UNIVERSITY OF STRATHCLYDE

DEPARTMENT OF MECHANICAL AND PROCESS ENGINEERING

DIVISION OF MECHANICS OF MATERIALS

A STUDY ON SIMPLE PIPING ELBOW FINITE ELEMENTS

by

Donald Mackenzie B.Sc.

March 1990

Thesis presented for the
Degree of Doctor of Philosophy

ABSTRACT

Three new simple beam-shell piping elbow finite elements have been formulated and programmed as User Elements for the ANSYS finite element program. The elements share a common beam bending model, based on an exact solution of Vlasov's thin-walled beam theory, but adopt different ovalisation interpolation schemes.

In the first element, PB1, ovalisation deformation is assumed constant with respect to axial position and interpolated circumferentially by an even Fourier series. The second element, PB2, extends the PB1 formulation to include linear interpolation of ovalisation with respect to axial position. In the third element, PB3, constant ovalisation deformation is interpolated piecewise around the cross-section by four quintic polynomial functions.

Several sample analyses of single bends were performed, and the results compared with published theoretical and experimental results. PB1 and PB2 were found to give good agreement with stresses from alternative solutions for a range of elbow geometries. The element PB2 solution converged slightly more rapidly than PB1, but the additional degrees of freedom required for linear interpolation negated the advantage of the higher order scheme. Element PB3 performed poorly in comparison with the other elements, especially in the analysis of elbows of low bend parameter. It was concluded that the polynomial interpolation scheme of element PB3 was less effective than Fourier interpolation of ovalisation.

Of the three elements presented, element PB1 was chosen in preference to PB2 for general piping analysis as its complete stiffness matrix had been obtained in closed form, making it computationally less expensive.

The use of element PB1 in piping system analysis was investigated by performing several sample analyses of systems and comparing results with flexibility and finite element analysis. It was found that the element gave accurate results at low computing costs, indicating its suitability for general elastic analysis of piping systems.

ACKNOWLEDGEMENTS

I would like to extend my thanks to the following people and organisations for their contributions to this thesis:

Dr. J T Boyle for his assistance, advice and encouragement whilst supervising this study.

Professor J Spence, head of Division, for providing the opportunity and facilities required to carry out the work.

My fellow research students and research assistants for their valuable advice and encouragement.

Stress Analysis Services - Computing, for the use of their facilities in preparing the manuscript.

Swanson Analysis Systems Inc., Hibbitt Karlsson and Sorensen Inc. and the Inference Corporation for the use of the computer programs ANSYS, ABAQUS and SMP under academic licence.

Finally, I would especially like to thank my Parents and Janet for all the support and encouragement they have given me over the years.

CONTENTS. PAGE

APPENDICES.

CHAPTER 1.

INTRODUCTION.

1 INTRODUCTION.

Pipework is used extensively in industry for transporting material between items of plant. In many applications, for example in the petro-chemical and power industries, the material may be intrinsically hazardous or in a hazardous thermodynamic state. Safety considerations require such piping to have high structural integrity, whilst economic considerations require minimum structural redundancy in the design. In practice piping safety and economy are ensured by designed systems according to a particular design code or specification.

Piping codes generally allow two approaches to system design. In one approach, a standard design procedure is adopted and the design drawn up according to a set of rules defined in the code. Alternatively, an analysis of a proposed design may be carried out in order to demonstrate that the stresses within the pipes and forces at connections to items of plant are within specified limits. If the code limits are exceeded, the pipework designer modifies the design and repeats the analysis until the requirements are met.

The main problem in piping analysis is the size and complexity of piping systems. Piping components such as elbows, bellows and branch connections are essentially shell structures. A rigorous analysis of a piping system would therefore be based on shell theory. However, the size and cost of a full shell analysis of a general system makes such an approach impracticable at present, and a simplified analysis must be used.

In contemporary practice, most piping design and analysis is done using computer pipe stress analysis programs in which the piping system is modelled as a framework assembled from simple beams. The model is analysed to obtain the forces on the components, from which the pipe stresses are evaluated.

The engineers theory of bending makes two fundamental assumptions about the behaviour of long slender straight or curved beams:

i) Plane sections remain plane during bending.

ii) The cross-section of the beam does not deform during bending, (Poisson effects being neglected).

Under these assumptions, only longitudinal stress and strain are induced in beams under bending.

The beam bending assumptions are generally valid for straight pipes, but are not applicable in the case of pipe bends. Cross-sectional ovalisation, violating assumption (ii) above, is a fundamental deformation mode of a piping elbow under bending. This is called the Von Karman effect; it causes a pipe bend to be more flexible than a curved beam of the same nominal dimensions and induces higher and more complex longitudinal and hoop stress in the elbow.

It is not, therefore, possible to directly model pipe bends as curved beams, and in practice factors correcting elbow flexibility and stress intensification (derived from shell analyses of bends) are required in the simplified beam analysis. The above approach is generally referred to as piping flexibility analysis.

However, a more realistic alternative analysis method for piping systems has been available since the early 1970's. This is finite element analysis using special purpose pipe bend elements which incorporate elbow ovalisation as a fundamental deformation mode. Such elements require neither flexibility nor stress intensification factors, and give more detailed and accurate stress results for pipe bends. A number of ovalising element formulations have been presented in the literature, several of which have been incorporated in commercial finite element programs. However, these elements have not generally been accepted for elastic piping analysis, as required by the piping design codes. This is mainly due to two factors:

1) The flexibility analysis is a tried and tested piping analysis method, which has been used successfully for many years.

2) Finite element analysis using special piping elements is more expensive, as the elbow elements are computationally larger and require numerical integration of the element stiffness matrices.

1.1 Scope of Present Study

The object of this present study is to review the elbow element literature, to present possible formulations for simple elbow elements suitable for the elastic analysis of piping systems and to investigate the performance of the elements proposed. The elements are two-node displacement based finite elements, with axial, bending and torsional deformation modes based on beam theory and elbow ovalisation modes based on a reduced, 2-dimensional, shell theory.

The proposed elements have been programmed as "user elements" in the commercial Finite Element system ANSYS. ANSYS is a powerful general purpose finite element program offering a wide range of analysis capabilities, including linear and non linear statics, dynamics, heat transfer, fluid flow, electrostatics and magnetostatics. The program offers a large library of elements for problems of the above types, and in addition allows the user to define his own element which interacts with the program in the same way as the standard ANSYS elements.

Programming the elbow elements as ANSYS user elements has two significant advantages over using a finite element program written specifically for element development. Firstly, the ANSYS pre-processor can be used to create finite element models interactively and the post-processor used to display analysis results. Secondly, the elements are accessible to other workers using ANSYS, facilitating further study and application of the elements.

There is no official ANSYS documentation describing the user element capability, other than a brief mention in Appendix P of the ANSYS User Manual. The user element capability is described an ANSYS program called USER.ROUTINES, which is supplied with main-frame and workstation versions of ANSYS. USER.ROUTINES includes documented source code for an example user element: a 3-D spar element.

In order to clarify the user element programming procedure, an extensive "User Element Programming Manual" was written specifically to accompany this thesis and is included as Appendix 1. The Appendix is limited to the programming of linear elastic structural elements. User element source code for the elbow elements formulated in the thesis are given in Appendix 4.

In the body of the thesis, a discussion of piping analysis methods, including flexibility analysis and the finite element method, is presented in Chapter 2.

A review of the piping elbow finite element literature to date is presented in Chapter 3, and salient features of such elements discussed with a view to formulating simple elements for elastic analysis.

In Chapter 4, a number of elbow ovalisation models are studied by investigating the axial deformation of a semi-toroidal bellows expansion joint, which is similar to the ovalisation deformation mode of a piping elbow. Four bellows finite element formulations are presented.

Three elbow element formulations are presented in Chapter 5. The elements share a common beam model, based on an exact curved beam solution. In elements PB1 and PB2, Fourier interpolation of a two-dimensional ovalisation strain field (both constant and linear with respect to axial position) is investigated. In element PB3, piecewise quintic polynomial interpolation of the two dimensional ovalisation strain field is studied.

In Chapter 6 a number of sample analyses of bellows, elbows and piping systems are presented, in order to assess the accuracy and applicability of the element formulations presented in Chapters 4 and 5. Results are compared with published analytical, experimental and finite element results. In addition, a number of flexibility analyses and finite element analyses using commercial finite elements were performed for comparison purposes.

Finally, conclusions of the study are presented in Chapter 7, and recommendations for further investigations in the field of finite element analysis of piping systems given.

CHAPTER 2.

DESIGN AND ANALYSIS OF PIPING SYSTEMS AND PIPING ELBOWS.

2 DESIGN AND ANALYSIS OF PIPING SYSTEMS AND PIPING ELBOWS.

In many industries plant safety and reliability considerations require industrial piping to retain its structural integrity throughout its operational life. In order to meet this requirement piping systems are built to a specified piping code which defines rules for design, materials, construction and inspection of the system.

Codes, standards and regulations covering the design of piping systems are issued by a number of Standards bodies, including the British Standards Institute (BSI) in the UK, and the American National Standards Institute (ANSI) and the American Society of Mechanical Engineers (ASME) in the USA. For example, three of the most popular codes in the UK petro-chemical and power industries are:

BS806: Ferrous Pipes and Piping for and in connection with Land Boilers [2.1].

ANSI/ASME B31.1: Code for Power Piping [2.2].

ASME Pressure Vessel & Boiler Code Section III: Nuclear power plant components [2.3]

These codes will be used to provide specific examples of some of the general points discussed in this chapter.

Industrial piping is subject to many different kinds of loading, but for the purposes of code design three categories are usually considered. These are sustained loads, occasional loads, and expansion loads.

Sustained loads arise from the mechanical forces present throughout normal operation of the system, and include self weight, fluid weight, insulation weight and internal (or external) pressure.

Occasional loads also arise from mechanical forces, but are expected to occur during only a small proportion of the plant life. Occasional loads include wind loading, seismic loading, loads arising from possible plant accidents and intermittent operational loads such as relief valve discharge.

Expansion loads arise when piping systems experience changes in operating temperature, causing thermal expansion of the piping material. In an unconstrained system such expansion is stress-free. However, in a constrained system, thermal stresses are set up in the piping components and reaction forces occur at connections to plant, supports and anchors.

The object of design codes is to guard against failures arising from these loadings. The codes may identify several failure modes, but most commonly consider three: excessive plastic deformation or bursting, ratchetting and fatigue.

Excessive plastic deformation is considered to arise from sustained and occasional loads. These loads give rise to stresses in the wall of the pipe which, if high enough, may cause the piping material to yield. Ultimately, if the loading is high enough to cause the yielding to spread through the cross-section of the pipe, failure occurs due to rupture or burst.

Ratchetting failure may occur when a system subject to sustained and occasional loading also experiences expansion loads, causing cyclic stresses in the pipe wall. The magnitude of the stresses in the first loading cycle determines the behaviour of the system under subsequent cycles. If the maximum combined sustained occasional and expansion stresses are within the elastic range of the piping material, the piping system will exhibit wholly elastic behaviour in all loading cycles. If, however, the stresses occurring during the first cycle cause the piping to yield, one of two types of behaviour will occur on subsequent cycles. If the initial yielding sets up a state of residual stress in the pipe wall such that subsequent cycling causes no further plastic deformation, the system is said to exhibit shakedown. However, if the residual stresses are not great enough to accommodate subsequent cycles elastically, successive cycles lead to cumulative plastic deformation or ratchetting of the piping material, eventually causing local plastic failure in the wall of the pipe.

Fatigue failure occurs when stress raisers cause peak stresses to arise in the system. The peak values may be greater than yield, but, as they are by definition very highly localised, they do not lead to failure due to bursting or yielding. However, repeated thermal cycling may cause localised damage to occur, leading to crack formation, propagation and eventually leakage in the system.

At the most basic level design against burst is achieved by ensuring that the piping is of sufficient wall thickness to ensure that averaged or "membrane" stress in the wall due to pressure loading is below a specified fraction of the material yield strength. Failure due to gross yielding is then precluded by restricting stresses due to other sustained and occasional loads to certain limits.

Protection against failure due to thermal loading is achieved by ensuring that the proposed design has sufficient flexibility to absorb thermal strains without giving rise to excessive stresses, deformations or connection forces in the system. In code design it must be demonstrated that the system has sufficient flexibility to absorb the design thermal loads safely. To this end a "flexibility analysis" of the proposed design is carried out, in which the thermal displacements, stresses and reaction forces are evaluated and compared with allowable limits specified in the code.

Due to the size and complexity of piping systems, a number of simplified flexibility analysis methods have been developed over the years. In general these simplifications are intended to result in a conservative analysis, (although Carmichael and Edwards have shown that this is not always the case [2.4]).

A more advanced approach to piping assessment is the "design by analysis" procedure allowed for in the ASME Code Section III. Design by analysis requires a detailed elastic stress analysis of the piping system to be performed, the results of which are then compared with specified allowable stress levels etc. Such an analysis is more complex, time consuming and expensive than a flexibility analysis of the same system but gives more accurate and detailed stress, force and displacement results. As computer costs have fallen design by analysis has become a more viable alternative to flexibility analysis for a wide range of design work.

2.1 Flexibility Analysis of Piping Systems.

The size and complexity of piping systems have in the past precluded detailed stress analysis of such systems, and traditionally a number of simplifying assumptions have been used in piping analysis in order to reduce the problem to a manageable size. Most significantly, expansion effects have been investigated by performing a flexibility analysis of the system.

In flexibility analysis the problem is simplified by considering the piping system to be an assembly of straight and curved beams. In the Engineer's theory of bending it is assumed that straight and curved beams deform such that sections initially plane and normal to the centroid of the beam remain so after bending, and that the cross-section of the beam does not deform. Under these conditions a beam subject to a pure bending moment experiences direct strain in the longitudinal fibres only, which, neglecting the Poisson effect, gives rise to uniaxial longitudinal stress in the beam.

In practice beam bending assumptions can be applied in the analysis of straight pipes with reasonable accuracy. However this is not the case in the analysis of pipe bends, bellows expansion joints and branch connections, which exhibit more complex behaviour than simple beams.

When a piping elbow is subjected to bending loading its cross-section tends to flatten or ovalise, violating the "rigid section" beam assumption as shown in Figure 2.1. Deformation of the cross-section reduces the bending stiffness of the elbow and, consequently, the flexibility of a pipe bend can be considerably higher than that calculated for a rigid section beam of similar geometry. The ovalisation also causes a more complex state of stress than that predicted by beam theory. The longitudinal stress is redistributed, and high circumferential bending stress is introduced into the problem. Bellows and branch connections also violate the basic beam bending assumptions. Bellows convolutions deform in a manner similar to bend cross-section ovalisation, and branches exhibit complex local effects at the pipe intersections.

In flexibility analysis the true behaviour of elbows, bellows and branch connections is usually accounted for by the use of correction factors. Correction factors for flexibility and stress intensification are obtained by comparing the response predicted by complex analysis of the components to that given by beam theory and by experimental investigations. In the following section the analysis of piping elbows is considered in some detail.

2.1.1 Elbow Analysis: Flexibility and Stress Intensification Factors.

The first theoretical analysis of a pipe bend *per se* was published by Von Karman in 1911, when he presented an analysis of an elbow subject to a constant in-plane bending moment [2.5]. The previous year Bantlin had shown experimentally

that elbows are more flexible than predicted by beam theory but attributed this behaviour to manufacturing defects [2.6]. Von Karman showed that the enhanced flexibility was due to cross-sectional deformation of the elbow.

The geometry and co-ordinate system of a general elbow is shown in Figure 2.2.

Von Karman applied the minimum potential energy method to a section of elbow under in-plane bending moment loading, simplifying the analysis by invoking the following assumptions:

a) As in beam theory, sections initially plane and normal to the centroid of the elbow remain so upon loading.

b) The applied bending load gives rise to a bending moment which is constant with reference to position around the axis of the elbow. This results in constant ovalisation deformation of the cross-section with respect to axial position and is generally known as the constant bending or constant ovalisation assumption.

c) The circumferential cross-section of the bend is inextensible: ovalisation arises solely from transverse bending of the pipe wall.

d) The elbow geometry is such that the bend radius R is much greater than the pipe mid-surface radius r.

e) the pipe mid-surface radius r is much greater than the wall thickness t.

f) Stresses normal to the shell mid-surface are negligible.

g) Shear strains are negligible.

Assumptions (b) and (g) arise from the pure bending of an axisymmetric body. Assumption (c) gives a coupling condition between the radial and tangential deformation of the elbow cross-section, allowing the deformation of the cross-section to be defined in terms of a single displacement component. Assumption (d) limits the solution to long radius bends but simplifies analysis calculations as the distance of a point on the mid-surface of the pipe wall at angle θ can be assumed to be the bend radius R. That is

$$R - r\cos\theta = R$$

Assumptions (e) and (f) conform with standard thin shell theory assumptions.

In the Von Karman analysis the radial deformation w of the cross-section is represented by the Fourier series:

$$w = \sum_{n=1}^{N} a_{2n} \cos 2n\theta \qquad (1)$$

By invoking the circumferential inextensibility assumption (c) above, the radial displacement v is also defined in terms of the Fourier coefficients a_{2n}. The elbow's potential energy expression may therefore be defined in terms of the coefficients of a single Fourier series and the work done by the applied moment in causing end rotation of the elbow.

Minimising the potential energy of the elbow with respect to the Fourier coefficients and end rotation yields an expression relating the rotation and applied moment. Comparing the rotation values given by this expression to values obtained by an analysis in which ovalisation is neglected, it is found that the flexibility is higher by a constant called the flexibility factor of the elbow. Von Karman found that the flexibility factor and geometry of elbows could be related through a dimensionless bend parameter λ, where:

$$\lambda = \frac{Rt}{r^2}$$

If one term is taken in the ovalisation displacement series the Von Karman analysis yields a bend flexibility factor of:

$$k = \frac{12\lambda^2 + 10}{12\lambda^2 + 1}$$

Taking more terms in the series yields more accurate expressions for flexibility factor. The variation of flexibility factor with bend parameter for one, two and three term series is given in Figure 2.3, which shows that the bend flexibility factor increases as the bend parameter decreases. However, as the bore increases for a given bend radius the long radius assumption (d) above is violated. In practice bend behaviour is governed not only by the bend parameter but also by the radius ratio of the bend.

The stress distribution around the circumference of the bend may also be obtained from the Von Karman analysis. Again, considering a single term solution, the normalised stress distribution is given by the equations:

$$\frac{\sigma_\theta}{\sigma_n} = \nu(k\cos\theta - k_s\cos^3\theta) \pm \frac{3}{2}k_s\lambda\cos 2\theta$$

$$\frac{\sigma_\phi}{\sigma_n} = (k\cos\theta - k_s\cos^3\theta) \pm \frac{3}{2}\nu k_s\lambda\cos 2\theta$$

where the positive sign refers to the outer surface of the pipe, k is the flexibility factor as before, and

$$k_s = \frac{12}{12\lambda^2 + 1}$$

The normalising stress is the stress occurring at the outer fibres of a straight beam of the same nominal dimensions subject to the same bending moment. From bending theory

$$\sigma_n = \frac{My}{I} = \frac{M}{\pi t r^2}$$

The longitudinal and circumferential stress distribution at the outer surface of a general elbow is compared to the beam stress distribution in Figure 2.4. Figure 2.5 shows the stress distribution at the outer and inner surfaces for an elbow of bend parameter $\lambda = 0.5$ and Poisson's ratio $\nu = 0.3$. From these plots it is seen that ovalisation significantly affects the stress levels and distribution in the elbow. In a beam subject to a closing moment maximum longitudinal stress occurs at the extrados (tension) and intrados (compression). In an elbow, ovalisation reduces the stress at these locations and the maximum value occurs in fibres nearer the neutral axis of the bend. The maximum longitudinal stress may be greater than that in a similar curved beam, but it is not as high as the circumferential stress introduced by the ovalisation. The maximum circumferential stress may be up to four times the maximum equivalent beam longitudinal stress.

Most of the elbow analyses published between 1911 and the late 1940's concerned improvement and refinement of the basic Von Karman model. Probably the most significant advance during this period was the extension of the Von Karman

analysis to out-of-plane bending by Vigness in 1943 [2.7]. In a potential energy analysis similar to that of Von Karman, Vigness assumed the radial displacement of the cross-section of a bend subject to an out-of-plane bending moment to be represented by the series:

$$w = \sum_{n=1}^{N} b_{2n} \sin 2n\theta$$

This is equivalent to the in-plane displacement function of Von Karman rotated 45° about the centroid of the section. For a single term solution the expression for out-of-plane flexibility factor is identical to the in-plane expression. Stress intensification is also similar in form to the in-plane case, but with the maxima rotated through 45°.

The first major departure from the Von Karman approach to bend analysis was made by Reissner in 1949, when he defined the governing differential equations for in-plane bending of an elbow based on rotationally symmetric thin shell theory [2.8]. The first pipe bend solution offering flexibility and stress intensification factors based on shell analysis was published by Clark and Reissner later that year [2.9], when the following expressions for flexibility and stress intensification factors of low bend parameter elbows were presented:

$$k = \frac{1.65}{\lambda}$$

$$\frac{\sigma_\theta}{\sigma_n} = \pm \frac{1.892}{\lambda^{\frac{2}{3}}} - \frac{0.480}{\lambda^{\frac{1}{3}}}$$

The Clark and Reissner flexibility factor is plotted against λ in Figure 2.6, and shows good agreement with a Von Karman three term Fourier series solution. The stress intensification factors for the inner and outer surfaces of the elbow are plotted against λ in Figure 2.6.

Although the Clark and Reissner solution offered the first real alternative to the Von Karman approach to elbow analysis, many of Von Karman's original assumptions were retained. Most notably, the elbow was still considered to exhibit constant ovalisation deformation with respect to axial position. However, in the discussion of a 1945 Beskin paper [2.11], Pardue and Vigness presented experimental results which indicated that connection to flanges or straight piping

runs could significantly stiffen the bend. This work was substantiated by Pardue and Vigness in 1951 when they published the results of an experimental investigation into the effect of such end constraints on piping elbows [2.12].

In practice attachment to straight pipes and flanges constrains the ovalisation deformation of the elbow at the ends and the ovalisation varies with respect to axial position. This reduces the bend flexibility, alters the stress distribution and reduces the maximum stress in the elbow.

For some time the effect of end constraints on flexibility was dealt with empirically. The first detailed results of a theoretical analysis including end effects was not published until 1978, when Whatham presented the first in a series of papers on end effects based on a series solution of the governing shell equations [2.13]. Since then a number of papers extending the classic Von Karman analysis to include end constraints have been published by Thomson and Spence, who reviewed end constraints in piping elbows in reference [2.14].

Elbow flexibility and stress intensification factors which include end effects lead to smaller displacements and stresses in a flexibility analysis but few of the design codes actually include such factors. Most are still based on the constant ovalisation approach.

Another significant factor not considered in most codes is the effect of internal pressure on bend behaviour. Internal pressurisation of a pipe bend subject to bending loading tends to force the ovalised cross-section back to its original circular form, effectively stiffening the bend. In 1957 Rodabaugh and George published an elbow analysis in which the effect of pressure was included as a linear work term in a Von Karman type energy analysis [2.15]. However, in [2.16] Crandall and Dahl showed that the relationship between pressure and ovalisation is non-linear, even for small cross-section displacements. Thus the linearised small displacement theory used in analyses such as [2.15] cannot describe the true nature of pressure-bending coupling. The reduction in bend flexibility due to pressure-bending coupling is recognised in some codes but, in general, it is considered too complex for inclusion in flexibility analysis.

The origins and limitations of flexibility and stress intensification factors used in some specific piping codes are discussed below. Although complex piping models including end effects and pressure-bending coupling have been presented in the literature, most codes are based on constant bending analysis of elbows.

2.1.2 Flexibility and Stress Intensification Factors in Piping Codes.

The flexibility and stress intensification factors used in the codes are derived from several sources.

In BS806 a single curve is presented for both in-plane and out-of-plane flexibility factor. The curve is the average of in-plane values from Turner and Ford [2.17] and out-of-plane values from Smith [2.18]. Turner and Ford reviewed contemporary analytical methods for pipe bends in 1957 and provided a detailed numerical analysis of the problem based on shell theory. Smith obtained flexibility factors for out-of-plane bending by an extension of the Vigness analysis. The BS806 stress intensification factors are also drawn from these sources.

The ANSI/ASME B31.1 flexibility factors are based on the Clark and Reissner shell solution [2.9]; however, the stress intensification factors are derived from fatigue tests performed by Markl in 1952 [2.19]. These tests defined so-called "i-factors" which are approximately half the value of the stress intensification factors given in BS 806. The stresses obtained using these factors are not true elastic stresses, but an indication of the fatigue strength of the elbow. The reasons for using these factors are somewhat obscure, but they are retained in the code as they have been applied successfully in the past. Separate i-factors are specified for in-plane and out-of-plane loading, and also for flanged bends.

ASME Section III for Nuclear Vessels defines three classes of piping. Subsection NB gives the rules for Class 1 components including piping within the reactor coolant pressure boundary. Subsection NC concerns Class 2 components important to safety and designed for emergency core cooling etc. Subsection ND concerns Class 3 components found in the cooling water and auxiliary feedwater systems.

Classes 2 and 3 have design rules based on the ANSI B31.1 approach, whereas two approaches are permitted for Class 1 piping. One method is Design by Analysis, as discussed below, and the other is a conventional flexibility analysis based on beam theory. As in the case of ANSI B31.1, flexibility factors based on Clark and Reissner are available; however, alternative factors taking into account the effect of end constraints and internal pressure may be used if preferred. These alternative flexibility factors are based on the work of Rodabaugh *et al* [2.20].

Stress intensification for Class 1 piping is accounted for by the use of three types of stress intensification indices, each related to a particular failure mechanism: gross yield, shakedown and localised peak stresses. These stress indices are essentially different to the i-factors used in ANSI B31.1, being based on limit stresses as opposed to fatigue results.

2.1.3 Computer-Based Flexibility Analysis: The Matrix Displacement Method.

Flexibility analysis is most commonly performed using piping analysis computer programs based on code assessment procedure. Most of these programs model the piping system as a three dimensional framework of simple beams, using standard skeletal structure analysis techniques. In commercial software the most popular analysis method is the matrix displacement method.

In the matrix displacement method a mathematical model of a complex piping system is built up by assembling a number of straight and curved beam models of system components called "members" or "elements". Here the term "element" is used for consistency with the finite element method which is discussed below.

The behaviour of an element is fully defined in terms of a finite number of degrees of freedom by the element stiffness equation, the form of which is:

$$\{F\} = [K]\{d\}$$

where {d} is a vector of degrees of freedom, {F} a corresponding vector of forces, and [K] the stiffness matrix of the element. The element degrees of freedom are generally translational and rotational displacements at designated "nodal points" or simply "nodes" located at the ends of the element.

The beam element stiffness matrices are defined by classical beam theory or by inversion of the flexibility equation obtained by methods such as Castigliano's theorem. In the case of elbows, bellows and branch connections, the beam element stiffness matrices must be modified to represent the actual behaviour of these components. To this end flexibility factors are introduced into the analysis at element level to increase the flexibility of the elements. The flexibility factors are automatically evaluated by the programs according to the element geometry and the specified design code.

As the element stiffness matrices are generated, thermal loads arising from the design temperature variation are evaluated in the form of an element load vector. The element matrices and load vectors are then assembled to form the system or global stiffness equation, by enforcing equilibrium and compatibility at nodes common to adjacent elements. The system anchors, connections to plant, supports and so on are then applied to the global displacement vector as displacement boundary conditions. The global stiffness equation is thus generated in the form:

$$\{F_g\} = [K_g]\{d_g\} \qquad (2.1)$$

which is then solved for the unknown global displacements. Symbolically the solution may be written:

$$\{d_g\} = [K_g]^{-1}\{F_g\} \qquad (2.2)$$

Once the displacement solution is complete, the vector of element reaction forces may be evaluated by back-substituting {d} into (2.1). Stress calculations are then performed in which the element nodal stresses are evaluated from nodal forces and moments according to bending theory.

Flexibility analysis programs based on the matrix displacement method may produce large and seemingly detailed amounts of information, generally in a form required by the piping codes. However, it must be emphasized that these results are obtained by a simplified analysis and are primarily intended to be conservative in nature.

2.2 Finite Element Analysis of Piping Systems.

As computing costs and limitations have fallen in recent years, finite element analysis has become more popular in the analysis of piping systems. Finite element analysis is a powerful numerical analysis technique which, given adequate computing facilities, may be used to obtain a detailed elastic stress analysis of piping systems. A significant advantage of finite element analysis over flexibility analysis is that the need for flexibility and stress correction factors is removed, as the piping model more fully represents the true behaviour of the piping components.

The stresses obtained by finite element analysis are "real" pipe stresses, as opposed to the beam stresses evaluated in flexibility analysis. However, the stress results given by the finite element method do not conform to the traditional piping stresses postulated in the design codes, and cannot be applied directly to such codes. In order to make use of such analysis, it is necessary to specify an alternative design philosophy to the traditional flexibility analysis approach. Such an alternative has already been included in some of the more advanced piping codes in the form of the "design by analysis" procedure. Design by analysis originated in ASME Section III and Section VIII Division 2, and has subsequently been adapted by other standards bodies including the BSI in BS 5500 Appendix A [2.21]. In the design by analysis approach, the detailed elastic stresses given by finite element analysis are processed and categorised in a form similar to traditional pipe stresses. These stresses are then compared with allowable limits specified in the codes.

In the remainder of this chapter the finite element method is discussed first in general, and then with respect to piping analysis applications.

2.2.1 The Finite Element Method.

The finite element method is a powerful numerical analysis technique which is used in many areas of science and engineering for the solution of field problems. The method is defined in a large body of literature, for example references [2.22,2.23,2.24,2.25,2.26], but in terms of structural analysis it may be viewed as a generalisation of standard structural analysis techniques.

The object of finite element analysis is to produce and solve a mathematical model of a real structure in terms of a finite number of degrees of freedom. This is done by considering the complex structure to be an assembly of simpler components or "finite elements", for which general mechanical models can be obtained in terms of a limited number of degrees of freedom. The elements may represent real components, such as beams and spars in a frame structure, or they may be conceptional sub-regions of a continuum, such as a sub-area of plate or shell, or a sub-volume of a three-dimensional solid.

In the finite element method the behaviour of an element of structure is defined by a system of simultaneous linear algebraic equations. This is done by applying the well known Rayleigh-Ritz and minimum potential energy methods to the

element. Thus the behaviour of the element is defined in terms of displacement and displacement derivative degrees of freedom located at specific element positions called nodes. Nodes are usually located on the boundary of the element, although some elements include nodes at internal locations.

As in the matrix displacement method, complex structures are modelled by assembling a number of elements connected together at common nodes so as to satisfy equilibrium and compatibility requirements, resulting in a global stiffness equation similar in form to that of the matrix displacement method discussed above:

$$\{F_g\} = [K_g]\{d_g\} \qquad (2.1)$$

where $\{F_g\}$ is a vector of applied forces, $\{d_g\}$ a corresponding vector of structural degrees of freedom (displacements and displacement derivatives) and $[K_g]$ the global stiffness matrix of the structure.

The form of element stiffness equations is also common to both methods; that is

$$\{F\} = [K]\{d\} \qquad (2.2)$$

where {F} is a vector of forces applied to the element, {d} the corresponding degree of freedom vector and [K] the element stiffness matrix. However, the methods differ significantly in how the stiffness matrix of the elements are obtained. In the displacement-based finite element method, equation (2.2) is obtained approximately by prescribing the deformation of the element by means of interpolation functions and applying the principle of minimum potential energy and the Rayleigh-Ritz method.

Potential Energy.

The potential energy of an general finite element is given by the expression:

$$\pi = U - W$$

where U is the strain energy of the element and W the work done by forces acting on the element. From elasticity theory, the strain energy of an element of volume V is given by:

$$U = \frac{1}{2}\int_V \{\sigma\}^T \{\epsilon\} dV \tag{2.3}$$

where $\{\sigma\}$ and $\{\epsilon\}$ are vectors of stress and strain respectively at a point within the element domain; that is within or on the boundaries of the element.

Forces acting on the element arise from three sources: body forces {R}, surface tractions {T} and point loads {p}. These forces give rise to the work term W in the potential energy expression:

$$W = \int_V \{u\}^T \{R\} dV + \int_S \{u\}^T \{T\} dS + \{u\}^T \{p\} \tag{2.4}$$

where V is the element volume, S the surface area and {u} the vector of displacements at a point in the element.

The Rayleigh-Ritz Procedure: Minimising Potential Energy.

Equations (2.3) and (2.4) represent the strain energy and work contributions to the potential energy of a continuum. Thus:

$$\pi = U - W = \frac{1}{2}\int_V \{\sigma\}^T \{\epsilon\} dV - \int_V \{u\}^T \{R\} dV - \int_S \{u\}^T \{T\} dS - \{u\}\{p\} \tag{2.5}$$

As the strains (and hence stresses) at a point in the element are defined in terms of partial derivatives of displacement, equation (2.5) represents the potential energy of the element in terms of an infinite number of degrees of freedom. In the finite element method a Rayleigh-Ritz procedure is invoked to reduce the problem to one with a finite number of freedoms.

The strain-displacement relationships for the element are defined by the particular structural theory used in the formulation: for example, three-dimensional elasticity, beam theory, shell theory etc. In general:

$$\{\epsilon\} = \Delta(\{u\}) \tag{2.6}$$

where Δ is a differential operator. The first step in the procedure is to assume a displacement field which describes the possible deformation patterns of the element in terms of a finite number of degrees of freedom. Deformation within the element domain is defined by interpolating displacements and displacement derivatives at discrete nodal points within the element. The interpolation equation is of the general form:

$$\{u\} = [N]\{d\} \tag{2.7}$$

where [N] is a matrix of interpolation or shape functions relating the vector of displacements at a point, {u}, to the selected nodal degrees of freedom of the element, {d}.

[N] is required to meet internal displacement continuity and compatibility requirements, whilst maintaining inter-element continuity as far as possible. [N] may be obtained directly from well-known interpolation functions such as Lagrangian polynomials or by defining a displacement function explicitly. In the latter case, a general displacement function is of the form

$$\{u\} = [f(x,y,z)]\{a\} \tag{2.8}$$

where [f(x,y,z)] is a matrix of functions of position and {a} a vector of unknown constants. By applying the element boundary conditions to (2.8), {a} can be expressed in terms of {d}, the nodal degrees of freedom. For example, typical boundary conditions at node *i*, co-ordinates (x_i,y_i,z_i), may be $u = u_i$, $v = v_i$ etc. Thus:

$$\{d\} = [A]\{a\}$$

where [A] is a matrix of constants. This equation may then be inverted to give:

$$\{a\} = [A]^{-1}\{d\}$$

Substituting for {a} into (2.8):

$$\{u\} = [f(x,y,z)][A]^{-1}\{d\}$$

Thus from (2.7) the shape function matrix is:

$$[N] = [f(x,y,z)][A]^{-1}$$

In the finite element procedure the strain vector is obtained in terms of the element degrees of freedom by substituting (2.7) into (2.6):

$$\{\epsilon\} = \Delta([N]\{d\})$$

As {d} is a vector of constants, this may be written

$$\{\epsilon\} = \Delta([N])\{d\}$$

or

$$\{\epsilon\} = [B]\{d\} \qquad \text{where} \quad [B] = \Delta([N]) \tag{2.9}$$

[B] is called the strain-displacement matrix of the element and relates strain at a point in the element to the nodal degrees of freedom.

The stresses at a point in the element are related to strains by the appropriate constitutive relationship for the element; for example, plane stress, plane strain etc. Generally:

$$\{\sigma\} = [D]\{\epsilon\} \tag{2.10}$$

where [D] is the element constitutive (or elasticity) matrix.

Equations (2.9) and (2.10) can be substituted into equation (2.3) to define the element strain energy in terms of the element degrees of freedom:

$$U = \frac{1}{2}\int_v \{\epsilon\}^T[D]\{\epsilon\}dV = \frac{1}{2}\int_v \{d\}^T[B]^T[D][B]\{d\}dV \tag{2.11}$$

Also, substituting (2.9) into (2.4) defines the work in terms of the element degrees of freedom:

$$W = \int_v \{d\}^T[N]\{R\}dV + \int_s \{d\}^T[N]\{T\}dS + \{d\}\{P\} \tag{2.12}$$

Note that the vector {P} is not the same vector as {p} in the continuum work equation (2.4); only point forces {P} applied at the nodes and corresponding to appropriate degrees of freedom are now admissible. Thus from (2.11) and (2.12) the total potential energy of the element is given by:

$$\pi = \frac{1}{2}\int_V \{d\}^T [B]^T [D][B]\{d\} dV - \int_V \{d\}^T [N]\{R\} dV - \int_S \{d\}^T [N]\{T\} dS - \{d\}\{P\} \tag{2.13}$$

The principle of minimum potential energy states that the total potential energy of a structure meeting all boundary conditions is at a minimum when the structure is in static equilibrium. When a structure is in static equilibrium its governing differential equations are satisfied; thus, by defining the minimum potential energy state of the structure, the governing equations are satisfied indirectly.

The approximate minimum potential energy state of the element is defined by systematically minimising (2.13) with respect to each degree of freedom in {d}. For a total of *n* degrees of freedom this procedure results in a system of *n* linear algebraic equations. The minimisation is written in matrix form:

$$\frac{\partial \pi}{\partial \{d\}} = \int_V [B]^T [D][B]\{d\} dV - \int_V [N]\{R\} dV - \int_S [N]\{T\} dS - \{P\} = \{0\}$$

or

$$\int_V [B]^T [D][B] dV \ \{d\} = \int_V [N]\{R\} dV + \int_S [N]\{T\} dS + \{P\} \tag{2.14}$$

The solution is approximate in nature due to the finite number of degrees of freedom chosen to describe the element deformation in the interpolation equation (2.7). In effect the shape functions constrain the mathematical model of the structure to deform in a particular way, rather than to respond with the full freedom of the real structure.

Comparing (2.14) with the general element stiffness equation (2.2), it is found by inspection that:

$$[k] = \int_V [B]^T [D][B] dV \tag{2.15}$$

and

$$[f] = \int_V [N]\{R\}dV + \int_S [N]\{T\}dS + \{P\} \qquad (2.16)$$

Nodeless Degrees of Freedom, Static Condensation and Recovery.

In the formulation procedure described above, element behaviour is fully defined in terms of degrees of freedom located at specific nodal points in the element domain. It is, however, possible to formulate elements with degrees of freedom which are not associated specifically with particular nodes. Such "nodeless" degrees of freedom have been used to define the ovalisation deformation of elbow elements, and are discussed in Chapter 3.

As "nodeless" degrees of freedom are unique to their associated element, continuity is not required between elements and nodeless degrees of freedom can be "statically condensed" from the element matrices before assembly. The resulting element is, therefore, fully defined by its nodal degrees of freedom.

Static condensation is a general finite element technique which may be applied, if desired, to remove any degree of freedom not required for inter-element compatibility. In order to apply the process of static condensation the element stiffness matrix is required in the following partitioned form:

$$\begin{Bmatrix} \{F_b\} \\ \{F_o\} \end{Bmatrix} = \begin{bmatrix} [K_{bb}] & [K_{bo}] \\ [K_{ob}] & [K_{oo}] \end{bmatrix} \begin{Bmatrix} \{d_b\} \\ \{d_o\} \end{Bmatrix} \qquad (2.17)$$

where $\{d_b\}$ and $\{d_o\}$ are vectors of nodal and nodeless degrees of freedom, and $[F_b]$ and $[F_o]$ are corresponding force vectors. The stiffness sub-matrices are: $[K_{bb}]$, relating nodal forces and displacements; $[K_{oo}]$, relating nodeless degrees of freedom and forces; and $[K_{bo}]$ and $[K_{ob}]$, which are coupling matrices.

The stiffness sub-matrices may be evaluated by partitioning the [B] matrix as follows:

$$[B] = [[B_{bb}][B_{oo}]]$$

where the sub-matrix $[B_{bb}]$ relates strain to nodal displacements, and $[B_{oo}]$ relates strain to the nodeless degrees of freedom.

From (2.15), the element stiffness matrix is given by:

$$[k] = \int_V [B]^T [D][B] dV$$

Performing the above integration using the partitioned form of [B] gives

$$\begin{Bmatrix} \{F_b\} \\ \{F_o\} \end{Bmatrix} = \begin{bmatrix} [K_{bb}] & [K_{bo}] \\ [K_{ob}] & [K_{oo}] \end{bmatrix} \begin{Bmatrix} \{d_b\} \\ \{d_o\} \end{Bmatrix}$$

where $[K_{bb}]$ is the beam bending stiffness matrix,

$$[K_{bb}] = \int_V [B_b]^T [D][B_b] dV \tag{2.18}$$

$[K_{oo}]$ the ovalisation stiffness matrix,

$$[K_{oo}] = \int_V [B_o]^T [D][B_o] dV \tag{2.19}$$

and $[K_{bo}]$ and $[K_{ob}]$ bending ovalisation coupling matrices such that

$$[K_{bo}] = [K_{ob}]^T = \int_V [B_b]^T [D][B_o] dV \tag{2.20}$$

The element stiffness matrix of (2.17) is of order (m+n)x(m+n), where m and n are the number of nodal and nodeless degrees of freedom respectively. This "full" stiffness matrix may be reduced to an mxm matrix by applying the process of static condensation. From (2.17), two sub-matrix equations may be written:

$$\{F_b\} = [K_{bb}]\{d_b\} + [K_{bo}]\{d_o\} \tag{2.21}$$

$$\{F_o\} = [K_{ob}]\{d_b\} + [K_{oo}]\{d_o\} \tag{2.22}$$

From (2.22),

$$\{d_o\} = -[K_{oo}]^{-1}\{[K_{ob}]\{d_b\} - \{F_o\}\} \tag{2.23}$$

Which upon substitution into (2.21) gives

$$\{F_b\} = [K_{bb}]\{d_b\} + [K_{bo}]\{-[K_{oo}]^{-1}\{[K_{ob}]\{d_b\} - \{F_o\}\}\}$$

That is

$$\{F_b\} = [[K_{bb}]-[K_{ob}]^T[K_{oo}]^{-1}[K_{ob}]]\{d_b\}+[K_{ob}]^T[K_{oo}]^{-1}\{F_o\}$$

This may be written

$$\{F_R\} = [K_R]\{d_b\} \tag{2.24}$$

where $[K_R]$ is the reduced element stiffness matrix given by

$$[K_R] = [[K_{bb}]-[K_{ob}]^T[K_{oo}]^{-1}[K_{ob}]] \tag{2.25}$$

and $\{F_R\}$ is the reduced load vector:

$$\{F_R\} = \{ \{F_b\}-[K_{ob}]^T[K_{oo}]^{-1}\{F_o\} \} \tag{2.26}$$

After the assembled structural equation has been solved, the nodeless degrees of freedom $\{d_O\}$ may be required for stress evaluation in the reduced element. If so, they can be "recovered" from the nodal degrees of freedom through equation (2.23).

