
1080  K=0
1090  FOR I=1 TO Nnode-Nk
```

```
1100  J=4-2*Test(I)
1110  K=K+5-2*Test(I)
1120  IF Kode(I,1)=0 THEN GOTO Test
1130  IF (Kode(I,1)=2) OR (Kode(I,1)=6) THEN GOTO Kc2
1140  IF Kode(I,1)=7 THEN GOTO Kc3
1150  CALL Geombc(A$(*),Ulx(I,1),Q(*),K(*),Add(*),Nfree,Lim(*)
      ,K-J,1,Seg)
1160  IF Kode(I,1)=3 THEN GOTO Kc3
1170  IF Kode(I,1)=5 THEN GOTO Test
1180 Kc2: CALL Geombc(A$(*),Vly(I,1),Q(*),K(*),Add(*),Nfree,Li
     m(*),K-J+1,1,Seg)
1190  IF (Kode(I,1)=4) OR (Kode(I,1)=6) THEN GOTO Test
1200 Kc3: CALL Geombc(A$(*),Wlz(I,1),Q(*),K(*),Add(*),Nfree,Li
     m(*),K-J+2,1,Seg)
1210 Test: IF (J<>4) OR (Kode1(I,1)=0) THEN GOTO Next
1220  CALL Geombc(A$(*),Thetxz(I,1),Q(*),K(*),Add(*),Nfree,Lim
      (*),K-J+3,1,Seg)
1230  CALL Geombc(A$(*),Thetxz(I,1),Q(*),K(*),Add(*),Nfree,Lim
      (*),K-J+4,1,Seg)
1240 Next: NEXT I
1250  PRINT LIN(3)
1260  LINK "MSYVIM",1610
1270  PRINT SPA(10);"------Symvbsol begins------"
1280  CALL Segsol(A$(*),K(*),Q(*),Add(*),Nfree,Lim(*),Nsetfs,S
      eg,Col(*),Row(*))
1290  IF Nskw=0 THEN GOTO 1360
1300  IF Nk=0 THEN GOTO 1360
1310  FOR I=1 TO L
1320  Qb(I)=Q(I,1)
1330  NEXT I
1340  LINK "SKWIM2",1610
1350  CALL Skewim2(Q(*),Angsk(*),Qb(*),Alp,Nskw,Test(*),Nsetfs
      ,Nosk(*),St(*),Nk,Nskewd,Nskewc)
1360  PRINT ,SPA(4);"Nodal displacements"
1370  PRINT ,SPA(3);"----------------------"
1380  PRINT LIN(2)
1390  PRINT "Node";"Disp in X-dir",SPA(5),"Disp in Y-dir",SPA(
      5),"Disp in Z-dir"
1400  K=0
1410  FOR I=1 TO Nnode
1420  J=4-2*Test(I)
1430  K=K+5-2*Test(I)
1440  PRINT USING "3D,3X,3(MD.8DE,3X)";I,Q(K-J,1),Q(K-J+1,1),Q
      (K-J+2,1)
1450  PRINT
1460  NEXT I
1470  PRINT
1480  PRINT SPA(4);"Rotations at loof nodes"
1490  PRINT SPA(4);"--------------------------"
1500  K=0
1510  PRINT
1520  FOR I=1 TO Nnode
1530  K=K+5-2*Test(I)
1540  J=4-2*Test(I)
1550  IF Test(I)<>0 THEN GOTO N
1560  PRINT USING "3D,3X,2(MD.8DE,3X)";I,Q(K-J+3,1),Q(K-J+4,1)
1570  PRINT
1580 N: NEXT I
1590  PRINT
1600  IF (Prnt<>1) AND (Prnt<>3) THEN GOTO 1630
1610  LINK "SMELST",1610
1620  CALL Elstrs(E$,Q(*),B(*),C(*),Node(*),Test(*),Nelemt,Nno
      de,Qort,Princ)
1630  IF (Prnt<>2) AND (Prnt<>3) THEN GOTO 1660
1640  LINK "SMNDST",1610
1650  CALL Nodstr(N$,Q(*),B(*),C(*),Node(*),Test(*),Nelemt,Nno
      de,Qort,Princ)
```

```
930  SUB Maddarray(Add(*),Lim(*),INTEGER Row(*),Col(*),Test(*)
     ,Nelemt,Nnode,Node(*),Qort,Nfree,Seg,Ck,Nk,St(*))
940   OPTION BASE 1
950   COM Ck1,Ck2,Ck3,Ck4,Chta,Chtp,Size
960   INTEGER Itemp,L,I,K,M,A(21)
970   Add(1)=1
980   FOR I=2 TO 11
990   Add(I)=Add(I-1)+I
1000  NEXT I
1010  Addtemp=Add(11)
1020  Itemp=12
1030  FOR I=4 TO Nnode-Nk
1040  K=I
1050  I2=0
1060  FOR Z=1 TO Nelemt
1070  FOR L=1 TO 6+2*Qort
1080  IF Node(Z,L)<>I THEN GOTO L1
1090  I2=I2+1
1100  A(I2)=Z
1110 L1: NEXT L
1120  NEXT Z
1130  I3=1
1140  IF I2=1 THEN GOTO 1160
1150  FOR I3=1 TO I2
1160  FOR I1=1 TO Nk
1170  FOR L=1 TO 6+2*Qort
1180  IF Node(A(I3),L)<>St(I1,2) THEN GOTO L2
1190  IF K>St(I1,1) THEN K=St(I1,1)
1200  GOTO L3
1210 L2: NEXT L
1220 L3: NEXT I1
1230  IF I2=1 THEN GOTO 1241
1240  NEXT I3
1241  FOR Z=1 TO I2
1242  M=A(Z)
1250  FOR L=1 TO 6+2*Qort
1260  IF Node(M,L)<K THEN K=Node(M,L)
1270  NEXT L
1271  NEXT Z
1280  M=0
1290  IF K=1 THEN GOTO 1330
1300  FOR L=1 TO K-1
1310  M=M+5-2*Test(L)
1320  NEXT L
1330 Add(Itemp)=Addtemp+Itemp-M
1340  Add(Itemp+1)=Add(Itemp)+(Itemp+1)-M
1350  Add(Itemp+2)=Add(Itemp+1)+(Itemp+2)-M
1360  IF Test(I)<>1 THEN GOTO Aud
1370  Addtemp=Add(Itemp+2)
1380  Itemp=Itemp+3
1390  GOTO Next
1400 Aud: Add(Itemp+3)=Add(Itemp+2)+(Itemp+3)-M
1410  Add(Itemp+4)=Add(Itemp+3)+(Itemp+4)-M
1420  Addtemp=Add(Itemp+4)
1430  Itemp=Itemp+5
1440 Next: NEXT I
1450  Lim(0)=0
1460  Row(0)=0
1470  Seg=INT(Add(Nfree)/10000)+1
1480  W=INT(Add(Nfree)/Seg)
1490  FOR I=1 TO Seg
1500  IF I=Seg THEN GOTO A
1510  Lim(I)=W+Lim(I-1)
1520  FOR J=0 TO Nfree
1530  IF Lim(I)>=Add(INT(Nfree/Seg)*I+J) THEN GOTO Nextj
```

```
1540  Lim(I)=Add(INT(Nfree/Seg)*I+J)
1550  Row(I)=INT(Nfree/Seg)*I+J
1560  GOTO Ad2
1570 Nextj: NEXT J
1580 Ad2: NEXT I
1590 A: Lim(I)=Add(Nfree)
1600  Row(Seg)=Nfree
1610  Size=Lim(1)
1620  FOR I=2 TO Seg
1630  IF Size<Lim(I)-Lim(I-1) THEN Size=Lim(I)-Lim(I-1)
1640  NEXT I
1650  FOR I=1 TO Seg
1660  Change=Nfree
1670  FOR J=Row(I-1)+1 TO Row(I)
1680  Col(I)=J-(Add(J)-Add(J-1))+1
1690  IF Col(I)<Change THEN Change=Col(I)
1700  NEXT J
1710  FOR J=1 TO Seg
1720  IF Change>Row(J) THEN GOTO Njj
1730  Col(I)=J
1740  GOTO Ad3
1750 Njj: NEXT J
1760 Ad3: NEXT I
1770  Ck1=Ck3=Lim(0)
1780  Ck2=Ck4=Lim(1)
1790  Chta=Chtp=1
1800  Ck=1
1810   DISP "Note:the one dimensional stiffness matrix has";Ad
      d(Nfree);"elements"
1820 SUBEND !END OF Maddarray
```

```
10    ! RE-SAVE"ASEMIM"
20 SUB Massembly(N$,E$,A$(*),Q(*),Xx(*),Yy(*),Zz(*),C(*),Th(*)
   ,T(*),As(*),Q3(*),K(*),Add(*),Lim(*),INTEGER Node(*),Tes
      t(*),Nosk(*),Qort,Nk,St(*),Seg)
30    OPTION BASE 1
40    DIM Elxyzt(9,4),Wshel(13,45),Xita(2),Ro(3,3),Felvec(24+8
      *Qort),Shear(11,43),Xyzpre(8,4),Vloof(3,36),A(5),Poin(3)
      ,Bod(3),B(6,24+8*Qort)
50    DIM Ke(24+8*Qort,24+8*Qort)
60    INTEGER J,M,Z,Lvabz,Nozpre,Swop(6),Lnodz,Itest,Icoun1,Ic
      oun2,Jf1,Sub1,Sub2,Sub3,Sub4,Sub5,Sub6,An(6+2*Qort)
70    COM Ck1,Ck2,Ck3,Ck4,Chta,Chtp,Size
80    COM Alp,Pr,Nfree,Nskw,Nelemt,Ch1,Ch2,Omg
90    Lnodz=6+2*Qort
100   Lvabz=4*Lnodz
110   FOR I=1 TO Seg
120   ASSIGN #I TO A$(I)
130   MAT PRINT #I;K
140   NEXT I
150   IF (Pr=1) OR (Pr=3) THEN ASSIGN #9 TO N$
160   IF (Pr=2) OR (Pr=3) THEN ASSIGN #10 TO E$
170   IF Qort=0 THEN GOTO 230
180   A(1)=A(2)=A(3)=A(4)=.95
190   A(5)=.2
200   W(1,1)=W(1,2)=W(2,2)=W(4,1)=-.592348878
210   W(2,1)=W(4,2)=W(3,1)=W(3,2)=.592348878
220   GOTO 270
230   A(1)=A(2)=A(3)=A(4)=1
240   W(1,1)=W(2,2)=0
250   W(1,2)=W(2,1)=W(3,1)=W(3,2)=.5
260   W(4,1)=W(4,2)=.333333
270   MAT K=ZER
280   MAT Q=ZER
290   Mod(1)=1
300   Mod(2)=4+Qort
310   Mod(3)=2
320   Mod(4)=5+Qort
330   Mod(5)=3
340   Mod(6)=6+Qort
350   Mod(7)=4
360   Mod(8)=8
370   FOR Z=1 TO Nelemt
380   FOR I=1 TO 6+2*Qort
390   A1(I)=Node(Z,I)
400   NEXT I
410   FOR I=1 TO 6+2*Qort
420   Node(Z,I)=A1(Mod(I))
430   NEXT I
440   MAT Ke=ZER
450   MAT Xyzpre=ZER
460   MAT Elxyzt=ZER
470   Nozpre=0
480   MAT Shear=ZER
490   Thick=T(Z)
500  FOR I=1 TO Lnodz
510   K=INT(ABS(Node(Z,I)))
520   Elxyzt(I,4)=Thick
530   Elxyzt(I,1)=Xx(K)
540   Elxyzt(I,2)=Yy(K)
550   Elxyzt(I,3)=Zz(K)
560   NEXT I
570   FOR M=1 TO 4+Qort
580   IF M<>5 THEN GOTO 610
590   Xita(1)=Xita(2)=0
600   GOTO 630
```

```
610    Xita(1)=W(M,1)
620    Xita(2)=W(M,2)
630    CALL Haloof(Xita(*),Area,Thik,Ro(*),Poin(*),Elxyzt(*),Ws
       hel(*),Thick,Shear(*),Xyzpre(*),Vloof(*),Nelemt,Swop(*),
       Node(*),Lnodz,Nozpre,Z,Qort)
640    Jn=Node(Z,7+2*Qort)
650    FOR I=1 TO 4*Lnodz
660    B(1,I)=Wshel(4,I)
670    B(2,I)=Wshel(7,I)
680    B(3,I)=Wshel(5,I)+Wshel(6,I)
690    B(4,I)=Wshel(10,I)
700    B(5,I)=Wshel(12,I)
710    B(6,I)=2*Wshel(11,I)
720    NEXT I
730    Detj=Area