Element Co-ordinate Transformations.

Finite elements may be formulated in terms of a local element co-ordinate system and transformed into a common "global" co-ordinate system before assembly by use of a rotation or transformation matrix.

Transformation matrices are discussed in some detail in Appendix 2, but essentially a transformation matrix consists of direction cosines relating the axes of the local and global co-ordinate systems. Each type of finite element - spar, beam, plate etc. - has its own form of matrix, depending on the geometry and degrees of freedom of the element. However, in general, the transformation matrix is defined implicitly in the expression:

$$\{\text{local vector}\}=[TR]\{\text{global vector}\}$$

or conversely

$$\{\text{global vector}\}=[TR]^{-1}\{\text{local vector}\}$$

where [TR] is the transformation matrix. From this definition it can be shown that the element matrices are transformed from the local to global co-ordinate system according to the general relationship:

$$[\text{global matrix}] = [TR]^{-1}[\text{element matrix}][TR]$$

The transformation matrix has the very useful property of orthogonality; that is, its inverse is equivalent to its transpose. Thus:

$$[TR]^T = [TR]^{-1}$$

Therefore the local to global transformation equations for the vectors and matrices of a general finite element may be written:

$$\{d_g\} = [TR]^T \{d_l\}$$

$$\{F_g\} = [TR]^T \{F_l\}$$

$$[K_g] = [TR]^T [K_l][TR]$$

$$[M_g] = [TR]^T [M_l][TR]$$

Where d = displacement
F = Force
K = Stiffness
M = Mass

2.2.2 Finite Element Modelling of Piping Systems.

In practice the finite element method may be applied to piping systems in two ways: standard shell element analysis, in which the piping system is modelled using general purpose shell finite elements, or piping element analysis, in which the system is modelled by specific piping finite elements.

A piping system may be discretized using standard thin shell elements as shown for a simple example in Figure 2.7. Provided an adequate finite element mesh is used this is the most accurate piping analysis method currently possible. Detailed displacement and stress results are obtained throughout the system, and pipe bends, bellows and branches can be modelled with high accuracy. At present the piping design codes do not require this level of detail and, in addition, there are several practical disadvantages in applying such an approach to general piping analysis:

i) Finite element modelling even using state of the art pre-processors is time-consuming and hence expensive.

ii) Computing costs may be prohibitive.

iii) Finite element models of complex systems may be too large to run on the available computer.

iv) Post-processing of results is time consuming and hence expensive.

Problem (iii) above may be circumvented by adopting a substructuring approach to the problem. In substructuring, large and complex systems can be analysed on fairly modest machines by dividing the single structure up into a number of smaller component structures or substructures. Models of the individual substructures are created in the usual way, but only a partial solution of the model is carried out to evaluate the forces and stiffness at nodes lying on the boundaries at which the substructure will be joined to others. This procedure is repeated for all the substructures and they are then assembled to form a model of the complete structure. The size of the substructured model may be a fraction of the size of a single model with similar finite element meshing; thus very large jobs can be run.

The main disadvantages of the substructuring approach are:

a) The total computing costs are greater than for a single model analysis of the system.

b) Additional pre-processing and post-processing is required.

Therefore, although the technique may be used to overcome disadvantage (iii) above, (i), (ii) and (iv) are worsened. In general, piping systems are simply too big to be analysed in this way, and an alternative approach is required. To this end, most finite element analyses of piping systems make use of special elements formulated specifically for piping analysis. The main requirements of such elements is that they simplify finite element modelling and post-processing of results and reduce model size in comparison with general shell element models.

In the discussion of flexibility analysis presented above, it was noted that straight pipes can be modelled as beam elements with acceptable accuracy. However elbows, bellows and branches cannot be modelled adequately using simple beams, due to their more complex deformation behaviour. In flexibility analysis this was allowed for by the use of correction factors in the element stiffness and stress calculations. This situation persists for bellows and branch connections in all current commercial finite element systems offering piping elements; however,

since the early 1970s a number of finite elements formulated specifically for modelling pipe bends have been presented in the literature. Piping elbow elements were originally proposed for analysis of material non-linearity, such as plasticity and creep, which cannot be adequately represented by simplified beam models. The elements account for ovalisation effects by including shell deformation modes in the element formulation. To date, two basic types of elbow element formulations have been proposed: beam-shell elements and shell-ring elements.

In beam-shell elements the ovalisation behaviour of the elbow is modelled by superposing shell type ovalisation onto a curved beam element formulation, thus removing the need for stiffness and stress correction factors. Complete piping systems are modelled by using straight beams to represent the straight piping runs and beam-shell elbow elements for bends, as shown in Figure 2.8a. In the case of large bore small λ bends it has been shown that some ovalisation may occur in straights connected to bends, and in some cases curved elbow elements can degenerate to straight pipe elements to include this behaviour in the finite element model.

Shell-ring elements model the behaviour of bends wholly in terms of shell theory. As the name suggests, the element has a ring type geometry and, depending on the formulation, may be used either to model entire branchless systems as shown in Figure 2.8b, or to model elbows only, with straight beams used to represent straight piping runs.

Although several commercial finite element programs such as MARC, ABAQUS and ADINA include elbow elements in their element libraries, the use of such elements has not been universally accepted for general elastic analysis of piping systems. Most of the elbow elements presented in the literature were intended for detailed non-linear analysis of piping systems, and in some respects may be regarded as over-complex and computationally expensive for analysis of general piping systems. Also, programs such as ABAQUS are not the most "user-friendly" on the market, and in the main are used only by non-linear analysis specialists. Commercial flexibility analysis programs have specialist pre-processors which allow the system geometry, supports, material properties and loading to be specified in a manner familiar to piping design engineers.

Taking a broad view, a successful elbow element must offer acceptable accuracy at acceptable computing costs and be incorporated in a user-friendly, piping analysis orientated, finite element system. The object of this thesis is to study simple elbow elements and propose suitable formulations for general elastic analysis. The first step in the study is to review the elbow element formulations presented in the literature, and this is done in the next Chapter.

2.3 References

[2.1] British Standards Institution
BS 806, "Ferrous Pipes and Piping Installations for and in Connection with Land Boilers," (1986).

[2.2] American National Standards Institution
ANSI B31.1, "Power Piping," (1989).

[2.3] American Society of Mechanical Engineers
ASME "Boiler and Pressure Vessel Code", (1989).

[2.4] G D T CARMICHAEL and G EDWARDS
"Some Observations on the Analysis of High Temperature Steam Piping Systems," Proc. Inst. Mech. Eng., Vol 193, (1979).

[2.5] T Von KARMAN
"Uber die Formanderung Dunnwandiger Rohre, Insbesonders Federnder Ausleichrohre," Z., V.D.I., Vol 55, p1889-1895 (1911).

[2.6] A BANTLIN
"Formanderung und Beanspruchung Federnder Ausgleichrohre," Z., V.D.I., Vol 54, p43-49 (1910).

[2.7] I VIGNESS
"Elastic Properties of Curved Tubes," Trans. ASME, Vol 65, p105-120 (1943).

[2.8] E REISSNER
"On Bending of Curved Thin-Walled Tubes," Proc. Nat. Acad. Sci. US, Vol 35, p204-209 (1949).

[2.9] R A CLARK and E REISSNER
"Bending of Curved Tubes," Proc. Nat. Acad. Sci. US, Vol 35, p204-209 (1949).

[2.10] E C RODABAUGH and H H GEORGE
"Effect of Internal Pressure on Flexibility and Stress Intensification Factors of Curved Pipies or Welding Elbows," Trans. ASME, Vol 79, p939 (1975).

[2.11] L BESKIN
"Bending of Curved Thin Tubes," Trans. ASME, Vol 65, p105-120 (1943).

[2.12] T E PARDUE and I VIGNESS
"Properties of Thin-Walled Curved Tubes of Short-Bend Radius," Trans. ASME, Vol. 73, p77-87 (1951).

[2.13] J F WHATHAM
"In-Plane Bending of Flanged Pipe Elbows," Proc. Metal Struc. Conf, Perth, Australia (1978).

[2.14] G THOMSON and J SPENCE
"A survey of End Constraints in Piping Elbows," Proc. 3rd Simposio Brasileiro Sobre Tubulacoes e Vasos de Pressao, p33-55 (1984).

[2.15] E C Rodabaugh and H H George
"Effects of Internal Pressure on Flexibility and Stress Intensification Factors of Curved Pipe or Welding Elbows," J. Appl. Mech., Vol 79, pp 939-948, (1956).

[2.16] S H Crandall and N C Dahl
"The Influence of Pressure on the Bending of Curved Tubes," Proc 9th Int. Conf. App. Mech., Vol 6, pp331-343, Brussels, (1957).

[2.17] C E TURNER and H FORD
"Examination of the Theories for Calculating the Stresses in Pipe Bends Subject to In-Plane Bending," Proc. IMechE, Vol 171, p513 (1957).

[2.18] R T SMITH
"Theoretical Analysis of the Stresses in Pipe Bends Subjected to Out-of-Plane Bending," J. Mech. Eng. Sci., Vol 9, p115 (1967).

[2.19] A R C MARKL
"Fatigue Tests of Piping Components," Trans. ASME, Vol. 74, p287, (1952).

[2.20] E C RODABAUGH, S E MOORE and S K ISKANDER
"End Effects on Elbows Subject to Moment Loadings," ORNL Rep. 2913-0, 1977. (ASME Special Publ. No. H00213,1982).

[2.21] BRITISH STANDARDS INSTITUTION
BS 5500 Pressure Vessel Code Appendix A, (1988).

[2.22] R W CLOUGH
"The Finite Element Method in Structural Analysis," Stress Analysis, Ed. O C Zienkiewicz and G S Holister, John Wiley & Sons Ltd., London, (1965).

[2.23] R D COOK
Concepts and Applications of Finite Element Analysis, 2nd. Ed., John Wiley & Sons Ltd., New York, (1981).

[2.24] O C ZIENKIEWICZ AND R L TAYLOR
The Finite Element Method, 4th. Ed., McGraw-Hill, London,(1977).

[2.25] D J DAWE
Matrix and Finite Element Displacement Analysis of Structures, Clarendon Press, Oxford, (1984).

[2.26] B IRONS and S AHMAD
Techniques of Finite Elements, Ellis Horwood, Chichester, (1980).

[2.27] P R SMITH and T J VAN LAAN
Piping and Pipe Support Systems Design and Engineering, McGraw-Hill Book Company, New York, (1987).

[2.28] S KANNAPPAN
Introduction to Pipe Stress Analysis, Wiley-Interscience, John Wiley & Sons, (1986).

[2.29] Developments in Pressure Vessel Technology-5
Pressure Vessel Codes and Standards, Ed. R W NICHOLS, Elsevier Applied Science Publishers Ltd, (1987).

[2.30] A R C MARKL
"Piping-Flexibility Analysis," Trans ASME, (1955).

[2.31] J T BOYLE
Design of Piping Systems Course Notes, University of Strathclyde, (1988).

[2.32] R KITCHING
"Smooth and Mitred Pipe Bends," Chap. 7, The Stress Analysis of Pressure Vessels and Pressure Vessel Components" Ed. SS Gill, Pergamon, (1970).

[2.33] RAE K
"A Study on Pipe Bends," Engineers PhD Thesis, University of Strathclyde, (1984).

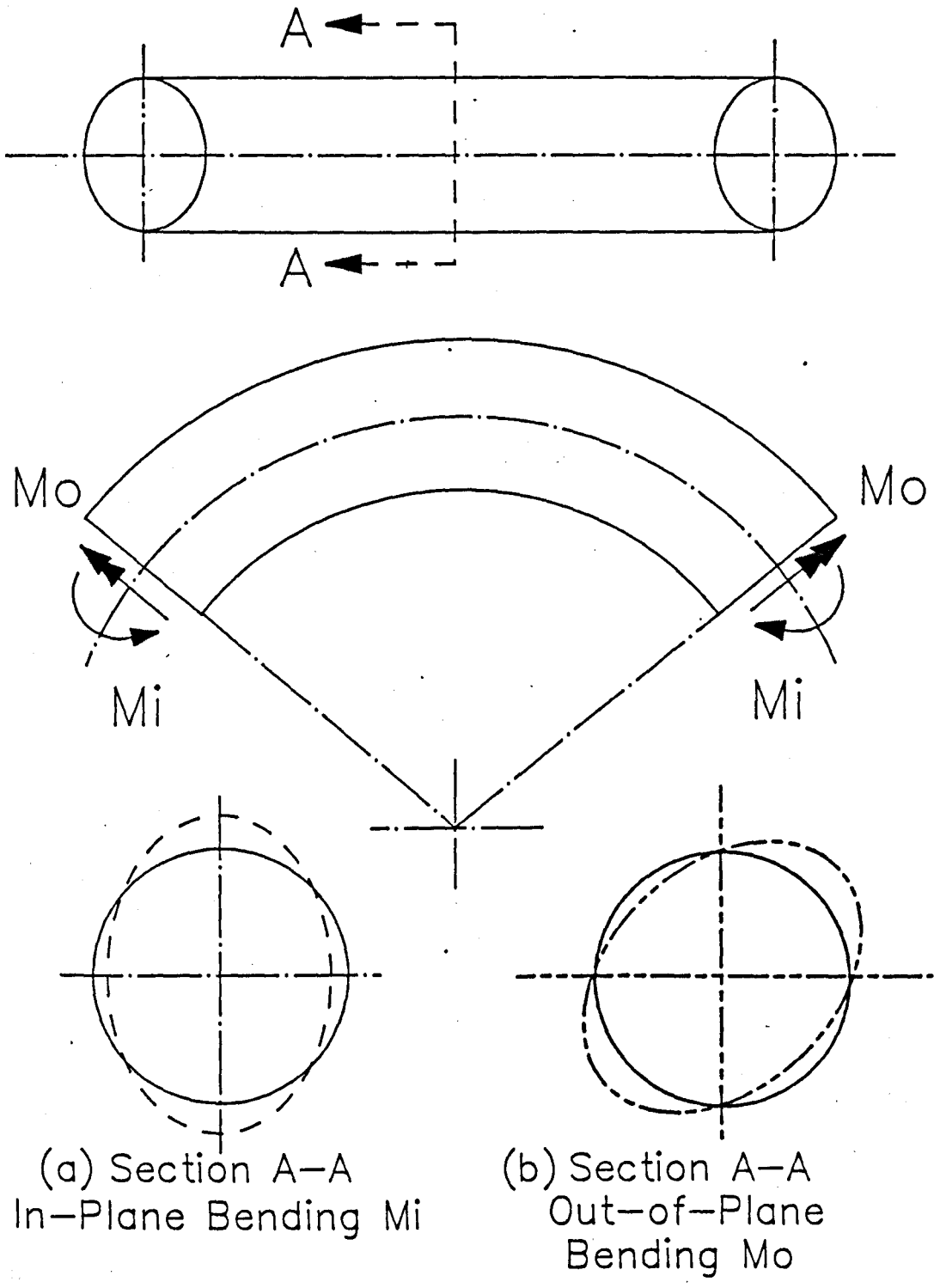


Figure 2.1 Ovalisation of a piping elbow cross-section due to (a) in-plane bending moment Mi and (b) out-of-plane bending moment Mo.

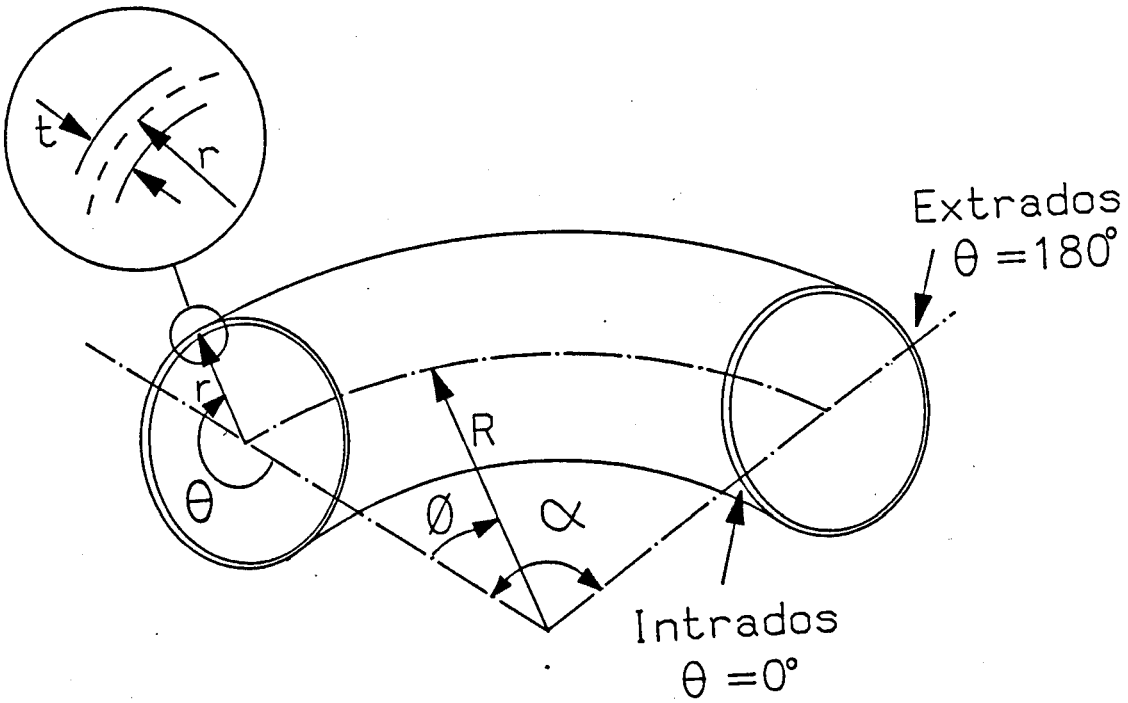


Figure 2.2 Piping elbow geometry and coordinate system.

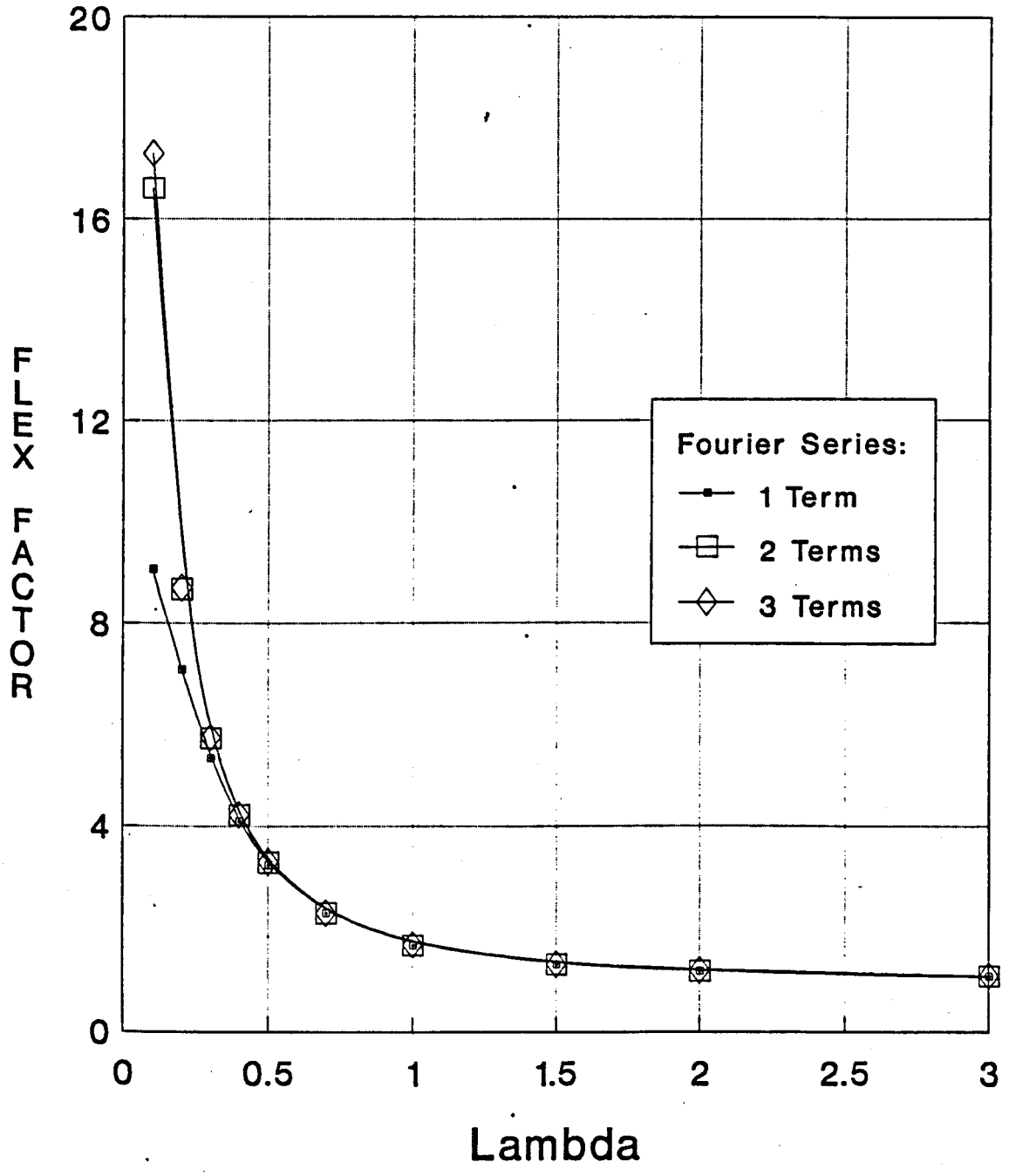


Figure 2.3 Von Karman flexibility factor versus bend parameter λ.

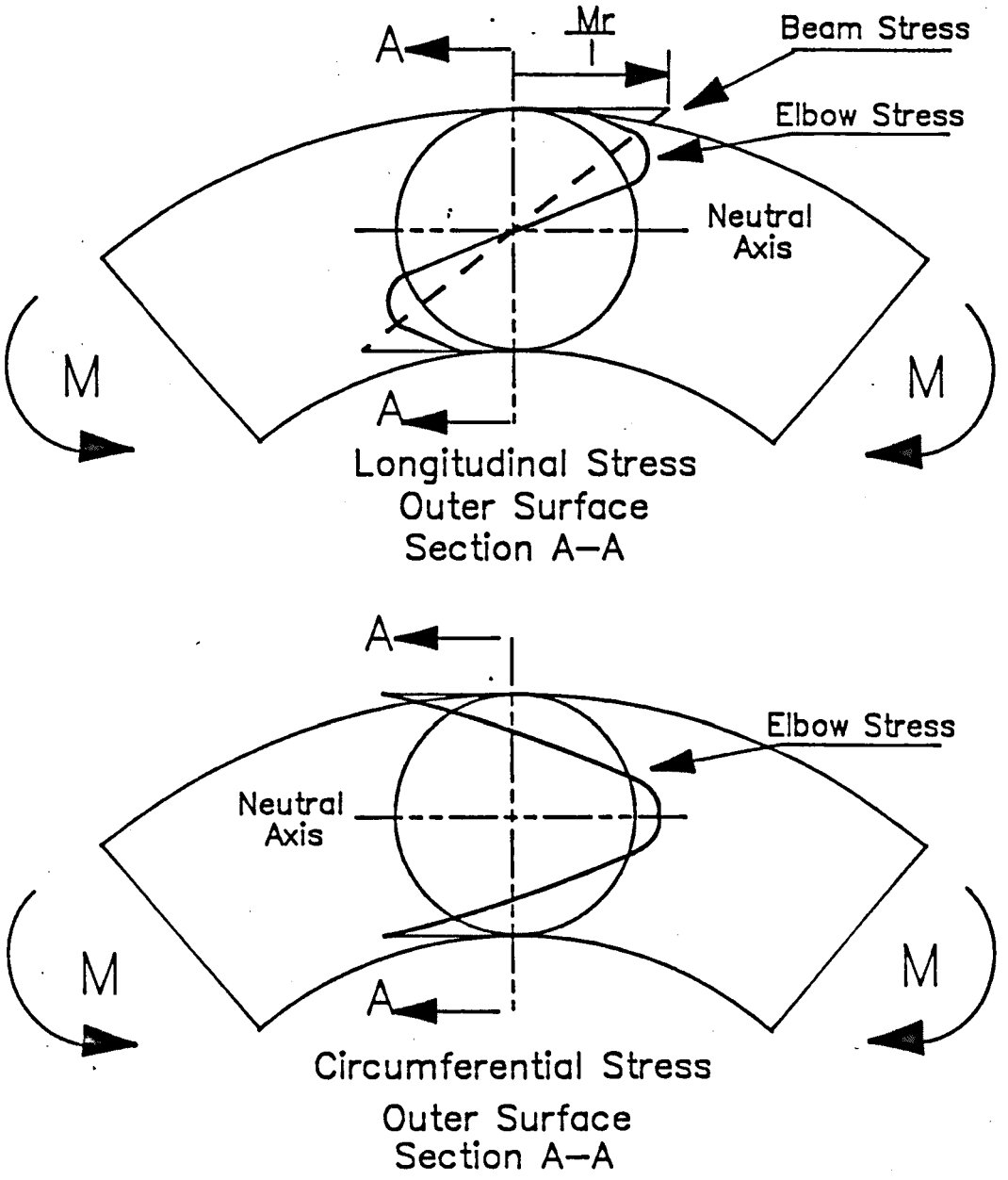
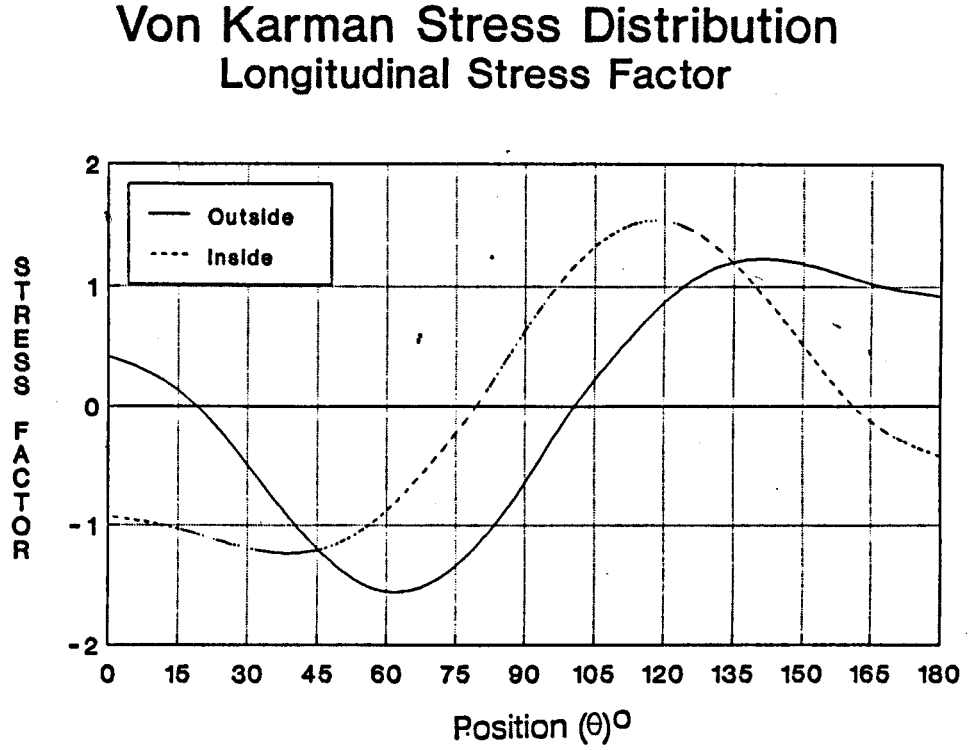


Figure 2.4 General elbow stress distribution due to in-plane bending.



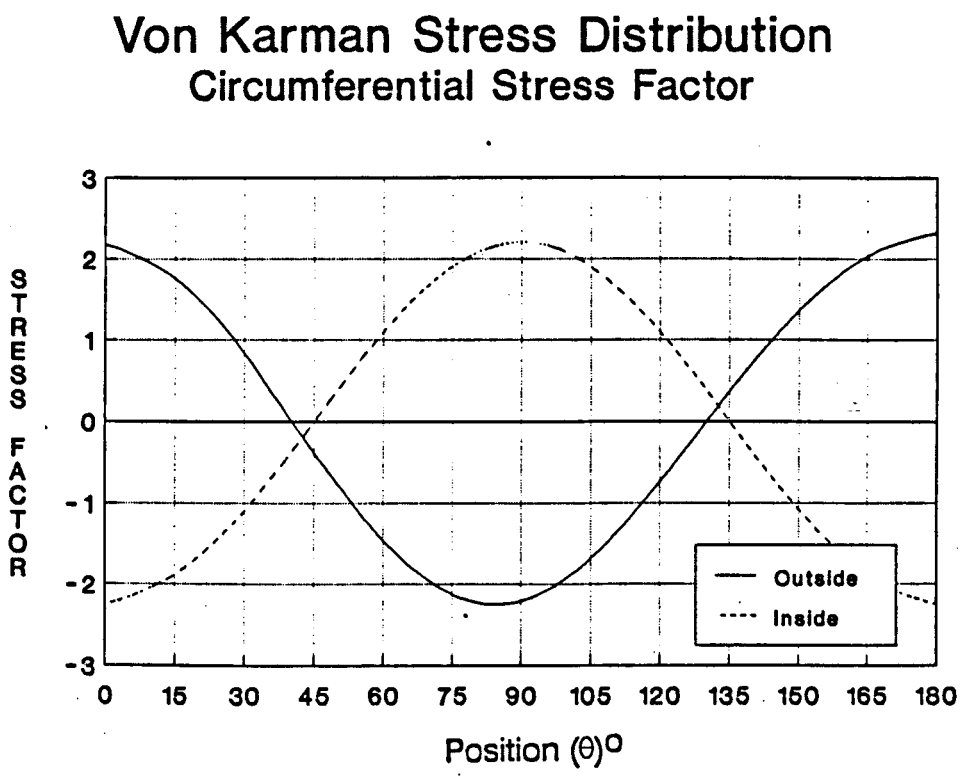
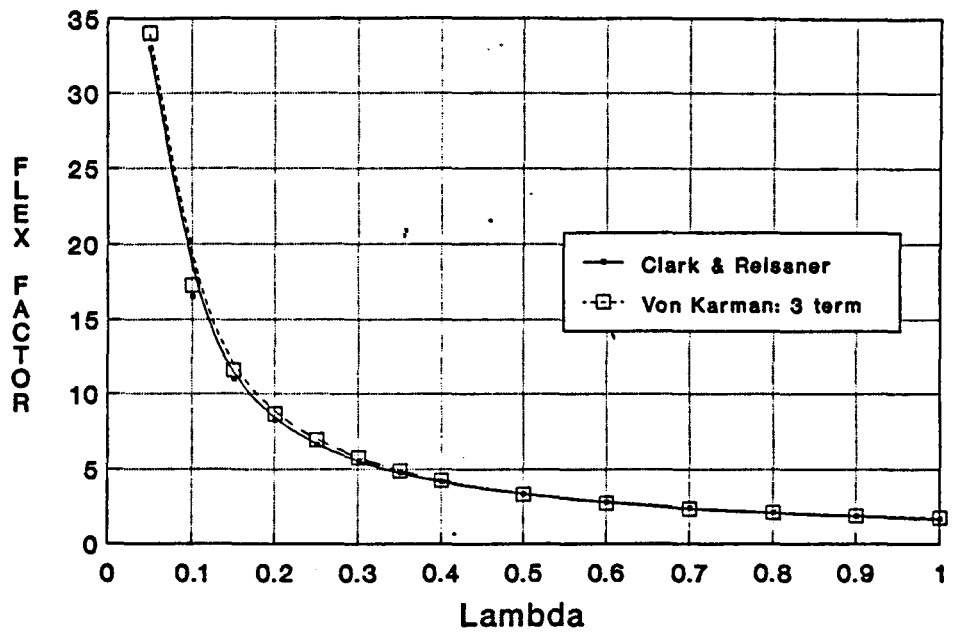


Figure 2.5 Von Karman analysis stress distribution for an elbow of $\lambda = 0.5$ and $\nu = 0.3$



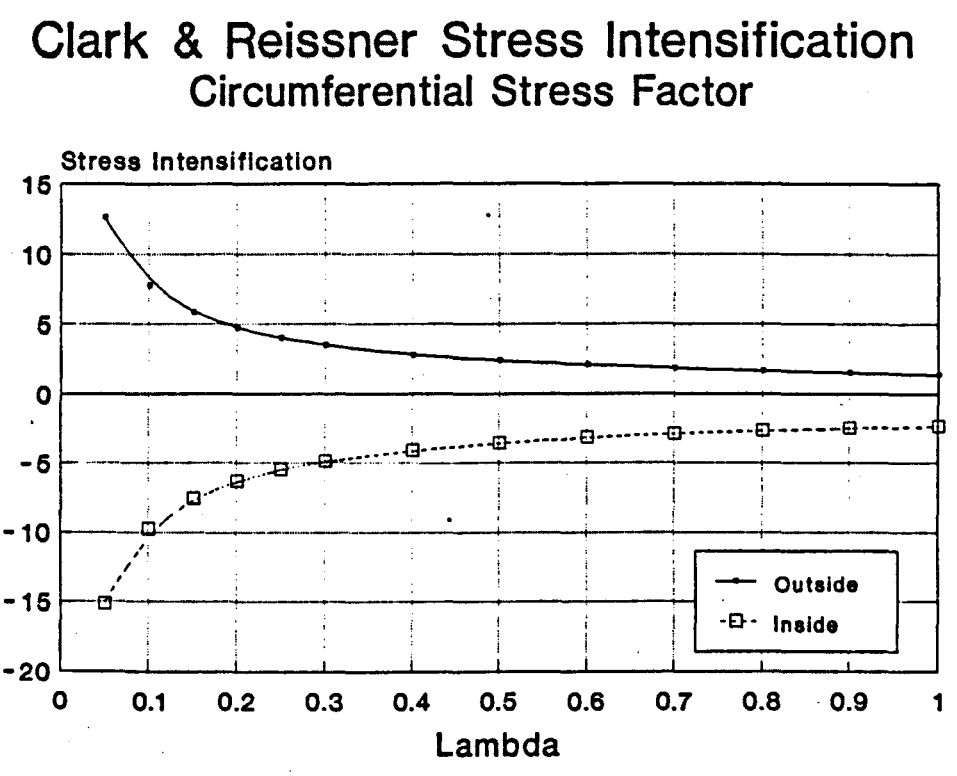


Figure 2.6 Clark and Reissner analysis flexibility and stress intensification factors for an elbow of $\lambda = 0.5$ and $\nu = 0.3$

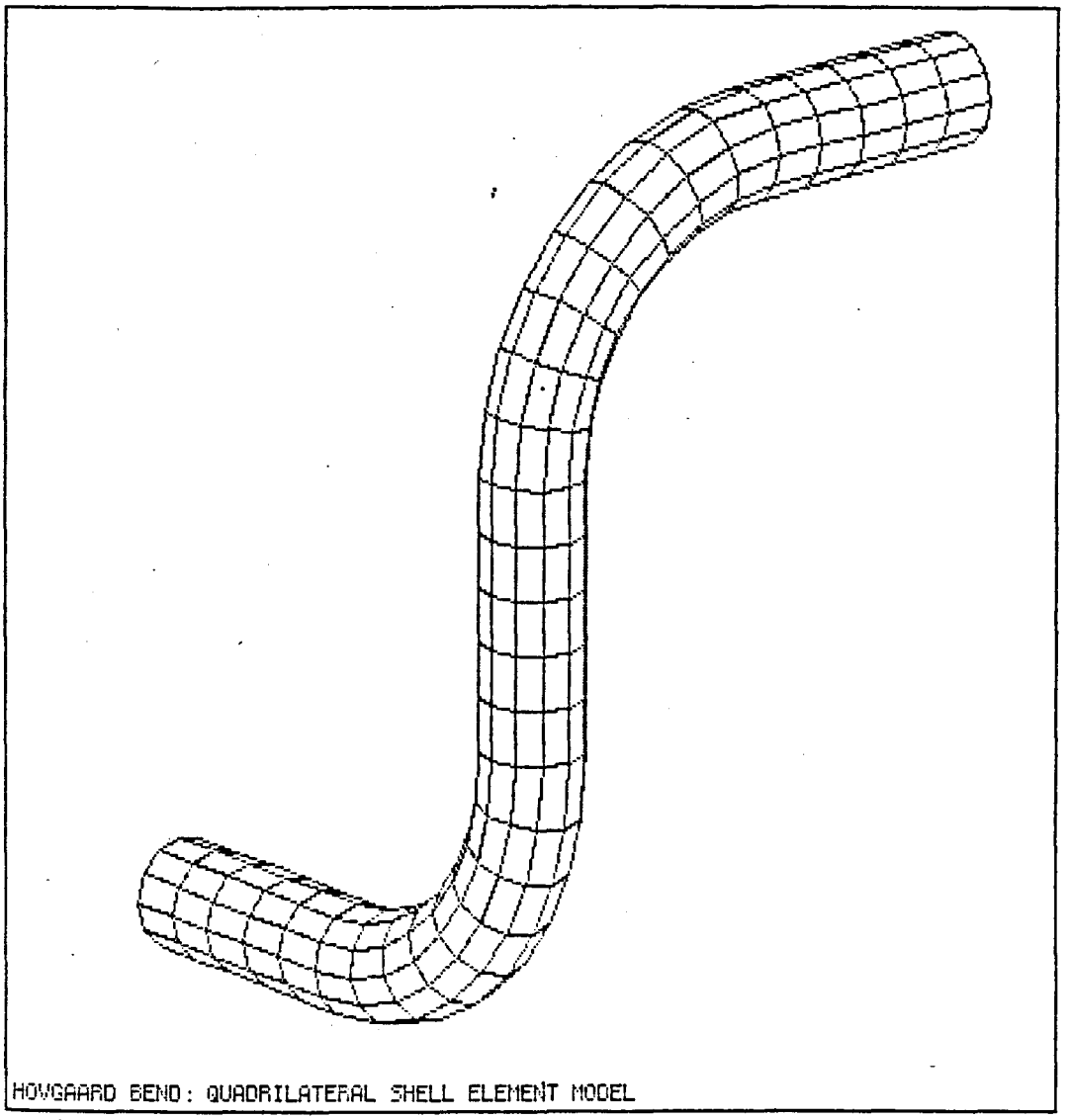


Figure 2.7 Thin shell finite element model of a piping system.

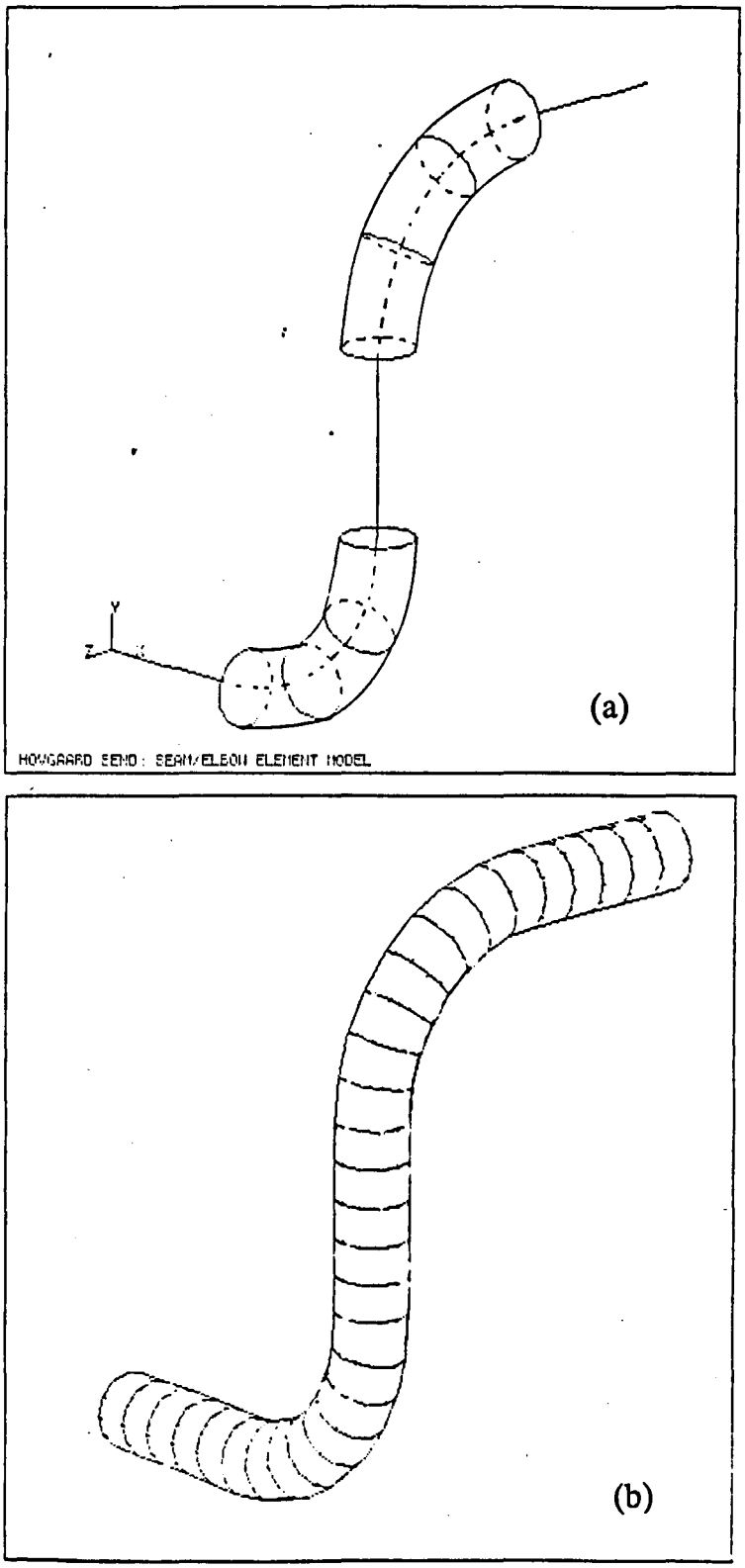


Figure 2.8 Schematic finite element models of a piping system using (a) beam-shell elbow elements and straight beam elements and (b) shell-ring elements.