740    IF (M=4) AND (Qort=0) THEN GOTO 1230
750    FOR K=1 TO 4
760    FOR I=K TO 21+8*Qort+(K-1) STEP 4
770    FOR J=I TO 21+8*Qort+(K-1) STEP 4
780    A=C(Jn,1)*B(1,I)+C(Jn,2)*B(2,I)+C(Jn,3)*B(3,I)
790    B=C(Jn,2)*B(1,I)+C(Jn,7)*B(2,I)+C(Jn,8)*B(3,I)
800    C=C(Jn,3)*B(1,I)+C(Jn,8)*B(2,I)+C(Jn,12)*B(3,I)
810    D=(C(Jn,16)*B(4,I)+C(Jn,17)*B(5,I)+C(Jn,18)*B(6,I))*Thik
       *Thik
820    E=(C(Jn,17)*B(4,I)+C(Jn,19)*B(5,I)+C(Jn,20)*B(6,I))*Thik
       *Thik
830    F=(C(Jn,18)*B(4,I)+C(Jn,20)*B(5,I)+C(Jn,21)*B(6,I))*Thik
       *Thik
840    Ke(J,I)=Ke(I,J)=Ke(J,I)+A(M)*(B(1,J)*A+B(2,J)*B+B(3,J)*C
       +B(4,J)*D+B(5,J)*E+B(6,J)*F)*Detj*Thik
850    IF J=24+8*Qort THEN GOTO L6
860    Ke(I,J+1)=Ke(J+1,I)=Ke(J+1,I)+A(M)*(B(1,J+1)*A+B(2,J+1)*
       B+B(3,J+1)*C+B(4,J+1)*D+B(5,J+1)*E+B(6,J+1)*F)*Detj*Thik
870    IF J=23+8*Qort THEN GOTO L6
880    Ke(I,J+2)=Ke(J+2,I)=Ke(J+2,I)+A(M)*(B(1,J+2)*A+B(2,J+2)*
       B+B(3,J+2)*C+B(4,J+2)*D+B(5,J+2)*E+B(6,J+2)*F)*Detj*Thik
890    IF J=22+8*Qort THEN GOTO L6
900    Ke(I,J+3)=Ke(J+3,I)=Ke(J+3,I)+A(M)*(B(1,J+3)*A+B(2,J+3)*
       B+B(3,J+3)*C+B(4,J+3)*D+B(5,J+3)*E+B(6,J+3)*F)*Detj*Thik
910 L6:  NEXT J
920    NEXT I
930    NEXT K
940    IF (Ch1=0) AND (Ch2=0) THEN GOTO 1100
950    IF Ch1=0 THEN GOTO Press
960    Weight=Area*Ch1*A(M)
970    FOR K=1 TO Lvabz
980    Gash=Felvec(K)
990    FOR I=1 TO 3
1000   Gash=Gash+Weight*Ro(3,I)*Wshel(I,K)
1010   NEXT I
1020   IF Ch2<>0 THEN GOTO 1070
1030   Felvec(K)=Gash
1040   GOTO 1100
1050 Press: FOR K=1 TO Lvabz
1060 Press=Ch2
1070   Gash=Felvec(K)
1080   Felvec(K)=Gash+Press*Area*A(M)*Wshel(3,K)
1090 NEXT K
1100   IF Omg=0 THEN GOTO 1230
1110   Ro=Th(Node(Z,7+2*Qort))
1120   FOR I=1 TO 3
1130   Bod(I)=Ro*Omg*Omg*Poin(I)*Thik*Area*A(M)
1140   Bod(3)=0
1150   NEXT I
1160   FOR K=1 TO Lvabz
1170   Gash=Felvec(K)
```

```
1180  FOR I=1 TO 3
1190  Gash=Gash+(Bod(1)*Ro(1,I)+Bod(2)*Ro(2,I)+Bod(3)*Ro(3,I))
      *Wshel(I,K)
1200  NEXT I
1210  Felvec(K)=Gash
1220  NEXT K
1230  IF Pr=0 THEN GOTO 1340
1240 IF (M=4+Qort) OR (Pr=2) THEN GOTO 1290
1250  MAT PRINT #9;B
1260  PRINT #9;Thik
1270  MAT PRINT #9;Ro
1280  MAT PRINT #9;Poin
1290  IF (M<>4+Qort) OR (Pr=1) THEN GOTO 1340
1300  MAT PRINT #10;B
1310  PRINT #10;Thik
1320  MAT PRINT #10;Ro
1330  MAT PRINT #10;Poin
1340  NEXT M
1350  IF Nskw=0 THEN GOTO 1370
1360  CALL Mskewdcon(As(*),Ke(*),Alp,Nskw,Nosk(*),Node(*),Test
      (*),Z,Qort)
1370  Icoun1=1
1380  Icoun2=0
1390  Itest=1
1400  IF Nk=0 THEN GOTO 1490
1410  FOR I=1 TO 6+2*Qort
1420  An(I)=Node(Z,I)
1430  FOR J=1 TO Nk
1440  IF Node(Z,I)<>St(J,2) THEN GOTO Nextj
1450  Node(Z,I)=St(J,1)
1460  GOTO Nexti
1470 Nextj: NEXT J
1480 Nexti: NEXT I
1490  FOR I=1 TO 6+2*Qort
1500  N=0
1510  FOR K=1 TO Node(Z,I)
1520  N=N+5-2*Test(K)
1530  NEXT K
1540  FOR J=1 TO 6+2*Qort
1550  Jf1=0
1560  L=0
1570  FOR K=1 TO Node(Z,J)
1580  L=L+5-2*Test(K)
1590  NEXT K
1600  FOR K=1 TO J
1610  Jf1=Jf1+5-2*Test(Node(Z,K))
1620  NEXT K
1630  FOR M=4-2*Test(Node(Z,J)) TO 0 STEP -1
1640  Sub1=N-4+2*Test(Node(Z,I))
1650  Sub2=N-3+2*Test(Node(Z,I))
1660  Sub3=L-M
1670  Sub4=N-2+2*Test(Node(Z,I))
1680  IF Test(Node(Z,I))<>0 THEN GOTO 1710
1690  Sub5=N-1
1700  Sub6=N
1710  IF Sub1<Sub3 THEN GOTO Lab1
1720  An2=Add(Sub1)-Sub1+Sub3
1730  CALL Test(Ck1,Ck2,Chta,Chtp,Ck3,Ck4,Ck1,Ck2,K(*),K(*),An
      2,Lim(*),#1,#2,#3,#4,#5,#6,#7,#8,#9,#10,1,Seg)
1740  An3=An2-Ck1
1750  An1=INT(An3)
1760  K(An1)=K(An1)+Ke(Itest,Jf1-M)
1770 Lab1: IF Sub2<Sub3 THEN GOTO Lab2
1780  An2=Add(Sub2)-Sub2+Sub3
1790  CALL Test(Ck1,Ck2,Chta,Chtp,Ck3,Ck4,Ck1,Ck2,K(*),K(*),An
      2,Lim(*),#1,#2,#3,#4,#5,#6,#7,#8,#9,#10,1,Seg)
```

```
1800  An3=An2-Ck1
1810  An1=INT(An3)
1820  K(An1)=K(An1)+Ke(Itest+1,Jf1-M)
1830 Lab2: IF Sub4<Sub3 THEN GOTO Lab3
1840  An2=Add(Sub4)-Sub4+Sub3
1850  CALL Test(Ck1,Ck2,Chta,Chtp,Ck3,Ck4,Ck1,Ck2,K(*),K(*),An
      2,Lim(*),#1,#2,#3,#4,#5,#6,#7,#8,#9,#10,1,Seg)
1860  An3=An2-Ck1
1870  An1=INT(An3)
1880  K(An1)=K(An1)+Ke(Itest+2,Jf1-M)
1890 Lab3: IF Test(Node(Z,I))<>0 THEN GOTO Lab
1900  IF Sub5<Sub3 THEN GOTO Lab4
1910  An2=Add(Sub5)-Sub5+Sub3
1920  CALL Test(Ck1,Ck2,Chta,Chtp,Ck3,Ck4,Ck1,Ck2,K(*),K(*),An
      2,Lim(*),#1,#2,#3,#4,#5,#6,#7,#8,#9,#10,1,Seg)
1930  An3=An2-Ck1
1940  An1=INT(An3)
1950  K(An1)=K(An1)+Ke(Itest+3,Jf1-M)
1960 Lab4: IF Sub6<Sub3 THEN GOTO Lab
1970  An2=Add(Sub6)-Sub6+Sub3
1980  CALL Test(Ck1,Ck2,Chta,Chtp,Ck3,Ck4,Ck1,Ck2,K(*),K(*),An
      2,Lim(*),#1,#2,#3,#4,#5,#6,#7,#8,#9,#10,1,Seg)
1990  An3=An2-Ck1
2000  An1=INT(An3)
2010  K(An1)=K(An1)+Ke(Itest+4,Jf1-M)
2020 Lab: NEXT M
2030  NEXT J
2040  IF Icoun2>Icoun1 THEN GOTO 2090
2050  Icoun1=Icoun1-1
2060  Icoun2=Icoun2+1
2070  Itest=Itest+3
2080  GOTO Nexti
2090  Icoun1=Icoun1+1
2100  Icoun2=Icoun2-1
2110  Itest=Itest+5
2120 Nexti: NEXT I
2130  IF (Ch1=0) AND (Ch2=0) AND (Omg=0) THEN GOTO 2700
2140  J=1
2150  FOR I=1 TO 6+2*Qort STEP 2
2160  M=N=L1=L2=Icoun1=Icoun2=0
2170  IF Nk=0 THEN GOTO 2320
2180  FOR L=1 TO Nk
2190  IF An(I)<>St(L,2) THEN GOTO Next1
2200  L1=L1+1
2210  IF L=1 THEN GOTO N1
2220  FOR Sub1=1 TO L-1
2230  Icoun1=Icoun1+5-2*Test(St(Sub1,2))
2240  NEXT Sub1
2250  GOTO N1
2260 Next1: NEXT L
2270  GOTO 2320
2280 N1: Q3(Icoun1+1)=Q3(Icoun1+1)+Felvec(J)
2290  Q3(Icoun1+2)=Q3(Icoun1+2)+Felvec(J+1)
2300  Q3(Icoun1+3)=Q3(Icoun1+3)+Felvec(J+2)
2310  GOTO 2370
2320  IF Node(Z,I)=1 THEN GOTO 2370
2330  FOR K=1 TO Node(Z,I)-1
2340 M=M+5-2*Test(K)
2350  NEXT K
2360  IF Nk=0 THEN GOTO Az1
2370  FOR L=1 TO Nk
2380  IF An(I+1)<>St(L,2) THEN GOTO N2
2390  L2=L2+1
2400  FOR Sub1=1 TO L-1
2410  Icoun2=Icoun2+5-2*Test(St(Sub1,2))
2420  NEXT Sub1
```

```
2430  GOTO 2460
2440 N2: NEXT L
2450  GOTO Az1
2460  Q3(Icoun2+1)=Q3(Icoun2+1)+Felvec(J+3)
2470  Q3(Icoun2+2)=Q3(Icoun2+2)+Felvec(J+4)
2480  Q3(Icoun2+3)=Q3(Icoun2+3)+Felvec(J+5)
2490  Q3(Icoun2+4)=Q3(Icoun2+4)+Felvec(J+6)
2500  Q3(Icoun2+5)=Q3(Icoun2+5)+Felvec(J+7)
2510  GOTO 2560
2520 Az1: FOR K=1 TO Node(Z,I+1)-1
2530  N=N+5-2*Test(K)
2540 NEXT K
2550  IF Nk=0 THEN GOTO 2570
2560  IF L1<>0 THEN GOTO 2610
2570  Q(M+1,1)=Q(M+1,1)+Felvec(J)
2580  Q(M+2,1)=Q(M+2,1)+Felvec(J+1)
2590  Q(M+3,1)=Q(M+3,1)+Felvec(J+2)
2600  IF Nk=0 THEN GOTO 2620
2610  IF L2<>0 THEN GOTO 2670
2620  Q(N+1,1)=Q(N+1,1)+Felvec(J+3)
2630  Q(N+2,1)=Q(N+2,1)+Felvec(J+4)
2640  Q(N+3,1)=Q(N+3,1)+Felvec(J+5)
2650  Q(N+4,1)=Q(N+4,1)+Felvec(J+6)
2660  Q(N+5,1)=Q(N+5,1)+Felvec(J+7)
2670  J=J+8
2680  NEXT I
2690  MAT Felvec=ZER
2700  FOR I=1 TO 6+2*Qort
2710  Node(Z,I)=A1(I)
2720  NEXT I
2730  NEXT Z
2740  IF (Pr=3) OR (Pr=1) THEN ASSIGN #9 TO *
2750  IF (Pr=3) OR (Pr=2) THEN ASSIGN #10 TO *
2760  SUBEND !End of Massembly
```

```
10    ! RE-SAVE"SKWIM1"
20    SUB Skewim1(Q(*),Angsk(*),Qb(*),Alp,Nskew,INTEGER Test(*
      ),J,Nosk(*),St(*),Nk,Nskewd,Nskewc)
30    OPTION BASE 1
40    DIM S(5,5)
50    REM ----Transformation of disc started
60    FOR I=1 TO Nskew
70    A=A1=L=0
80    FOR J1=1 TO Nk
90    IF Nosk(I)<>St(J1,2) THEN GOTO 120
100   L=L+1
110   GOTO N2
120   NEXT J1