CHAPTER 3.

A REVIEW OF PIPING ELBOW FINITE ELEMENTS.

3 A REVIEW OF PIPING ELBOW FINITE ELEMENTS.

Several piping elbow finite element formulations have been presented in the literature since the early 1970's. Most of these elements were intended for inelastic analysis of high temperature piping systems and model effects such as plasticity and creep which can not be adequately represented in simplified beam-based analyses. Although elbow elements are included in several commercial finite element programs, their use has not been universally accepted for general elastic piping analysis. Elbow elements intended for non-linear analysis are in some respects more complex than required for elastic analysis and have been regarded as being too expensive (in terms of computating requirements) for general use. In this Chapter a review of the piping elbow finite element literature is presented. The object of the review is to identify possible formulations for simple elbow elements for elastic analysis of piping systems.

The geometry of a general pipe bend is defined in Figure 2.2. The distance from the axis of symmetry Z to the centroid is the bend radius R. The bend subtends a bend angle Φ. The distance from the centroid of the bend to the wall mid-surface is the pipe radius r.

A point $p(\phi, \theta, \zeta)$ within the shell wall is defined by the angle ϕ along the axis of the bend, the angle θ around the circumference of the bend and the radial through thickness position ζ, positive outwards, such that $\frac{-t}{2} \leq \zeta \leq \frac{t}{2}$ where t is the wall thickness.

In the literature two conventions are commonly used for the origin of the circumferential angle θ: θ is either measured from the intrados (as in Figure 2.2) or from the crown of the bend ($\theta = 90°$ in Figure 2.2). In the following discussion of elbow elements the origin of θ for each element is consistent with that used in the original formulations.

3.1 The MARC Beam-Shell Elements.

The first pipe-bend element was formulated by Hibbitt, Sorensen and Marcal in the early 1970's, and installed as Element 17 in the MARC non-linear finite element program [3.1,3.2]. The element extends a bend analysis technique developed by Marcal in [3.3].

A piping elbow may be regarded as an incomplete shell of revolution, obtained by rotating a circle of radius r an angle $\phi = \alpha$ around an axis of symmetry distance R from the centre of the circle. This results in the bend geometry described above and shown in Figure 2.2. If the circle radius r is rotated through an angle $\alpha = 2\pi$ radians a complete axisymmetric toroidal shell is obtained. If such a torus is loaded axisymmetrically it may be modelled by a reduced shell theory, which takes account of the axisymmetric geometry and loading of the torus.

Axisymmetric shell analysis is considerably simpler than full two-dimensional shell analysis, but is not directly applicable in the analysis of pipe bends, where the torus is incomplete and the loading not axisymmetric. Marcal, however, proposed that the behaviour of an incomplete torus could be approximated by superposing a beam model, representing the non-axisymmetric modes of bending and stretch, on an axisymmetric shell finite element model representing axisymmetric modes such as ovalisation. This assumption is valid if the variation in strain in the Φ direction of the bend is negligible and is, effectively, a finite element approach to the clasic Von Karman constant bending analysis of elbows.

Marcal's original work was confined to analysis of single elbows, but the technique was developed by Hibbett *et al* in the formulation of a piping elbow element intended for use in the analysis of complete piping systems in which straight pipes are modelled by straight beam elements.

The geometry and co-ordinate system of MARC Element 17 is defined in Figure 3.1. The element has pipe radius r, bend radius R and subtends a bend angle α. Each bend in the finite element model is discretized into a number of elbow elements over which the constant bending assumption is approximately valid. The cross-section of each element in the bend is discretized as a ring of axisymmetric shell elements and an isoparametric coordinate system is used to define both shell displacement and position in terms of a local "parent" co-ordinate ζ as shown in Figure 3.1. For an arbitrary element with nodes 1-2, the isoparametric mapping is defined as

$$q = \sum_{i=1}^{2} \left(N_{1i} q_i + N_{2i} \left(\frac{dq}{d\zeta} \right)_i \right) = [N]\{q\}$$

where q is u_g, v_g, r or z. u_g and v_g are shell displacements defined in the global co-ordinate system. N_1 and N_2 are complete cubic polynomial functions of ζ given by

$$N_{1i} = \frac{1}{4}(\zeta_o \zeta^2 - 3\zeta_o + 2) \qquad N_{2i} = \frac{1}{4}(1 - \zeta_o)^2 (1 + \zeta_o)$$

where

$$\zeta_o = \zeta_i \zeta \quad \text{and} \quad -1 \le \zeta \le 1$$

Thus the axisymmetric element interpolation is obtained in the form:

$$\begin{Bmatrix} u_g \\ v_g \end{Bmatrix} = [N(\zeta)]\{d\}$$

where {d} is the degree of freedom vector consisting of the nodal displacements and their first derivatives with respect to ζ.

The axisymmetric shell model is required to describe local ovalisation effects only, as the element rigid body modes are included in the element's beam model. The shell model rigid body modes are suppressed by applying displacement constraints at discrete degrees of freedom on the cross-section.

The element beam deformation modes are represented by classical curved beam theory. Three beam modes are included in the element: relative stretch, relative in-plane rotation and relative out-of-plane rotation of the elbow end planes. No torsional deformation mode is included. The beam is constrained to deform to the arc of a circle, and all three deformation modes are assumed to give rise to longitudinal strain only, constant through the wall of the elbow. The total beam strain due to in-plane bending, out-of-plane bending and axial stretch is given by the expression:

$$\epsilon_{\phi b} = \frac{1}{\alpha}\left(\frac{\Delta u}{r} + \left(1 - \frac{R}{r}\right)\Delta\alpha + \frac{z}{r}\Delta\psi\right)$$

The complete elbow element is formulated by superposing the ovalisation and beam models and the stiffness matrix evaluated by numerical integration. Static condensation is then applied, so that only beam freedoms are retained in what is effectively a single node element defined in terms of relative displacement and rotation degrees of freedom $\{\Delta u \quad \Delta\alpha \quad \Delta\psi\}^T$.

At this stage the element is not directly compatible with straight beam elements, and kinematic constraint equations are used to relate the single node degrees of freedom to two additional beam-type nodes A and B, defined at the ends of the element centroid as shown in Figure 3.2. The kinematic relationships are based on the assumption that:

i) The elbow deforms to the arc of a circle.

ii) Stretch is not accompanied by radial motion.

iii) There is no axial displacement at the mid-span of the element.

iv) There is no rotation at the mid-span of the element.

v) The element is torsionally rigid.

From these assumptions the following relationships can be defined:

$$u_x^B - u_x^A = 2\left(\sin\frac{\alpha}{2}\right)\frac{\Delta u}{\alpha} + R\left(\cos\frac{\alpha}{2} - \frac{2}{\alpha}\left(\sin\frac{\alpha}{2}\right)\right)\Delta\alpha$$

$$u_y^B - u_y^A = 2R\left(\sin\frac{\alpha}{2}\right)\Theta_z^A - R\left(\sin\frac{\alpha}{2}\right)\Delta\alpha$$

$$u_z^B - u_z^A = -R\left(\sin\frac{\alpha}{2}\right)\Delta\psi - 2R\left(\sin\frac{\alpha}{2}\right)\Theta_A$$

$$\Theta_y^B - \Theta_y^A = \Delta\psi$$

$$\Theta_z^B - \Theta_z^A = -\Delta\alpha$$

$$\Theta_x^B - \Theta_x^A = 0$$

In piping system finite element analysis a 90^o bend is typically modelled by three or four elbow elements, with 16 to 32 axisymmetric elements around the cross-section [3.4].

Although MARC Element 17 has been used in industry for many years, there are several basic deficiencies in the formulation. The use of constraint equations for inter-element connection means that end effects cannot be included and a consistent mass matrix cannot be formulated. The constant bending assumption

also gives rise to compatibility problems, particularly in out-of-plane bending where the moment varies rapidly with respect to axial position. The out-of-plane response of the element is further compromised by the omission of torsion effects, as, in practice, there is a high degree of coupling between the out-of-plane moment and twisting of the bend.

The element provided what Hibbitt termed "a workable tool", but in order to circumvent the problems arising from the kinematic coupling in particular, an "improved" element formulation was presented in [3.2]. In the second element the ovalisation response is superposed on a conventional 12 degrees of freedom curved beam element, thus dispensing with the need for applying kinematic coupling. As in the original element the cross-section is modelled using cubic polynomials; however the ovalisation is interpolated linearly along the axis of the bend. Ovalisation degrees of freedom are statically condensed from the element before assembly as in the original element.

Whilst the use of a conventional beam element introduced the effects of torsion to the elbow element, many of the original limitations such as lack of inter-element ovalisation compatibility persisted. In practice the element was never implemented in the MARC program.

3.2 The Ohtsubo and Watanabe Ring Element.

In 1977 Ohtsubo and Watanabe suggested an alternative approach to beam/beam-shell modelling of piping systems when they presented their "finite ring element method of analysis", in which an entire branchless piping system is discretized into straight and curved rings of shell [3.5,3.6]. The element is based on a complete shell theory and includes axial variation in strain and interaction effects between adjoining elements.

In [3.5] a curved ring element as shown in Figure 3.3 was presented. The element subtends an angle α and bends are modelled by using sufficient ring elements to give convergence in the usual finite element manner.

The element geometry is a segment of toroidal shell, the behaviour of which is defined by Washizu's linearised small displacement shell theory under the Love-Kirchhoff hypothesis [3.7]. The deformation of a point $p(\phi, \theta, \zeta)$ in the shell is defined by the linearised equations

$$u_\phi = u(\phi,\theta) + \zeta\Theta_\phi(\phi,\theta)$$

$$u_\theta = v(\phi,\theta) + \zeta\Theta_\theta(\phi,\theta)$$

$$u_\zeta = w(\phi,\theta)$$

where $u(\phi,\theta)$, $v(\phi,\theta)$, and $w(\phi,\theta)$ are translational displacements of a point $q(\phi,\theta,0)$ on the mid-surface of the element as shown in Figure 3.4. Θ_ϕ and Θ_θ are the rotations of the shell at point q in the θ and ϕ directions respectively.

Before invoking the Love-Kirchhoff hypothesis the state of strain at p is defined by the expressions:

$$\epsilon_\phi = \epsilon_{o\phi} + \zeta\kappa_{o\phi} \qquad \gamma_{\phi\zeta} = \text{constant}$$

$$\epsilon_\theta = \epsilon_{o\theta} + \zeta\kappa_{o\theta} \qquad \gamma_{\theta\zeta} = \text{constant}$$

$$\gamma_{\phi\theta} = \epsilon_{o\phi\theta} + \zeta\kappa_{o\phi\theta}$$

where $\epsilon_{o\phi}, \epsilon_{o\theta}$, and $\epsilon_{o\phi\theta}$ are the mid-surface membrane strains, $\kappa_{o\phi}, \kappa_{o\theta}$, and $\kappa_{o\phi\theta}$ are bending strains, and $\gamma_{\phi\zeta}$ and $\gamma_{\theta\zeta}$ are transverse shear strains.

The strains are related to the shell displacements and rotations by the following equations:

$$\epsilon_{o\phi} = \frac{1}{R + r\sin\theta}\left(\frac{\partial u}{\partial \phi} + v\cos\theta + w\sin\theta\right) \tag{3.1a}$$

$$\kappa_{o\phi} = \frac{1}{R + r\sin\theta}\left(\frac{\partial \Theta_\theta}{\partial \phi} - \Theta_\phi\cos\phi\right) \tag{3.1b}$$

$$\epsilon_{o\theta} = \frac{1}{r}\left(\frac{\partial v}{\partial \theta} + w\right) \tag{3.1c}$$

$$\kappa_{o\theta} = -\frac{1}{r}\frac{\partial \Theta_\phi}{\partial \theta} \tag{3.1d}$$

$$\epsilon_{o\phi\theta} = \frac{1}{R + r\sin\theta}\left(\frac{\partial v}{\partial \phi} - u\cos\theta\right) + \frac{1}{r}\frac{\partial \Theta_\theta}{\partial \theta} \tag{3.1e}$$

$$\kappa_{o\phi\theta} = \frac{-1}{R + r\sin\theta}\left(\frac{\partial \Theta_\phi}{\partial \phi} + \Theta_\theta \cos\theta\right) + \frac{1}{r}\frac{\partial \Theta_\theta}{\partial \theta} \qquad (3.1f)$$

$$+ \frac{\sin\theta}{R + r\sin\theta}\frac{1}{r}\frac{\partial u}{\partial \theta} + \frac{1}{R + r\sin\theta}\frac{1}{r}\left(\frac{\partial v}{\partial \phi} - u\cos\theta\right)$$

$$\gamma_{\phi\zeta} = -\Theta_\phi + \frac{1}{r}\left(\frac{\partial w}{\partial \theta} - v\right) \qquad (3.1g)$$

$$\gamma_{\theta\zeta} = \Theta_\theta + \frac{1}{R + r\sin\theta}\left(\frac{\partial w}{\partial \phi} - u\sin\theta\right) \qquad (3.1h)$$

Thus it is seen that the state of strain in the elbow is defined by the translations *u, v,* and *w,* the rotations Θ_ϕ and Θ_θ, and certain first derivatives of these with respect to the shell angles θ and ϕ.

Under the Love-Kirchhoff hypothesis, plane fibres initially straight and perpendicular to the mid-surface are constrained to remain so upon deformation. Such deformation can occur only if the transverse shear strains are zero and therefore, from (3.2g) and (3.2h) above, the shell rotations are fully defined in terms of translation derivatives by the equations:

$$\Theta_\phi = \frac{1}{r}\left(\frac{\partial w}{\partial \theta} - v\right) \qquad (3.2a)$$

$$\Theta_\theta = \frac{-1}{R + r\sin\theta}\left(\frac{\partial w}{\partial \phi} - u\sin\theta\right) \qquad (3.2b)$$

and the state of strain at point p is fully defined in terms of the mid-surface displacements u, v, and w and derivatives thereof.

In the Ohtsubo and Watanabe formulation the mid-surface displacements u, v and w are interpolated around the circumferential direction θ by a Fourier series and along the meridinal direction ϕ by second order (cubic) Hermitian polynomials. The combined interpolation functions are:

$$u = \sum_{m=0}^{M}\left(H_1(\phi)\left(\frac{du}{d\phi}\right)_i^{(m)} + H_2(\phi)u_i^{(m)} + H_3(\phi)\left(\frac{du}{d\phi}\right)_j^{(m)} + H_4(\phi)u_j^{(m)}\right)G_u^{(m)}(\theta)$$

$$v = \sum_{m=0}^{M}\left(H_1(\phi)\left(\frac{dv}{d\phi}\right)_i^{(m)} + H_2(\phi)v_i^{(m)} + H_3(\phi)\left(\frac{dv}{d\phi}\right)_j^{(m)} + H_4(\phi)v_j^{(m)}\right)G_v^{(m)}(\theta)$$

$$w = \sum_{m=0}^{M} \left(H_1(\phi) \left(\frac{dw}{d\phi} \right)_i^{(m)} + H_2(\phi) w_i^{(m)} + H_3(\phi) \left(\frac{dw}{d\phi} \right)_j^{(m)} + H_4(\phi) w_j^{(m)} \right) G_w^{(m)}(\theta)$$

The element degrees of freedom are the Fourier coefficients u^m, v^m and w^m for m = 1 to 6 and their first derivatives with respect to ϕ at the element end planes $\phi = 0$ and $\phi = \alpha$, denoted as sections *i* and *j* respectively. This effectively defines nodal rings (as opposed to points) at the element end planes, around which the displacements vary according to the active Fourier modes.

H_k, k = 1,2,3,4 are the Hermitian shape functions

$$H_1(\phi) = \phi - \frac{2\phi^2}{\alpha} + \frac{\phi^3}{\alpha^2} \qquad H_2(\phi) = 1 - \frac{3\phi^2}{\alpha^2} + \frac{2\phi^3}{\alpha^3}$$

$$H_3(\phi) = \frac{-\phi^2}{\alpha} + \frac{\phi^3}{\alpha^2} \qquad H_4(\phi) = \frac{3\phi^2}{\alpha^2} - \frac{2\phi^3}{\alpha^3}$$

G_u, G_v and G_w are trigonometric (Fourier) functions, given for in-plane and out-of-plane bending by Tables 3.1 and 3.2 respectively.

m	0	1	2	3	4	5	6
$G_u^{(m)}(\theta)$	1	$\sin\theta$	$\cos 2\theta$	$\sin 3\theta$	$\cos 4\theta$	$\sin 5\theta$	$\cos 6\theta$
$G_v^{(m)}(\theta)$	0	$\cos\theta$	$\sin 2\theta$	$\cos 3\theta$	$\sin 4\theta$	$\cos 5\theta$	$\sin 6\theta$
$G_w^{(m)}(\theta)$	1	$\sin\theta$	$\cos 2\theta$	$\sin 3\theta$	$\cos 4\theta$	$\sin 5\theta$	$\cos 6\theta$

Table 3.1. In-plane bending ovalisation functions.

m	0	1	2	3	4	5	6
$G_u^{(m)}(\theta)$	0	$\cos\theta$	$\sin 2\theta$	$\cos 3\theta$	$\sin 4\theta$	$\cos 5\theta$	$\sin 6\theta$
$G_v^{(m)}(\theta)$	1	$\sin\theta$	$\cos 2\theta$	$\sin 3\theta$	$\cos 4\theta$	$\sin 5\theta$	$\cos 6\theta$
$G_w^{(m)}(\theta)$	0	$\cos\theta$	$\sin 2\theta$	$\cos 3\theta$	$\sin 4\theta$	$\cos 5\theta$	$\sin 6\theta$

Table 3.2. Out-of-plane plane bending ovalisation functions.

Separate elbow elements were programmed for in-plane and out-of-plane loading, apparently because of limited computer access when the elements were developed. Each element has 12 x m degrees of freedom, with a maximum of 72 for m = 6, and the stiffness matrix is integrated numerically by a Gauss quadrature rule. 32 points are used around the circumference of the element and 4 in the axial direction.

The full element stiffness matrix is assembled at global level, enforcing inter-element continuity of all degrees of freedom. Thus ovalisation is continuous between adjacent elements and ovalisation constraints can be applied at rigid connections such as flanges. However, in finite element terminology the Ohtsubo and Watanabe element is said to have "excessive" continuity, as derivatives of higher order than theoretically required are continuous between elements.

The continuity requirements of a general finite element can be defined by considering the potential energy expression of the element. The potential energy is generally a function of displacement and displacement derivatives up to order n. Therefore, to adequately represent the element energy, the displacement field must be such that derivatives up to n are non-zero and continuous throughout the element. However, the element stiffness equations are obtained by integrating the potential energy expression and, consequently, the highest order of derivative which must be continuous between "conforming" elements is n-1. An element satisfying this condition is said to be C_{n-1} continuous.

If the Love-Kirchhoff hypothesis is not invoked, the highest order of derivative in the strain displacement relations, (and hence the element strain energy), is $n = 1$. Therefore C_0 continuity is required between elements; that is, continuity of displacements and rotations u,v,w,Θ_ϕ and Θ_θ. However, under Love-Kirchhoff the rotations and rotation derivatives are defined in terms of displacements and displacement derivatives, and from (3.2):

$$\frac{\partial \Theta_\phi}{\partial \theta} = \frac{1}{r}\frac{\partial}{\partial \theta}\left(\frac{\partial w}{\partial \theta} - v\right) = \frac{1}{r}\left(\frac{\partial^2 w}{\partial \theta^2} - \frac{\partial v}{\partial \theta}\right)$$

and

$$\frac{\partial \Theta_\theta}{\partial \phi} = \frac{-1}{R + r\sin\theta}\left(\frac{\partial^2 w}{\partial \phi^2} - \frac{\partial u}{\partial \phi}\sin\theta\right)$$

The highest order of derivative in the strain equations is now $n=2$ for the out-of-plane displacement w, and hence C_1 continuity of w is required. As only first derivatives of u and v exist no derivatives of these displacements are required to be continuous for a conforming element. However, due to the use of cubic interpolation of all three displacements in the ϕ direction, first derivatives of u, v and w are included in the element degrees of freedom in the form of Fourier coefficients and derivatives. As this is in excess of the theoretical requirements the element is said to have excessive continuity.

Although the ring element is applicable to general loading problems and can model interaction effects, it is subject to several disadvantages. In practice, use of the element has been limited by its size. Convergence studies show that a 90^o bend of pipe parameter $\lambda = 0.0924$ subject to constant bending, (in-plane or out-of-plane), requires six elements around the bend and six ovalisation modes. Therefore, with up to 72 degrees of freedom per element, an expensive 32x4 point numerical integration scheme and with a minimum of six elements per 90^o bend, the use of such an element is limited by computing costs and model size.

It has also been found that unless the mesh is sufficiently fine spurious error stresses arise in the element, due to the inability of the Fourier-Hermitian displacement field to fully encapsulate the rigid body modes of the curved element in a global cartesian system. This problem vanishes in the limit as the mesh is refined. In [3.5] the "error stress" is quantified, and recommendations for appropriate mesh sub-division to minimise the problem are given.

3.3 The PAMEL Ring Element.

The ring element approach was pursued by Lazzeri in [3.8] in an element called PAMEL (Plastic Analysis Membrane Elbows), intended principally for elasto-plastic analysis of Pressurised Water Reactors (PWR) using the PAULA (Plastic Analysis Using Library and ADINA) finite element program.

The element is based on Vlasov's thin shell theory under the Love-Kirchhoff hypothesis, which is similar in form to that of Washizu discussed above. The element is intended for use with straight beam elements representing straight runs, as opposed to the complete ring discretization proposed by Ohtsubo and Watanabe.

The element mid-surface displacements u, v and w are interpolated around the element cross-section by the Fourier series:

$$u(\phi) = u_o(\phi) + \sum_{m=1}^{M} (u_1^{(m)}(\phi)\cos m\theta + u_2^{(m)}(\phi)\sin m\theta)$$

$$v(\phi) = v_o(\phi) + \sum_{m=1}^{M} (v_1^{(m)}(\phi)\cos m\theta + v_2^{(m)}(\phi)\sin m\theta)$$

$$w(\phi) = w_o(\phi) + \sum_{m=1}^{M} (w_1^{(m)}(\phi)\cos m\theta + w_2^{(m)}(\theta)\sin m\phi)$$

As in the earlier ring element this effectively defines nodal rings at the element end planes. Applying Vlasov's strain-displacement relationships the ring strains are defined in terms of the Fourier coefficients, their first derivatives with respect to ϕ, and the second derivative of w with respect to ϕ.

The derivatives of the Fourier coefficients in the strain displacement matrix are obtained by a finite difference technique, in which the axis of the element is meshed by N difference stations, where the n^{th} derivative of an general coefficient, say q, at station *i* is given by the difference equation

$$\left(\frac{\partial^n q}{\partial \phi^n}\right)_i = \sum_{i=1}^{N} \beta_i^n q_i$$

where the β terms are evaluated from à Lagrangian polynomial.

The complete element has 9xMxN degrees of freedom, where M is the number of Fourier terms used in the interpolation and N the number of difference stations along the axis of the bend. In practice both M and N are generally taken as 4, giving 144 degrees of freedom per element. The element stiffness matrix is evaluated by numerically integrating around the cross-section by a 14 point Gaussian quadrature rule, and through thickness by a generalised constitutive matrix.

The element stiffness matrix is statically condensed to a 12 degree of freedom matrix compatible with standard beam based formulations. The degrees of freedom are the Fourier coefficients which mimic beam bending modes, namely:

u_0 - axial displacement
v_1 - displacement parallel to the axis of symmetry
v_2 - radial displacement
u_1 - in-plane rotation
u_2 - out-of-plane rotation
v_0 - axial rotation

at the end planes of the element.

Although static condensation makes the final element more economically attractive, it precludes the element from ovalisation compatibility with its neighbours. Further, the "plane sections remain plane" assumption imposed on the end planes in order to mimic beam modes effectively suppresses warping at the element ends. In these respects the element offers no advantage over MARC Element 17.

The performance of the Lazzeri element is compared with that of Ohtsubo and Watanabe for a bend with a rigid flange and with MARC for a bend free to ovalise in [3.8]. Good agreement is found in both cases.

3.4 The FINAS Quadrilateral Shell Ring Elements.

In [3.9,3.10,3.11] Takeda *et al* present formulations for three elbow elements developed for the Japanese Power Reactor and Nuclear Fuel Development Corporation finite element program FINAS (FBR Inelastic Structural Analysis System). The elements are called ELBOW6, ELBOW6R and ELBOW3R.

ELBOW 6 and ELBOW 6R are finite ring elements intended for discretization of complete branchless piping systems. In fact both could be more accurately described as element rings, as each ring in the model is assembled circumferentially from doubly curved thin shell elements. The element formulations are presented for curved (elbow) rings, which degenerate to cylindrical rings for modelling straight sections of pipe. ELBOW 3R is a constant bending element similar to MARC Element 17, but allowing axial ovalisation interpolation.

The geometries of ELBOW 6 and ELBOW 6R are identical and shown in Figure 3.4. Elements rings are assembled from a number of doubly curved quadrilateral

shell elements. Each element in the assembly has seven associated nodes: four on the mid-surface at the quadrilateral corners, one at each end of the ring centroid, and one at the centre of curvature of the bend.

ELBOW6 is based on Washizu's shell theory under the Love-Kirchhoff hypothesis as used by Ohtsubo and Watanabe. The circumferential and radial displacements v and w are interpolated in terms of the mid-surface node degrees of freedom by a two dimensional cubic Hermitian polynomial. Axial displacement is interpolated around the cross-section by a Fourier series, and the Fourier coefficients are interpolated along the axis in terms of the centroid nodes' degrees of freedom by a one-dimensional cubic Hermitian polynomial. The remaining node defines the orientation of the bend.

Considering the element geometry shown in Figure 3.4, nodes 1, 2, 4 and 5 are the surface or shell nodes, nodes 3 and 6 the end-plane or beam nodes, and node 7 the orientation node. The mid-surface axial strain is interpolated by the Fourier series

$$u(\xi,\theta) = u_1(\xi) + \sum_{m=1}^{M} u_{2m}(\xi)\sin m\theta + u_{2m+1}(\xi)\cos m\theta)$$

where ξ is the axial isoparametric variable: $-1 \le \xi \le 1$, as defined in Figure 3.5.

The Fourier coefficients $u_{(i)}$, where i = 1, 2m etc., are interpolated in terms of the axial degrees of freedom of the beam nodes, u^3 and u^6, by a one-dimensional cubic Hermitian polynomial:

$$u_i(\zeta) = H_{30}(\xi)u_i^3 + H_{31}(\xi)\left(\frac{\partial u_i}{\partial \xi}\right)^3 + H_{60}(\xi)u_i^6 + H_{61}(\xi)\left(\frac{\partial u_i}{\partial \xi}\right)^6$$

where

$$H_{30} = \frac{1}{4}(2 - 3\xi + \xi^3) \qquad H_{60} = \frac{1}{4}(2 + 3\xi - \xi^3)$$

$$H_{31} = \frac{1}{4}(1 - \xi - \xi^2 + \xi^3) \qquad H_{61} = \frac{1}{4}(-1 - \xi + \xi^2 + \xi^3)$$

The ovalisation degrees of freedom v^i and w^i of the shell nodes i=1,2,4,5 are interpolated by complete bicubic expansion of the above Hermitian functions, extended to two dimensions in the isoparametric system (ξ, η). The quadratic shell ovalisation field is therefore expressed:

$$v(\xi,\eta) = \sum_{i=1,2,4,5} \left(H_{i00}v^i + H_{i10}\left(\frac{\partial v}{\partial \xi}\right)^i H_{i01}\left(\frac{\partial v}{\partial \eta}\right)^i + H_{i11}\left(\frac{\partial^2 v}{\partial \xi \partial \eta}\right)^i\right.$$

$$w(\xi,\eta) = \sum_{i=1,2,4,5} \left(H_{i00}w^i + H_{i10}\left(\frac{\partial w}{\partial \xi}\right)^i H_{i01}\left(\frac{\partial w}{\partial \eta}\right)^i + H_{i11}\left(\frac{\partial^2 w}{\partial \xi \partial \eta}\right)^i\right.$$

where the polynomials H_{ijk} are products of the one-dimensional functions defined above, according to the scheme

$$H_{ijk} = H_{li}(\xi) H_{mk}(\eta)$$

with corresponding values for i ,l and m as given in Table 3.

i	l	m
1	3	3
2	3	6
4	6	3
5	6	6

Table 3. Hermitian polynomial multiplication integer values.

The element degrees of freedom are displacements u, v and w, their first derivatives with respect to both isoparametric directions, and the twist of the in-plane displacements, u and v, given by $\partial / \partial \xi \partial \eta$. The element has excessive continuity in both shell directions due to the use of bicubic interpolation.

In convergence tests it was found that a large number of ELBOW 6 elements are required to model a single elbow, making it too expensive for general use. In an attempt to improve the element performance a modified ring element with relaxed continuity requirements, ELBOW 6R, was formulated.

ELBOW 6R is identical to ELBOW 6 in geometry, and again is based on Washizu's shell theory. However the Love-Kirchhoff hypothesis is not invoked, thus reducing the element continuity requirements to C_0 as discussed in section 3.2. Transverse shear effects are included and the mid-surface of the shell has freedoms u, v, w, Θ_ϕ and Θ_θ.

Axial displacement is interpolated by the same Fourier series as ELBOW 6, but the Fourier coefficients are now interpolated along the axis in terms of the beam nodes 3 and 6 by a Lagrangian scheme, such that

$$u_i = \frac{1}{2}(1-\xi)u_i^3 + \frac{1}{2}(1+\xi)u_i^6$$

The shell displacements v and w and rotations Θ_ϕ and Θ_θ are interpolated bilinearly in the isoparametric system in terms of their nodal values. Thus

$$v = \sum_{i=1,2,4,5} N_i v^i \qquad w = \sum_{i=1,2,4,5} N_i w^i$$

$$\Theta_\phi = \sum_{i=1,2,4,5} N_i \Theta_\phi^i \qquad \Theta_\theta = \sum_{i=1,2,4,5} N_i \Theta_\theta^i$$

where N_i i=1,2,4,5 is the function

$$N_i = \frac{1}{4}(1+\xi\xi_i)(1+\eta\eta_i)$$

Convergence tests for ELBOW 6R show that even with the relaxed continuity requirement a large number of degrees of freedom are required for each bend. In an attempt to improve performance, reduced integration of the element stiffness matrix was introduced. Both selective (shear terms) and uniform reduced integration schemes were examined and it was found that, whilst both improved convergence, the uniform reduction was significantly more effective.

ELBOW 3R is a simplified version of ELBOW 6R, originally programmed to study integration requirements around the cross-section of the ring. The state of strain is assumed to be constant with respect to axial position, thus the element is effectively a constant bending element similar to MARC Element 17.

The general formulation is that of ELBOW 6R, but the constant ovalisation assumption sets $\Theta_\theta = 0$ and the remaining shell displacements are interpolated in terms of the co-planar shell nodes 1 and 2. Thus

$$v = \sum_{i=1}^{2} N_i v^i \qquad w = \sum_{i=1}^{2} N_i w^i \qquad \Theta_\phi = \sum_{i=1}^{2} N_i \Theta_\phi^i$$

where

$$N_1 = \frac{1}{2}(1-\eta) \qquad N_2 = \frac{1}{2}(1+\eta)$$

Although ELBOW 6 and ELBOW 6R have been shown to be accurate in published results, neither is suitable for general piping analysis due to the element size, where no substantial benefit is gained over elbow modelling using standard doubly curved shell elements. ELBOW 3R could be used in general applications provided arrangements were made for joining to straight beam elements, for example using MARC Element 17 type constraint equations.

3.5 The ADINA-P Beam-Shell Element.

The beam-shell element approach proposed by Hibbitt in [3.2] was developed and extended by Bathe and Almeida in the early 1980's, when they presented the formulation of a beam-shell element for the ADINA-P program [3.12,3.13,3.14]. The element geometry and co-ordinate systems are shown in Figure 3.5.

The beam model is a 4 node isoparametric thick beam based on cubic Lagrangian interpolation. Each node has six degrees of freedom: translations u_1, u_2, and u_3 in the global directions X_1, X_2 and X_3 respectively, and corresponding global rotations Θ_1, Θ_2 and Θ_3. The cross-section of the beam is assumed to be rigid, and sections originally plane and normal to the centroid of the beam are assumed to remain plane but not necessarily normal upon deformation.

A point in the beam element is defined by an isoparametric co-ordinate system ζ, η, ξ. The displacement of a point $p(\zeta, \eta, \xi)$ is defined by components of the global displacement in the local isoparametric directions. Thus p has freedoms

$$u_{1\zeta} u_{1\eta} u_{1\xi} u_{2\zeta} \ \ldots\ u_{3\xi}$$

These freedoms are interpolated in terms of the global degrees of freedom at the four nodes of the element by the equation

$$\begin{Bmatrix} u_{i\xi} \\ u_{i\eta} \\ u_{i\zeta} \end{Bmatrix} = \sum_{k=1}^{4} h_k \begin{bmatrix} 1 & [g]_{1i}^k & [g]_{2i}^k & [g]_{3i}^k \\ 0 & [\hat{g}]_{1i}^k & [\hat{g}]_{2i}^k & [\hat{g}]_{3i}^k \\ 0 & [\bar{g}]_{1i}^k & [\bar{g}]_{2i}^k & [\bar{g}]_{3i}^k \end{bmatrix} \begin{Bmatrix} u_i^k \\ \begin{bmatrix} \Theta_1^k \\ \Theta_2^k \\ \Theta_3^k \end{bmatrix} \end{Bmatrix}$$

h_k, k=1,2,3,4 are the cubic Lagrangian interpolation functions

$$h_1 = (-9\zeta^3 + 9\zeta^2 + \zeta - 1)/16 \qquad h_3 = (27\zeta^3 - 9\zeta^2 - 27\zeta + 9)/16$$

$$h_2 = (9\zeta^3 + 9\zeta^2 - \zeta - 1)/16 \qquad h_4 = (-27\zeta^3 - 9\zeta^2 + 27\zeta + 9)/16$$

The [g] matrices are local to global co-ordinate rotation matrices, such that

$$[\hat{g}] = r_o \begin{bmatrix} 0 & -e_{\eta 3}^m & e_{\eta 2}^m \\ e_{\eta 3}^m & 0 & e_{\eta 1}^m \\ -e_{\eta 2}^m & e_{\eta 1}^m & 0 \end{bmatrix}$$

$$[\hat{g}] = r_o \begin{bmatrix} 0 & -e_{\xi 3}^m & e_{\xi 2}^m \\ e_{\xi 3}^m & 0 & e_{\xi 1}^m \\ -e_{\xi 2}^m & e_{\xi 1}^m & 0 \end{bmatrix}$$

and

$$[g]_{ij}^m = \eta[\hat{g}]_{ij}^m + \xi[\bar{g}]_{ij}^m$$

where $e_{\eta i}^m$ and $e_{\xi i}^m$ are components of the unit vectors in the local η and ξ directions in the global system.

The global translations are obtained from the above interpolation functions in terms of the nodal degrees of freedom. The global displacement derivatives are obtained by performing the Jacobian transformation

$$\frac{\partial}{\partial x} = [J]^{-1} \frac{\partial}{\partial \zeta}$$

where the Jacobian matrix [J] contains the derivatives of the global co-ordinates with respect to the isoparametric co-ordinates.

The global strains are obtained in terms of the nodal degrees of freedom from the above interpolation. These are then transformed into the local strain components ϵ_{ϕ}, $\gamma_{\phi,\theta}$ and $\gamma_{\phi,n}$.

The element ovalisation model is an extension of the MARC Element 17 constant bending analysis. Assuming that the cross-section ovalisation is constant with respect to axial position, the state of strain is constant around the bend and plane sections remain plane upon deformation. This allows the shell theory to be simplified considerably, as all derivatives with respect to axial position can be neglected. Further, longitudinal strain due to bending and axial stretch and shear strain due to torsion are omitted from the shell theory and incorporated in the elbow model as beam bending strains. Von Karman's long radius assumption that $R >> r$ is not invoked, and the Washizu shell theory under the Love-Kirchhoff hypothesis discussed above reduces to

$$\epsilon_{\phi} = \epsilon_{o\phi} = \frac{1}{R - r\cos\theta}(v\sin\theta - w\cos\theta) \tag{3.3a}$$

$$\epsilon_{\theta} = \epsilon_{o\theta} + \zeta\kappa_{o\theta} = \epsilon_{o\theta} - \frac{\zeta}{r^2}\left(\frac{\partial^2 w}{\partial\theta^2} + \frac{\partial v}{\partial\theta}\right) \tag{3.3b}$$

where

$$\epsilon_{o\theta} = \frac{1}{r}\left(\frac{\partial v}{\partial\theta} + w\right) \tag{3.3c}$$

By invoking the Von Karman inextensibility assumption the circumferential membrane strain (3.3c) is equated to zero. This establishes a coupling condition relating the radial and tangential displacements of the shell:

$$w = -\frac{\partial v}{\partial\phi} \tag{3.4}$$

The circumferential stain is thus due to transverse bending only, and (iii) becomes:

$$\epsilon_{\theta} = -\frac{\zeta}{r^2}\left(\frac{\partial^2 w}{\partial\phi^2} + w\right)$$

The Von Karman constant ovalisation strains are interpolated cubically around the axis of the bend in terms of ovalisation degrees of freedom located at the beam nodes. The ovalisation radial displacement w_o is interpolated by the Fourier series

$$w(\zeta,\theta)=\sum_{n=1}^{N_i}\sum_{m=1}^{4}h_m a_n^m \sin 2n\theta+\sum_{n=1}^{N_o}\sum_{m=1}^{4}h_m b_n^m \cos 2n\theta$$

where h_m (m=1,2,3,4) are the cubic interpolation functions of the beam model. The tangential displacement is obtained in terms of the same Fourier coefficients by invoking the displacement coupling equation (3.4).

Up to three in-plane and three out-of-plane even Fourier terms may be included in the element, giving a maximum of 12 degrees of freedom per node: 6 standard beam degrees of freedom and up to 6 Fourier coefficients. Thus a single bend element has up to 48 degrees of freedom.

By including the nodal Fourier coefficients as degrees of freedom at assembly level, a degree of inter-element ovalisation continuity is achieved. However, as no account is taken of ovalisation slope with respect to the axial direction, continuity is limited to ovalisation displacement only.

In order to force full ovalisation continuity between elements an improved version of the element was proposed in [3.13]. Novozhilov's thin shell theory [3.15] was adopted to identify strain terms due to variation of ovalisation not included in the original formulation. Two strains were added to the formulation: longitudinal bending of the elbow wall

$$\epsilon^I=-\zeta\left(\frac{1}{R-r\cos\theta}\right)^2\frac{d^2w}{d\phi^2}$$

and a shear strain term

$$\gamma^I_{\phi\theta}=\left(\frac{1}{R-r\cos\theta}\right)\frac{dv}{d\theta}$$

where superscript *I* denotes interaction effect strains.

A second derivative of the radial displacement now occurs in the strain energy expression for the elbow, and C_1 continuity of displacement w is required for inter-element compatibility. However, the interpolation scheme cannot accomodate higher order connectivity directly, as Fourier coefficient gradients are not available as nodal degrees of freedom.

This problem was circumvented by the adoption of a penalty function method, which modifies the potential energy expression of the element by including constraint equations relating the degrees of freedom. This is analogous to including a finite stiffness coupling between elements to force continuity. The penalty parameter, which may be regarded as the coupling stiffness, must be high enough to ensure adequate coupling, but not so high as to cause numerical errors at the solution phase. Suitable parameter values are generally established empirically, and in this application Bathe and Almeida recommend a value in the order of the greatest ovalisation term in the element stiffness matrix.

The element stiffness matrix is integrated numerically. Through thickness and axial integration is by a 3x5 point Newton-Cotes rule, with the composite trapezoidal rule around the circumference: 12 points for in-plane loading, 24 for out-of-plane loading.

Although the element extends the MARC idea to include cubic variation of Von Karman ovalisation, with inter-element ovalisation compatibility, it is, in some respects, a simpler element than the MARC Element 17. The choice of even Fourier terms only limits the ovalisation to symmetric and antisymmetric modes which is valid only if $R>>r$; an assumption not invoked in the element formulation. However, published analyses using the ADINAP element have indicate that the element is "capable of calculating the static modal and transient thermal behaviour .. [of specific pipelines].. adequately" [3.16].

3.6 The Kanarachos and Koutsides Beam-shell Element.

A further beam-shell elbow element was presented by Kanarachos and Koutsides in [3.17]. The element geometry is defined in Figure 3.6. The beam model for the element is an "exact" solution, established from the curved beam theory of Bickford and Strom [3.18]. The actual beam displacement field is not presented

in [3.17], but was apparently obtained either analytically, by solution of the governing differential equations of the beam, or by extracting the shape functions numerically from a piecewise-linear curved beam finite element solution.

The ovalisation behaviour of the elbow is defined by a linear thin shell theory under the Love-Kirchhoff hypothesis. The tangential ovalisation displacement is interpolated by the function

$$v(x,\zeta) = \sum_{i=1}^{4} N_i v_i(x)$$

where the co-ordinate ζ defines the circumferential position such that $\zeta = \theta/2\pi$. v_i: i = 1,2,3,4 are ovalisation degrees of freedom corresponding to discrete values of v at four positions around the cross-section, stepped at 90° intervals from the intrados of the bend.

The ovalisation shape functions N_i: i = 1,2,3,4 are defined piecewise around the cross-section from $\zeta = 0$ to 0.5, and $\zeta = 0.5$ to 1 by cubic splines. N_1 is defined in Figure 3.6. The remaining three ovalisation shape functions, N_2, N_3, and N_4 are obtained by considering the rotational symmetry of the cross-section, that is rotating N_1 by 90°, 180° and 270° respectively.

The tangential displacement is interpolated along the axis of the bend by a cubic Hermitian polynomial

$$v_i(x) = H_1 v_i^1 + H_2 R\alpha \left(\frac{\partial v}{\partial s}\right)_i^1 + H_3 v_i^2 + H_4 R\alpha \left(\frac{\partial v}{\partial s}\right)_i^2$$

where the functions H_i are

$$H_1 = (1 - 3\eta^2 + 2\eta^3) \qquad H_3 = (3\eta^2 - 2\eta^3)$$

$$H_2 = (\eta - \eta^2 + \eta^3) \qquad H_4 = (-\eta^2 + \eta^3)$$

and η is an axial co-ordinate from zero at end 1 to unity at end 2.