130 N2:  FOR K=1 TO Nosk(I)
140   A=A+5-2*Test(K)
150   NEXT K
160   IF L=0 THEN GOTO 200
170   FOR K=1 TO J1
180   A1=A1+5-2*Test(St(K,2))
190   NEXT K
200   IF I>Nskewd THEN GOTO N1
210   S(1,1)=COS(Angsk(I)*.0174533)
220   S(1,2)=SIN(Angsk(I)*.0174533)
230   S(2,1)=-SIN(Angsk(I)*.0174533)
240   S(2,2)=COS(Angsk(I)*.0174533)
250   S(3,3)=1
260   GOTO N3
270 N1:  Phi=Angsk(I)*.0174533
280   S(1,1)=-COS(Phi)*COS(Alp*.0174533)
290   S(1,2)=-SIN(Phi)*COS(Alp*.0174533)
300   S(1,3)=-SIN(Alp*.0174533)
310   S(2,1)=-COS(Phi)*SIN(Alp*.0174533)
320   S(2,2)=-SIN(Phi)*SIN(Alp*.0174533)
330   S(2,3)=COS(Alp*.0174533)
340   S(3,1)=-SIN(Phi)
350   S(3,2)=COS(Phi)
360   S(3,3)=0
370 N3:  IF L=0 THEN GOTO 420
380   M=Qb(A1-4+2*Test(St(J1,2)))
390   N=Qb(A1-3+2*Test(St(J1,2)))
400   O=Qb(A1-2+2*Test(St(J1,2)))
401   Qb(A1-4+2*Test(St(J1,2)))=S(1,1)*M+S(1,2)*N+S(1,3)*O
402   Qb(A1-3+2*Test(St(J1,2)))=S(2,1)*M+S(2,2)*N+S(2,3)*O
403   Qb(A1-2+2*Test(St(J1,2)))=S(3,1)*M+S(3,2)*N+S(3,3)*O
410   GOTO 480
420   M=Q(A-4+2*Test(Nosk(I)),J)
430   N=Q(A-3+2*Test(Nosk(I)),J)
440   O=Q(A-2+2*Test(Nosk(I)),J)
450   Q(A-4+2*Test(Nosk(I)),J)=S(1,1)*M+S(1,2)*N+S(1,3)*O
460   Q(A-3+2*Test(Nosk(I)),J)=S(2,1)*M+S(2,2)*N+S(2,3)*O
470   Q(A-2+2*Test(Nosk(I)),J)=S(3,1)*M+S(3,2)*N+S(3,3)*O
480   NEXT I
490   SUBEND! End of Skewim1
```

```
2000 SUB Segsol(A$(*),A(*),B(*),S(*),Lim(*),INTEGER N,R,Seg,Co
     l(*),Row(*))
2001 COM I1,I2,I3,I4,Chta,I5,Size
2010 DIM L(Size)
2020 INTEGER I,J,J1,J2
2030 H=0
2031 FOR I=1 TO Seg
2032 ASSIGN #I TO A$(I)
2033 NEXT I
2040 READ #Chta,1
2050 MAT PRINT #Chta;A
2060 READ #1,1
2070 MAT READ #1;A
2080 FOR I=1 TO Seg
2090 FOR W=Col(I) TO I
2100 IF W=I THEN GOTO 2130
2110 READ #W,1
2120 MAT READ #W;L
2130 FOR Z=Row(I-1)+1 TO Row(I)
2140 P=S(Z-1)
2150 T=Z-(S(Z)-P)+1
2160 G=P+1
2170 Q=Row(W-1)+1
2180 IF Z+1-Q>=Z+1-T THEN GOTO 2210
2190 H=Q-T
2200 GOTO 2220
2210 H=0
2220 H=H+P
2230 IF W<>I THEN GOTO 2260
2240 P1=Z-1
2250 GOTO 2270
2260 P1=Row(W)
2270 IF T>Q THEN Q=T
2280 FOR J=Q TO P1
2290 U=G
2300 H=H+1
2310 K=J-(S(J)-S(J-1))+1
2320 V=H-S(J)
2321  J1=INT(H-Lim(I-1))
2330 Y=A(J1)
2340 IF K>T THEN U=U+K-T
2350 FOR U=U TO H-1
2360 IF W<>I THEN GOTO 2381
2361  J1=INT(U-Lim(I-1))
2362  J2=INT(U-V-Lim(I-1))
2370 Y=Y-A(J1)*A(J2)
2380 GOTO 2400
2381  J1=INT(U-Lim(I-1))
2382  J2=INT(U-V-Lim(W-1))
2390 Y=Y-A(J1)*L(J2)
2400 NEXT U
2410 IF W<>I THEN GOTO 2431
2411  J1=INT(H-V-Lim(I-1))
2420 Y=Y/A(J1)
2430 GOTO 2441
2431  J1=INT(H-V-Lim(W-1))
2440 Y=Y/L(J1)
2441  J2=INT(H-Lim(I-1))
2450 A(J2)=Y
2460 FOR M=1 TO R
2470 B(Z,M)=B(Z,M)-B(J,M)*Y
2480 NEXT J
2490 IF W<>I THEN GOTO 2630
2491  J1=INT(H+1-Lim(I-1))
2500 Y=A(J1)
```

```
2510 FOR U=G TO H
2511  J1=INT(U-Lim(I-1))
2520 Y=Y-A(J1)^2
2530 NEXT U
2540 IF Y>0 THEN GOTO 2570
2550 PRINT "****PROGRAM FAILED IN SYMVBS****"
2560 STOP
2570 Y=SQR(Y)
2580 H=H+1
2581  J1=INT(H-Lim(I-1))
2590 A(J1)=Y
2600 FOR M=1 TO R
2610 B(Z,M)=B(Z,M)/Y
2620 NEXT M
2630 NEXT Z
2640 NEXT W
2650 IF I=Seg THEN GOTO Reduced
2660 READ #I,1
2670 MAT PRINT #I;A
2680 READ #I+1,1
2690 MAT READ #I+1;A
2700 Reduced:NEXT I
2710 REM REDUCTION COMPLETE
2720 H=S(N)
2730 FOR J=Seg TO 1 STEP -1
2740 FOR I=Row(J) TO Row(J-1)+1 STEP -1
2741 J1=INT(H-Lim(J-1))
2750 Y=A(J1)
2760 FOR M=1 TO R
2770 B(I,M)=B(I,M)/Y
2780 NEXT M
2790 IF I=1 THEN GOTO Complete
2800 V=I
2810 P=S(I-1)
2820 FOR H=H-1 TO P+1 STEP -1
2830 V=V-1
2831 J1=INT(H-Lim(J-1))
2840 Y=A(J1)
2850 FOR M=1 TO R
2860 B(V,M)=B(V,M)-B(I,M)*Y
2870 NEXT H
2880 H=P
2890 NEXT I
2900 READ #J-1,1
2910 MAT READ #J-1;A
2920 Complete:NEXT J
2930 SUBEND!END of Segsol
```

```
10    ! RE-SAVE"SKWIM2"
20    SUB Skewim2(Q(*),Angsk(*),Qb(*),Alp,Nskew,INTEGER Test(*
      ),J,Nosk(*),St(*),Nk,Nskewd,Nskewc)
30    OPTION BASE 1
40    DIM S(5,5)
50    REM ----Transformation of disc started
60    Count=0
70    FOR I=1 TO Nskew
80    IF I>Nskewd THEN Count=1
90    A=A1=L=0
100   FOR J1=1 TO Nk
110   IF Nosk(I)<>St(J1,2) THEN GOTO 140
120   L=L+1
130   GOTO N2
140   NEXT J1
150 N2:  FOR K=1 TO Nosk(I)
160   A=A+5-2*Test(K)
170   NEXT K
180   IF (Count=0) AND (I>Nskewd/2) THEN GOTO 240
190   IF (Count=1) AND (I>Nskewd+Nskewc/2) THEN GOTO 231
200   M=Q(A-4+2*Test(Nosk(I)),J)
210   N=Q(A-3+2*Test(Nosk(I)),J)
220   O=Q(A-2+2*Test(Nosk(I)),J)
230   GOTO 310
231   Coun=Coun+1
232   IF Coun>1 THEN GOTO 240
233   Jn=Nskewd
240   Jn=Jn+1
250   FOR K=1 TO St(Jn,1)
260   A1=A1+5-2*Test(St(K,1))
270   NEXT K
280   M=Qb(A1-4+2*Test(Nosk(I)))
290   N=Qb(A1-3+2*Test(Nosk(I)))
300   O=Qb(A1-2+2*Test(Nosk(I)))
310   IF Count=1 THEN GOTO N1
320   S(1,1)=COS(Angsk(I)*.0174533)
330   S(1,2)=SIN(Angsk(I)*.0174533)
340   S(2,1)=-SIN(Angsk(I)*.0174533)
350   S(2,2)=COS(Angsk(I)*.0174533)
360   S(3,3)=1
370   GOTO N3
380 N1:  Phi=Angsk(I)*.0174533
390   S(1,1)=-COS(Phi)*COS(Alp*.0174533)
400   S(2,1)=-SIN(Phi)*COS(Alp*.0174533)
410   S(3,1)=-SIN(Alp*.0174533)
420   S(1,2)=-COS(Phi)*SIN(Alp*.0174533)
430   S(2,2)=-SIN(Phi)*SIN(Alp*.0174533)
440   S(3,2)=COS(Alp*.0174533)
450   S(1,3)=-SIN(Phi)
460   S(2,3)=COS(Phi)
470   S(3,3)=0
480 N3:  Q(A-4+2*Test(Nosk(I)),J)=S(1,1)*M+S(1,2)*N+S(1,3)*O
490   Q(A-3+2*Test(Nosk(I)),J)=S(2,1)*M+S(2,2)*N+S(2,3)*O
500   Q(A-2+2*Test(Nosk(I)),J)=S(3,1)*M+S(3,2)*N+S(3,3)*O
510   NEXT I
520   SUBEND! End of Skewim2
```

REFERENCES

1. Timoshenko, S. P. and Woinowsky-Krieger, S. — "Theory of plates and shells" 2nd Ed., McGraw-Hill Book Company, New York, 1959.

2. Haerle, H. — "The strength of rotating discs" Engineering Vol. CVI. 1918, p. 131.

3. Stodola, A. — "Steam and gas turbines", Vol. 1, McGraw-Hill Company, 1945.

4. Meriam, J. L. — "Stresses and displacements in a rotating conical shells", J. App. Mech. A.S.M.E., 1943.

5. Bell, R. — "Theoretical and experimental stress analysis of centrifugal fan impellers" Ph.D. Thesis, The Queen's University of Belfast, 1975.

6. Thurston, G. A. — "A numerical solution for thin conical shells under asymmetric loads", 4th Midwestern Conf. on Solid Mechanics, University of Texas, Austin, 1959.

7. Schile, R. — "Analysis of thin conical shells under asymmetric loading", 4th Midwestern Conf. on Solid Mechanics, University of Texas, Austin, 1959.

8. Osborne, W.C. — "Fans", Pergamon Press, 1966.

9. Patton, R. G. — "Stresses and displacements in a rotating conical shells", J. App. Mech. A.S.M.E., 1943.

10. Glessner, J. W. — "A method of analysing the stresses in centrifugal impellers", Am. Soc. of Mech. Eng., Paper 54-A-167.

11. Thurgood, D. A. — "Stresses in asymmetric discs", Journal of Strain Analysis, Vol. 4, No. 1, 1969.

12. Brooking, R. L., Brown, J. and Attkins, B.R. — "A direct easily computable method for the evaluation of centrifugal and thermal stresses in discs, and stress changes due to plastic flow and creep", National Gas Turbine Establishment Report R122, July 1956.

13. Bell, R., and Benham, P.P. "Theoretical and experimental stress analysis of centrifugal fan impellers", Journal of Strain Analysis, Vol. 13, No. 3, 1978.