The radial ovalisation displacement is defined by invoking the Von Karman inextensibility assumption (3.4)

No information concerning how ovalisation rigid body modes are suppressed is presented in [3.17], nor is the integration rule stated for the element formation. The element is interesting in that it introduces polynomial interpolation of Von Karman ovalisation, but insufficient application examples have been published to assess how effective this interpolation scheme is.

3.7 The TEDEL Beam-Shell Element.

In 1980 Millard and Hoffman presented the formulation of a beam-shell element included in the French "Commissarait a L'Energie Atomique" non-linear beam program TEDEL [3.19]. The element is essentially an extension of the classic Von Karman analysis. The shell behaviour is represented by a thin shell theory under the Love-Kirchhoff hypothesis, in which the Von Karman assumption of circumferential mid-surface inextensibility is invoked. The radial ovalisation displacement is interpolated by the Fourier series

$$w(\theta,\phi)=\sum_{n=1}^{N}(a_n(\phi)\cos 2n\theta+b_n(\phi)\sin 2n\theta)$$

where the Fourier coefficients are chosen to be either constant or linear around the bend. The tangential displacement $v(\theta,\phi)$ is obtained from the inextensibility condition.

The element beam model includes pure in-plane and out-of-plane bending modes only; no axial extension or torsion modes are included. This is a simplification of the MARC Element 17 beam model, which included axial extension as well as bending modes.

The full element stiffness matrix, including beam and ovalisation modes, is statically condensed to produce an element with two degrees of freedom: relative in-plane and out-of-plane rotations of the elbow ends. Thus ovalisation compatibility is not enforced between elements.

Example analyses presented in [3.19] are limited to the study of local effects in single bends subject to pure moment and thermal loads. In order to use the element in general pipeline analysis, kinematic coupling such as that used in MARC Element 17 would be required.

3.8 The ABAQUS Elements.

The non-linear finite element program ABAQUS offers a number of elbow elements, described by the authors as "beams with deforming section". The elements have been upgraded since the inception of the program and two generations of element families are considered.

The original ABAQUS pipe bend elements are based on small displacement thin shell theory ovalisation modes superposed on a thin walled beam formulation [3.20]. The ovalisation is represented by a small displacement shell theory, again under the Love-Kirchhoff hypothesis, in which warping of the elbow cross-section is neglected. The ovalisation displacements v and w are interpolated independently by Fourier series, as in the Ohtsubo and Watanabe ring element. The functions are

$$v(\theta,\phi) = \sum_{n=1}^{N}\sum_{m=1}^{M} H_n(\phi) v^i_{nm} \sin m\theta + \sum_{n=1}^{N}\sum_{m=2}^{M} H_n(\phi) v^o_{nm} \cos m\theta$$

$$w(\theta,\phi) = \sum_{n=1}^{N} H_n(\phi) w_{no} \sum_{n=1}^{N} + \sum_{m=2}^{M} H_n(\phi) w^i_{nm} \sin m\theta + \sum_{n=1}^{N}\sum_{m=1}^{M} H_n(\phi) w^o_{nm} \cos m\theta$$

where v^i_m and w^i_m are in-plane ovalisation Fourier coefficients, and v^o_m and w^o_m are out-of-plane Fourier coefficients. The zero Fourier mode w_o is included in the radial displacement function, introducing a uniform radial expansion mode to the element. Other ovalisation rigid body modes are prevented by omitting zero and first Fourier mode terms as appropriate. H_n is a polynomial interpolation function, chosen to be linear, quadratic or cubic.

Several interpolation variations of the formulation were available in the original issue of ABAQUS, ranging from a linear beam with linear axial variation of cross-section through to a cubic beam with quadratic cross-section variation. Up to 16 Fourier terms could be included in these elements.

The second generation of ABAQUS elements extend the original formulation to include warping effects and a non-linear shell theory [3.21,3.22]. The behaviour of the elbow is defined by considering two co-ordinate systems; the reference (undeformed) system, and the current (deformed) system.

In the reference system a point on the axis of the elbow is defined in terms of the axial angle ϕ, whilst a point on the mid-surface is defined by both ϕ and θ. In this case θ is measured from the crown of the bend as shown in Figure 3.7. Points on the axis and mid-surface have reference co-ordinates $X_A(\phi)$ and $X(\theta,\phi)$ respectively. Corresponding co-ordinates in the current (deformed) system are $x_A(\phi)$ and $x(\theta,\phi)$.

The deformation of the bend is defined by considering two sets of right-handed orthogonal unit vectors, **A** in the reference system and **a** in the current system. The directors $\mathbf{A}=\{A_1, A_2, A_3\}$ are defined such that A_1 points towards the crown of the bend, A_2 towards the extrados and A_3 along the axis of the bend. After deformation the corresponding directors are $\mathbf{a}=\{a_1, a_2, a_3\}$, where a_3 is approximately tangent to the bend axis, but a_1 and a_2 are not generally coincident with A_1 and A_2.

The offset of a point on the pipe mid-surface from the bend axis in the current system is given by

$$x_o = x(\theta,\phi) - x_A(\phi) = \{x_{o1}, x_{o2}, x_{o3}\}$$

which is written in terms of **a** as

$$x_o = (r+w)t_r + vt_\theta + x_{o3}a_3$$

where w and v are radial and tangential displacements, and x_{03} is a warping freedom. The unit vectors t are:

$$t_r = a_1\cos\theta + a_2\sin\theta$$

$$t_\theta = -a_1\sin\theta + a_2\cos\theta$$

The rotation of the mid-surface of the elbow is obtained by defining a unit vector normal to the deformed surface, which is related to the current director system by a rotation denoted γ_{o3}. The deformed elbow is, therefore, fully defined by the displacements v and w, the warping freedom x_{03}, and the rotation γ_{o3}.

The beam modes of the element are defined by interpolating the current axial position $x_A(\phi)$ and a corresponding rotation set $\Omega_A(\phi)$ between nodal values by the polynomial functions

$$x_A(\phi) = \sum_{n=1}^{N} H_n(\phi) x_A^n$$

$$\Omega_A(\phi) = \sum_{n=1}^{N} H_n(\phi) \Omega_A^n$$

where x_A^n and Ω_A^n are values at node n, and $H_n(\phi)$ are polynomials of order (N-1).

The cross-sectional deformation is interpolated by Fourier series similar to those used in the earlier elements, but now including warping deformation:

$$x_{o3} = \sum_{n=1}^{\hat{N}} \sum_{m=2}^{M} \hat{H}_n(\phi) Q_{sym}(m\theta) u_{nm}^i + \sum_{n=1}^{\hat{N}} \sum_{m=2}^{M} \hat{H}_n(\phi) Q_{asym}(m\theta) u_{nm}^o$$

$$w = \sum_{n=1}^{\hat{N}} \hat{H}_n(\phi) w_{no} + \sum_{n=1}^{\hat{N}} \sum_{m=1}^{M} \hat{H}_n(\phi) Q_{sym}(m\theta) w_{nm}^i + \sum_{n=1}^{\hat{N}} \sum_{m=2}^{M} \hat{H}_n(\phi) Q_{asym}(m\theta) w_{nm}^o$$

$$v = \sum_{n=1}^{\hat{N}} \sum_{m=2}^{M} \hat{H}_n(\phi) Q_{sym}(m\theta) v_{nm}^i + \sum_{n=1}^{\hat{N}} \sum_{m=2}^{M} \hat{H}_n(\phi) Q_{asym}(m\theta) v_{nm}^o$$

$$\gamma_{o3} = \sum_{n=1}^{\hat{N}} \hat{H}_n(\phi) \gamma_{no} + \sum_{n=1}^{\hat{N}} \sum_{m=1}^{M} \hat{H}_n(\phi) Q_{sym}(m\theta) \gamma_{nm}^i + \sum_{n=1}^{\hat{N}} \sum_{m=2}^{M} \hat{H}_n(\phi) Q_{asym}(m\theta) \gamma_{nm}^o$$

where $\hat{H}(\phi)$ are polynomials of the same or lower order as the H_n used in the beam interpolation and

$$Q_{sym}(m\theta) = \begin{cases} \cos m\theta & m \text{ even} \\ \sin m\theta & m \text{ odd} \end{cases}$$

$$Q_{asym}(m\theta) = \begin{cases} \sin m\theta & m \text{ even} \\ \cos m\theta & m \text{ odd} \end{cases}$$

The superscripts i and o denote in-plane and out-of-plane ovalisation deformation. Again Fourier terms are selected so as to omit rigid body modes.

The element strain-displacement relationships are evaluated according to a discrete Kirchoff formulation using the Koiter-Sanders generalised large displacement section strains at each point on the mid-surface. It is assumed that the warping deformation is small.

Three elements options are available in the current ABAQUS version. ELEMENT 31 is based on linear interpolation of strain along the axis of a linear beam. ELEMENT 31B simplifies the above, and a constant state of strain with respect to axial position (and no warping of the cross-section) is assumed; ELBOW 31B is in effect a constant bending element. ELEMENT 32 is the most complex of the elements and is based on quadratic interpolation of both beam and ovalisation modes.

ABAQUS offers the most complete and best behaved elbow elements currently available. All elements allow non-symmetric extensional ovalisation and through thickness effects, and the higher order elements allow inter-element compatibility. However, it is interesting to note that the most widely used element is the simplest: the constant bending ELEMENT 31B.

3.9 The See Beam-Shell Element.

In 1984 See presented a constant bending element based on an exact curved beam solution with superposed Von Karman ovalisation [3.23].

The element was intended for inclusion in a micro-computer piping analysis program, but the inefficient element formulation and large numerical integration scheme used in the programming meant that it was computationally too large for its intended use. The results presented in [3.23] indicate that the element performed reasonably well for in-plane analysis, but was poor in the case of out-of-plane loading. An alternative formulation of a similar element, including closed form integration of the element stiffness matrix, is presented in Chapter 5.

3.10 The de Melo and de Castro Semi-Membrane Element.

The most recent elbow element to appear in the literature is the semi-membrane ring element of de Melo and de Castro intended for in-plane linear elastic analysis only [3.23].

Semi-membrane theory is a simplified shell theory which postulates that the response of certain types of shells can be considered to consist of two parts. The main part of the response allows full membrane and bending action in one co-ordinate direction, but membrane action only in the other. Thus this part of the response can be said to be semi-momentless or semi-membrane, as bending stresses are supported on only one of the two co-ordinate directions. The remaining part of the response is an edge effect in the membrane direction. This dies out rapidly away from the edge and is regarded as mostly unimportant.

In the Von Karman strain model of an elbow, axial bending of the elbow wall is assumed to be negligible and, therefore, the Von Karman model may be regarded as a simple semi-membrane analysis. The Von Karman model introduces the further assumption that membrane strain is negligable in the circumferential direction and that all shear strains are also negligible.

In terms of element formulation, the Melo and Castro element can be regarded as an advanced Von Karman element, as the formulation superposes beam and shell responses. The beam model assumes axial inextensibility and plane sections remain-plane, so that the beam deformation can be fully defined by polynomial interpolation of rotational degrees of freedom at nodes located at the beam ends. Rotation of the beam centroid is interpolated linearly around the bend in terms of end node rotations.

The ovalisation and warping modes are interpolated around the cross-section by Fourier series, with linear interpolation of the Fourier coefficients between nodes. The Von Karman circumferential inextensibility assumption is invoked to relate radial and tangential ovalisation displacements.

The element stiffness matrix was integrated by hand for axial integrals and a trapezoidal rule with up to 50 subintervals used for circumferential integrals.

The element degree of freedom vector consists of nodal rotations and Fourier coefficients. The complete stiffness matrix is assembled at global level thus enforcing semi-membrane ovalisation and warping interaction; that is, edge effects are neglected. Each element has a total of $2+4n$ degrees of freedom, where n is the number of Fourier terms in the ovalisation and warping interpolations.

Results of analysis of a number of single bend problems are presented and they show good agreement with theoretical solutions by Ory and Wilczek [3.27], Wilczek's experimental results and semi-loof shell finite element analysis results.

The element is similar to the MARC Element 17 and TEDEL elements but, unlike these elements, is limited to in-plane bending only. The beam modes are restricted to pure bending and the elbow element is not directly compatible with standard straight beam elements.

The consistant adoption of a semi-membrane approach introduces a simplified way to include consistant interaction effects between elements. From the results presented in [3.24] this approach appears to be adequate for certain pipe elbow configurations, however, a more detailed investigation of a wider range of geometries is required in order to fully ascertain the effectiveness of the semi-membrane approach.

3.11 Discussion

The above review has shown that a number of piping elbow elements of varying complexity and capabilities have been presented in the literature. In general, the more complex elements give more accurate results but, in practice, the use of many of these elements has been limited by their high computing costs. The ring elements in particular offer no significant economic advantage over shell finite element descretisation of elbows.

Possibly the most significant factor to be considered when formulating an elbow element is treatment of end effects and the continuity of ovalisation between adjacent elements. In constant bending elements, such as MARC Element 17 and ABAQUS ELBOW 31B, ovalisation is assumed to be constant with respect to axial position along the element. This assumption considerably simplifies the ovalisation shell theory and significantly reduces the number of degrees of freedom of the element in comparison with more complex formulations. However, constant bending elements have been shown to give good agreement with more complex elbow models and experimental results for a range of geometries and therefore end effects and interelement ovalisation compatibility will be neglected in the simple elements presented in this thesis.

If end and interaction effects are neglected, the obvious approach to elbow element formulation is the simple beam-shell approach, in which the shell type ovalisation response of an elbow under bending is superposed on a beam bending model.

Several types of beam models have previously been used as the basis of elbow elements. A classical curved beam approach has been used in MARC Element 17 and the TEDEL element. This offers the advantage of a very simple beam model, but leads to compatibility problems and connection to straight beam elements can only be made by use of constraint equations.

In the ADINAP element an isoparametric beam formulation was used. This approach allows bends of non-circular geometry to be modelled but at the expense of greatly added complexity in the element formulation.

Thin wall beam theory has been used in several element formulations, namely the ABAQUS elements and See's element. The ABAQUS elements are based on an original beam model which, although complex in formulation, has proven very effective in practice. See's element is based on Vlasov's thin wall curved beam theory. The programmed element had little demontrated success, however the Vlasov model is attractive as a basis for elbow elements because if approached properly it is possible to formulate a closed form solution for the beam model. This dispenses with computationally expensive numerical integration, making the element financially attractive. For this reason Vlasov's theory is adopted as the basic beam model for the simple elements presented in this thesis.

Several ovalisation interpolation functions have been proposed in the literature but these essentially fall into two camps: Fourier series and polynomial functions.

Fourier series have been used in several formulations, for example the ADINAP, Ohtsubo and Watanabe, ABAQUS and TEDEL elements. In the ADINAP and TEDEL elements truncated even Fourier series were used. In the Ohtsubo and Watanabe and ABAQUS elements truncated complete series were used.

Polynomial functions have been used to interpolate ovalisation implicitly in MARC Element 17 (in which the cross-section is modelled by polynomial-based axisymmetric elements) and directly in the Kanarachos and Koutsides element. MARC Element 17 has been demonstrated to have acceptable accuracy for a large number of published analyses; however, it requires a larger number of

ovalisation degrees of freedom than constant bending Fourier ovalisation elements. Insufficient examples were given in [3.17] to critically assess the performance of the Kanarachos element.

In order to investigate the relative merits of Fourier and polynomial interpolation of ovalisation, both approaches will be investigated. This may be done by formulating a number of elbow elements based on the same beam model but with different ovalisation interpolation schemes. However, a simpler approach to the investigation is possible by considering the deformation of expansion bellows.

It was stated in Chapter 2 that the deformation mode of expansion bellows is essentially similar to the ovalisation deformation of an elbow under bending. It is therefore possible to investigate interpolation schemes for elbow ovalisation by applying such schemes to simple bellows analyses, which are simpler to formulate and require much less programming than elbow elements.

In the next chapter four semi-toroidal bellows finite element formulations are presented. The principle objective in formulating these elements is to investigate ovalisation interpolation. However, a secondary objective is to introduce the concept of bellows elements *per se*. Present piping analyses consider bellows to be straight beams with flexibility and stress correction factors. A bellows element based on the true bellows deformation mode removes the requirement for such correction factors and gives a more detailed and accurate stress analysis of the component.

3.12 References.

[3.1] H D HIBBITT, E P SORENSEN and P V MARCAL
"The Elastic-Plastic and Creep Analysis of Pipelines by Finite Elements," Proc. 2nd Int. Conf. on Pressure Vessel Technology, San Antonio (1973).

[3.2] H D HIBBITT
"Special Structural Elements for Piping Analysis," ASME Special Publ. 'Pressure Vessels and Piping: Analysis and Computers' (1974).

[3.3] P V MARCAL
"Elastic-Plastic Behaviour of Pipe Bends with In-Plane Bending," J. Strain Analysis, 1967, Vol. 2, No. 1. pp 86, (1967).

[3.4] L SOBEL
"In-Plane Bending of Elbows," Computers and Structures, Vol. 7, pp 701-715, (1977).

[3.5] H OHTSUBO and O WATANABE
"Flexibility and Stress Factors of Pipe Bends - An Analysis by the Finite Ring Method," J. Pressure Vessel Tech., Vol. 99, pp 281-290, (1977).

[3.6] H OHTSUBO and O WATANABE
"Stress Analysis of Pipe Bends by Ring Elements," J. Pressure Vessel Tech., Vol. 100, pp 112-122, (1978).

[3.7] K WASHIZU
Variational Methods in Elasticity and Plasticity, 2nd Edition, Pergamon Press, (1975).

[3.8] L LAZZERI
"An Elasto-plastic Elbow Element - Theory and Applications," Int. J. Pre. Ves. & Piping, Vol. 8, pp 197-213,1980.
Based on presentation, No. F 3/5 at SMiRT 5, (1979).

[3.9] H TAKEDA and K IWATA
"A Finite Element for Detailed Analysis of Piping Systems Considering End Effects," 56th Annual Meeting of the JSME, (1978). (In Japanese).

[3.10] H TAKEDA and K IWATA
"A New Finite Element for Structural Analysis of Piping Systems," Paper M5/5, proc. 5th Int. Conf. Struct. Mech. in Reactor Tech., Berlin, (1979).

[3.11] H TAKEDA, K IWATA and S ASAI
"Finite Elements for Detailed Stress Analysis of Piping Systems," JSME preprint No. 805, (1982). (In Japanese).

[3.12] K J BATHE and C A ALMEIDA
"A Simple and Effective Pipe Elbow Element-Linear Analysis," J. App. Mech., 1980, Vol. 47, pp 93, (1982).

[3.13] K J BATHE and C A ALMEIDA
"A Simple and Effective Pipe Elbow Element-Interaction Effects," J. App. Mech., 1982, Vol. 49, pp 165, (1980).

[3.14] C A ALMEIDA
"A Simple New Element for Linear and Non-Linear Analysis of Piping Systems," Engineers PhD thesis, Massachusetts Institute of Technology, (1982).

[3.15] V V NOVOZHILOV
"Thin Shell Theory, Translated by P G lowe, P Noordhoff Ltd., Gronigen, Netherlands, (1964).

[3.16] W CHRISTOPH MULLER
"Piping Analysis of Large Scale Experiments with ADINA and DAPSY," Computers and Structures, Vol. 26, no1/2, pp 111-121, (1987).

[3.17] A KANARACHOS and R N KOUTSIDES
"A New Approach to Shell Displacement in a Beam-Type Pipe Element," Paper B9/2, Proc. 8th Int. Conf. Struct. Mech. in Reactor Tech., Brussels, (1985).

[3.18] W B BICKFORD and B T STROM
"Vibration of Plane Curved Beams," J. Sound and Vibration, 39(2), pp 135-146, (1975).

[3.19] A MILLARD and A HOFFMANN
"Refined Inelastic Analysis of Piping Systems Using a Beam-Type Program," Paper M10/2, Proc. 6th Int. Conf. Struct. Mech. in Reactor Tech.,Paris, (1981).

[3.20] 'ABAQUS'
ABAQUS Theory Manual, Hibbitt, Karlssen and Sorensen Inc., (1981).

[3.21] H D HIBBITT and E K LEUNG
"An Approach to Detailed Inelastic Analysis of Thin-Walled Pipelines," ASME App. Mech. Div. Pub. AMD-48, pp 83-118, (1981).

[3.22] 'ABAQUS'
ABAQUS Theory Manual, Hibbitt, Karlssen and Sorensen Inc., (1987).

[3.23] P S SEE
"The Analysis of Piping Systems on a Microcomputer Using Pipe Bend Elements," Engineers PhD Thesis, Strathclyde University, (1984).

[3.24] F J M Q de MELO and M S T de CASTRO
"A Semi-membrane Ring Element for the Linear Elastic Stress Analysis of Pipe Elbows under In-plane Bending," Int. J. Pres. Ves. & Piping Vol. 38, pp 309-319, (1989).

[3.25] E L AXELRAD
"Elastic Tubes-Assumptions, Equations, Edge Conditions," Thin-Walled Structures, Vol. 3, pp 193-215, (1985).

[3.26] F A EMMERLING
"Nonlinear Bending of Curved Tubes,"
pp 175-191, Flexible Shells, Theory and Applications, E L Axelrad and F A Emmerling, Eds., Springer-Verlag, Berlin (1984)

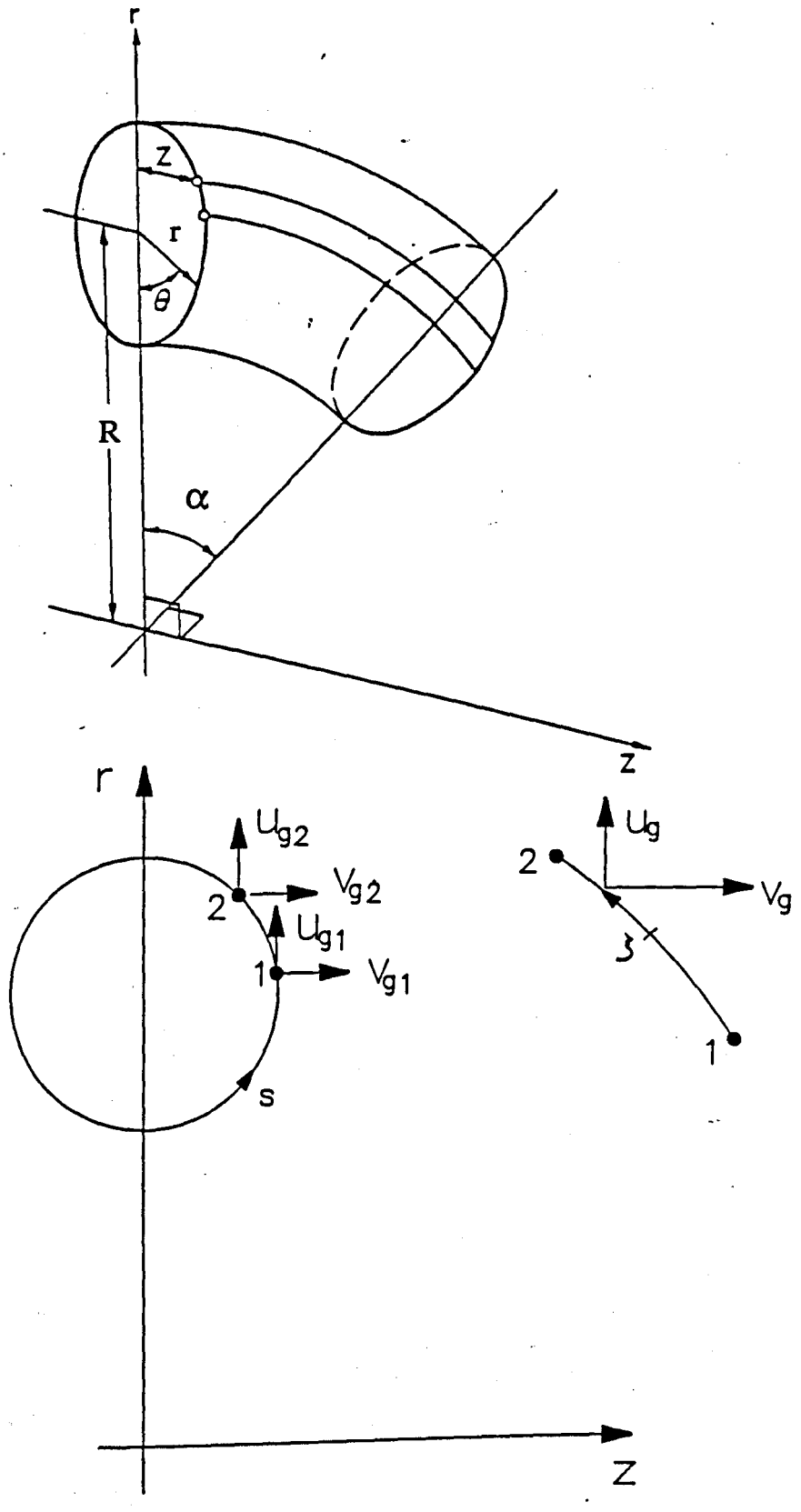


Figure 3.1 MARC Element 17 geometry, coordinate system and ovalisation displacements.

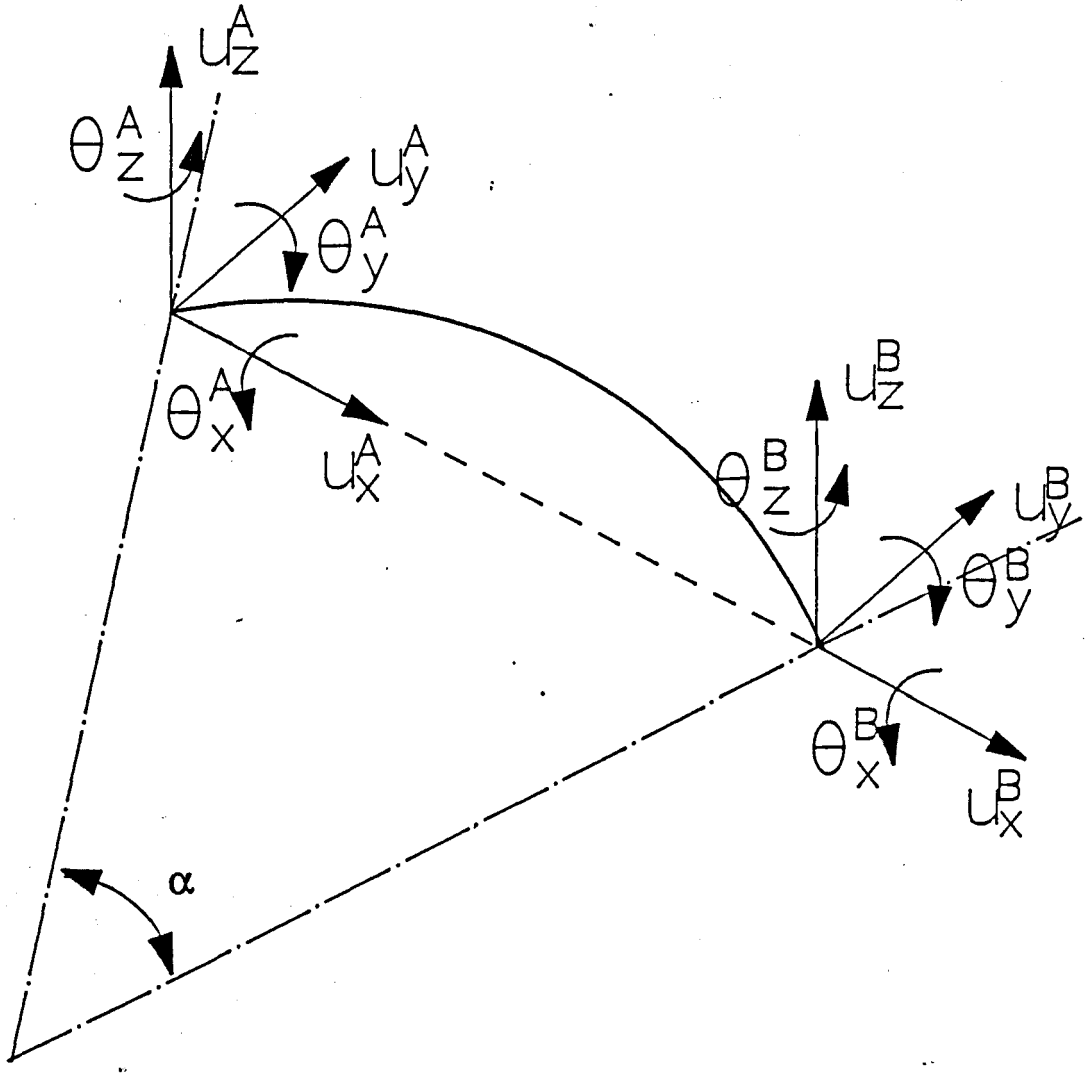


Figure 3.2 MARC Element 17 kinematically coupled degrees of freedom.

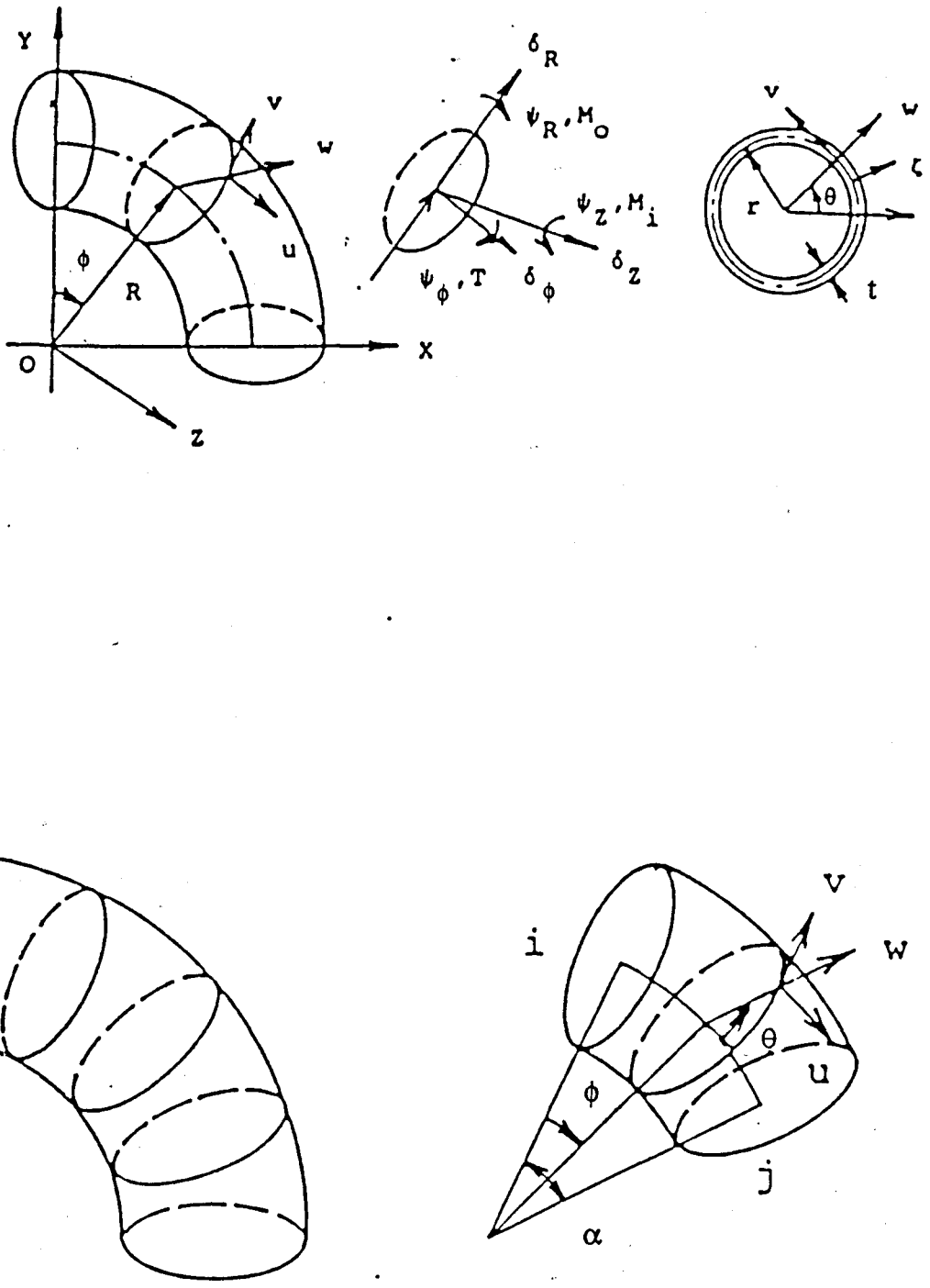


Figure 3.3 Ohtsubo and Watanabe ring element geometry, coordinate system and elbow descretisation. (Adapted from [3.5]).

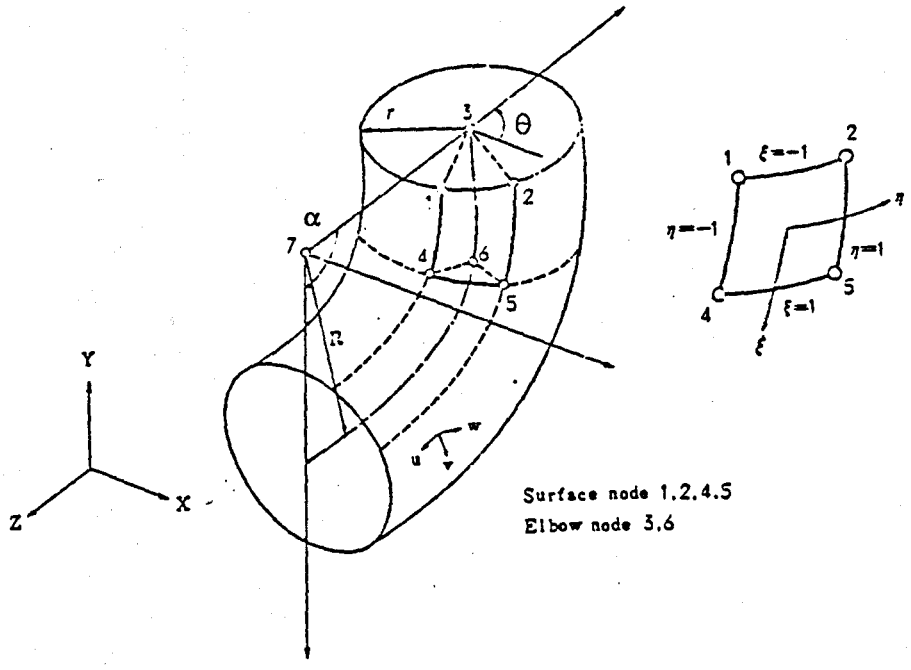


Figure 3.4 FINAS ELBOW 6 and ELBOW 6R geometry and coordinate system. (Adapted from[3.10]).

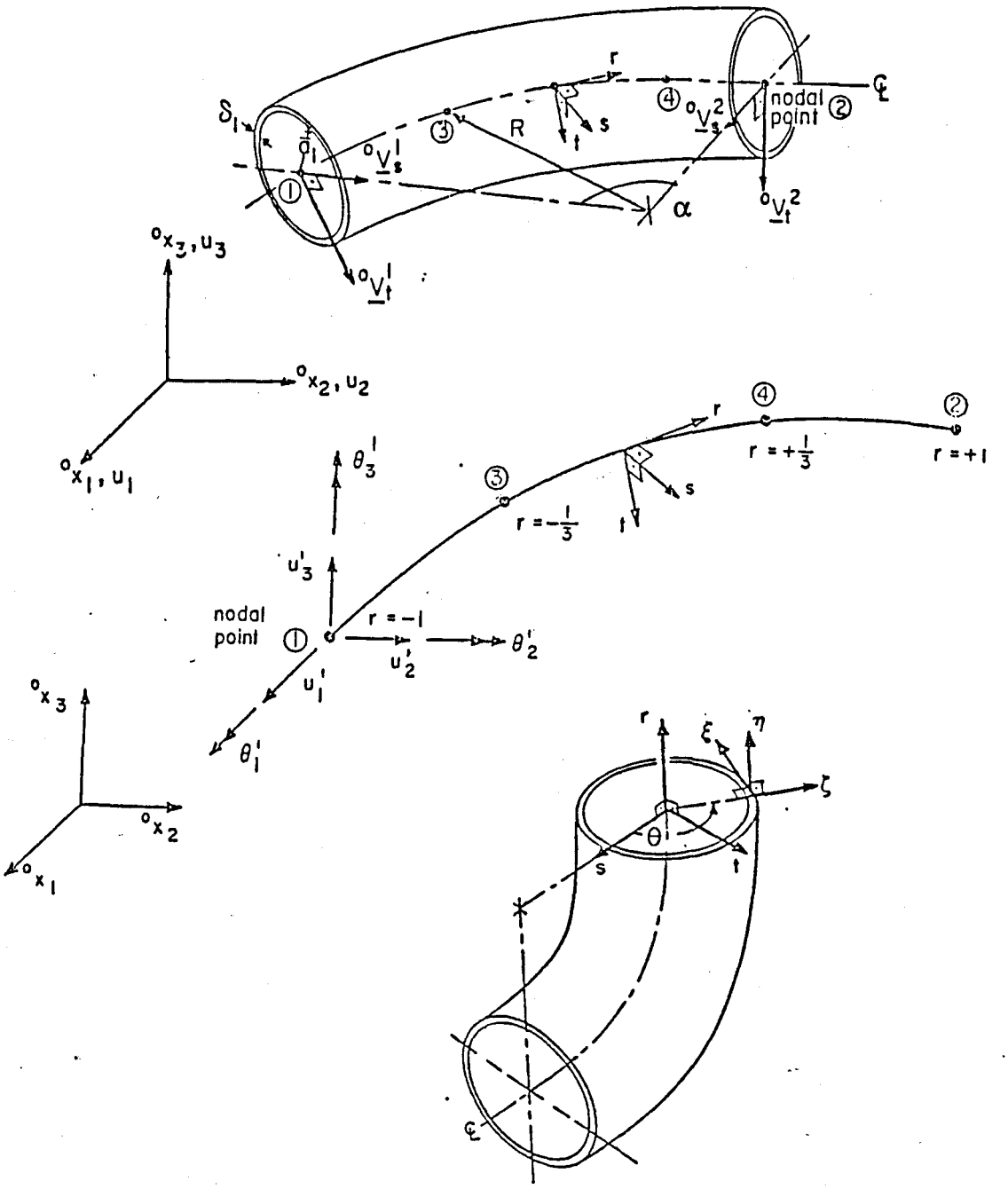
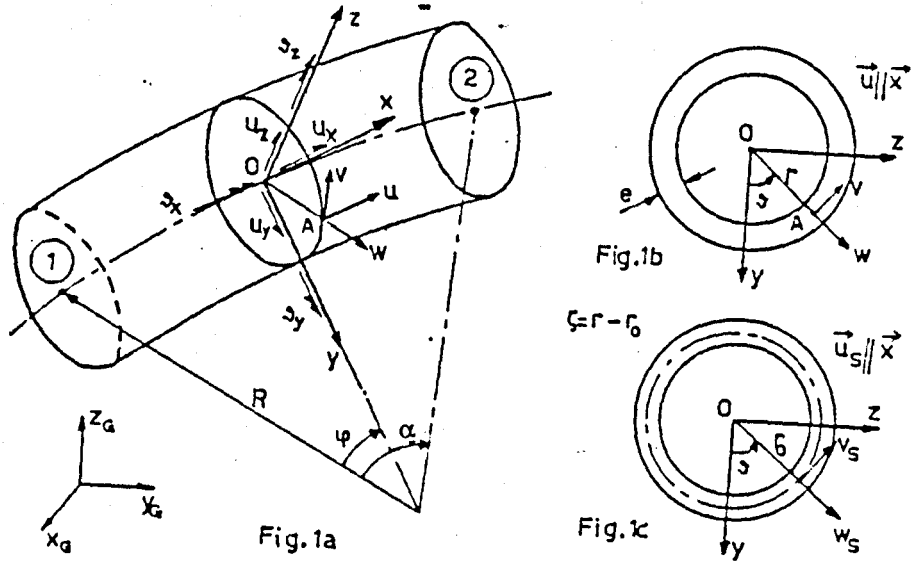
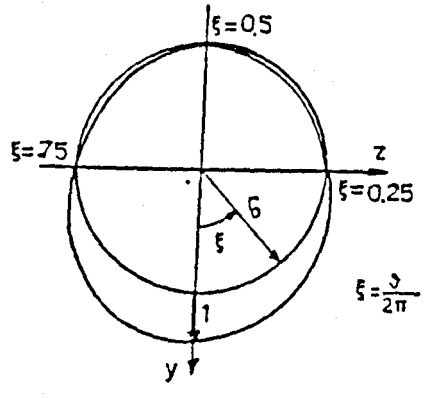


Figure 3.5 ADINA-P element geometry, coordinate systems and beam degrees of freedom. (Adapted from [3.14]).





$$N_1(\xi)=\begin{cases} -128\xi^4+144\xi^3-44\xi^2+1 & :0\leq\xi\leq 0.5 \\ -128(1-\xi)^4+144(1-\xi)^3-44(1-\xi)^2+1 & :0.5\leq\xi\leq 1 \end{cases}$$

Figure 3.6 Kanarachos and Koutsides element geometry, coordinate system and ovalisation interpolation scheme. (Adapter from [3.17]).