14. Richards, T. H. "Essential of stress analysis implementation in approximation techniques", Journal of Strain Analysis, Vol. 14, No.2, 1979.

15. Patterson, C "On the efficient use of numerical model data by the designer", I. Mech. Engineering Conference Publications 1981-7 Engineering Research and Design Bridging the Gap.

16. Delves, B. "A semi-analytic approach to stress analysis of circular plates", M.Phil Thesis, University of Aston in Birmingham, 1977.

17. Mahabaliraja "Dynamic response of noncircular stiffened cylindrical shells with rectangular cutouts", Faculty of the Graduate College of the Oklahoma State University, Ph.D. Thesis, 1974.

18. Sheba k, S. A. M. "Mechanical vibration characteristics of bell-type pump mountings", Ph.D. Thesis, University of Aston in Birmingham, 1981.

19. Richards, T. H. "Energy methods in stress analysis", Ellis-Horwood series in Engineering Science, London, 1977.

20. Zienkiewicz, O. C. "The finite element method in engineering science", McGraw-Hill Book Company, New York, 1971.

21. Bazely, G. P., Cheung, Y. K., Irons, B. M. and "Triangular elements in plate elements, conforming and nonconforming solutions", Proc. (first) Conf. on Matrix Methods in Structural Mech., Wright Patterson, AFB Ohio, 1965.

22. Irons, B. M. and Razzaque, A. "Experience with the patch test for the convergence of finite elements", Conf. Math. found. finite element method, University of Maryland (Baltimore), 1972, Academic Press, New York and London pp. 557-587.

23. Irons, B. M. "The patch test for engineers", paper presented at Finite Elements Symposium, Atlas Computing Laboratory, Chilton Didcot, Berks, 27th March 1979.

24. Stummel, F. "The limitations of the patch test", Int. J. of Num. Methods in Engineering, Vol. 15, pp. 177-188, 1980.

25. Zienkiewicz, O. C. and Cheung, Y. K. "Finite element method of analysis of arch and dam shells and comparison with finite difference", Proc. Symp. on the Theory of Arch Dams. Pergamon Press, New York, 1965.

26. Carr, A. J. "A refined finite element analysis of thin shell structures including dynamic loadings", Struct. Eng. Lab., Report No. 67-9, University of California, Berkerly, 1967.

27. Chu, T. C. and Schnobich, W. C. "Finite element analysis of translational shells", Technical Report of Civil Engineering Dept., University of Illinois, 1970.

28. Dawe, D. J. "The analysis of thin shells using a facet element", Berkerly Nuclear Lab., C.E.G.B., Report No. RD/B/N2038. 1971

29. Mervyn, D. O. and Terrence, W. B. "A simple flat triangular shell element revised", Int. J. Num. Methods in Engineering Vol. 14, pp. 51-68, (1979).

30. Klaus, B. J. and Lee-Wing, H. , "A simple and effective element for analysis of general shell structures", Comput. and Structures, Vol. 13, pp. 673-681, 1981.

31. Connor, J., and Brebbia, C. "Stiffness matrix for shallow rectangular shell element", Proc. A.S.C.E., J. of the Eng. Mech. Div., EM-2, 1968.

32. Gallagher, R. H., and Yang, H. "Elastic Instability predictions for doubly curved shells", Proc. (second) Conf. on Matrix Methods in Struct. Mech., Wright- Patterson AFB, Ohio, 1968.

33. Cantin, R., and Clough, R. W. "A curved cylindrical discrete elements", J. AIAA, Vol. 6, No. 5, 1968.

34. Olsen, M. and Lindberg, G. "Vibration analysis of cantilevered curved plates using a new cylindrical shell finite element", Proc. (second) Conf. on Matrix Methods in Struct. Mech., Wright-Patterson AFB - Ohio, 1968.

35. Love, A. E. N. "A treatise on mathematical theory of elasticity", Dover Publications 4th Ed., 19

36. Gionnini, M., and Miles, G. A. "A curved element approximation to the analysis of axisymmetric curved shells", Int. J. Num. Methods in Engineering, Vol. 2, pp. 459-476, 1970.

37. Henshell, R. D. "New cylindrical shell finite element", J. of Sound and Vibration, 16(4), pp. 519-531, 1971.

38. Birkhoff, G. and Garabedian, H. "Smooth surface interpolation", J. of Math. Physics, Vol. 39, pp. 358-368, 1960.

39. Fonder, G. A. "Studies in doubly-curved elements for shells of revolution", Conference on finite elements applied to thin shells and curved members, University College, Cardiff, Wales 1974. Edited by D. G.Ashwell and R. H. Gallagher.

40. Key, S. W. and Beisinger, Z. E. "The analysis of thin shells with transverse shear strains by the finite element method", Proc. (second) Conf. on Matrix Methods in Struct. Mech., Wright-Patterson, AFB - Ohio, 1968.

41. Ergatodis, J., Irons, B. M. and Zienkiewicz, O. C. "Curved isoparametric quadrilateral element for finite element analysis", Int. J. of Solids and Structures, Vol. 4, pp. 31-42, 1968.

42. Greene, B. E., Jones, R. E. and Strome, D. R. "Dynamic analysis of shells using doubly curved finite elements", Proc (second) Conf. on Matrix Methods in Struct. Mech., Wright-Patterson, AFB - Ohio, 1968.

43. Greene, B. E., Jones, R. E., McLay, R. W., and Strome, D. R. "On the application of generalised variational principles in the finite element method", AIAA Paper No. 68-290, AIAA/ASME gth structures, structural dynamics and materials Conference, Palms Springs, California, 1968.

44. Wempner, G., Oden, J. T. and Kros, D. "Finite element analysis of thin shells", Proc. ASCE J. of Eng. Mech. Div. EM-6, 1968.

45. Argyris, J. H. and Scharpf, D. "The SHEBA family of shell elements for the matrix displacement method", Aero. Journal, pp. 873-883, 1968.

46. Argyris, J. H. and Lochner, N. "On the application of the SHEBA shell element", Comp. Methods in Applied Mechanics and Engrg., Vol. 1, pp. 317-347, 1972.

47. Ahmad, S. "Curved finite elements in the analysis of solid shell and plate structures", Ph.D. Thesis, University of Wales, Swansea, 1969.