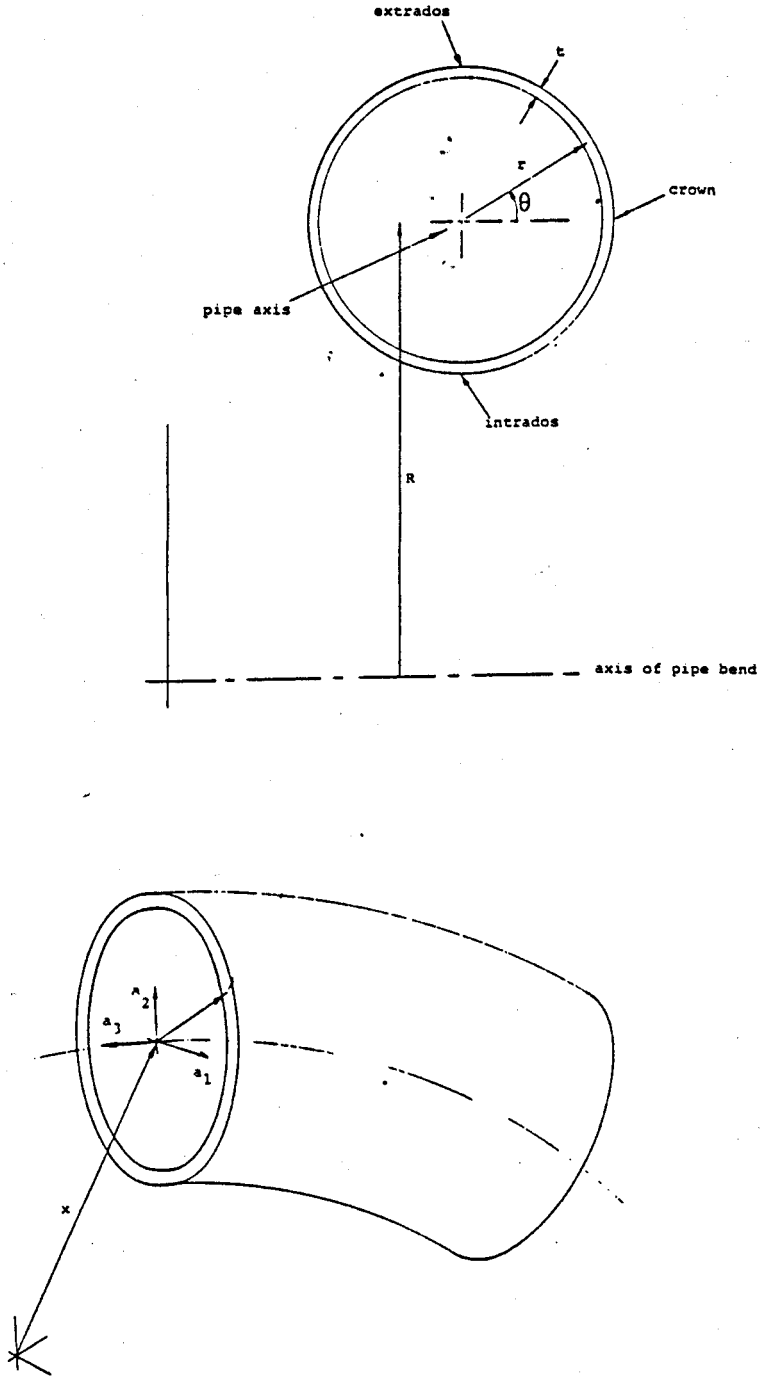


Figure 3.7 ABAQUS elbow element coordinate system. (Adapted from [3.21]).

CHAPTER 4.

SEMI-TOROIDAL BELLOWS FINITE ELEMENT FORMULATIONS.

4 SEMI-TOROIDAL BELLOWS FINITE ELEMENT FORMULATIONS.

Bellows expansion joints are often used to absorb thermal expansion in piping systems when lack of space prohibits the preferred use of expansion loops. Several types of expansion bellows are available commercially; however, in this thesis only bellows of semi-toroidal geometry, as shown in Figure 4.1, are considered.

When a semi-toroidal bellows is subject to an axial load it deforms by near inextensional bending of the bellows convolutions, as illustrated in Figure 4.2. The deformation of the convolution is similar to the ovalisation deformation of a piping elbow under bending. Therefore, because of the similarity in deformation modes, ovalisation models for pipe bends may be investigated by considering the simpler case of expansion bellows.

In this chapter four bellows finite element formulations are presented, primarily in order to study suitable ovalisation models for piping elbow elements. However, a secondary objective is also identified: that is, to formulate a bellows element capable of accurately modelling true bellows behaviour. In flexibility analysis, bellows are modelled using straight beam elements. As in the case of flexibility analysis elbow elements, such "bellows elements" do not represent the true behaviour of the component and, consequently, flexibility and stress intensification factors are required in order to evaluate approximately the bellows stiffness and stresses. An element based on the true bellows deformation would not require the use of such correction factors.

The bellows geometry and sign conventions used in this chapter are defined in Figure 4.1. Like a piping elbow, semi-toroidal bellows are shells of revolution. The radius from the bellows axis of symmetry to the shell mid-surface is a. The mean axisymmetric radius of the bellows is R, and the convolution mid-surface radius is r. The (hoop) angle about the axis of symmetry is ϕ. The position along the convolution in the axial direction is defined for each 90^{o} arc of by θ.

4.1 Element BEL1: A Bellows Element Based on the Laupa and Weil Analysis.

In [4.1], Laupa and Weil published an analysis of U-shaped bellows subject to axial force and internal pressure loading based on the potential energy method. Semi-toroidal geometry was considered as a special case of U-shape bellows in

[4.1], and a simplified version of this analysis (in which only axial force loading was considered) was later presented by Findlay and Spence in [4.2]. Bellows element BEL1 is based on this simplified semi-toroidal bellows analysis.

4.1.1 Bellows Deformation and Displacement Interpolation.

The co-ordinate system and boundary conditions of an outer quadrant of bellows is shown in Figure 4.3. The position of a point on the wall mid-surface is defined by angle $\eta = \frac{\pi}{2} - \theta$. Mid-surface deformation is described by displacement y parallel to the axis of symmetry, displacement z perpendicular to the axis in an outward direction and rotation ψ.

The boundary conditions of an outer 90^0 section of bellows are:

$$y = \psi = 0 \text{ at } \eta = 0 \tag{4.1}$$

$$z = 0 \text{ at } \eta = \frac{\pi}{2} \tag{4.2}$$

The normalised displacements of an arbitrary point P on the section are given by:

$$\frac{y}{r} = \int_0^{\eta} \psi \sin\eta d\eta \tag{4.3}$$

$$\frac{z}{r} = -\int_{\eta}^{\frac{\pi}{2}} \psi \cos\eta d\eta \tag{4.4}$$

where ψ is the tangent angle of the mid-surface. ψ is interpolated around the section of convolution by the Fourier series:

$$\psi = C_1 \sin\eta + \sum_1^p C_{2p} \sin 2p\eta \tag{4.5}$$

The axial displacement Y_1 at $\eta = \pi/2$ may be found from (4.3) by substituting (4.5) for ψ:

$$\frac{Y_1}{r} = \int_0^{\frac{\pi}{2}} \left(C_1 \sin^2\eta + \sum_1^p C_{2p} \sin\eta \sin 2p\eta \right) d\eta$$

which may be written:

$$\frac{Y_1}{r} = \{B_n\}\{C_n\} \tag{4.6}$$

that is

$$\{B_n\} = \int_0^{\frac{\pi}{2}} \left(\sin^2\eta, \sum_1^p \sin\eta \sin 2p\eta \right) d\eta$$

and

$$\{C_n\} = \left\{ C_1, \sum_1^p C_{2p} \right\}^T$$

Integrating $\{B_n\}$ gives:

$$\{B_n\} = \left\{ \frac{\pi}{4} \quad \frac{2}{3} \quad -\frac{4}{15} \quad \frac{6}{35} \ldots\ldots \right\}$$

4.1.2 Formulation of the stiffness matrix.

The values of the coefficients C_n in the shape function are found by applying the principle of minimum potential energy to the bellows.

The potential energy, V, of the bellows is defined by the expression:

$$V = U - W$$

where U is the strain energy and W the work done by the axial load.

The state of strain in the bellows is assumed to be direct strain in the axisymmetric (hoop) direction and inextensional bending of the convolution wall in the axial direction. Thus the total strain energy in a quarter section of bellows consists of two components: membrane energy due to the direct hoop strain and bending energy due to the bending of the convolution wall. That is:

$$U = U_m + U_b$$

The membrane strain energy of a differential element of shell is given by the expression [4.3]:

$$dU_m = \frac{1}{2} N_\phi \epsilon_\phi r a d\phi d\eta$$

where the circumferential stress resultant N_ϕ is given by:

$$N_\phi = \frac{Et}{1-\nu^2}\epsilon_\phi$$

The membrane strain energy is obtained by integrating the elemental energy equation through the volume. From the long radius assumption R>>r, radius a is assumed constant such that a=R. Noting that ϵ_ϕ is constant with respect to ϕ, integration with respect to ϕ results in the following expression for membrane strain energy:

$$U_m = \frac{Et}{1-\nu^2}\pi R r \int_0^{\frac{\pi}{2}} \epsilon_\phi^2 d\eta$$

where

$$\epsilon_\phi = \frac{z}{a} = \frac{z}{r}\frac{r}{a} = -\frac{r}{R} \int_\eta^{\frac{\pi}{2}} \psi \cos\eta d\eta \qquad (4.7)$$

Substituting for (z/r) from (4.4), U_m becomes:

$$U_m = \frac{Et}{1-\nu^2}\pi R r \left(\frac{r}{R}\right)^2 \int_0^{\frac{\pi}{2}} \hat{\epsilon}_\phi^2 d\eta \qquad (4.8)$$

where

$$\hat{\epsilon}_\phi = -\frac{1}{2}\left\{C_1 \cos^2\eta + \sum_1^p C_{2p} \frac{\cos(2p+1)\eta}{2p+1} + \frac{\cos 2(p-1)\eta}{2p-1}\right\}$$

Adopting the identities:

$$\lambda = \frac{Rt}{r^2} \qquad \text{and} \qquad I = \frac{\pi R t^3}{6}$$

equation (4.8) may be written:

$$U_m = \frac{6EI}{(1-\nu^2)r} \int_0^{\frac{\pi}{2}} \left(\frac{\hat{\epsilon}_\phi}{\lambda}\right)^2 d\eta \qquad (4.9)$$

An expression for bending strain energy is obtained in a similar manner from the differential bending energy equation:

$$dU_b = \frac{1}{2} M_\theta k_\theta r a d\phi d\eta \qquad M_\theta = \frac{Et^3}{12(1-\nu^2)} k_\theta$$

and

$$k_\theta = \frac{\psi'}{r} = \frac{1}{r}\left\{ C_1 \cos\eta + \sum_1^p C_{2p} 2p \cos 2p\eta \right\} \tag{4.10}$$

Thus the bending strain energy is:

$$U_b = \frac{6EI}{(1-\nu^2)r} \int_0^{\frac{\pi}{2}} \left(\frac{\hat{k}_\theta^2}{12} \right) d\eta \tag{4.11}$$

where

$$\hat{k}_\theta = C_1 \cos\eta + \sum_1^n C_{2n} 2n \cos 2n\eta$$

The total work done on the section of bellows by the axial force is this force P multiplied by the displacement Y_1. Thus the potential energy equation of the quarter section is:

$$V = U_m + U_b - PY_1$$

which, upon substituting (4.9) and (4.11) becomes:

$$V = \frac{6EI}{(1-\nu^2)r} D - PY_1 \tag{4.12}$$

where:

$$D = \int_0^{\frac{\pi}{2}} \left(\left(\frac{\hat{\epsilon}_\phi}{\lambda} \right)^2 + \frac{\hat{k}_\theta^2}{12} \right) d\eta \tag{4.13}$$

Equation (4.13) was obtained in a closed form using the symbolic algebra computer program SMP (Symbolic Manipulation Program) [4.4].

In equation (4.6), Y_1 is expressed in terms of the interpolation function coefficients C_n. The values of these coefficients are obtained by minimising the potential energy of the section with respect to each coefficient in turn:

$$\frac{\partial V}{\partial C_n} = 0 \qquad \text{for } n = 1,2,4,6,8....$$

It is seen from (4.12) that the potential energy is a function of D and also of Y_1, both of which are functions of the coefficients C_n. Therefore, partial differentiation of (4.13) with respect to C_n gives:

$$\frac{\partial V}{\partial C_n} = \frac{6EI}{(1-\nu^2)r}\frac{\partial D}{\partial C_n} - P\frac{\partial Y_1}{\partial C_n} = 0$$

or:

$$\frac{\partial D}{\partial C_n} = \frac{P(1-\nu^2)r}{6EI}\frac{\partial Y_1}{\partial C_n} \qquad (4.14)$$

The relationship between Y_1 and $\{C_n\}$ is defined by (4.6) which, upon differentiating with respect C_n, gives:

$$\frac{\partial Y_1}{\partial C_n} = \{B_n\}^T r \qquad (4.15)$$

The relationship between the energy term D and $\{C_n\}$ is more complex, and is defined in equation (4.14). Integrating this equation and subsequently differentiating with respect to the coefficients yields a system of equations of the form:

$$\frac{\partial D}{\partial C_n} = = [A]\{C_n\} \qquad (4.16)$$

where [A] is a symmetric matrix of constants for a given value of λ, the upper triangle of which is:

$$A_{11} = 0.1309 + \left(\frac{1.178097}{4\lambda^2}\right) \qquad A_{22} = 0.523599 + \left(\frac{2.871682}{4\lambda^2}\right)$$

$$A_{12} = 0.111111 + \left(\frac{1.659685}{4\lambda^2}\right) \qquad A_{23} = \frac{0.213333}{4\lambda^2}$$

$$A_{13} = -0.044444 + \left(\frac{0.25418}{4\lambda^2}\right) \qquad A_{24} = \frac{0.016327}{4\lambda^2}$$

$$A_{14} = 0.028571 + \left(\frac{0.001814}{4\lambda^2}\right) \qquad A_{25} = \frac{0.014109}{4\lambda^2}$$

$$A_{15} = -0.021164 - \left(\frac{0.000641}{4\lambda^2}\right) \qquad A_{26} = \frac{-0.012489}{4\lambda^2}$$

$$A_{16} = -0.016835 + \left(\frac{0.000283}{4\lambda^2}\right)$$

$$A_{33} = 2.094395 + \left(\frac{0.173873}{4\lambda^2}\right) \qquad A_{44} = 4.712389 + \left(\frac{0.070646}{4\lambda^2}\right)$$

$$A_{34} = \frac{0.059259}{4\lambda^2} \qquad A_{45} = \frac{0.028132}{4\lambda^2}$$

$$A_{35} = \frac{0.017316}{4\lambda^2} \qquad A_{46} = \frac{0.011005}{4\lambda^2}$$

$$A_{36} = 0.0$$

$$A_{55} = 8.37758 + \left(\frac{0.38751}{4\lambda^2}\right) \qquad A_{66} = 13.089969 + \left(\frac{0.024577}{4\lambda^2}\right)$$

$$A_{56} = \frac{0.016433}{4\lambda^2}$$

The vector $\{C_n\}$ is as previously defined.

Substituting (4.15) and (4.16) into (4.14) and rearranging gives:

$$\{C_n\} = \frac{(1-\nu^2)r^2}{6EI} [A]^{-1} \{B_n\}^T P \qquad (4.17)$$

which defines $\{C_n\}$ in terms of [A] and $\{B_n\}$. On obtained $\{C_n\}$ from (4.17), the axial displacement Y_1 of the section of bellows due to load P is given by equation (4.6):

$$Y_1 = r\{B_n\}\{C_n\} \qquad (4.18)$$

Substituting for $\{C_n\}$ from (4.17):

$$Y_1 = \frac{(1-\nu^2)r^3}{6EI} \{B_n\} [A]^{-1} \{B_n\}^T P \qquad (4.19)$$

Comparing (4.19) with the general structural stiffness equation **F = Kd**, it is seen that the axial stiffness of the quarter toroidal section is given by:

$$K_q = \frac{6EI}{(1-\nu^2)r^3 \{B_n\} [A]^{-1} \{B_n\}^T} \qquad (4.20)$$

From the long radius assumption, it may be assumed that a similar inner quadrant of bellows has the same axial stiffness as an outer quadrant [4.1]. It is therefore possible to define the complete bellows stiffness from (4.20). The stiffness of one convolution of bellows, that is four quarter sections, is the series sum of four such stiffnesses:

$$K_{corr} = \frac{K_q}{4}$$

Similarly, the stiffness of a bellows comprising of N convolutions is:

$$K_b = \frac{K_q}{4N} \qquad (4.21)$$

Thus the stiffness equation of a one dimensional bellows element is:

$$[K] = \begin{bmatrix} K_b & -K_b \\ -K_b & K_b \end{bmatrix}$$

Expanding into three dimensions gives:

$$[K] = \begin{bmatrix} K_b & 0 & 0 & -K_b & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ -K_b & 0 & 0 & K_b & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \qquad (4.22)$$

4.1.3 Bellows flexibility factor.

Bellows flexibility is generally discussed in terms of a flexibility factor, which is defined as the ratio of bellows displacement to comparable pipe displacement for a given load P. The straight pipe is considered to be of length r, the bellows radius.

The flexibility factor is defined mathematically by the expression:

$$Flex = \frac{3EI}{(1-\nu^2)r^3}\frac{Y_1}{P}$$

Comparing this expression with (4.21), it is found that the flexibility factor may also be expressed as:

$$Flex = \frac{\{B_n\}[A]^{-1}\{B_n\}^T}{2} \tag{4.23}$$

4.1.4 Stress and Strain.

In section 2.1.2 the state of strain in the bellows was assumed to be circumferential direct strain and meridional bending.

From (4.10) the axial strain (due to bending) at position ζ, positive outwards from the wall mid-surface, is given by

$$\epsilon_\theta = \frac{\zeta}{r}\left\{ C_1\cos\eta + \sum_1^p C_{2p}\ 2p\cos 2p\eta \right\}$$

The hoop strain is obtained by integrating (4.7), and is given by

$$\epsilon_\phi = -\frac{r}{2R}\left\{ C_1\cos^2\eta + \sum_1^p C_{2p}\frac{\cos(2p+1)\eta}{2p+1} + \frac{\cos 2(p-1)\eta}{2p-1} \right\}$$

These equations are expanded and arranged into the matrix equation:

$$\{\epsilon\} = [B]\{C_n\} \tag{4.24}$$

where

$$\{\epsilon\} = \{\epsilon_\phi \quad \epsilon_\theta\}^T \qquad \{C_n\} = \{C_1 \quad C_2 \quad C_4 \ \ldots\ldots\}^T$$

(4.24) is the element strain displacement equation and the coefficients C_n are effectively nodeless degrees of freedom. The element stresses are defined by the stress strain relationship:

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

where [D] is the constitutive matrix .

$$[D] = \frac{E}{(1-\nu^2)}\begin{bmatrix} 1 & \nu \\ \nu & 1 \end{bmatrix}$$

4.1.5 BEL1 Programming.

The BEL1 theory was programmed to evaluate bellows flexibility factor and normalised stress distribution in the FORTRAN routine BEL1PROG.FOR, given in Appendix 3.1. Bellows geometry and material properties are entered interactively in response to screen prompts. Results are written to the screen.

The BEL1 formulation was programmed as a "user element" for the ANSYS finite element program [4.5]. ANSYS is a powerful general purpose finite element package which allows the user to define elements which will interact with the program pre-processing, solution and post-processing routines in a manner similar to standard ANSYS elements. ANSYS user element programming is discussed in detail in Appendix 1. The BEL1 user element is based on the BEL1PROG routine. The user element source code is given in Appendix 4.1. The element has the following parameters:

i). 2 node straight line element.

ii) 6 degrees of freedom: three translations per node.

iii) 4 input constants: mean axisymmetric radius R, convolution radius r, thickness t, and number of complete corrugations N (integer).

iv) Input material properties : Elastic modulus, Poisson's ratio, density.

The geometry of the bellows is valid on condition:

a) Length > 0.0

b) Length is an integer multiple of 4 x convolution radius. (An error of 1% is allowed).

4.2 An Axisymmetric Shell Bellows Formulation.

In the remainder of this chapter three elements based on an axisymmetric shell bellows model are presented. The common formulation procedure is defined in section 4.2, and specific formulations for three elements based on the procedure developed in sections 4.3, 4.4 and 4.5.

4.2.1 Shell Theory: Strain-Displacement Equations.

An outer quarter section of axisymmetric bellows shell, as shown in Figure 4.4, is considered. The position of a point along the convolution is defined by the angle θ. The displacement of the mid-surface of the section is described by tangential displacement v, radial displacement w, and rotation ψ.

In Chapter 3 the thin shell theory of Washizu was discussed in relation to elbow element formulation, with the strain-displacement equations for an axisymmetric shell under the Love-Kirchhoff hypothesis defined in equations (3.3 a-c). Assuming that the axial deformation of the bellows is inextensional bending of the convolution wall the axisymmetric strain displacement equations are obtained from equations (3.3) as:

$$\epsilon_\theta = \frac{-\zeta}{r^2}\left(\frac{d^2 w}{d\phi^2} + w\right) \tag{4.25}$$

$$\epsilon_\phi = \frac{1}{R}(w\sin\theta + v\cos\theta) \tag{4.26}$$

The inextensibility coupling condition is:

$$w = -\frac{dv}{d\theta} \tag{4.27}$$

and, under the Love-Kirchhoff hypothesis, the shell mid-surface rotation and displacements are related as in (3.2a):

$$\psi = \frac{1}{r}\left(\frac{dw}{d\theta} - v\right) \tag{4.28}$$

The above shell strain-displacement equations have previously been used for bellows analysis by Findlay and Spence in [4.2].

4.2.2 Interpolation.

The quarter section degrees of freedom are displacements w and v and rotation ψ at nodes located at the ends of the arc as shown in Figure 4.4. The element degree of freedom vector $\{\Delta\}$ is therefore:

$$\{\Delta\} = \{v_1 \quad w_1 \quad \psi_1 \quad v_2 \quad w_2 \quad \psi_2\}^T$$

Displacements w,v and ψ at a point θ around the element are defined in terms of a vector of unknown constants {a}. As w,v and ψ are related by (4.27) and (4.28), assuming a distribution for one intrinsically defines the distribution of the others in terms of the same constants. In general the displacement field is written:

$$\{u\} = [\Theta]\{a\} \tag{4.29}$$

where

$$\{u\} = \{v \quad w \quad \psi\}^T \qquad \{a\} = \{a_1 \quad a_2 \quad a_3 \quad a_4 \quad a_5 \quad a_6\}^T$$

and $[\Theta]$ is a matrix of functions of position θ.

{a} is defined in terms of the nodal displacements Δ by applying the boundary conditions to (4.29): that is

$$\text{at}\ \theta = 0 : v = v_1, \quad w = w_1, \quad \psi = \psi_1$$

$$\text{at}\ \theta = \frac{\pi}{2} : v = v_2, \quad w = w_2, \quad \psi = \psi_2$$

which yields the equation

$$\{\Delta\} = [C]\{a\} \tag{4.30}$$

where [C] is a 6x6 matrix of constants. The unknown coefficients {a} are defined in terms of the element degrees of freedom by inverting (4.30):

$$\{a\} = [C]^{-1}\{\Delta\} \tag{4.31}$$

Displacement at a point Θ on the mid-surface is obtained in terms of the degrees of freedom by substituting {a} from (4.31) into (4.29). However, it is possible to reduce the size of the element matrices by invoking the specific boundary conditions of the element for the bellows application. In order to construct a complete bellows from a single element stiffness matrix by series stiffness addition, the same boundary conditions as used in BEL1 are invoked:

$$v_1 = 0 \quad v_2 = 0 \quad \psi_2 = 0$$

Thus (4.31) is reduced to

$$\{a\} = [G]\{d\} \tag{4.32}$$

where {a} is as before, {d} is the reduced degree of freedom vector:

$$\{d\} = \{w_1 \quad \psi_1 \quad w_2\}$$

and [G] is a 6x3 matrix of constants comprising of columns 2,3 and 5 of $[C]^{-1}$.

The local and nodal displacements are now related by substituting {a} from (4.32) into (4.29) to give:

$$\{u\} = [\Theta][G]\{d\} \tag{4.33}$$

or

$$[N] = [\Theta][G] \tag{4.34}$$

where [N] is the element shape function matrix.

4.2.3 Strain Displacement.

The interpolation equation (4.33) may be written as three separate equations by partitioning the $[\Theta]$ matrix as follows:

$$\begin{Bmatrix} v \\ w \\ \psi \end{Bmatrix} = \begin{bmatrix} \{\Theta_1\} \\ \{\Theta_2\} \\ \{\Theta_3\} \end{bmatrix} [G]\{d\}$$

Thus the displacements can be defined as:

$$v = \{\Theta_1\}[G]\{d\} \tag{4.35a}$$

$$w = \{\Theta_2\}[G]\{d\} \tag{4.35b}$$

$$\psi = \{\Theta_3\}[G]\{d\} \tag{4.35c}$$

The element strain displacement equation is

$$\{\epsilon\} = [B]\{d\} \tag{4.36}$$

where

$$\{\epsilon\} = \{\epsilon_\theta \quad \epsilon_\phi\}^T \qquad\qquad \{d\} = \{v \quad w \quad \psi\}^T$$

and [B] is the strain-displacement matrix obtained by substituting equations (4.35) into (4.25) and (4.26). Noting that [G] and {d} are arrays of constants, (4.35b) gives

$$\frac{d^2 w}{d\theta^2} = \left\{\frac{d^2}{d\theta^2}\{\Theta_2\}\right\}[G]\{d\} \tag{4.37}$$

Thus the axial strain is:

$$\epsilon_\theta = \frac{-\zeta}{r^2}\left\{\frac{d^2}{d\theta^2}\{\Theta_2\} + \{\Theta_2\}\right\}[G]\{d\} \tag{4.38}$$

and the hoop strain is

$$\epsilon_\phi = \frac{1}{R}\{\{\Theta_2\}\sin\theta + \{\Theta_1\}\cos\theta\}[G]\{d\} \tag{4.39}$$

where ζ is the element through thickness co-ordinate, positive radially outwards from mid-surface of the shell.

Arranging the strain equations (4.38) and (4.39) into the form of (4.36), the strain displacement vector is given by:

$$[B] = [A][G] \tag{4.40}$$

where

$$[A] = \begin{bmatrix} \dfrac{-\zeta}{r^2}\left(\dfrac{d^2}{d\theta^2} + 1\right)\{\Theta_2\} \\ \dfrac{1}{R}(\{\Theta_2\}\sin\theta + \{\Theta_1\}\cos\theta) \end{bmatrix}$$

4.2.4 Quarter Section Stiffness matrix.

In Chapter 2 it was shown that the stiffness matrix of an element is given by the equation

$$[K] = \int_V [B]^T [D][B] dV \tag{4.41}$$

where [B] is the strain displacement matrix and [D] is the element constitutive matrix:

$$[D] = \frac{E}{1-\nu^2}\begin{bmatrix} 1 & \nu \\ \nu & 1 \end{bmatrix} = \frac{E}{1-\nu^2}[D_1] \tag{4.42}$$

Therefore (4.41) may be written

$$[K] = \int_0^{2\pi} \int_{-\frac{t}{2}}^{\frac{t}{2}} \int_0^{\frac{\pi}{2}} [B]^T [D][B] r R \, d\phi \, d\zeta \, d\theta$$

However, it is possible to simplify the above equation by considering the integration limits more closely. As [B] is constant with respect to ϕ, this simplifies to

$$[K] = \frac{2\pi E r R}{1-\nu^2} \int_{-\frac{t}{2}}^{\frac{t}{2}} \int_0^{\frac{\pi}{2}} [B]^T [D_1][B] \, d\zeta \, d\theta$$

where $$[D_1] = \begin{bmatrix} 1 & \nu \\ \nu & 1 \end{bmatrix}$$

Further, as the circumferential strain is constant through-thickness, the integral of products of axial and hoop strains with respect to ζ is zero, as

$$\int_{-\frac{t}{2}}^{\frac{t}{2}} \zeta \, d\zeta = [\zeta^2]_{-\frac{t}{2}}^{\frac{t}{2}} = 0$$

Such terms can therefore be omitted from stiffness calculations and the stiffness matrix equation reduces to

$$[K] = \frac{2\pi E r R}{1-\nu^2} \int_{-\frac{t}{2}}^{\frac{t}{2}} \int_{0}^{\frac{\pi}{2}} [B]^T [B] d\zeta d\theta \qquad (4.43)$$

4.2.5 The Element Stiffness Matrix.

In order to assemble a bellows element by series stiffness addition of single 90° convolutions, only the axial stiffness of a single convolution is required. This is obtained by applying the static condensation procedure defined in Chapter 2 to the quarter section stiffness matrix. The stiffness matrix is partitioned as follows, and degrees of freedom $\{\psi_1 \quad w_1\}$ statically condensed from the equation:

$$\begin{Bmatrix} F \\ - \\ 0 \\ 0 \end{Bmatrix} = \begin{bmatrix} K_{11} & | & K_{12} & K_{13} \\ - & + & - & - \\ K_{21} & | & K_{22} & K_{23} \\ K_{31} & | & K_{32} & K_{33} \end{bmatrix} \begin{Bmatrix} w_1 \\ - \\ \psi_1 \\ w_2 \end{Bmatrix} \qquad (4.44)$$

The axial stiffness equation is thus obtained in the form

$$F = K_q w_1$$

where K_q is the axial stiffness of the quarter section of bellows. The element stiffness matrix is obtained from the axial stiffness as in element BEL1. The stiffness of a single convolution is given by

$$K_{corr} = \frac{K_q}{4}$$

and the stiffness of a bellows of N convolutions is:

$$K_b = \frac{K_q}{4N}$$

In finite element form, the stiffness equation of a one dimensional bellows element is therefore

$$[K] = \begin{bmatrix} K_b & -K_b \\ -K_b & K_b \end{bmatrix}$$

which is expanded into three dimensions as in BEL1.

4.2.6 Stress and Strain.

The state of strain at a point in the section of bellows is defined by the strain displacement equation, (4.36):

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

The stresses are evaluated from the constitutive relationship of the section:

$$\{\sigma\} = [D]\{\epsilon\} = [D][B]\{d\}$$

where $\{\sigma\} = \{\sigma_\theta \quad \sigma_\phi\}^T$ and [D] is the constitutive matrix (4.42).

4.3 Element BEL2: Trigonometric Interpolation.

Element BEL2 is the first of three bellows elements based on the above formulation. The element interpolates radial displacement in terms of the five term complete Fourier series:

$$w = a_1 - \sum_{n=2}^{5} a_n \sin n\theta$$

A similar interpolation function has previously been used by Findlay and Spence in [4.2], in conjunction with the strain-displacement equations defined in section 4.2, although the series was not limited to a set number of terms as it is here.

The radial displacement, v, is obtained by integrating the inextensibility equation (4.27) to give

$$v = -\int w d\theta = -a_1\theta - \sum_{n=2}^{5} \frac{a_n}{n} \cos n\theta + a_6$$

where a_6 is a constant of integration.

The rotation ψ of the mid-surface is related to v and w by the Love-Kirchhoff hypothesis as defined in (4.28). Thus the displacement field of the element is defined by the equation

$$\{u\} = [\Theta]\{a\}$$

where

$$\{u\} = \{v w \psi\}^T \qquad \{a\} = \{a_1 a_2 a_3 a_4 a_5 a_6\}^T$$

and

$$[\Theta] = \begin{bmatrix} -\theta & -\cos\theta & -\frac{1}{2}\cos 2\theta & -\frac{1}{3}\cos 3\theta & -\frac{1}{4}\cos 4\theta & 1 \\ 1 & -\sin\theta & -\sin 2\theta & -\sin 3\theta & -\sin 4\theta & 0 \\ \frac{\theta}{r} & 0 & -\frac{3}{2r}\cos 2\theta & -\frac{8}{3r}\cos 3\theta & -\frac{15}{4r}\cos 4\theta & -1 \end{bmatrix}$$

applying the boundary conditions yields

$$\{\Delta\} = [C]\{a\}$$

which is solved for {a} by inverting the equation. Applying the specific bellows boundary conditions gives

$$\{a\} = [G]\{d\}$$

where {a} is as before, {d} is the reduced degree of freedom vector and [G] is a 6x3 matrix of constants comprising of columns 2,3 and 5 of $[C]^{-1}$:

$$[G] = \begin{bmatrix} 1 & 0 & 0 \\ 3.46238898 & \frac{3}{4}r & \frac{5}{4} \\ -2.71238898 & -r & -2 \\ 2.46238898 & \frac{3}{4}r & \frac{9}{4} \\ -1.35619449 & \frac{1}{2}r & -1 \\ 2.587942194 & \frac{3}{8}r & \frac{3}{4} \end{bmatrix}$$

Thus the shape function matrix [N] of the element is fully defined:

$$[N] = [\Theta][G]$$

Applying the strain-displacement procedure of section 4.2.3 defines the strain-displacement matrix:

$$[B] = [A][G]$$

where the elements of the [A] matrix are:

$$A_{11} = \frac{-\zeta}{r^2} \qquad A_{21} = \frac{1}{R}(\sin\theta - \theta\cos\theta)$$

$$A_{12} = 0 \qquad A_{22} = -\frac{1}{R}(\sin^2\theta + \cos^2\theta)$$

$$A_{13} = -3\frac{\zeta}{r^2}\sin 2\theta \qquad A_{23} = \frac{1}{R}\left(\sin 2\theta \sin\theta + \frac{1}{2}\cos 2\theta\cos\theta\right)$$

$$A_{14} = -8\frac{\zeta}{r^2}\sin 4\theta \qquad A_{24} = \frac{1}{R}\left(\sin 3\theta \sin\theta + \frac{1}{3}\cos 3\theta\cos\theta\right)$$

$$A_{15} = -15\frac{\zeta}{r^2}\sin 4\theta \qquad A_{25} = \frac{1}{R}\left(\sin 4\theta \sin\theta + \frac{1}{4}\cos 4\theta\cos\theta\right)$$

$$A_{16} = 0 \qquad A_{26} = \frac{1}{R}\cos\theta$$

4.3.1 BEL2 Programming

Two FORTRAN programs based on the above theory have been programmed. BEL2FLEX.FOR evaluates the bellows flexibility factor and BEL2STR.FOR evaluates the normalised stress distribution over a 90^o section of bellows. The FORTRAN code of these programs is given in Appendix 3.2

The 3x3 stiffness matrix of the bellows is obtained by integrating (4.43). This is done numerically, using a 5x3x3 Gaussian quadrature rule. The axial stiffness of the bellows is obtained by static condensation of the 3x3 matrix and the belows flexibility factor is evaluated according to the definition of Chapter 4.1.

4.4 Element BEL3: Polynomial Interpolation.

Polynomials are the most popular interpolation functions used in finite element formulations. Many polynomial interpolation schemes have been proposed for shell elements in the literature. These have often been assessed by investigating the behaviour of simpler two dimensional arch or curved beam elements, which may be regarded as the limiting case of a shell in which one of the area dimensions reduces to the order of the shell thickness.

Early investigation into polynomial interpolation in arch elements was carried out by Ashwell *et al* [4.6,4.7,4.8]. Combinations of polynomials up to cubics were investigated, however these elements generally proved to be ineffective. Similar work by Dawe based on higher order polynomials (up to quintics) [4.9,4.10] indicated that independent quintic interpolation for both radial and tangential displacement gave the most effective element, however, as noted in [4.9], such elements introduce "....some waviness in the [axial] force distribution ... in deep, thin, nearly inextensional applications". In [4.11] Meck presented the formulation of a curved beam based on coupled polynomial interpolation for tangential and radial displacement. This produced a well behaved element in which the specified coupling between the radial and tangential displacement explicitly ensured that the element could represent inextensible strain modes.

In this section Meck's quintic polynomial interpolation scheme is adopted as the basis of bellows element BEL3. The tangential displacement v is interpolated along the section by a quintic polynomial in terms of angle β, where $\theta = \beta + \frac{\pi}{4}$ as defined in Figure 4.5. Thus,

$$v = a_1 + a_2\beta + a_3\beta^2 + a_4\beta^3 + a_5\beta^4 + a_6\beta^5$$

The radial displacement is coupled to the tangential displacement by the inextensibility condition (4.27). Noting that

$$\frac{d}{d\theta} = \frac{d}{d\beta}$$

it follows from (4.27) and (4.28) that

$$w = -\frac{dv}{d\beta} \qquad \text{and} \qquad \psi = \frac{1}{r}\left(\frac{dw}{d\beta} - v\right)$$

Thus the displacement field function matrix $[\Theta]$ of (4.29) is found to be:

$$[\Theta] = \begin{bmatrix} 1 & \beta & \beta^2 & \beta^3 & \beta^4 & \beta^5 \\ 0 & -1 & -2\beta & -3\beta^2 & -4\beta^3 & -5\beta^4 \\ -\frac{1}{r} & -\frac{\beta}{r} & -\frac{(2+\beta^2)}{r} & -\frac{(6\beta+\beta^3)}{r} & -\frac{(12\beta^2+\beta^4)}{r} & -\frac{(20\beta^3+\beta^5)}{r} \end{bmatrix}$$

Considering the definition of the angle β, the general boundary conditions for element BEL3 are:

$$\beta = \frac{-\pi}{4}: \quad v = v_1, \quad w = w_1, \quad \psi = \psi_1$$

$$\beta = \frac{\pi}{4}: \quad v = v_1, \quad w = w_1, \quad \psi = \psi_1$$

The boundary conditions are applied and {a} obtained as before. Subsequently applying the specific boundary conditions, the matrix [G] of (4.31) is found to be:

$$[G] = \begin{bmatrix} -0.2454369261 & -0.03855314219r & 0.2454369261 \\ \frac{7}{16} & 0.04908738521r & \frac{7}{16} \\ 0.4774648293 & \frac{1}{8}r & -0.4774648293 \\ -1.013211836 & -0.1591549431r & -1.013211836 \\ -0.1290061377 & -0.1013211836r & 0.1290061377 \\ -0.4984044817 & 0.1290061377r & 0.4984044817 \end{bmatrix}$$

Thus the shape function matrix [N] of (4.34) is fully defined:

$$[N] = [\Theta][G]$$

Applying the strain-displacement procedure of Section 4.2 defines the quarter section strain-displacement matrix

$$[B] = [A][G]$$

where the elements of [A] are:

$$A_{11} = 0 \qquad A_{21} = \frac{1}{R}\cos\theta$$

$$A_{12} = \frac{\zeta}{r^2} \qquad A_{22} = \frac{1}{R}\left(\left(\theta - \frac{\pi}{4}\right)\cos\theta - \sin\theta\right)$$

$$A_{13} = -2\frac{\zeta}{r^2}\left(\theta - \frac{\pi}{4}\right) \qquad A_{23} = \frac{1}{R}\left(\left(\theta - \frac{\pi}{4}\right)^2\cos\theta - 2\left(\theta - \frac{\pi}{4}\right)\sin\theta\right)$$

$$A_{14} = \frac{-\zeta}{r^2}\left(6 + 3\left(\theta - \frac{\pi}{4}\right)^2\right) \qquad A_{24} = \frac{1}{R}\left(\left(\theta - \frac{\pi}{4}\right)^3\cos\theta - 3\left(\theta - \frac{\pi}{4}\right)^2\sin\theta\right)$$

$$A_{15} = \frac{-\zeta}{r^2}\left(24\left(\theta - \frac{\pi}{4}\right) + 4\left(\theta - \frac{\pi}{4}\right)^3\right) \qquad A_{25} = \frac{1}{R}\left(\left(\theta - \frac{\pi}{4}\right)^4\cos\theta - 4\left(\theta - \frac{\pi}{4}\right)^3\sin\theta\right)$$

$$A_{16} = \frac{-\zeta}{r^2}\left(60\left(\theta-\frac{\pi}{4}\right)^2 + 5\left(\theta-\frac{\pi}{4}\right)^4\right) \qquad A_{26} = \frac{1}{R}\left(\left(\theta-\frac{\pi}{4}\right)^5\cos\theta - 5\left(\theta-\frac{\pi}{4}\right)^4\sin\theta\right.$$

The element stiffness matrix is obtained as in BEL2.

4.4.1 BEL3 Programming.

Two FORTRAN programs based on the above theory have been programmed. BEL3FLEX.FOR evaluates the bellows flexibility factor and BEL3STR.FOR evaluates the normalised stress distribution over a 90^0 section of bellows. The FORTRAN code for these programs is given in Appendix 3.3.

As for BEL2, the 3x3 stiffness matrix of the bellows is obtained by integrating (4.43). This is done numerically, using a 5x3x3 Gaussian quadrature rule. The axial stiffness of the bellows is obtained by static condensation of the 3x3 matrix, and the bellows flexibility factor evaluated according to the definition of Chapter 4.1.3.

4.5 Element BEL4: Polynomial interpolation with Constant Axial Extension.

Element BEL4 is an extension of BEL3 which allows for constant direct axial strain in the bellows. Removing the inextensibility assumption uncouples the radial and tangential displacements, as:

$$\epsilon_{\theta o} = \frac{1}{r}\left(\frac{dv}{d\beta} + w\right) \neq 0$$

However, a coupling condition between w and v can be defined by considering the differential equations of extensional deformation. In [4.11] Meck extended his inextensional beam to include constant extensional strain based on a simple relationship derived by Cheng and Hoff for the bending of thin circular rings [4.12]. The radial and tangential displacements are coupled according to the equation

$$\frac{d}{d\beta}\left(\frac{dv}{d\beta} + w\right) = 0$$

Rearranging and differentiating the axial membrane strain gives

$$\frac{d}{d\beta}(r\epsilon_{\theta o}) = \frac{d}{d\beta}\left(\frac{dv}{d\beta} + w\right) = 0 \Leftrightarrow \epsilon_{\theta o} = \epsilon_o = \text{constant}$$

Therefore the displacements are coupled according to the equation

$$w - r\epsilon_o = \frac{dv}{d\beta} \tag{4.45}$$

where ϵ_o is a constant, and the rotation of the mid-surface is again given by (4.28):

$$\psi = \frac{1}{r}\left(\frac{dw}{d\beta} - v\right)$$

The tangential displacement v is again interpolated by the quintic polynomial:

$$v = a_1 + a_2\beta + a3_\beta^2 + a_4\beta^3 + a_5\beta^4 + a_6\beta^5$$

from which w and ψ are obtained according to (4.45) and (4.28) respectively. Thus the relationship between mid-surface displacement and unknown coefficients {a}

$$\{u\} = [\Theta]\{a\}$$

is obtained. This equation is identical to the corresponding BEL3 equation, except that {u} is now defined as

$$\{u\} = \{v \quad (w - r\epsilon_o) \quad \psi\}^T \tag{4.46}$$

The constant membrane strain has been introduced into the formulation as an additional degree of freedom. Applying the general boundary conditions to (4.46) yields an equation equivalent to (4.30), of the form

$$\{\Delta\} = [C]\{a\} \tag{4.47}$$

where in this case

$$\{\Delta\} = \{v_1 \quad (w_1 - r\epsilon_o) \quad \psi_1 \quad v_2 \quad (w_2 - r\epsilon_o) \quad \psi_2\}^T$$

Solving (4.47) for {a} and applying the specific boundary conditions of the quarter section yields the [G] matrix of (4.31). Again, this is identical to that obtained for BEL3 but in this case the reduced degree of freedom vector $\{\delta\}$ is

$$\{\delta\} = \{(w_1 - r\epsilon_o) \quad \psi_1 \quad (w_2 - r\epsilon_o)\}$$

In order to consider ϵ_o as a degree of freedom of the element, it must be included in the {d} vector. This is done by partitioning the matrix equation

$$\{a\} = [G]\{\delta\}$$

in the form

$$\{a\} = [\{G_1\}\{G_2\}\{G_3\}]\begin{Bmatrix}(w_1 - r\epsilon_o)\\ \psi_1 \\ (w_2 - r\epsilon_o)\end{Bmatrix}$$

This is rearranged as:

$$\{a\} = [J]\{d\} \tag{4.48}$$

where

$$\{d\} = \{w_1 \quad \psi_1 \quad w_2 \quad r\epsilon_o\}^T$$

and

$$[J] = [\{G_1\}\{G_2\}\{G_3\} - \{G_1 + G_3\}]$$

Substituting (4.48) into (4.29), the displacement vector {u} is given in terms of the four degrees of freedom {d} by:

$$\{u\} = [\Theta][J]\{d\} = \begin{bmatrix}\{\Theta_1\}\\ \{\Theta_2\}\\ \{\Theta_3\}\end{bmatrix}[J]\{d\} \tag{4.49}$$

The element strain-displacement matrix equation may now be obtained by considering (4.25) and (4.26).