48. Too, J. T. M. "Two dimensional plate, shell and finite prism isoparametric elements and their applications", Ph.D. Thesis, University of Wales, Swansea, 1971.

49. Cowper, G. R., Lindberg, G. M. and Olson, M. D. "A shallow shell finite element of triangular shape", Int. J. Solids and Structures, Vol. 6, pp. 1133-1156, 1970.

50. Ernst, L. J. "A finite element approach to shell problems", IUTAM Symposium on Shell Theory, W. T. Koiter and Mikhailov, North Holland Publishing Company, 1980.

51. Mohr, G. "A doubly curved isoparametric triangular shell element" Computers and Structures, Vol. 14, No. 1-2, pp. 9-13, 1981.

52. Sander, G. and Idelsohn, S. "A family of conforming finite elements for deep shell analysis" Int. J. for Num. Methods in Engineering, Vol. 18, pp. 363-380, 1982.

53. Irons, B. M. "The semiloof shell element", Conference on Finite Elements Applied to Thin Shells and Curved Members, Chapter 11, University College, Cardiff, Wales, 1974, Edited by D. G. Ashwell and R. H. Gallagher.

54. Gallagher, R. H. "Shell elements", Proc. World Conf. Finite Element Method in Structural Mechanics, Bournemouth, England, 1975.

55. Ergatoudis, J. G. "Isoparametric finite elements in two and three dimensional anlaysis", Ph.D. Thesis, University of Wales, Swansea, 1968.

56. Zienkiewicz, O. C. "Reduced integration technique in general analysis of plates and shells", Int. J. for Num. Methods in Engineering, Vol. 3, pp. 278-290, 1971.

57. Martins, R. A. F. "Structural instability and natural vibration analysis of thin arbitrary shells by use of the semiloof shell element", Int. J. Num. Methods in Engineering, Vol.11, pp. 481-498,1977.

58. Javaherian, P. J., Dowling and Lyons, L. P. R. "Nonlinear finite element analysis of shell structures using the semiloof shell element", Computers and Structures, Vol.12, pp. 147-159, 1980.

59. Martins, R. A. F. and Owen, D. K. J. "Elastic plastic and geometrically non-linear thin shell structures by the semiloof shell elements", J. of Computers and Structures, 1981.

60. Denis, L. M., and Owen, D. R. J. "Elasto-viscoplastic and Elasto-plastic large deflections analysis of thin plates and shells", Int. J. Num. Methods in Engineering, Vol. 18, pp. 591-607, 1982.

61. Martins, R. A. F. "Analysis of plates and arbitrary shells by the use of the semiloof shell element", Ph.D. Thesis, University of Wales, 1976.

62. Irons, B. M. "The semiloof shell elements". Lecture notes for the NATO research seminar on the theory and application of finite element methods, University of Calgary, July-August 1973.

63. Jennings, A., and Tuff, A. D. "A direct method for the solution of large sparse symmetric simultaneous equations", Proc. Oxford Conf. of the Institute of Mathematics and its Applications (I.M.A.), pp. 97-104, April 1970.

64. Irons, B. M., and Ahmad, S. "Techniques of finite elements", Ellis Horwood Series in Engineering Science, London, 1980.

65. Robertson, A. W. "On the stress analysis of cracked bodies by means of finite elements" Ph.D. Thesis, University of Aston in Birmingham, 1976.

66. Cook, R. "Concepts and applications of finite element analysis", John Wiley, 1974.

67. Hinton, E., Scott, F. C. and Ricketts, R. E. "Local least squares stress smoothing for parabolic isoparametric elements", Int. J. Num. Methods in Engineering, Vol. 9, pp.235-256, 1975.

68. Wood, P. C. "Application of finite element method to problems in Fracture Mechanics", Ph.D. Thesis, University of Aston in Birmingham, 1979.

69. Scordellis, A. C. and Lo, K. S. "Computer analyiss of cylindrical shells", J. Am. Concr. Inst., 61, May 1964.

70. Fellipa, C. A., and Clough, R. W. "A refined quadrilateral element for analysis of plate bending", 2nd Conference on matrix methods in struct. Mechanics, Wright-Patterson Base, Dayton, Ohio, October 1968.

71. Megard, G. "Planar and curved shelles", published in Finite Element Method in Stress Analysis, Edited by I. Holland and K. Bell, Tapir, Trondhium, Norway, 1969.

72. Flugge, W. "Stresses in Shells", Springer-Velay, Berlin, 1962.

73. MacNeal, R. L., Harder, J. B. and Mason "NASTRAN cyclic symmetry capability", Nastran Users Experiences, 3rd Colloquium, Langley Research Centre Hampton, Virginia, U.S.A, 11-12 September, pp. 395-421, 1973. (NASA Tech. Memo. NASA TMX-2893).

74. Thomas, D. L. "Dynamics of rotationally periodic structures", Int. J. of Num. Methods in Engineering, Vol. 14, pp. 81-102, 1979.

75. Nelson, R. L. "Investigation of shell and axisymmetric shell structures", Ph.D. Thesis, City University, London, 1978.

76. McEwan, J. R. "Stresses in centrifugal impellers", Ph.D. Thesis, University of Strathclyde, 1981.

77. Meriam, J. L. "Stresses and displacement in a rotating conical shell", J. Applied Mech. A.S.M.E., June 1943.

78. Richard, C. B. "Advanced strength and applied stress analysis", McGraw-Hill Book Company, New York, 1977.

79. Gall, M. H. W. "Strain gauge slip ring circuits", Journal of "Strain", January 1979.

80. King, I. P. "Triangular mesh generation program", Centre for Numerical Methods in Engineering, Swansea, C.P.R.2, 1967.

81. Zienkiewicz, O. C. and Phillips, D. V. "An automatic mesh generation scheme for plane and curved surfaces by isoparametric co-ordinates", Int. J. Num. Meth. Engng., 3, pp 519-528, 1971.

82. Steinmueller, G. "Restrictions in the application of automatic mesh generation schemes by isoparametric co-ordinates", Int. J. Num. Meth. Eng., 8, pp 289-294, 1974.

83. Sadek, E. A. "A scheme for the automatic generation of triangular finite elements", Int. J. Num. Meth. Eng., 15, pp. 1813-1822, 1980.

84. Dupis., V. and Goel., J. "A curved finite element for thin elastic shells". Int. J. Solids and Structures. V.6, 1970, pp 987-996.