From (4.49), the radial displacement may be expressed:

$$w = \{\Theta_2\}[J]\{d\} + r\epsilon_o = \{\{\Theta_2\}[J] + \{0 \quad 0 \quad 0 \quad 1\}\}\{d\}$$

As this introduces only constant terms into the radial displacement equations, differentials are as in the BEL3 formulation. Thus:

$$\frac{d^2w}{d\theta^2} = \left\{\frac{d}{d\theta^2}\{\Theta_2\}\right\}\{d\}$$

However, the axial strain now contains a membrane contribution and is given by:

$$\epsilon_\theta = \epsilon_o - \frac{\zeta}{r^2}\left(\frac{d^2 w}{d\phi^2} + w\right)$$

Noting that

$$\epsilon_o = \left\{0 \quad 0 \quad 0 \quad \frac{1}{r}\right\}\{d\}$$

this may be written

$$\epsilon_\theta = \left\{[A][J] + \left\{0 \quad 0 \quad 0 \quad \left(\frac{1}{r} - \frac{h}{r^2}\right)\right\}\right\}\{d\}$$

The circumferential strain

$$\epsilon_\phi = \frac{1}{R}(w\sin\theta + v\cos\theta)$$

here becomes

$$\epsilon_\phi = \left\{[A][J] + \left\{0 \quad 0 \quad 0 \quad \frac{\sin\theta}{R}\right)\right\}\{d\}$$

Thus the strain displacement equation is fully defined as

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

where

$$\{\epsilon\} = \{\epsilon_\theta \epsilon_\phi\}^T$$

and

$$[B] = [[A][J] + \begin{bmatrix} 0 & 0 & 0 & \left(\frac{r-h}{r^2}\right) \\ 0 & 0 & 0 & \frac{\sin\theta}{R} \end{bmatrix}$$

4.5.1 BEL4 Programming.

Two FORTRAN programs based on the above theory have been programmed. BEL4FLEX.FOR evaluates the bellows flexibility factor and BEL4STR.FOR evaluates the normalised stress distribution over a 90^{o} section of bellows. The FORTRAN code of these programs is given in Appendix 3.4.

The 3x3 stiffness matrix of the bellows is obtained by integrating (4.43). This is done numerically, using a 5x3x3 Gaussian quadrature rule. The axial stiffness of the bellows is obtained by static condensation of the 3x3 matrix and the bellows flexibility factor evaluated according to the definition of Chapter 4.1.3.

4.6 Discussion.

Sample analyses using the above bellows element formulations will be presented in Chapter 6, where it will be shown that the even Fourier series approach of BEL1 gives the best results when compared to alternative analyses. The polynomial based elements BEL3 and BEL4 will also be shown to give reasonably good results, although no significant improvement over BEL3 is made by including the constant extension term in BEL4. BEL2 will be shown to give relatively poor results.

In the next chapter formulations are presented for three elbow elements in which the interpolation schemes of BEL1 and BEL3 are developed for pipe bend applications. The first element adopts even Fourier series interpolation of ovalisation displacement in a constant bending element. The second element extends the first to include linear interpolation of ovalisation with respect to axial position, in order to investigate the relative convergence characteristics of the two approaches. The third element ovalisation model is based on the polynomial interpolation scheme of BEL3.

4.7 References.

[4.1] A LAUPA and N A WEIL
"Analysis of U-Shaped Expansion Joints," J. App. Mech. (Trans. ASME), Vol. 19, (1962).

[4.2] G E FINDLAY and J SPENCE
"On the elastic flexibility of axially loaded omega and toroidal bellows," Nuclear Eng. and Design, Vol 52, pp 83-89, (1979).

[4.3] W.Flugge
Stresses in shells
Springler-Verlag (1973).

[4.4] 'SMP'
SMP a Symbolic Manipulation Program. Reference Manual.
Inference Corporation, (1983).

[4.5] ANSYS
User's Manual
Swanson Analysis Systems, Inc., Houston, Pennsylvania, 15342 (USA).

[4.6] D G Ashwell and A B Sabir
"Limitations of Certain Curved Finite Elements When Applied to Arches,"
Int. J. Mech. Sci., Vol 13, pp 133-139, (1971).

[4.7] D G Ashwell A B Sabir and T M Roberts
"Further Studies in the application of Curved Finite Elements to Circular Arches,"
Int. J. Mech. Sci., Vol 13, pp 507-517, (1971).

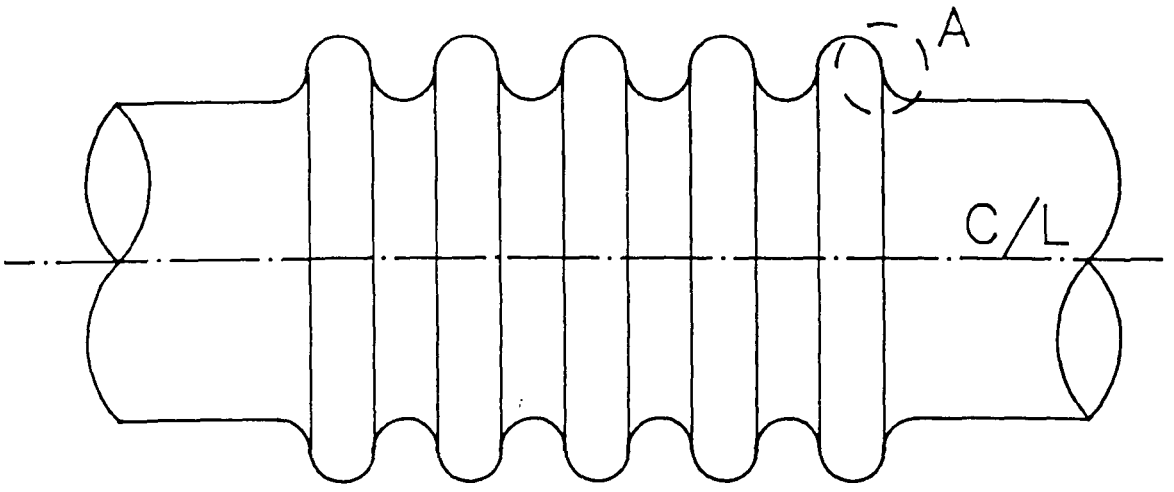
[4.8] A B Sabir and D G Ashwell
"A corrected Assesment of the Cylindrical Shell Finite Element of Bogner, Fox and Schmit When Applied to Arches,"
Int. J. Mech. Sci., Vol 13, pp 133-139, (1971).

[4.9] D J Dawe
"Curved Finite Elements for the Analysis of Shallow and Deep Arches,"
Comp. & Struct., Vol 4, pp 729-740, (1970).

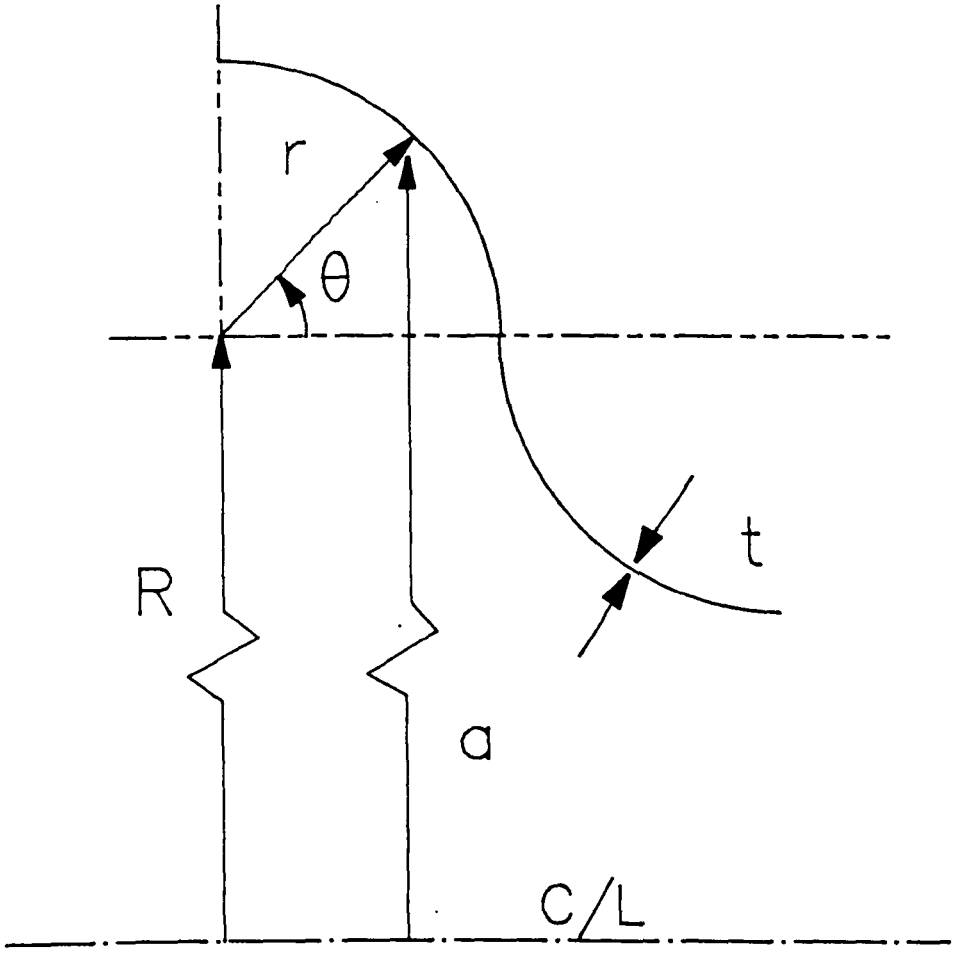
[4.10] D J Dawe
"Numerical Studies Using Circular Arch Finite Elements,"
Comp. & Struct., Vol 4, pp 729-740, (1974).

[4.11] H R Meck
"An Accurate Polynomial Displacement Function for Finite Ring Elements," Comp. & Str., Vol 11, pp 265-269, (1980).

[4.12] S Cheng N J Hoff
"Bending of Thin Circular Rings," Int. J. Solids Str., Vol 11, pp 139-152, (1975).



Semi–toroidal bellows



Sectional view of A

Figure 4.1 Semi-toroidal bellows expansion joint geometry.

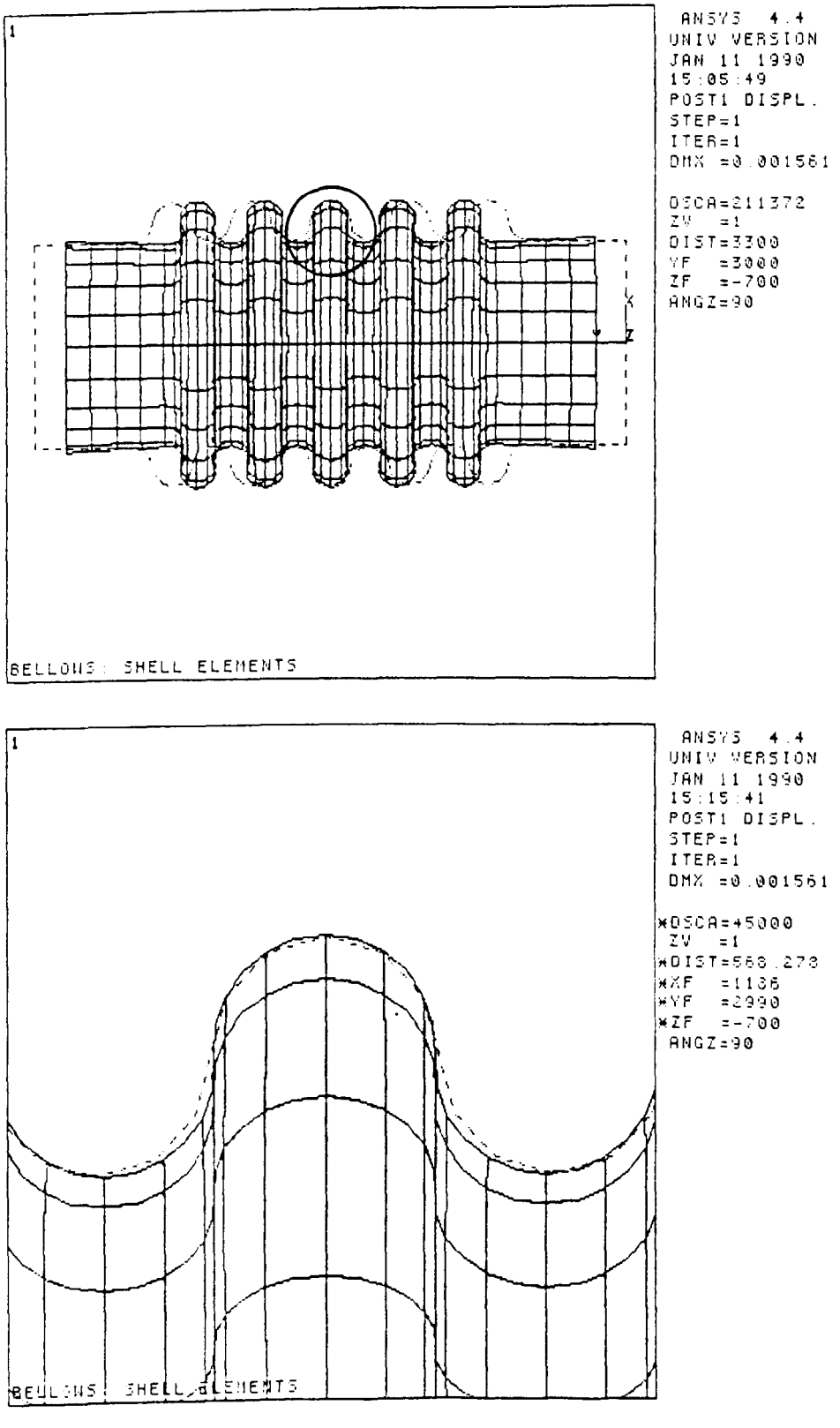


Figure 4.2 Semi-toroidal bellows deformation due to axial force, (thin shell finite element analysis).

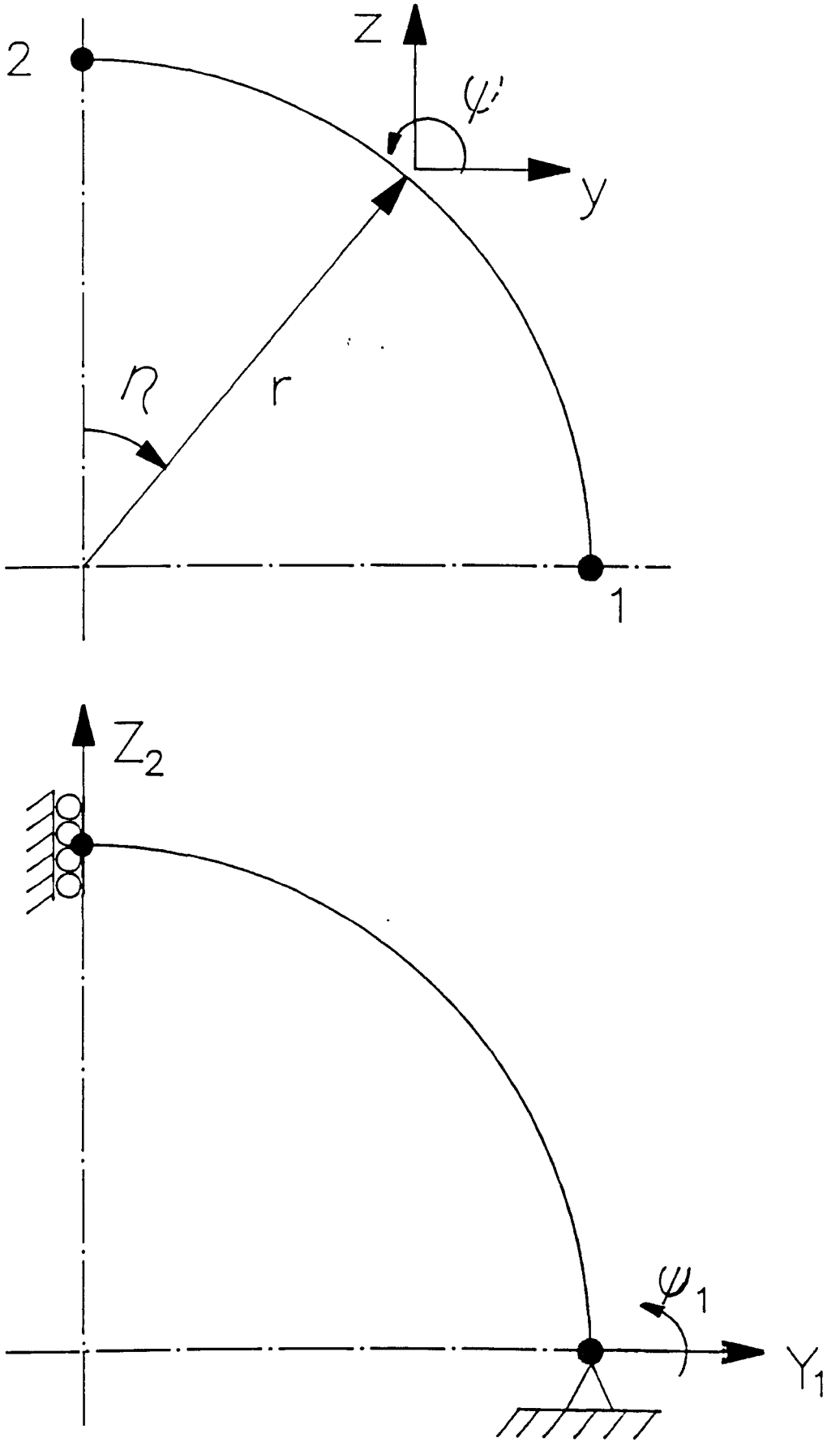


Figure 4.3 Bellows element BEL1 coordinate systems, shell mid surface displacements and nodal degrees of freedom.

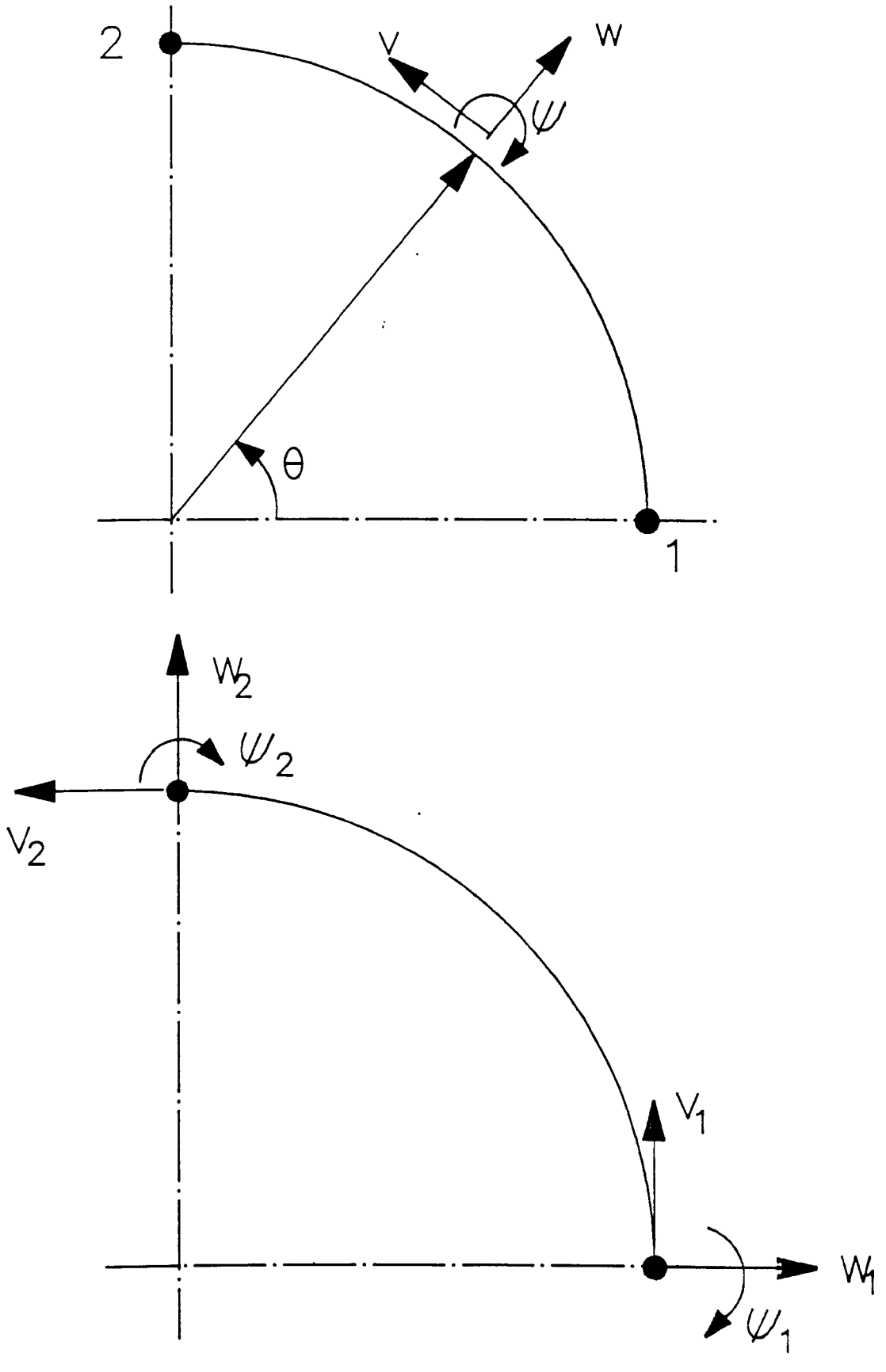


Figure 4.4 Bellows elements BEL2, BEL3 and BEL4 shell displacements and nodal degrees of freedom.

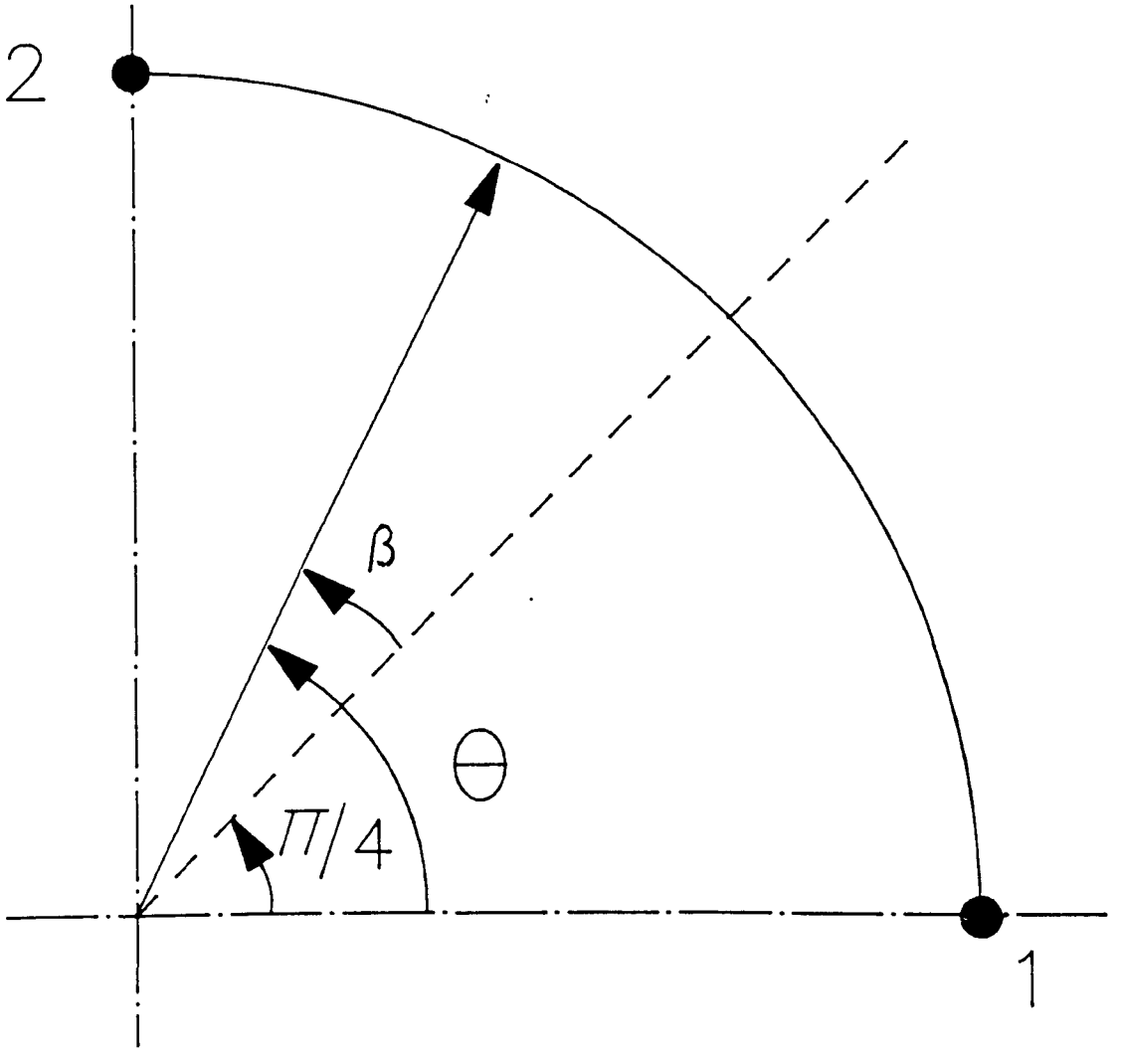


Figure 4.5 BEL3 and BEL4 interpolation coordinate system.

CHAPTER 5.

THREE SIMPLE PIPING ELBOW FINITE ELEMENT FORMULATIONS.

5 THREE SIMPLE PIPING ELBOW FINITE ELEMENT FORMULATIONS.

The basic requirements for simple piping elbow elements were discussed in Chapter 3 and elbow ovalisation investigated in Chapter 4 by considering the deformation of semi-toroidal bellows. In this chapter three simple elbow element formulations are presented. The proposed elements are based on the beam-shell (or beam with deforming section) approach, and share a common beam-bending model based on the exact solution of Vlasov's thin-wall curved beam theory [5.1].

In the first element, PB1, the ovalisation behaviour of the element is based on the classic Von Karman analysis discussed in Chapter 2. The ovalisation displacement is constant with respect to axial position and interpolated around the cross-section of the elbow by a three even term Fourier series similar to that used in the bellows element BEL1 of Chapter 4.

The second, PB2, extends the PB1 formulation is to include linear variation of ovalisation in the axial direction. This allows the convergence characteristics of a simple element with linear ovalisation to be investigated.

The third element, PB3, was formulated in order to asses the performance of polynomial interpolation of ovalisation. As in PB1, the ovalisation is constant with respect to axial position. Four quintic polynomials are used to interpolate the ovalisation displacements piecewise around the cross-section of the bend. The interpolation scheme is similar to that of the bellows element BEL3.

All three elements have been programmed as user elements for the ANSYS finite element program. In order to minimise the computing time required in element formulation, the common beam bending stiffness matrix has been obtained in closed form by analytical integration. The ovalisation stiffness of element PB1 is also given in closed form, but in PB2 and PB3 the ovalisation stiffness matrices are integrated numerically.

5.1 PB1: An Elbow Element with Constant Fourier Ovalisation.

In this chapter a constant ovalisation beam-shell element, referred to as element PB1, is presented. The element is based on an exact solution of Vlasov's thin wall curved beam theory, upon which classic Von Karman ovalisation is superposed.

An element based on similar beam and ovalisation models has previously been presented by See in [3.23]. See's element was shown to give good agreement with comparable analyses for in-plane bending of elbows, but poor agreement for out-of-plane bending. The poor out-of-plane performance was not adequately accounted for in [3.23], where it was concluded that the difference between results given by the element and ANSYS STIF18 (a conventional flexibility analysis curved beam element) was due to the limitations of flexibility analysis and not the element formulation. However, comparison with results given by other analyses indicate that the out-of-plane results given by the element were in fact poor.

In Chapter 6 it will be shown that the actual reason for the poor out-of-plane performance was that See failed to consider the element convergence requirements for out-of-plane bending.

In the formulation of a Vlasov-Karman element proposed by See, the element stiffness matrix was evaluated by numerical integration. In practice 24x12x3 point Gaussian quadrature was used, making the element expensive in comparison with flexibility analysis elements. However, the basic Vlasov and Von Karman models offer potential for programming an inexpensive element. In the formulation presented below, the total number of integrals required to evaluate the 12x12 stiffness matrix is 21, of which 16 are double integrals and 5 triple integrals. The integrals are given in closed form, dispensing with the need for numerical integration. The integration was performed using the symbolic algebra program SMP [4.4].

5.1.1 Overview of Element PB1 Formulation.

The element beam bending model is based on Vlasov's thin wall curved beam theory as defined in reference [5.1]. Beam elements based on the exact solution of Vlasov's differential equations have been presented in the literature by Chaundhuri and Shore [5.2] and by Yoo [5.3, 5.4]. In element PB1, (and subsequently elements PB2 and PB3), the Chaundhuri and Shore solution has been modified and adopted as the beam bending model for the elbow element. The elbow ovalisation deformation, which is superposed on the beam model, is represented by an extension of the classic Von Karman model.

In both Vlasov's beam theory and Von Karman's ovalisation analysis the in-plane and out-of-plane responses are uncoupled. Thus the element may be formulated by developing independent in-plane and out-of-plane models, which are then superposed to give a fully representative elbow element. In the proposed element formulation in-plane and out-of-plane stiffness equations are obtained in the form

$$\{F\} = [K]\{d\} \tag{5.1}$$

where {F} is the load vector, {d} the generalised displacement vector, and [K] the stiffness matrix, given by the equation

$$[k] = \int_V [B]^T [D][B] dV \tag{5.2}$$

[B] is the element strain-displacement matrix and [D] the constitutive matrix.

By partitioning the in-plane and out-of-plane strain-displacement matrices [B] into nodal and nodeless degree of freedom sub-matrices, the stiffness matrix is obtained in the form:

$$\begin{Bmatrix} \{F_b\} \\ \{F_o\} \end{Bmatrix} = \begin{bmatrix} [K_{bb}] & [K_{bo}] \\ [K_{ob}] & [K_{oo}] \end{bmatrix} \begin{Bmatrix} \{d_b\} \\ \{d_o\} \end{Bmatrix} \tag{5.3}$$

where $\{d_b\}$ and $\{d_o\}$ are vectors of bending and ovalisation displacements, $[F_b]$ and $[F_o]$ bending and ovalisation forces and [Kbb], [Koo], [Kbo] and [Kob] are stiffness sub-matrices given by the equations:

$$[K_{bb}] = \int_V [B_b]^T [D][B_b] dV \tag{5.4}$$

$$[K_{oo}] = \int_V [B_o]^T [D][B_o] dV \tag{5.5}$$

$$[K_{bo}] = [K_{ob}]^T = \int_V [B_b]^T [D][B_o] dV \tag{5.6}$$

for bending, ovalisation and coupling respectively.

In-plane and out-of-plane elbow stiffness matrices are obtained according to (5.3), and then assembled to give a single, fully representative, elbow element stiffness matrix. The full matrix is of order (12+m)x(12+m), where m is the total number of ovalisation degrees of freedom. The full stiffness matrix is reduced to a 12x12 matrix compatible with standard beam elements by applying the process of static condensation as described in Chapter 2.2.1. In this case the ovalisation force vector is zero and static condensation results in the equation:

$$\{F_b\} = [K_R]\{d_b\} \tag{5.9}$$

where $[K_R]$ is the reduced element stiffness matrix given by

$$[K_R] = [[K_{bb}]-[K_{ob}]^T[K_{oo}]^{-1}[K_{ob}]] \tag{5.10}$$

5.1.2 In-plane Bending.

The elbow element's beam bending modes are represented by modifying the Chaundhuri and Shore curved beam element formulation [5.2]. In the Chaundhuri beam element it is assumed that:

i) Beam deformations are small with respect to the dimensions of the cross-section.

ii) The beam is thin walled and through-thickness stresses assumed negligible.

iii) Shearing deformation vanishes at the middle surface.

iv) The cross-section is open and symmetric about the beam's depth axis, (normal to the plane of curvature).

v) The cross-section remains undeformed; that is, the cross-section is constrained to be rigid.

In addition to the standard beam deformation modes of axial stretch, in-plane bending, out-of-plane bending, transverse shears, and torsional shear, assumptions (iv) and (v) give rise to longitudinal warping of the beam's cross-section, such that plane sections do not remain plane upon deformation.

In the Chaundhuri and Shore beam cross-sectional warping is represented by including beam degrees of freedom to define this behaviour. In the case of a

beam symmetric about its depth axis there are 7 stress resultants: 1 normal, 2 transverse, 3 bending and 1 warping. Hence 7 degrees of freedom are required to describe the beam behaviour.

However, in a beam of closed circular section warping of the cross-section is suppressed due to the higher torsional rigidity of the cross-section, as noted by Yoo in [5.4]. It is, therefore, possible to simplify the analysis of such beams by assuming cross-sectional warping to be negligible. Thus, in considering the beam bending behaviour of a pipe bend, the following assumption is added to the above

vi) Plane sections remain plane upon deformation: that is, warping deformation is negligible.

In the beam model proposed below, the Chaundhuri and Shore formulation is simplified by invoking assumption (vi) above.

Degrees of Freedom.

The basic beam model is a two-noded circularly curved beam of closed circular cross-section, as shown in Figure 5.1. The local X axis is along the beam's length, Y is perpendicular to the plane of the beam, and Z is radially inward to the centre of curvature, completing a right handed triad. The deformation of an arbitrary point on the axis of such a beam is defined by three translational degrees of freedom and a single rotation: displacements u in the X direction, v in the Y direction and w in the Z direction, and rotation ψ_Y about the Y axis.

The in-plane bending nodal degrees of freedom $\{d_{bi}\}$, shown in Figure 5.2, are:

$$\{d_{bi}\} = \{u_1 \quad w_1 \quad \psi_{Y1} \quad u_2 \quad w_2 \quad \psi_{Y2}\}^T$$

Governing differential equations.

Neglecting the effects of warping, the in-plane governing differential equations of the beam are given in [5.2] simplify to:

$$\frac{du}{ds} = \frac{w}{R} + R\left(\frac{I}{A}\right)\left(\frac{d^4w}{ds^4} + \frac{1}{R^2}\frac{d^2w}{ds^2}\right) \qquad (5.11)$$

$$\frac{d^5 w}{ds^5} + \frac{2}{R^2}\frac{d^3 w}{ds^3} + \frac{1}{R^4}\frac{dw}{ds} = 0 \qquad (5.12)$$

where s is the sole independent variable, denoting the position along the bend axis: $s = R\phi$. Here the dependant variable is taken instead to be angular position ϕ, where $\phi = s/R$

Displacement Field and Shape Function Matrix.

The displacement field for in-plane deformation is found by solving equations (5.11) and (5.12) for u and w. Neglecting cross-sectional warping (assumption (vi) above) the solution of Chaundhuri and Shore reduces to:

$$u(\phi, n) = Bl_{1,n}[\phi] + Bl_{2,n}[\sin\phi] \qquad (5.13)$$
$$+ Bl_{3,n}[\phi\sin\phi + (1-2d)\cos\phi] + Bl_{4,n}[\cos\phi]$$
$$+ Bl_{5,n}[\phi\cos\phi - (1-2d)\sin\phi] + Bl_{6,n}[1]$$

$$w(\phi, n) = Bl_{1,n}[1] + Bl_{2,n}[\cos\phi] + Bl_{3,n}[\phi\cos\phi] \qquad (5.14)$$
$$+ Bl_{4,n}[\sin\phi] + Bl_{5,n}[\phi\sin\phi] + Bl_{6,n}[0]$$

where

$$d = \frac{I}{AR^2} \qquad \text{and n = 1 to 6.}$$

In matrix notation, the displacement field equations may be written

$$u_{(\phi,n)} = \{\eta_1\} Bl \qquad (5.15)$$

$$w_{(\phi,n)} = \{\eta_2\} Bl \qquad (5.16)$$

where [B1] is a matrix of unknown in-plane constants and

$$\{\eta_1\} = \{\phi \quad \sin\phi \quad [\phi\sin\phi + (1-2d)\cos\phi]$$
$$\cos\phi \quad [\phi\cos\phi - (1-2d)\sin\phi] \quad 1\}$$

$$\{\eta_2\} = \{1 \quad \cos\phi \quad \phi\cos\phi \quad \sin\phi \quad \phi\sin\phi \quad 0\}$$

The in-plane rotation, ψ_Y, is given by:

$$\psi_{Y(\phi,n)} = -\frac{1}{R}\left\{\frac{dw}{d\phi} + u\right\}$$

As [B1] is constant, this may be written:

$$\psi_{Y(\phi,n)} = \frac{-1}{R}\left\{\frac{d\{\eta_2\}}{d\phi} + \{\eta_1\}\right\}Bl$$

or

$$\psi_{Y(\phi,n)} = \{\eta_3\}[Bl] \qquad (5.17)$$

where

$$\{\eta_3\} = \frac{-1}{R} \quad \{\phi \quad 0 \quad [2(1-d)\cos\phi] \quad 0 \quad [2(1-d)\sin\phi] \quad 1\}$$

Equations (5.15), (5.16) and (5.17) fully define the in-plane displacement field of the beam model. The subscript ϕ defines the position of a point along the axis and subscript n defines a specific degree of freedom from 1 to 6. The displacement $u_{(\phi,n)}$, for example, is the u displacement at a point ϕ along the axis due to degree of freedom *n* being given a unit displacement when all other degrees of freedom are fixed at zero.

The in-plane degrees of freedom correspond to the integers n as follows:

n	1	2	3	4	5	6
dof	u_1	w_1	ψ_{Y1}	u_2	w_2	ψ_{Y2}

The shape of the displacement field is defined by the vectors $\{\eta\}$. The magnitude of displacement is determined by the matrix of constants [B1].

In order to use this displacement field to formulate an element, we require it in the form of the interpolation equation given in (4.7):

$$\{u\} = [N]\{d\}$$

That is, the displacements at an arbitrary point are interpolated in terms of the nodal degrees of freedom by the shape function matrix [N]. [N] is obtained as follows.

Degree of freedom n is displaced by a given amount, which for simplicity is chosen to be unity. The displacement at an arbitrary point on the centroid of the beam due to this prescribed displacement is given by equations (5.15), (5.16) and (5.17). By choosing the arbitrary point to be at one of the nodes of the beam, the displacement at the point is defined by the boundary conditions: unity for degree of freedom n, and zero for the remaining five degrees of freedom. Repeating this procedure for all six n, a system of 36 simultaneous equations is obtained, which may be solved for the unknown constants [B1].

For example, consider n = 1. u_1 is given a value of 1 and the remaining degrees of freedom fixed at zero. Considering equations (5.15) to (5.17):

at node 1, $\phi = 0$, and for n = 1:

$$u_{(0,1)} = \{\eta_1\}_{\phi=0}[BI] = 1$$

$$w_{(0,1)} = \{\eta_2\}_{\phi=0}[BI] = 0$$

$$\psi_{Y(0,1)} = \{\eta_3\}_{\phi=0}[BI] = 0$$

at node 2, $\phi = \alpha$, and for n = 1:

$$u_{(\alpha,1)} = \{\eta_1\}_{\phi=\alpha}[BI] = 0$$

$$w_{(\alpha,1)} = \{\eta_2\}_{\phi=\alpha}[BI] = 0$$

$$\psi_{Y(\alpha,1)} = \{\eta_3\}_{\phi=\alpha}[BI] = 0$$

Hence six simultaneous equations are obtained. Repeating this procedure for all six n, a system of 36 simultaneous equations may be obtained in the form:

$$[CI][BI] = [I] \qquad (5.18)$$

where [I] is the 6x6 identity matrix, [B1] is a 6x6 matrix of constants and [C1] is the matrix:

$$[C1] = \begin{bmatrix} \{\eta_1\}_{\phi=0} \\ \{\eta_2\}_{\phi=0} \\ \{\eta_3\}_{\phi=0} \\ \{\eta_1\}_{\phi=\alpha} \\ \{\eta_2\}_{\phi=\alpha} \\ \{\eta_3\}_{\phi=\alpha} \end{bmatrix}$$

That is,

$$[C1] = \begin{bmatrix} 0 & 0 & (1-2d) & -1 & 0 & 1 \\ 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -B & 0 & 0 & -\frac{1}{R} \\ \alpha & s & C & -c & D & 1 \\ 1 & c & \alpha c & s & \alpha s & 0 \\ -\frac{\alpha}{R} & 0 & -Bc & 0 & -Bs & -\frac{1}{R} \end{bmatrix}$$

where

$$s = \sin\alpha \qquad c = \cos\alpha$$

$$B = \frac{2}{R}(1-d) \qquad C = \alpha s + (1-2d)c \qquad D = \alpha c + (1-2d)$$

By definition, equation (5.18) gives

$$[B1] = [C1]^{-1}$$

Therefore the displacement field is fully defined by inverting the [C1] matrix to evaluate [B1] and substituting into equations (5.15), (5.16) and (5.17). Hence the displacement interpolation equation for in-plane bending is:

$$\{u\} = [N1]\{d_{bi}\} \qquad (5.19)$$

where

$$\{u\} = \{u \quad w \quad \psi_Y\}^T$$

and

$$[N1] = \begin{bmatrix} \{\eta_1\} \\ \{\eta_2\} \\ \{\eta_3\} \end{bmatrix} [B1] \quad = \quad [\eta 1] \quad [B1]$$

[N1] is the in-plane shape function matrix of the beam model.

Strain-Displacement.

From Vlasov [5.1], the state of strain for in-plane bending is uniaxial meridional strain, given by the equation:

$$\epsilon_{\phi bi} = \epsilon_{\phi bio} + z k_{\phi bio} \tag{5.20}$$

where z denotes distance from the centroid in the element Z direction, as shown in Figure 5.3.

From Vlasov's small displacement theory of [5.1], neglecting all second order terms and noting the change in direction of the out-of-plane axis made here, the extension and bending terms are defined:

$$\epsilon_{\phi bi\ o} = \frac{1}{R}\left(\frac{du}{d\phi} - w\right)$$

$$k_{\phi bi\ o} = -\frac{1}{R^2}\left(\frac{d^2 w}{d\phi^2} + w\right)$$

Hence (5.20) becomes

$$\epsilon_{\phi bi} = \frac{1}{R}\left\{\frac{du}{d\phi} - \frac{r\cos\theta}{R}\left(\frac{d^2 w}{d\phi^2}\right) - \left(1 + \frac{r\cos\theta}{R^2}\right)w\right\}$$

From (5.19):

$$u = \{\eta_1\}[B1]\{d_{bi}\}$$

$$w = \{\eta_2\}[B1]\{d_{bi}\}$$

hence

$$\frac{du}{d\phi} = \frac{d\{\eta_1\}}{d\phi}[Bl]\{d_{bi}\}$$

$$\frac{d^2w}{d\phi^2} = \frac{d^2\{\eta_2\}}{d\phi^2}[Bl]\{d_{bi}\}$$

as [B1] and $\{d_{bi}\}$ are constant.

Thus

$$\epsilon_{\phi bi} = \frac{1}{R}\left\{\frac{d\{\eta_1\}}{d\phi} - \frac{r\cos\theta}{R}\frac{d^2\{\eta_2\}}{d\phi^2} - \left(1 + \frac{r\cos\theta}{R}\right)\{\eta_2\}\right\}[Bl]\{d_{bi}\}$$

or

$$\epsilon_{\phi bi} = \frac{1}{R}\{Al\}[Bl]\{d_{bi}\} \tag{5.21}$$

Differentiating and collecting the vector terms gives:

$$\{Al\} = \{\frac{-r\cos\theta}{R} \quad 0 \quad 2\left(d + \frac{r\cos\theta}{R}\right)\sin\phi \quad 0 \quad -2\left(d + \frac{r\cos\theta}{R}\right)\cos\phi \quad 0\} \tag{5.22}$$

Thus the in-plane beam strain-displacement equation is:

$$\epsilon_{\phi bi} = [B_{bi}]\{d_{bi}\}$$

where the strain-displacement matrix is given by:

$$[B_{bi}] = \frac{1}{R}\{Al\}[Bl] \tag{5.23}$$

Bending Stiffness Matrix.

The in-plane bending stiffness matrix is given by equation (5.4):

$$[K_{bb}] = \int_V [B_b]^T[D][B_b]dV$$

Noting that only axial strain is involved in the in-plane bending stiffness equation, the [D] matrix reduces to a single element:

$$[D] = \frac{E}{(1-\nu^2)}[1]$$

Thus, substituting for [D] and given [Bb] from (5.23):

$$[K_{bbi}] = \int_V \frac{1}{R}[BI]^T\{AI\}^T \frac{E}{(1-\nu^2)}\frac{1}{R}\{AI\}[BI]dV$$

Taking the constant terms out-with the integral and noting that the strain is constant through the thickness of the beam wall, the in-plane bending stiffness equation is given by:

$$[K_{bbi}] = \frac{Etr}{(1-\nu^2)R}[BI]^T[I_{bi}][BI]$$

where

$$[I_{bi}] = \int_0^{\alpha}\int_0^{2\pi}\{AI\}^T\{AI\}d\theta d\phi$$

Integrating:

$$[K_{bbi}] = \frac{EI}{(1-\nu^2)R^3}[BI]^T[J_{bi}][BI] \qquad (5.24)$$

where

$$[J_{bi}] = \begin{bmatrix} \alpha & 0 & 2(\cos\alpha-1) & 0 & 2\sin\alpha & 0 \\ & 0 & 0 & 0 & 0 & 0 \\ & & H(2\alpha-\sin 2\alpha) & 0 & H(\cos 2\alpha-1) & 0 \\ & \text{symm.} & & 0 & 0 & 0 \\ & & & & H(2\alpha+\sin 2\alpha) & 0 \\ & & & & & 0 \end{bmatrix}$$

and

$$H = 1 + \frac{1}{2}\left(\frac{r}{R}\right)^2$$

5.1.3 In-Plane Ovalisation

Elbow ovalisation is represented by the Von Karman constant bending or constant ovalisation model, as discussed in Chapters 2 and 3. The axial strain is assumed to be uniform through the wall thickness and the hoop strain is assumed to arise from inextensional bending of the pipe cross-section. Assuming R>>r and applying the Von Karman assumptions, the axisymmetric shell equations (3.3a-c) reduce to:

$$\epsilon_{\phi oi} = \frac{1}{R}(w_o \cos\theta - v_o \sin\theta) \tag{5.25}$$

$$\epsilon_{\theta oi} = -\frac{\zeta}{r^2}\left(\frac{d^2 w_o}{d\theta^2} + w_o\right) \tag{5.26}$$

under the inextensibility condition

$$w_o = -\frac{\partial v}{\partial \theta}$$

where w_O and v_O are ovalisation radial and tangential displacements of the elbow mid-surface, as shown in Figure 5.3.

The ovalisation displacements are interpolated by an even Fourier series as in the original Von Karman analysis. The radial displacement is interpolated by the series

$$w_o = \sum_{n=1}^{N} a_n \cos 2n\theta$$

The radial and tangential displacements are coupled by the inextensibility equation; thus the tangential displacement is:

$$v_o = -\sum_{n=1}^{N} \frac{1}{2n} a_n \sin 2n\theta$$

Substituting the interpolation functions into the strain displacement equations (5.25) and (5.26) gives

$$\epsilon_{\phi oi} = 1/R \sum a_n \left(\cos\theta \cos 2n\theta + \frac{1}{2n}\sin\theta \sin 2n\theta\right) \tag{5.27a}$$

$$\epsilon_{\theta oi} = -\frac{\zeta}{r^2}\sum a_n(1-4n^2)\cos 2n\theta) \qquad (5.27b)$$

The number of terms N taken in the Fourier series determines the accuracy and applicability of the element. It has been suggested that for elbows of parameter $\lambda \geq 0.1$, such as those encountered in general piping analysis, three terms are sufficient [3.12]. (5.27a) and (5.27b) are thus expanded and arranged in the form:

$$\{\epsilon_{oi}\} = [B_{oi}]\{a\} \qquad (5.28)$$

where

$$\{\epsilon_{oi}\} = \{\epsilon_{\phi oi} \quad \epsilon_{\theta oi}\}^T \qquad \{a\} = \{a_1 \quad a_2 \quad a_3 \quad \}^T$$

In-plane Ovalisation Stiffness.

The in-plane ovalisation stiffness matrix is defined by (5.5):

$$[K_{oo}] = \int_V [B_o]^T [D][B_o] dV$$

In this case the constitutive matrix is

$$[D] = \frac{E}{1-\nu^2}\begin{bmatrix} 1 & \nu \\ \nu & 1 \end{bmatrix} = \frac{E}{1-\nu^2}[D_1]$$

As the ovalisation is constant with respect to axial position the integral reduces to:

$$[K_{ooi}] = \frac{E r R \alpha}{1-\nu^2}\int_{-\frac{t}{2}}^{\frac{t}{2}}\int_0^{2\pi} [B_{oi}]^T [D_1][B_{oi}] d\theta d\zeta$$

However, as in the case of the bellows elements, the product of longitudinal and hoop strains integrate through-thickness to zero. Thus the stiffness integral reduces to:

$$[K_{ooi}] = \frac{E r R \alpha}{1-\nu^2}\int_{-\frac{t}{2}}^{\frac{t}{2}}\int_0^{2\pi} [B_{oi}]^T [B_{oi}] d\theta d\zeta$$

or

$$[K_{ooi}] = \frac{E r R \alpha}{1-\nu^2}[I_{oi}]$$

Integrating and collecting terms gives the in-plane ovalisation stiffness matrix in closed form:

$$[K_{ooi}] = \frac{E I \alpha}{(1-\nu^2)R^3}[J_{oi}] \tag{5.29}$$

where

$$[J_{oi}] = \begin{bmatrix} \left(\frac{3R^4t^2}{4r^6} + \frac{5R^2}{32r^2}\right) & \left(\frac{5R^2}{32r^2}\right) & 0 \\ & \left(\frac{75R^4t^2}{4r^6} + \frac{17R^2}{32r^2}\right) & \left(\frac{7R^2}{32r^2}\right) \\ symm. & & \left(\frac{1225R^4t^2}{12r^6} + \frac{37R^2}{72r^2}\right) \end{bmatrix}$$

5.1.4 In-plane Bending-Ovalisation Coupling.

The bending-ovalisation stiffness matrix is defined by (5.6):

$$[K_{bo}] = \int_V [B_b]^T[D][B_o]dV$$

As a 2-D constitutive matrix is required, {A1} is expanded by adding a row of zeros to give the matrix [A1]:

$$[AI] = \begin{bmatrix} A_1 & 0 & A_3 & 0 & A_5 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Thus, substituting for $[B_b]$ from (5.23),

$$[K_{boi}] = \int_V \frac{1}{R}[BI]^T[AI]^T[D][B_{oi}]dV$$

As before, $[B1]^T$ may be taken outside the integral to give

$$[K_{boi}] = \frac{1}{R}[BI]^T \int_V [AI]^T [D][B_{oi}] dV$$

which upon integration yields the in-plane bending-ovalisation coupling matrices

$$[K_{boi}] = [K_{obi}]^T = \frac{EI}{(1-\nu^2)R^3}[BI]^T [J_{boi}]$$

where:

$$[J_{boi}] = \begin{bmatrix} \frac{-3R\alpha}{4r} & 0 & 0 \\ 0 & 0 & 0 \\ \frac{3R}{2r}(1-\cos\alpha) & 0 & 0 \\ 0 & 0 & 0 \\ \frac{3R}{2r}(1-\cos\alpha) & 0 & 0 \end{bmatrix}$$

5.1.5 Out-of-Plane Bending.

The procedure for evaluating the out-of-plane bending stiffness matrix is identical to that for the in-plane case detailed in section 5.2.2. The principal equations are as follows:

Degrees of Freedom.

The out-of-plane beam degrees of freedom are shown in Figure 5.2.

$$\{d_{bo}\} = \{v_1 \quad \psi_{X1} \quad \psi_{Z1} \quad v_2 \quad \psi_{X2} \quad \psi_{Z2}\}^T$$

Governing differential equations.

$$\frac{d^2v}{ds^2} = \frac{-GJ}{EI+GJ} R \frac{d^2\psi_X}{ds^2} + \frac{EI}{EI+GJ}\frac{\psi_X}{R} \tag{5.30}$$

$$\frac{d^4\psi_X}{d\phi^4} + 2\frac{d^2\psi_X}{d\phi^2} + \psi_X = 0 \tag{5.31}$$

Displacement Field and Shape Function Matrix.

$$v_{(\phi,n)} = \{\eta_4\}[B2] \tag{5.32}$$

$$\psi_{X(\phi,n)} = \{\eta_5\}[B2] \tag{5.33}$$

$$\psi_{Z(\phi,n)} = \{\eta_6\}[B2] \tag{5.34}$$

$$\{\eta_4\} = \{\cos\phi \quad (\phi\cos\phi - b\sin\phi) \quad \sin\phi \quad (\phi\sin\phi + b\cos\phi) \quad -1 \quad -\phi\}$$

$$\{\eta_5\} = \{\cos\phi \quad \phi\cos\phi \quad \sin\phi \quad \phi\sin\phi \quad 0 \quad 0\}$$

$$\{\eta_6\} = \frac{d\{\eta_4\}}{ds}$$

$$\{\eta_6\} = \frac{1}{R}\{-\sin\phi \quad (-\phi\sin\phi + (1-b)\cos\phi) \quad \cos\phi$$

$$(\phi\cos\phi + (1-b)\sin\phi)0 \quad -1\}$$

Applying unit nodal displacements, the [C2] matrix corresponding to [C1] for in-plane bending is

$$[C2] = \begin{bmatrix} 1 & 0 & 0 & b & -1 & 0 \\ \frac{1}{R} & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{(1-b)}{R} & \frac{1}{R} & 0 & 0 & -\frac{1}{R} \\ c & (\alpha c - sb) & s & (\alpha s + cb) & -1 & -\alpha \\ \frac{c}{R} & \frac{\alpha c}{R} & \frac{s}{R} & \frac{\alpha s}{R} & 0 & 0 \\ \frac{-s}{R} & F & \frac{c}{R} & G & 0 & \frac{-1}{R} \end{bmatrix}$$

where

$$F = \frac{1}{R}(-\alpha s + c(1-b)) \qquad G = \frac{1}{R}(\alpha c + s(1-b))$$

The out-of-plane coefficient matrix [B2] is obtained by inverting [C2] as in the in-plane case. The out-of-plane displacement interpolation equation is thus:

$$\{u_{bo}\} = [N2]\{d_{bo}\}$$

where

$$\{u_{bo}\} = \{v \quad \psi_X \quad \psi_Z\}^T$$

$$[N2] = \begin{bmatrix} \{\eta_4\} \\ \{\eta_5\} \\ \{\eta_6\} \end{bmatrix} [B2] = [\eta 2] [B2]$$

[N2] is the out-of-plane bending shape function matrix.

Strain-Displacement

The out-of-plane bending strains are

i) A contribution to ϵ_ϕ due to bending.

ii) Shear strain γ.

from Vlasov [5.1]:

$$\epsilon_{\phi bo} = \frac{y}{R}\left\{\frac{1}{R}\frac{d^2v}{d\phi^2} + \psi_X\right\}$$

$$\gamma = \frac{r}{R}\left\{\frac{d\psi_X}{d\phi} - \frac{1}{R}\frac{dv}{d\phi}\right\}$$

where

$$y = r\sin\theta$$

as shown in Figure 5.3.

Differentiating and collecting the vector terms as before gives:

$$\epsilon_{\phi bo} = \frac{y}{R^2}\{0 \quad -(2-b)\sin\phi \quad 0 \quad (2-b)\cos\phi \quad 0 \quad 0\}[B2]\{d_{bo}\}$$

$$\gamma = \frac{r}{R^2}\{0 \quad b\cos\phi \quad 0 \quad b\sin\phi \quad 0 \quad 1\}[B2]\{d_{op}\}$$

Thus the strain-displacement equation may be written

$$\{\epsilon_o\} = \frac{r}{R^2}[A2][B2]\{d_{bo}\}$$

where

$$\{\epsilon_o\} = \{\epsilon_\phi \quad \gamma\}^T$$

and substituting for y

$$[A2] = \begin{bmatrix} 0 & [-(2-b)\sin\phi\sin\theta] & 0 & [(2-b)\cos\phi\sin\theta] & 0 & 0 \\ 0 & b\cos\phi & 0 & b\sin\phi & 0 & 1 \end{bmatrix}$$

The out-of-plane bending strain displacement matrix is

$$[B_{bo}] = \frac{r}{R^2}[A2][B2] \tag{5.35}$$

Bending Stiffness Matrix.

The out-of-plane bending stiffness matrix is given by (5.4):

$$[K_{bb}] = \int_V [B_b]^T [D][B_b]dV$$

Therefore:

$$[K_{bbo}] = \int_V \frac{1}{R}[BI]^T [A2]^T [D]\frac{1}{R}[A2][BI]dV$$

In this case shear strain is present and the constitutive matrix reduces to:

$$[D] = \frac{E}{(1-\nu^2)}\begin{bmatrix} 1 & 0 \\ 0 & \frac{(1-\nu)}{2} \end{bmatrix} = \frac{E}{(1-\nu^2)}[D']$$

Substituting in equation (5.35), taking the constant terms out-with the integral and noting that the strain is constant through the wall thickness, the out-of-plane bending stiffness equation is given by:

$$[K_{bbo}] = \frac{Etr^3}{(1-\nu^2)R^3}[B2]^T[I_{bo}][B2]$$

where

$$[I_{bo}] = \int_0^{\alpha}\int_0^{2\pi}[A2]^T[D'][A2]d\theta d\phi$$

Integrating and collecting terms gives:

$$[K_{bbo}] = \frac{EI}{(1-\nu^2)R^3}[B2]^T[J_{bo}][B2] \tag{5.36}$$

where

$$[J_{bo}] = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ & J_{22} & 0 & J_{24} & 0 & J_{26} \\ & & 0 & 0 & 0 & 0 \\ & symm. & & J_{44} & 0 & J_{46} \\ & & & & 0 & 0 \\ & & & & & J_{66} \end{bmatrix}$$

and

$$J_{22} = \frac{1}{4}(2-b)^2(2\alpha - \sin 2\alpha + \frac{1}{2}b^2\nu(2\alpha + \sin 2\alpha)$$

$$J_{24} = \left(\frac{1}{2}b^2\nu - \frac{1}{4}(2-b)^2\right)(1-\cos 2\alpha)$$

$$J_{26} = 2b\nu\sin\alpha$$

$$J_{44} = \frac{1}{4}(2-b)^2(2\alpha + \sin 2\alpha) + \frac{1}{2}b^2\nu(2\alpha - \sin 2\alpha)$$

$$J_{46} = 2b\nu(1-\cos\alpha)$$

$$J_{66} = 2\nu\sin\alpha$$

5.1.6 Out-of-Plane Ovalisation

The out-of-plane ovalisation stiffness matrix is obtained in a similar way to the in-plane matrix. The Von Karman ovalisation strains are again longitudinal extension and inextensional circumferential bending, as given by (5.25) and (5.26); however in the out-of-plane case, the radial displacement is represented by the Fourier series:

$$w_o = \sum_{n=1}^{N} b_n \sin 2n\theta$$

and the tangential displacement is obtained from the inextensibility condition as:

$$v_o = \sum_{n=1}^{N} b_n \frac{1}{2n} \cos 2n\theta$$

Substituting into the ovalisation strain-displacement equations:

$$\epsilon_{\phi oo} = \frac{1}{R}\sum b_n \left(\cos\theta \sin 2n\theta + \frac{1}{2n}\sin\theta\cos 2n\theta\right) \tag{5.37}$$

$$\epsilon_{\theta oo} = \frac{h}{r^2}\sum a_n \left(1-4n^2\right) \sin 2n\theta \tag{5.38}$$

As in the case of in-plane bending, N is chosen to be three and (5.27) and (5.28) are expanded and arranged in the form:

$$\{\epsilon_{oo}\} = [B_{oo}]\{b\} \tag{5.39}$$

where

$$\{\epsilon_{oo}\} = \{\epsilon_{\phi oo} \quad \epsilon_{\theta oo}\}^T \qquad \{b\} = \{b_1 \quad b_2 \quad b_3\}^T$$

Out-of-plane Ovalisation Stiffness.

The out-of-plane ovalisation stiffness matrix is defined by (5.5):

$$[K_{oo}] = \int_V [B_o]^T [D][B_o] dV$$

However, as in the in-plane case, this reduces to the integral

$$[K_{oo}o] = \frac{E r R \alpha}{1-\nu^2} \int_{-\frac{t}{2}}^{\frac{t}{2}} \int_0^{2\pi} [B_{oo}]^T [B_{oo}] d\theta d\zeta$$

which upon integration yields an out-of-plane ovalisation stiffness matrix identical to the in-plane matrix of equation (5.29).

5.1.7 Out-of-plane Bending-Ovalisation Coupling.

The bending-ovalisation stiffness matrix is defined by equation (5.6). However, in this case, the constitutive matrix includes torsional shear. As shear and ovalisation are not coupled in the Von Karman analysis, consideration of shear is omitted from (5.6). Thus the pertinent strain-displacement matrices are $[B_{oo}]$ from equation (5.39) and a reduced $[B_{bo}]$ from equation (5.35), with [A2] now given by:

$$[A2] = \begin{bmatrix} 0 & -(2-b)\sin\phi\sin\theta & 0 & (2-b)\cos\phi\sin\theta & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Thus

$$[K_{boo}] = \frac{E}{(1-\nu^2)R^2} \frac{r}{} [B2]^T \int_V [A2]^T [D][B_{oo}] dV$$

Integrating and rearranging gives

$$[K_{boo}] = \frac{EI}{(1-\nu^2)R^3} \frac{3R(2-b)}{4r} [B2]^T [J_{boo}]$$

where:

$$J_{boo} = \begin{bmatrix} 0 & 0 & 0 \\ (\cos\alpha - 1) & 0 & 0 \\ 0 & 0 & 0 \\ \sin\alpha & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

5.1.8 Static Condensation and Stiffness Matrix Assembly.

In order to reduce the element degrees of freedom to beam freedoms only, static condensation is applied to the in-plane and out-of-plane matrices. The procedure for both cases is identical, according to (5.9) and (5.10), where the stiffness matrices are in-plane or out-of-plane as appropriate.

Once the reduced 6x6 matrices are obtained they are assembled to give the fully representative 12x12 stiffness matrix, such that

$$\{F\} = [K]\{d\} \tag{5.40}$$

where

$$\{F\} = \{F_{X1} \quad F_{Y1} \quad F_{Z1} \quad M_{X1} \quad M_{Y1} \quad M_{Z1} \quad F_{X2} \quad \ldots\ldots \quad M_{Z2}\}^T$$

$$\{d\} = \{u_1 \quad v_1 \quad w_1 \quad \psi_{X1} \quad \psi_{Y1} \quad \psi_{Z1} \quad u_2 \quad \ldots\ldots \quad \psi_{Z2}\}^T$$

5.1.9 Stress and Strain Evaluation.

Element strain and stress are obtained from the equations

$$\{\epsilon\} = [B]\{d\} \tag{5.41}$$

and

$$\{\sigma\} = [D]\{\epsilon\} \tag{5.42}$$

upon solution for the global degree of freedom vector. As in-plane and out-of-plane modes are uncoupled, corresponding states of stress may be evaluated independently and summed.

The element displacement vector {d} in the stress and strain calculations is the full degree of freedom vector. The ovalisation degrees of freedom are recovered as described in Chapter 2 and the strain-displacement matrix of (5.41) is:

$$[B] = [[B_b][B_o]]$$

5.1.10 PB1 ANSYS User Element Programming.

Element PB1 was programmed as a user element for the ANSYS finite element package. ANSYS user elements are discussed in detail in Appendix 1.

The element position and orientation in three dimensional space is defined by two end nodes and an orientation angle Φ, where Φ defines the angle between the elbow local y axis and the global Y direction. The nodes are defined by standard ANSYS procedures. The element geometry definition is completed by defining bend radius, bend angle, cross-section mid-surface radius, wall thickness and orientation angle. Values are entered by the ANSYS **R** (Real constant) command. The ANSYS **R** input fields are as follows:

> **R**, element group number, bend radius R, bend angle α, mid-wall radius r, thickness t, orientation angle Φ

The required element material properties are Young's modulus EX and Poisson ratio NUXY. For thermal loading the coefficient of thermal expansion ALPX is also required.

The element thermal load vector is evaluated according to a matrix displacement method approach as follows. The original elbow length L is $L = R\alpha$. An applied temperature difference of ΔT gives rise to a thermal strain of

$$\epsilon^T = \alpha_T \Delta T$$

where α_T is the elbow material coefficient of thermal expansion. Consequently, the elbow experiences a change in length of u^T, where

$$u^T = L\epsilon^T = R\alpha\epsilon^T$$

This is equivalent to a change in bend radius of δ. The new length of the bend L' may therefore be written

$$L' = (R + \delta)\alpha = L + u^T$$

Therefore

$$R\alpha + \delta\alpha = L + u^T = R\alpha + R\alpha\epsilon^T$$

Hence the change in radius is given by the expression

$$\delta = R\epsilon^T = R\alpha_T \Delta T$$

For an unrestrained bend, the thermal expansion is equivalent to displacements of the bend radial degrees of freedom w_1 and w_2, and the local element degree of freedom vector is:

$$\{d\} = \{0 \quad 0 \quad -R\alpha_T \Delta T \quad 0 \quad 0 \quad -R\alpha_T \Delta T \quad 0 \quad 0 \quad 0\}^T$$

Converting to global co-ordinates,

$$\{d_g\} = [TR]^T \{d\}$$

In a constrained bend such thermal displacements give rise to a thermal load vector, which, from the standard stiffness equation, is given by the expression

$$\{F_g\} = [K_g]\{d_g\}$$

BEL1 displacement results may be printed and plotted in the usual ANSYS manner. Stress results for each element are evaluated at both nodes and written to a file PB1RES.DAT. In order to evaluate mechanical stress and strain in the element stress run, thermal displacements are subtracted from the total displacement vector.

The PB1 ANSYS user element source code is given in Appendix 4.1.

5.2 PB2: An Elbow Element with Linear Fourier Ovalisation.

In element PB1 the elbow ovalisation is constant with respect to axial position. However, under general loading, ovalisation varies with axial position and a number of elements (over which the constant ovalisation assumption is approximately valid) are required to give a converged solution. It is possible to reduce the number of elements required for convergence by increasing the order of axial interpolation of the element.

Element PB2 extends the PB1 formulation to include linear interpolation of ovalisation displacement in order to allow investigation of the relative

performance of the constant and linear assumptions. The beam bending stiffness matrix is identical to that of PB1 and the ovalisation and coupling stiffness matrices are defined in the following sections.

5.2.1 Ovalisation Stiffness.

The constant ovalisation strain-displacement equation for both in-plane and out-of-plane ovalisation may be written:

$$\{\epsilon_o\} = [B_{ocon}]\{a_{con}\}$$

where $\{a_{con}\}$ is the vector of constant Fourier coefficients a_1, a_2 and a_3 for in-plane bending and b_1, b_2 and b_3 for out-of-plane bending.

In element PB2 linear interpolation of the ovalisation deformation with respect to axial position is introduced. In the following derivation in-plane ovalisation is considered and the argument is applied to out-of-plane ovalisation in an identical fashion.

Choosing linear Lagrangian interpolation, a_n is written:

$$a_n = \left(1 - \frac{\phi}{\alpha}\right)a_{n1} + \left(\frac{\phi}{\alpha}\right)a_{n2}$$

or

$$a_n = N_1 a_{n1} + N_2 a_{n2}$$

where a_{n1} is the value of coefficient a_n at node 1, and a_{n2} the value of a_n at node 2. Thus $\{a_{con}\}$ becomes

$$\begin{Bmatrix} a_1 \\ a_2 \\ a_3 \end{Bmatrix} = \begin{Bmatrix} N_1 a_{11} + N_2 a_{12} \\ N_1 a_{21} + N_2 a_{22} \\ N_1 a_{31} + N_2 a_{32} \end{Bmatrix}$$

Defining the linear ovalisation degree of freedom vector $\{a\}$ to be

$$\{a\} = \{a_{11} \quad a_{21} \quad a_{31} \quad a_{12} \quad a_{22} \quad a_{32}\}^T$$

$\{a_{con}\}$ may be written

$$\begin{Bmatrix} a_1 \\ a_2 \\ a_3 \end{Bmatrix} = \begin{bmatrix} N_1 & 0 & 0 & N_2 & 0 & 0 \\ 0 & N_1 & 0 & 0 & N_2 & 0 \\ 0 & 0 & N_1 & 0 & 0 & N_2 \end{bmatrix} \{a\} = [N]\{a\}$$

The linear ovalisation strain-displacement equation may therefore be obtained by substituting the above relationship into (5.28):

$$\{\epsilon_o\} = [B_{ocon}]\{a_{con}\} = [B_{ocon}][N]\{a\} = [B_{ol}]\{a\}$$

so the linear ovalisation strain-displacement matrix is:

$$[B_{ol}] = [B_{ocon}][N] \tag{5.43}$$

The in-plane ovalisation stiffness matrix defined by (5.5) is

$$[K_{oo}] = \int_V [B_{ol}]^T [D][B_{ol}] dV$$

Substituting (5.43) into (5.5) and applying the integration limits gives

$$[K_{oo}] = \frac{E r R}{1-\nu^2} \int_o^{\alpha} \int_{-\frac{t}{2}}^{\frac{t}{2}} \int_0^{2\pi} [N]^T [B_{ocon}]^T [D][B_{ocon}][N] d\theta dh d\phi$$

In the case of in-plane ovalisation, the stiffness matrix is obtained by substituting the matrix $[B_{oi}]$ of (5.28) for $[B_{o\ con}]$. The out-of-plane ovalisation matrix is obtained in a similar manner and is found to be identical to the in-plane matrix.

5.2.2 Coupling Stiffness

The bending-ovalisation stiffness matrix is defined by equation (5.6):

$$[K_{bo}] = \int_V [B_b]^T [D][B_o] dV$$

which becomes:

$$[K_{boi}] = \int_V [B_b]^T [D][B_{ocon}][N] dV$$

In-plane and out-of-plane ovalisation and coupling matrices are obtained by substituting for the in-plane and out-of-plane bending and ovalisation matrices as appropriate.

5.2.3 Static Condensation.

Element PB2 has two nodes, each of which has 6 beam and 3 ovalisation degrees of freedom. In order to reduce the element degrees of freedom to beam freedoms only, static condensation is applied to the in-plane and out-of-plane matrices as in the case of element PB1. Once the reduced 6x6 matrices are obtained they are assembled to give the fully representative 12x12 stiffness matrix.

If the static condensation procedure were not applied and the full 18x18 element stiffness matrices assembled, a degree of ovalisation continuity similar to that of the original ADINAP element formulation [3.12] would be enforced between adjacent elements. However, it is not possible to program such a formulation as an ANSYS user element as ANSYS cannot accommodate such a nodal degree of freedom set, (see Appendix A1.3.4).

5.2.4 PB2 ANSYS User Element Programming.

The PB2 ANSYS user element source code is given in Appendix 4.2. The element modelling input requirements are identical to PB1, as described in Chapter 5.1.10. No thermal load vector has been programmed.

In element PB2 the ovalisation and coupling stiffness matrices are evaluated by numerical integration, using a 20x3x3 point Gaussian quadrature rule.

5.3 PB3: An Elbow Element with Constant Polynomial Ovalisation.

The majority of the beam-shell elbow elements reviewed in Chapter 3 employ Fourier interpolation of ovalisation displacement. The exceptions to this are the MARC element [3.1,3.2], and the Kanarachos and Koutsides element [3.23]

In the MARC element the elbow cross-section is modelled as a ring of axisymmetric shell elements, with discrete degrees of freedom at a variable

number of nodes around the circumference. Kanarachos and Koutsides used polynomial interpolation around the circumference, with discrete degrees of freedom at a set four nodes, stepped at 90^0 around the circumference.

In order to investigate the relative performance of Fourier and Polynomial interpolation of elbow ovalisation, element PB3 is proposed.

Element PB3 is a constant ovalisation element in which polynomial interpolation of ovalisation displacement is superposed on the beam model defined in Chapter 5.1. The ovalisation interpolation scheme is similar to that used in the formulation of the bellows element BEL3, as presented in Chapter 4. It is understood that Koutsides has formulated an elbow element using a similar interpolation scheme in [5.5]; however the writer was unable to obtain this reference[1]. As in the case of PB1, in-plane and out-of-plane responses are formulated independently and superposed to give a fully representative element. Ovalisation degrees of freedom are then statically condensed, resulting in a 12x12 element stiffness matrix.

5.3.1 In-Plane Ovalisation.

The ovalisation displacement is interpolated piecewise around the cross-section of the element by dividing it into four 90^0 arcs. Each arc has two associated end nodes, at which ovalisation degrees of freedom are defined. The ovalisation freedoms are tangential displacement v_0, radial displacement w_0 and rotation ψ as shown in Figure 5.4.

The tangential displacement v_0 is interpolated along each arc by quintic polynomial. (The choice of quintic polynomial interpolation is discussed in Chapter 4.4). Hence:

$$v_o = \alpha_1 + \alpha_2\beta + \alpha_3\beta^2 + \alpha_4\beta^3 + \alpha_5\beta^4 + \alpha_6\beta^5 \qquad (5.44)$$

The radial displacement w_0 is coupled to the tangential displacement by the inextensibility condition

$$w_o = -\frac{dv_o}{d\beta} \qquad (5.45)$$

1 Several letters to Dr. Koutsides remain unanswered.

and the rotation of the midsurface is defined under the Love-Kirchhoff hypothesis as:

$$\psi_o = \frac{1}{R}\left(\frac{dw_o}{d\beta} - v_o\right) \tag{5.46}$$

The ovalisation displacements are, therefore, defined in terms of the polynomial coefficients by the equation

$$\{u\} = [\beta]\{a\} \tag{5.47}$$

where

$$\{u\} = \{v_o \quad w_o \quad \psi_o\}^T$$

$$\{a\} = \{a_1 \quad a_2 \quad \ldots\ldots \quad a_6\}^T$$

and (5.48):

$$[\beta] = \begin{bmatrix} 1 & \beta & \beta^2 & \beta^3 & \beta^4 & \beta^5 \\ 0 & -1 & -2\beta & -3\beta^2 & -4\beta^3 & -5\beta^4 \\ -\frac{1}{r} & -\frac{\beta}{r} & -\frac{(2+\beta^2)}{r} & -\frac{(6\beta+\beta^3)}{r} & -\frac{(12\beta^2+\beta^4)}{r} & -\frac{(20\beta^3+\beta^5)}{r} \end{bmatrix}$$

The boundary conditions for a 90^0 arc with associated nodes *i* and *j* are

$$\text{at} \quad \beta = \frac{-\pi}{4} \qquad v_o = v_{oi} \quad w_o = w_{oi} \quad \psi_o = \psi_{oi}$$

$$\text{at} \quad \beta = \frac{+\pi}{4} \qquad v_o = v_{oj} \quad w_o = w_{oj} \quad \psi_o = \psi_{oj}$$

Thus the arc degrees of freedom $\{\Delta\}$ are defined in terms of the coefficients {a} by substituting the boundary conditions into (5.47)

$$\{\Delta\} = [C]\{a\} \tag{5.49}$$

where [C] is a 6x6 matrix of constants and

$$\{\Delta\} = \{v_{oi} \quad w_{oi} \quad \psi_{0i} \quad v_{oj} \quad w_{oj} \quad \psi_{oj}\}^T$$

The coefficients are defined in terms of the degrees of freedom by inverting (5.49) to give

$$\{\alpha\} = [C]^{-1}\{\Delta\} \tag{5.50}$$

The ovalisation displacements are then obtained in terms of the degrees of freedom by substituting for {a} in (5.47) to give:

$$\{u\} = [\beta][C]^{-1}\{\Delta\} = [\beta][E]\{\Delta\} \tag{5.51}$$

where [E] is the inverse of [C].

Thus, by definition, the shape function matrix for the 90^o arc, $[N_a]$, is given by:

$$[N_a] = [\beta][E]$$

However, this shape function is valid only for a single 90^o arc *i-j*. To interpolate around the section by four such arcs, continuity must be enforced at the cross-section nodes. This is done by creating a piecewise shape function for the entire cross-section of the bend in terms of one of displacement w_o.

By definition

> "A shape function N_i defines displacements within an element when the i^{th} degree of freedom has unit value and all other element degrees of freedom are zero" [2.23].

Assembling four 90^o arcs as in Figure 5.3, a ring element with degrees of freedom $\{d_r\}$ is formed, where:

$$\{d_r\} = \{v_{o1} \quad w_{o1} \quad \psi_{o1} \quad v_{o2} \quad \ldots\ldots \quad \psi_{o4}\}^T$$

Choosing w_o as the displacement to be interpolated around the circumference, the shape function N_1 for the ring is obtained by applying the nodal displacements:

$$w_{o1} = 1 \qquad w_{o2} = w_{o3} = w_{o4} = 0$$

By inspection, noting the symmetry of the deformation, the ring degree of freedom vector corresponding to N_1 is:

$$\{d_r\}_{N_1} = \left\langle 0 \quad 1 \quad 0 \quad -\frac{1}{2} \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad \frac{1}{2} \quad 0 \quad 0 \right\rangle^T$$

The corresponding ovalisation displacements {u} at a point between nodes 1 and 2 on the cross-section may be obtained by application of (5.51).

$$\{u_{12}\} = [\beta][F]\{\Delta_{12}\}$$

where

$$\{\Delta_{12}\} = \left\langle 0 \quad 1 \quad 0 \quad -\frac{1}{2} \quad 0 \quad 0 \right\rangle^T$$

Letting

$$\{G_{12}\} = [F]\{\Delta_{12}\}$$

this may be rewritten

$$\{u_{12}\} = [\beta]\{G_{12}\}$$

Similarly, in general

$$\{u_{ij}\} = [\beta]\{G_{ij}\}$$

$$\{G_{ij}\} = [E]\{\Delta_{ij}\}$$

where $\{\Delta_{ij}\}$ is a vector of degrees of freedom for nodes i and j.

Thus the piecewise displacement interpolation equations for shape function N_1 are obtained:

$$\{u_{12}\} = [\beta]\{G_{12}\} \qquad \text{(5.52 a-d)}$$

$$\{u_{23}\} = [\beta]\{G_{23}\}$$

$$\{u_{34}\} = [\beta]\{G_{34}\}$$

$$\{u_{41}\} = [\beta]\{G_{41}\}$$

Hence, for a general value of w_{o1}, the corresponding displacements are given by

$$\{u_{ij}\} = [\beta]\{G_{ij}\}w_{o1}$$

Shape functions N_2, N_3 and N_4 may be obtained as above, or more simply by considering the rotational symmetry of the cross-section. For example, the displacements corresponding to N_2 are given by:

$$\{u_{12}\} = [\beta]\{G_{41}\}w_{o2}$$

$$\{u_{23}\} = [\beta]\{G_{12}\}w_{o2}$$

$$\{u_{34}\} = [\beta]\{G_{23}\}w_{o2}$$

$$\{u_{41}\} = [\beta]\{G_{34}\}w_{o2}$$

N_3 and N_4 are obtained in a similar manner.

Considering the displacement contributions from all four selected degrees of freedom, the piecewise interpolation equations are:

$$\{u_{12}\} = [\beta][\{G_{12}\}\{G_{41}\}\{G_{34}\}\{G_{23}\}]\{w_o\}$$

$$\{u_{23}\} = [\beta][\{G_{23}\}\{G_{12}\}\{G_{41}\}\{G_{34}\}]\{w_o\}$$

$$\{u_{34}\} = [\beta][\{G_{34}\}\{G_{23}\}\{G_{12}\}\{G_{41}\}]\{w_o\}$$

$$\{u_{41}\} = [\beta][\{G_{41}\}\{G_{34}\}\{G_{23}\}\{G_{12}\}]\{w_o\}$$

where

$$\{w\} = \{w_{o1} \quad w_{o2} \quad w_{o3} \quad w_{o4}\}^T$$

These equations are written more briefly as:

$$\{u_{12}\} = [\beta][G_1]\{w_o\} \tag{5.53a}$$

$$\{u_{23}\} = [\beta][G_2]\{w_o\} \tag{5.53b}$$

$$\{u_{34}\} = [\beta][G_3]\{w_o\} \tag{5.53c}$$

$$\{u_{41}\} = [\beta][G_4]\{w_o\} \tag{5.53d}$$

5.3.2 Ovalisation Strain-Displacement.

By partitioning β as:

$$[\beta] = \begin{bmatrix} \{\beta_1\} \\ \{\beta_2\} \\ \{\beta_3\} \end{bmatrix}$$

the tangential and radial displacements between nodes *i* and *j* on the cross-section may be written

$$v_{oij} = \{\beta_1\}[G_i]\{w\} \tag{5.54}$$

$$w_{oij} = \{\beta_2\}[G_i]\{w\} \tag{5.55}$$

The general form of the element strain displacement equation is

$$\{\epsilon_{oi}\} = [B]\{w\}$$

where:

$$\{\epsilon_{oi}\} = \{\epsilon_{\theta oi} \quad \epsilon_{\phi oi}\}^T \qquad \{d\} = \{v \quad w \quad \psi\}^T$$

As in PB1, the Von Karman ovalisation strains are :

$$\epsilon_{\theta oi} = -\frac{\zeta}{r^2}\left(\frac{d^2 w_o}{d\theta^2} + w_o\right)$$

$$\epsilon_{\phi oi} = \frac{1}{R}(w_o \cos\theta - v_o \sin\theta)$$

Noting that $[G_i]$ and {w} are constant, substitution of (5.54) and (5.55) into the strain equations gives:

$$\epsilon_{\theta oi \quad ij} = \frac{-\zeta}{r^2}\left(\frac{d^2}{d\theta^2}\{\beta_2\} + \{\beta_2\}\right)[G_i]\{w\}$$

$$\epsilon_{\phi oi \quad ij} = \frac{1}{R}\{\{\beta_2\}\cos\theta + \{\beta_1\}\sin\theta\}[G_i]\{w\}$$

Thus the in-plane ovalisation [B] matrix is:

$$[B_{oi}] = [A][G_i]\{w\} \qquad (5.56)$$

where

$$[A] = \begin{bmatrix} \frac{-\zeta}{r^2}\left(\frac{d^2}{d\theta^2} + 1\right)\{\beta_2\} \\ \frac{1}{R}(\{\beta_2\}\sin\theta + \{\beta_1\}\cos\theta) \end{bmatrix}$$

5.3.3 In-Plane Ovalisation and Coupling Stiffness Matrices.

Having thus defined the ovalisation strain-displacement behaviour of an axisymmetric shell ring, the ovalisation stiffness matrix is given by:

$$[K] = \int_V [B_o]^T [D][B_o] dV$$

where [D] is the constitutive matrix:

$$[D] = \frac{E}{1-\nu^2}\begin{bmatrix} 1 & \nu \\ \nu & 1 \end{bmatrix} = \frac{E}{1-\nu^2}[D_1]$$

In effect the above formulation defines an axisymmetric shell ring, which includes both elastic and rigid body deformation modes. The rigid body modes are shown in Figure 5.5; the first is strain-free mode parallel to the axisymmetric axis, the second a straining mode perpendicular to the axisymmetric axis. However, in beam shell elbow elements the ovalisation model is required to include local shell deformation modes only, as rigid body modes are included in the element's beam model.

This problem is overcome in MARC Element 17 [3.1], by suppressing ovalisation rigid body modes at element level. However, here a different approach is taken. Rather than suppress the rigid body modes in the ring stiffness matrix, an elastic ovalisation mode is extracted from the stiffness equation.

The elastic ovalisation mode is illustrated in Figure 5.5. This constrains the ring degree of freedom vector of (5.53) to the form

$$\{w_o\} = \{-b_1 \quad b_2 \quad -b_1 \quad b_2\}^T$$

which is symmetric about the centroid of the beam. The ovalisation behaviour is therefore defined in terms of two nodeless degrees of freedom, b_1 and b_2. The general displacement interpolation equations (5.53) can now be rewritten in terms of these two degrees of freedom to give:

$$\{u_{12}\} = [\beta][H_i]\{b\} \tag{5.58}$$

where

$$[H_i] = \begin{bmatrix} -(G_{11}+G_{13}) & (G_{12}+G_{14}) \\ -(G_{21}+G_{23}) & (G_{22}+G_{24}) \\ etc. & etc. \end{bmatrix}$$

and

$$\{b\} = \begin{Bmatrix} b_1 \\ b_2 \end{Bmatrix}$$

Substituting (5.58) into the strain-displacement equations gives

$$[B_{oi}] = [A][H_i]\{b\}$$

where [A] is defined in (5.56).

The in-plane ovalisation and coupling stiffness matrices are obtained by evaluating the integrals

$$[K_{ooi}] = \int_V [B_{oi}]^T [D][B_{oi}] dV$$

$$[K_{boi}] = \frac{1}{R}[BI]^T \int_V [AI]^T [D][B_{oi}] dV$$

as in the PB1 procedure.

5.3.4 Out-of-Plane Ovalisation and Coupling Stiffness Matrices.

The out-of-plane ovalisation and coupling matrices may be obtained in the manner given above for the in-plane case. The out-of-plane bending ovalisation behaviour of the cross-section is equivalent to a rotation of the in-plane mode around the centroid of the elbow. In Fourier ovalisation models, the rotation angle is 45^o, and this value has been verified experimentally. It is therefore possible to evaluate the out-of-plane ovalisation stiffness and coupling matrices by rotating the in-plane model through 45^o. This is done simply by re-defining θ in the in-plane solution as $\theta + \pi/4$ in the out-of-plane analysis.

The resulting out-of-plane ovalisation stiffness matrix is identical to the in-plane matrix. The coupling matrix is given by

$$[K_{boo}] = \int_V [B_{bo}]^T [D][B_{oo}] dV$$

where $[B_{bo}]$ is the out-of-plane bending [B] matrix of (5.35) and $[B_{oo}]$ the two mode ovalisation matrix. Thus

$$[K_{boo}] = \frac{r}{R^2}[BI]^T \int_V [A2]^T [D][B_{oo}]dV$$

5.3.5 Static Condensation

The 8x8 in-plane and out-of-plane stiffness matrices are statically condensed to remove the ovalisation degrees of freedom, as described in Chapter 4.1.2, before the fully representative 12x12 elbow stiffness matrix is assembled.

5.3.6 PB3 ANSYS User Element Programming.

The PB3 ANSYS user element source code is given in Appendix 4.3. The element is defined in a similar manner to PB1, as described in Chapter 5.1.10, however no thermal load vector has been programmed at this time.

The element ovalisation and coupling stiffness matrices are integrated piecewise through the volume of the elbow, by applying a 5x3x3 point Gaussian quadrature rule to each of the four arcs making up the cross-section. Considering the in-plane matrices, it is seen from (5.14) that the circumferential position of a point in the element is defined by two angles; β and θ. In order to evaluate the element stiffness it is necessary to consider only one angle in the integration. For computing considerations it is chosen to substitute θ in terms of β for each section of circumference as follows:

$1-2 \qquad \theta = \beta + \frac{\pi}{4}$

$2-3 \qquad \theta = \beta + \frac{3\pi}{4}$

$3-4 \qquad \theta = \beta + \frac{5\pi}{4}$

$4-1 \qquad \theta = \beta + \frac{7\pi}{4}$

Hence the integration limits for each arc in the cross-section are from $-\pi/4$ to $\pi/4$ around each arc, -t/2 to t/2 through-thickness and 0 to β along the axis of the bend.

For the out-of-plane matrices θ of the in-plane solution is re-defined as $\theta + \pi/4$, and the matrices integrated as above.

5.4 Discussion.

In this chapter formulations for three simple elbow elements have been presented. The elements have been programmed as user elements for the ANSYS finite element program. The user element source code is given in Appendix 4.

The stiffness matrix of element PB1 was integrated analytically using a symbolic algebra program. This dispenses with the need for computationally expensive numerical integration. In the case of elements PB2 and PB3, the integrated beam stiffness matrix of PB1 was used but the ovalisation and ovalisation-bending coupling matrices are integrated numerically at element formation.

A number of piping elbow and piping system analyses using the above elements are presented in Chapter 6, where performance is assessed in comparison with flexibility analysis, finite element analysis and published theoretical and experimental solutions.

5.5 References

[5.1] V Z VLASOV
Thin-Walled Elastic Beams, 2nd Edition (Translated from Russian), Published for the Nat. Sci, Foundation, Washington DC, and the Department of Commerce, USA, by the Israel Program for Scientific Translations, Jerusalem, (1961).

[5.2] S K CHAUNDHURI and S SHORE
"Thin-Walled Curved Beam Finite Element," Proc. ASCE, J. Eng. Mech. Div., Vol 103, No. EM5, pp 921-937, (1977).

[5.3] C H YOO
"Matrix Formulation of Curved Girders," J. Eng. Mech. Div. ASCE, Vol. 105, pp 971-988, (1979).

[5.4] C H YOO
"A Consistent Discrete Elements Technique for Curved Members," Comp. & Str., Vol. 25, No. 1, pp 137-146, (1987).

[5.5] N KOUTSIDES
"New Effective Finite Elements for Thin-Walled Cross-Sections," PhD Thesis, National Technical University of Athens, (1986).

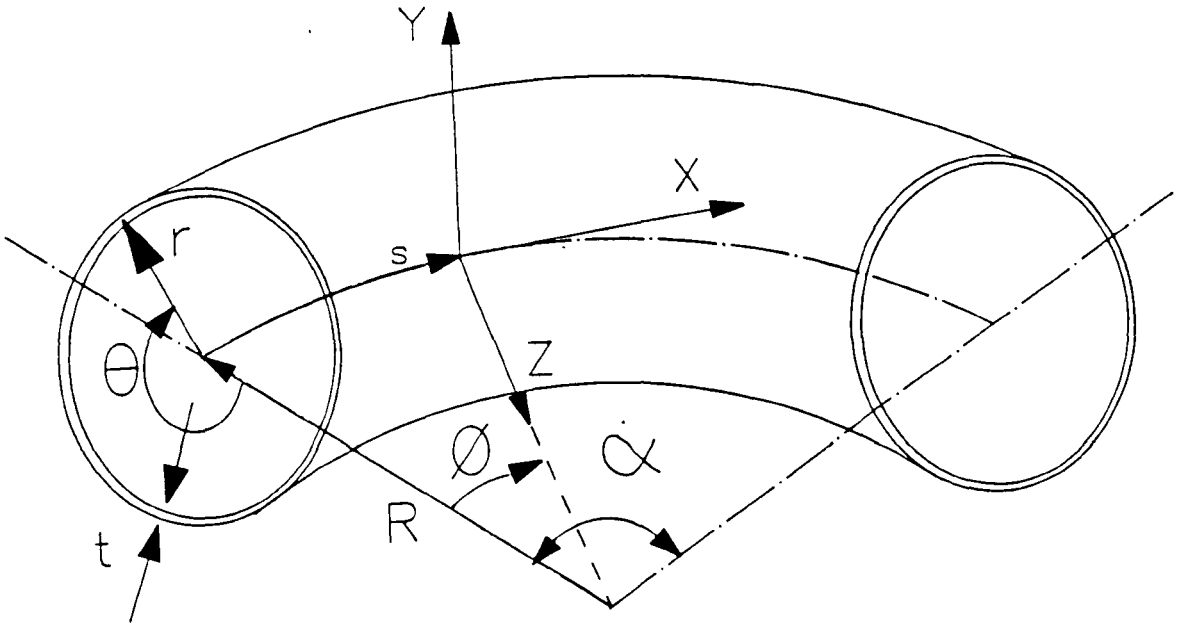


Figure 5.1 Elbow element beam model geometry and coordinate systems.

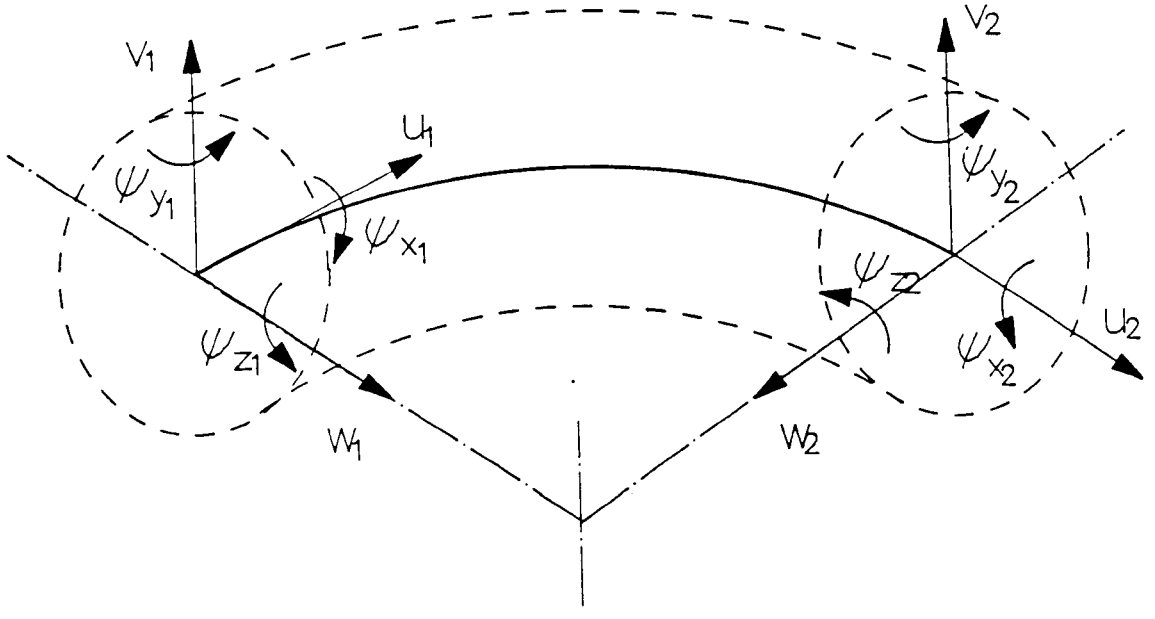


Figure 5.2 Elbow element beam model degrees of freedom.

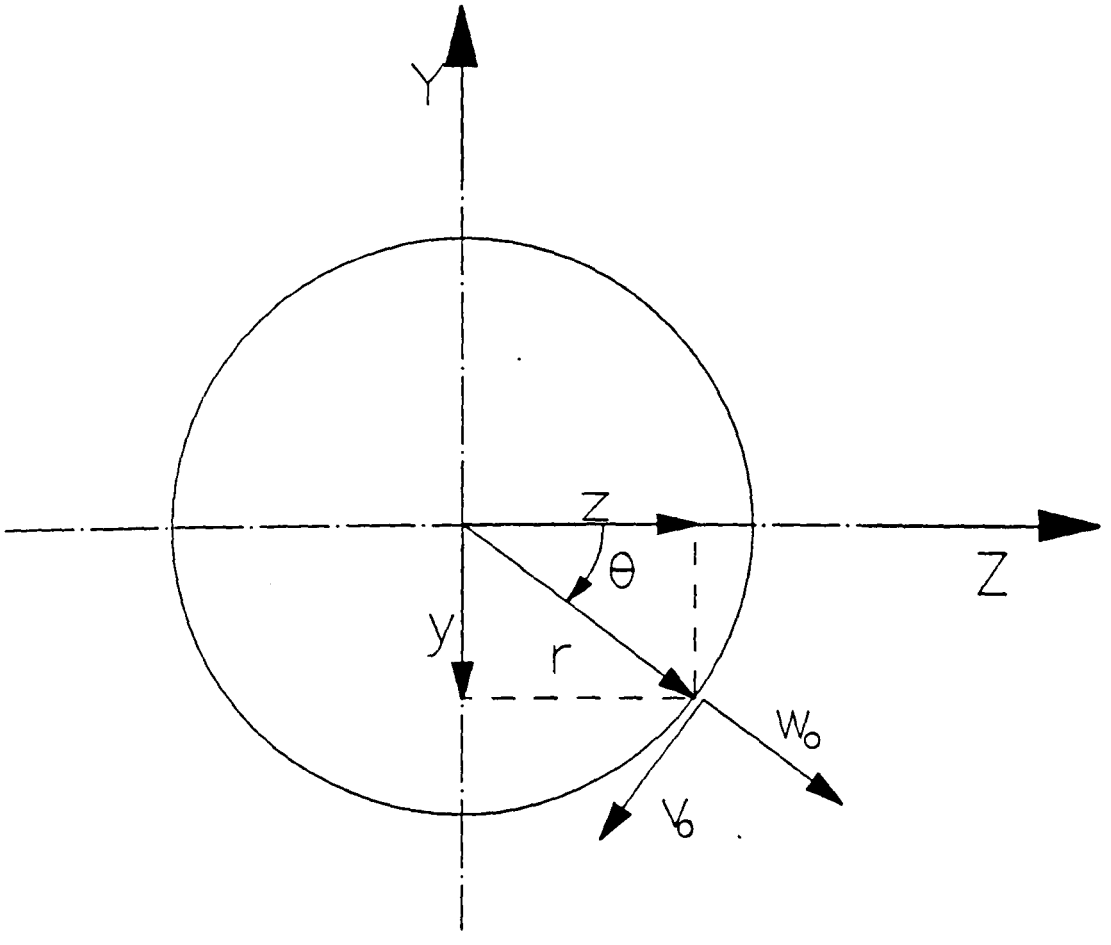


Figure 5.3 Element PB1/PB2 cross-section coordinate system and ovalisation degrees of freedom.

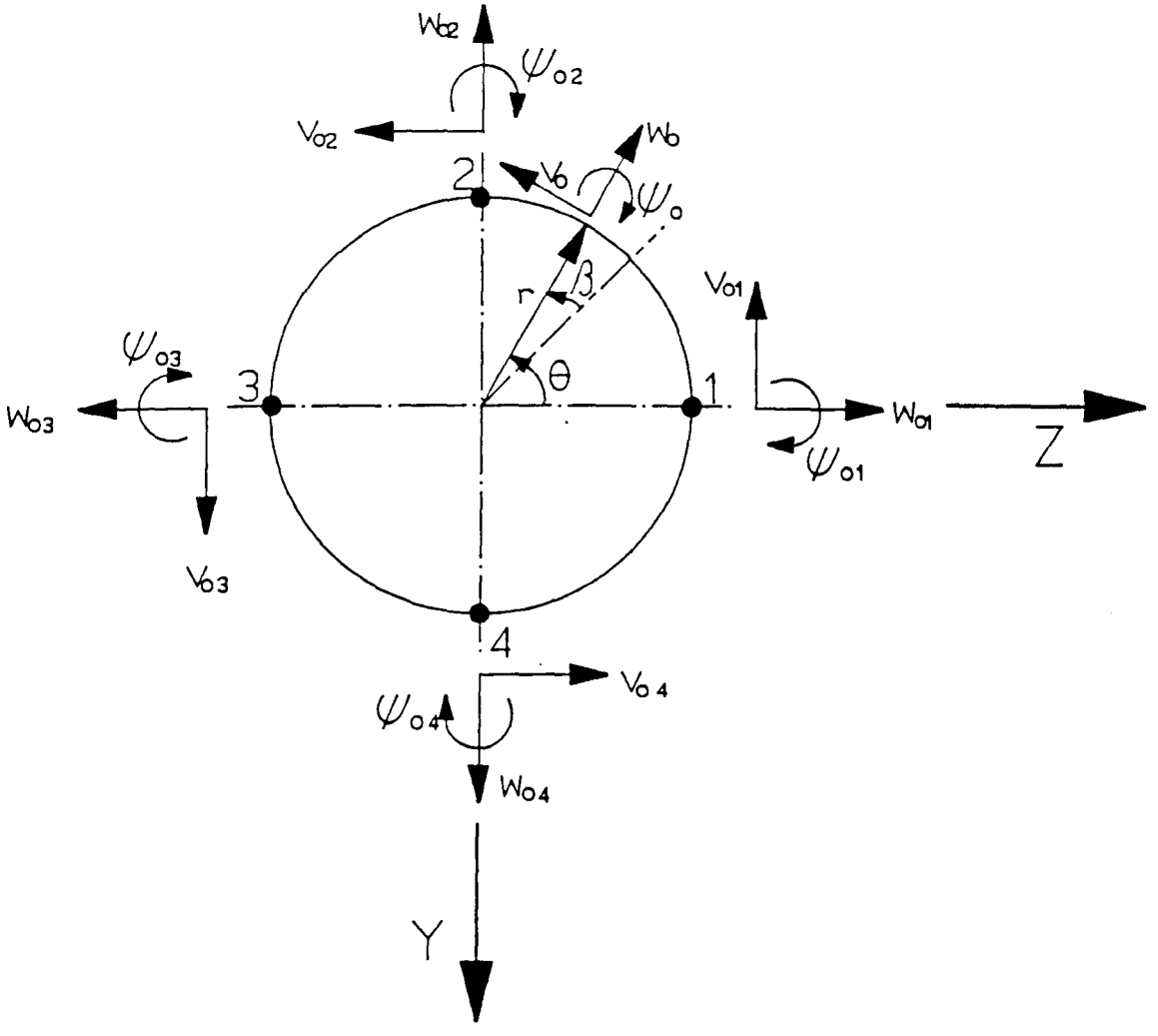


Figure 5.4 Element PB3 cross-section coordinate system and ovalisation degrees of freedom.

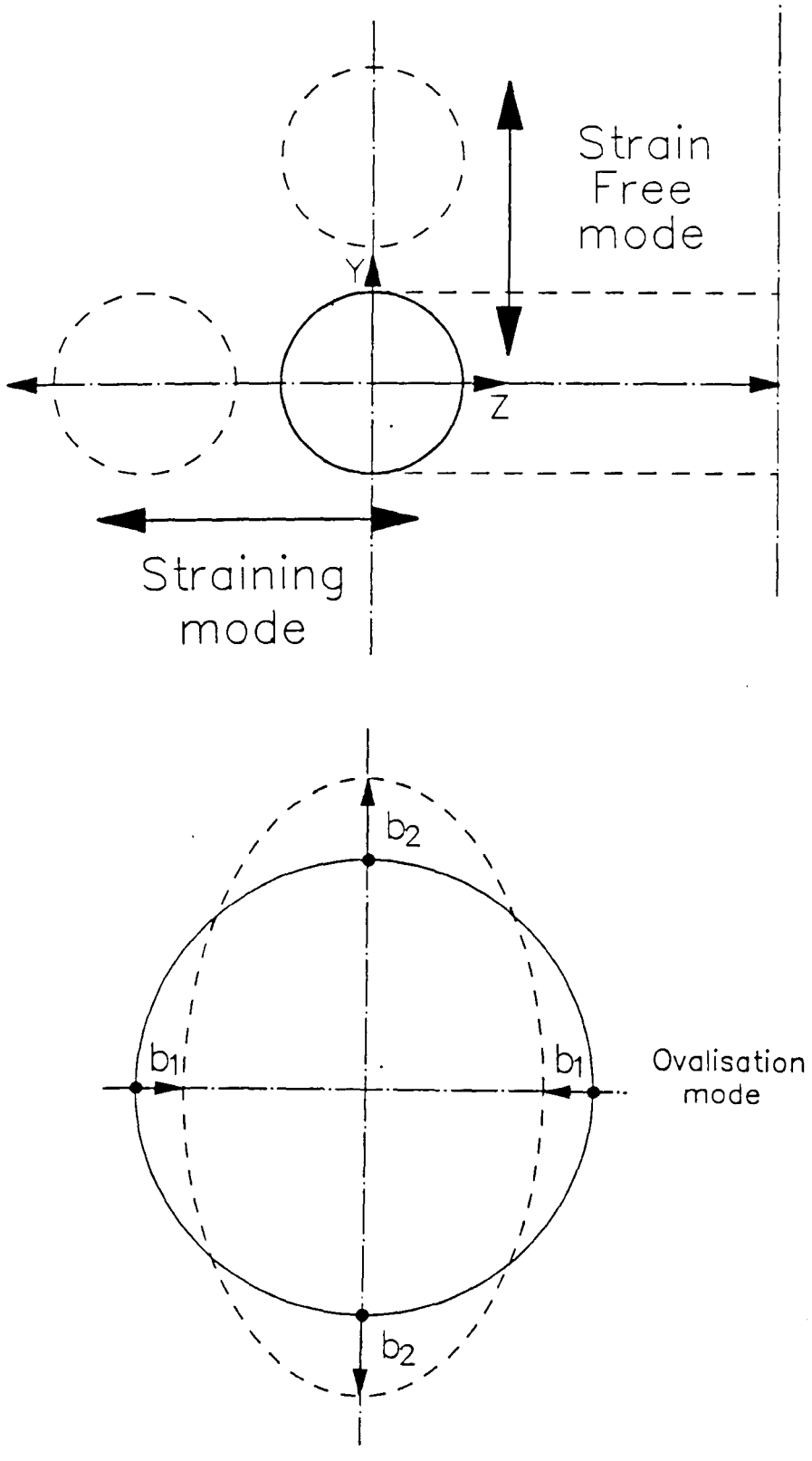


Figure 5.5 Element PB3 ovalisation model rigid body modes and elastic ovalisation mode.

CHAPTER 6.

SAMPLE ANALYSES OF BELLOWS, ELBOWS AND PIPING SYSTEMS.

6 SAMPLE ANALYSES OF BELLOWS, ELBOWS AND PIPING SYSTEMS.

In this chapter the accuracy, applicability and effectiveness of the bellows formulations of Chapter 4 and the elbow elements of Chapter 5 are assessed by presenting and discussing the results of sample analyses of bellows, elbows and piping systems.

In Section 6.1 several bellows of geometry parameter range $0.5 < \lambda < 5.0$ are analysed using elements BEL1, BEL2, BEL3 and BEL4. Flexibility and stress results are compared with published solutions and solutions obtained by axisymmetric thin shell finite element analysis. The relative performance of the elements is discussed and it is argued that element BEL1 gives the best performance of the four bellows elements.

In section 6.2 the performance of the curved beam model used in elbow elements PB1, PB2 and PB3 is verified by presenting two sample analyses of canti levered 90^{o} curved beams and comparing the results with ANSYS curved beam solutions.

In Section 6.3 several analyses of single 90^{o} pipe bends under moment loading are presented. Four bends under in-plane bending and three under out-of-plane bending are analysed. Displacements and stresses are compared with flexibility analysis and published experimental and theoretical results. The performance of the elbow elements is discussed and it is argued that element PB1 is the most effective element.

In Section 6.4 analyses of nine piping systems are presented. The first eight systems include at least two bends. Results obtained by analyses using elbow element PB1 in the ANSYS finite element program are compared with standard ANSYS flexibility analysis, ABAQUS elbow element analysis or thin shell finite element analysis results. The remaining system is a straight piping run which incorporates a semi-toroidal bellows unit. Results obtained using the bellows element BEL1 are compared with two ANSYS flexibility analyses; the first taking default values for bellows flexibility and stress intensification factors and the second based on axial stiffness and stress intensification factors derived from the bellows element BEL3 formulation.

6.1 Bellows Analyses.

The object of studying bellows under axial force loading is to establish the accuracy and applicability of the bellows element formulations BEL1, BEL2, BEL3 and BEL4 presented in Chapter 4.

Bellows of geometric parameter $0.5 < \lambda < 5.0$ and fixed radius ratio $R/r = 10$ were examined. Flexibility and stress results given by the bellows formulations are compared with published and axisymmetric finite element analysis results.

The axisymmetric models were created in the ANSYS finite element program. 18 STIF61 axisymmetric conical thin shell elements [4.5,6.1] were used to model a 90^{o} section of bellows and boundary conditions conforming to the element formulations applied. The axisymmetric bellows model is shown in Figure 6.1. In the remainder of this Chapter, results obtained by axisymmetric analysis are denoted FE(Axi).

6.1.1 Flexibility results.

In the literature, bellows deformation results are generally presented in the form of flexibility factors evaluated for particular values of λ. In order to allow direct comparison with such results, the bellows deformation results presented in this Chapter are given in this form.

Flexibility factors given by the bellows element formulations for a range of bellows parameters λ are presented in Figure 6.2.

In Figure 6.2a, flexibility factors from elements BEL1 and BEL2 are compared with axisymmetric shell finite element analysis FE(Axi) results and two solutions of Findlay and Spence.

The Findlay and Spence curves FS(1) and FS(2) were obtained by a minimum potential energy analysis of a semi-toroidal bellows [4.2]. FS(1) is based on the same strain-displacement relationships and displacement interpolation as the element BEL1. FS(2) is similar to the element BEL2 formulation in terms of shell theory and displacement function, but with more terms in the interpolation series.

Element BEL1 flexibility factors are in good agreement with both the FS(1) and FE(Axi) results. However, BEL2 gives lower flexibility factors more consistent with the FS(2) solution.

The flexibility factors given by the polynomial based elements BEL3 and BEL4 are shown in Figure 6.2b. The two elements give an almost identical solution, which shows good agreement with the BEL1 solution.

6.1.2 Stress results.

In practice, the largest stress occurring in an axially loaded bellows is the axial stress arising from transverse bending of the bellows convolution. For conciseness, stress results presented in this section are limited to such axial stresses.

Stress results given by the element formulations are presented in two forms: normalised maximum axial stresses and normalised axial stress distributions for given λ. The stresses are normalised according to the equation:

$$\sigma^* = \sigma \frac{t^2 R \pi}{3 r P}$$

where P is the applied axial force.

Maximum axial stresses are compared with solutions of Boyle and Spence, Hamada *et al* and FE(Axi) in Figure 6.3. The solutions of Boyle [6.2] and Hamada [6.3] are based on numerical analysis of thin shells of revolution.

Elements BEL3 and BEL4 were found to give almost identical maximum stress values and, for clarity, only BEL3 stresses are shown in Figure 6.3.

The best overall agreement with the FE(Axi) maximum stresses is given by BEL2, which also compares well with Boyle's solution. Elements BEL1, BEL3 and BEL4 give reasonable agreement with these solutions, although the stresses are slightly higher. Hamada's solution differs from the rest of the results, especially for low values of λ.

The axial stress distribution curves given by FE(Axi) and the bellows elements are shown in Figures 6.4 to 6.8.

The most consistent agreement with the FE(Axi) stress distribution is given by BEL1. Although BEL2 has been shown to give good maximum stress values, it clearly gives very poor agreement with FE(Axi) and BEL1 for stress distribution. The polynomial elements BEL3 and BEL4 give a slightly different form of stress distribution to FE(Axi), but there is reasonable agreement for stress magnitude. Finally, it is noted that in all cases the elements showed a marked deterioration in performance for values of bellows parameter $\lambda \leq 0.5$.

6.1.3 Discussion of Bellows Analyses.

From the above results it is concluded that element BEL1 gives the best overall performance in terms of flexibility and stress evaluation. At present, the element is limited to analysis of bellows of parameter λ greater than 0.5 but the range of geometries could be extended to include lower values of λ by increasing the number of terms taken in the interpolation series.

Element BEL2 performs poorly in terms of flexibility and stress distribution. As in BEL1, the element could be improved by taking more terms in the interpolation series, however, the Findlay and Spence solution FS(2) of [4.2] indicates that solution convergence will occur before the BEL1 flexibility is reached. In the limit, the full series of PB2 will contain all the terms in the even series used in PB1. Therefore, the poor performance is due to either the strain-displacement equations or the choice of radial displacement w as the interpolated variable, from which the tangential displacement v and rotation ψ are derived according to (4.27) and (4.28) respectively.

The polynomially interpolated elements, BEL3 and BEL4, are based on the same strain-displacement model as BEL2 but give better flexibility factors and stress distribution curves than the Fourier based element. This indicates that the strain-displacement model can adequately describe the bellows behaviour. However, unlike BEL2, in which radial displacement w was the interpolated variable, BEL2 and BEL3 are based on interpolation of tangential displacement v. This indicates that the choice of w as the interpolated variable was the cause of the poor performance of BEL2.

In considering the relative performance of BEL3 and BEL4, it is clear that the inclusion of constant axial extension in BEL4 gives little improvement over the inextensible element BEL3, indicating that constant membrane strain is negligible in the deformation of bellows of the geometries considered here.

The form of stress distribution given by the polynomially interpolated elements is smoother than the axisymmetric shell finite element distribution and, as with the other elements, performance deteriorates rapidly for values of λ less than 0.5. It would be possible to improve the performance of these elements by using more quintic polynomial "elements" in the 90^o section. Adopting this approach, degrees of freedom could be defined at internal nodes (that is $0 < \theta < 90$) and statically condensed at element level.

6.2 Curved Beam Verification.

In order to verify the curved beam element formulation used in elbow elements PB1, PB2 and PB3, in-plane and out-of-plane loading of two cantilevered 90^o curved beams was examined. The curved beam formulation was programmed as an ANSYS user element. (The user element code is incorporated in the elbow user element routines given in Appendix 4. These elements may be used for simple beam analyses by suppressing the Fourier ovalisation modes).

As the beam is based on an exact solution of Vlasov's curved beam theory, only a single element was required to model the beam. Results are compared with the ANSYS elbow element STIF18 with a unit flexibility factor [4.5,6.1]. The ANSYS STIF18 element stiffness matrix is evaluated by inverting a curved beam flexibility matrix, obtained by application of Castigliano's theorem [6.1,6.4].

The first beam examined was a slender member with dimensions R = 5000 mm, r = 100 mm, t = 10 mm and material properties E = 210E3 and ν = 0.3. The beam radius ratio and geometric parameter are

$$\frac{R}{r} = 50 \qquad \lambda = \frac{Rt}{r^2} = 5$$

The beam was loaded by in-plane and out-of-plane shear forces of 100 N applied at the free end. The corresponding displacements δ evaluated using single element PB1 (no ovalisation) and STIF18 (flexibility factor = 1) models are given in Table 6.1.

Element	$\delta_{i/p}$	$\delta_{o/p}$
PB1 (no ov.)	1.488	2.365
STIF18 (FF=1)	1.486	2.363
$\left(\frac{PB1}{STIF18}-1\right)\%$	0.1	0.1

Table 6.1 Curved Beam 1 displacements due to shear loading.

The second beam examined was less slender, with dimensions R = 1000 mm, r = 100 mm, t = 10 mm and material properties E = 210E3 and ν = 0.3. The beam radius ratio and geometric parameter are

$$\frac{R}{r} = 10 \qquad \lambda = \frac{Rt}{r^2} = 1$$

Two types of loading were applied to the beam; in-plane and out-of-plane shear forces of 1000 N and in-plane and out-of-plane moments of 1E6 Nmm applied at the free end. The corresponding displacements δ and rotations ψ given by single element PB1 (no ovalisation) and STIF18 (flexibility factor = 1) models are given in Tables 6.2 and 6.3.

Element	$\delta_{i/p}$	$\delta_{o/p}$
PB1 (no ov.)	0.1196	0.1892
STIF18 (FF=1)	0.1224	0.1950
$\left(\frac{PB1}{STIF18}-1\right)\%$	-2.3	-2.97

Table 6.2 Curved Beam 2 displacements due to shear loading.

Element	$\psi_{i/p}$	$\psi_{o/p}$
PB1 (no ov.)	0.1511	0.1743
STIF18 (FF=1)	0.1512	0.1739
$\left(\frac{PB1}{STIF18}-1\right)\%$	-0.0	0.2

Table 6.3 Curved Beam 2 displacements due to moment loading.

Tables 6.1, 6.2 and 6.3 show that the Vlasov curved beam element is slightly stiffer than the ANSYS curved beam element for shear loading of beam 2. However the beam 1 results and beam 2 rotations closely agree with the ANSYS curved beam solution.

6.3 Elbow Analyses.

In order to investigate the accuracy and applicability of the elbow elements formulated in Chapter 5, several single 90^{o} pipe bends under pure in-plane and out-of-plane moment loading were analysed.

The elbow elements presented in Chapter 5 are based on the assumption that the ovalisation is constant (PB1, PB3) or linear (PB2) over the length of the element. Therefore, under general loading, each bend must be discretized into several elements over which the assumption is approximately satisfied. The first stage in the study of the behaviour of the pipe bends is to determine the number of elements required to meet this condition for different bend geometries; that is, to determine the convergence requirements of the elements.

6.3.1 Element Convergence.

Two sets of convergence tests were carried out on 90^{o} bends of different bend parameter λ. In one set the long radius assumption $R >> r$ made in the element formulations was observed and a radius ratio of $R/r = 10$ considered. In the second set a less rigorous application of the assumption was made and a radius ratio of $R/r = 3$ used.

The applied loading in the convergence tests was in-plane or out-of-plane shear force. Shear loading sets up a more complex state of strain in the elbows than pure bending and is more representative of general loading conditions.

Typical in-plane and out-of-plane convergence test finite element models are shown in Figure 6.9. The models were fully fixed at one end and loaded by an applied shear force at the other, in-plane or out-of-plane as appropriate. Starting with a mesh of one element per bend, the number of elements was increased until convergence occurred. The convergence criteria was that the difference in the shear translational displacement (corresponding to the applied force) on increasing the number of elements was $< 0.5\%$.

Convergence plots for in-plane shear force loading of 90^o bends of different λ for elements PB1, PB2 and PB3 are shown in Figures 6.10 to 6.12 respectively. Corresponding plots for out-of-plane shear force loading are shown in Figures 6.13 to 6.15. Convergence is represented as displacement for a given number of elements, d, over converged displacement, dc.

6.3.2 Sample Analyses: In-Plane Bending of 90^o Bends

In order to study the accuracy and applicability of elements PB1, PB2 and PB3 under in-plane bending, four 90^o bends were analysed. The bends are identified by the names IP1, IP2, IP3 and IP4. Bend dimensions, material properties and applied moments used in the finite element analyses are given in Table 6.4.

BEND	E	ν	MY	R	r	t	λ	R/r
IP1	2.82E7 (lbf/in^2)	0.3	-1E4 (lbfin)	45 (in)	14.74 (in)	0.515 (in)	0.107	3.05
IP2	210E3 (N/mm^2)	0.3	-1E8 (Nmm)	1000 (mm)	173.01 (mm)	13.35 (mm)	0.446	5.78
IP3	210E3 (N/mm^2)	0.3	-1E8 (Nmm)	1000 (mm)	325.73 (mm)	15.66 (mm)	0.147	3.07
IP4	210E3 (N/mm^2)	0.3	-1E8 (Nmm)	1000 (mm)	347.22 (mm)	13.02 (mm)	0.108	2.88

Table 6.4. Elbow geometry for in-plane bending tests.

Experimental and theoretical analyses of a bend similar to bend IP1 were presented by Rodabaugh and George in [2.10]. Bend IP2 has been investigated experimentally by Smith and Ford in [6.5], where a theoretical solution according to Smith's in-plane bending theory [2.16] is also presented. Further analysis of bend IP2 was performed by Bathe and Almeida using the ADINAP elbow element in [3.12]. In [3.12] Bathe and Almeida analysed a bend similar to IP3 using the ADINAP element and compared the results with an earlier MARC Element 17 analysis of Sobel [3.4] and a solution obtained the Clark and Reissner shell theory solution [2.9]. A ring element analysis of a bend similar to IP4 was presented by Ohtsubo and Watanabe in [3.5], where the results were compared with experimental values from the Japan Welding Engineering Society.

In the following analyses, bend stress results are presented in normalised form. For consistency with published results, two different normalising equations have been used.

Stress results for bends IP1 and IP4 are expressed in terms of a stress factor σ^*; the ratio of the elbow stress to the stress at radius r in a straight pipe of equivalent nominal dimensions under the similar moment loading. The stress factor is therefore defined as

$$\sigma^* = \sigma\left(\frac{I}{Mr}\right)$$

where I is the second moment of area of the pipe and M the moment at the section at which the stress is measured or calculated; $M = M_y$ for in-plane bending.

Stress results for bends IP2 and IP3 are also presented in a normalised form but, for consistency with the Bathe and Almeida published results, a slightly different definition of stress factor is used:

$$\sigma_o^* = \sigma\left(\frac{I}{Mr_o}\right)$$

where r_o is the outer radius of the elbow cross-section.

Convergence requirements.

Convergence plots for 90^o bends loaded by in-plane shear force are given in Figures 6.10 to 6.12. The number of elements required for a converged solution for bend parameters of $\lambda = 0.1, 0.2, 0.5$ are summarised in Table 6.5.

Element	Number of Elements					
	λ = 0.1		λ = 0.2		λ = 0.5	
	R/r=10	R/r=3	R/r=10	R/r=3	R/r=10	R/r=3
PB1	5	5	4	4	3	4
PB2	3	3	3	3	2	3
PB3	5	4	4	4	4	4

Table 6.5 Element in-plane convergence requirements.

Considering the bend geometry given in Table 6.4, it is seen that between three and five elements of types PB1, PB2 and PB3 are required to give converged solutions for bends IP1, IP2, IP3 and IP4 under in-plane shear force. However, fewer elements are required under the simpler in-plane bending load case and

further tests indicated that three elements per bend are sufficient for all the above geometries. The three element model is shown in Figure 6.16, which defines the model co-ordinate system, applied force and displacement boundary conditions.

Displacement Results: Comparison with ANSYS STIF18.

Analysis results presented in the elbow element literature are in the main confined to stress distribution: displacements are not generally presented. It is not, therefore, possible to assess the displacement results given by elements PB1, PB2 and PB3 by comparison with alternative elbow element solutions from the literature. However, it is necessary to verify the element performance and this was done by comparing displacement results given by the elements with results obtained by flexibility analysis of the bends defined above.

The free-end rotations of the bends were evaluated using elements PB1, PB2 and PB3 and compared to results given by the ANSYS STIF18 flexibility analysis elbow element [4.5,6.1]. The basic STIF18 stiffness matrix is evaluated by inverting a curved beam flexibility matrix obtained by application of Castigliano's theorem. The resulting curved beam stiffness matrix is then modified by flexibility factors to account for the effect of elbow ovalisation. The default STIF18 flexibility factors are evaluated according to the ANSI B31.1 definition, as discussed in Chapter 2.1.2.

The elbow element and flexibility analysis results are given in Tables 6.6. It is found that for a converged solution elements PB1 and PB2 yield virtually identical results and for clarity a single rotation value is given for both these elements. The percentage difference between the elbow elements and ANSYS STIF18 results are also given in Table 6.6.

Element	IP1 ROTY x 10^5	IP2 ROTY x 10^2	IP3 ROTY x 10^2	IP4 ROTY x 10^2
PB1/PB2	-7.17	-1.17	-4.72	-6.40
PB3	-6.96	-1.16	-4.64	-6.21
STIF18	-7.48	-1.27	-4.91	-6.67
$\left(\frac{PB1}{STIF18}-1\right)\%$	-4.2%	-7.9%	-3.9%	-4.0%
$\left(\frac{PB3}{STIF18}-1\right)\%$	-7.0%	-8.7%	-5.5%	-6.9%

Table 6.6 In-plane bending rotations.

Considering the results given in Table 6.6, it is seen that the elements PB1, PB2 and PB3 give slightly lower rotations than ANSYS element STIF18. Considering the elbow element results alone, it is seen that PB1 and PB2 give greater rotations than PB3, indicating that the polynomial ovalisation formulation gives a stiffer element than the Fourier interpolation.

Stress Distribution

Normalised stress distributions for bends IP1, IP2, IP3 and IP4 given by the elbow elements PB1, PB2 and PB3 are compared with published results in Figures 6.17 to 6.21. For a converged solution elements PB1 and PB2 give virtually identical results and, for clarity, a single curve representing both of these elements is given in the Figures.

In Figure 6.17 the longitudinal and circumferential stress factors σ^* at the outside surface of bend IP1 are compared with the theoretical and experimental values of Rodabaugh and George.

Good general agreement is seen between the finite element solutions of PB1, PB2 and the Rodabaugh and George theoretical solution. There is some difference between the PB1/PB2 solution and the experimental results for longitudinal stress towards the intrados of the bend, but the circumferential stresses are similar throughout. Element PB3, however, is seen to give a high peak circumferential stress and the form of both longitudinal and circumferential distribution is different to both the theoretical and experimental distributions of Rodabaugh and George.

The outer surface longitudinal and circumferential stress factors σ^*_o for bend IP2 are compared with the ADINAP results and Smith and Ford experimental results in Figure 6.18. Three ADINAP elements were used to model the bend, with two Fourier terms in the ovalisation interpolation function.

PB1 and PB2 give slightly lower longitudinal stresses than the ADINAP solution, which is in better agreement with the Smith and Ford experimental results; however, the circumferential stresses are very similar and show good agreement with the experimental results. As in the case of IP1, PB3 again gives a high peak circumferential stress and the form of the stress distribution curve differs from the other solutions.

In Figure 6.19 the outer surface longitudinal and circumferential stress factors given by the elbow elements for bend IP3 are compared with results presented by Bathe and Almeida, in which the bend was modelled by four ADINAP elements with three Fourier ovalisation terms. MARC Element 17 results (using three elements around the bend and 16 axisymmetric elements around the cross-section) and a Clark and Reissner shell solution are also given. Mid-surface longitudinal stress results for bend IP3 are shown in Figure 6.20.

The performance of elements PB1, PB2 and PB3 for bend IP3 is similar to that observed for bends IP1 and IP2. In general, PB1 and PB2 are in good agreement with the ADINAP and MARC solutions, but PB3 gives a high peak circumferential stress and a different form of stress distribution curve.

In Figure 6.21, bend IP4 outer surface longitudinal and circumferential stress distributions are compared with ring element and experimental results given by Ohtsubo and Watanabe. Six ring elements with six Fourier terms in the ovalisation interpolation function were used to model the bend. PB1 and PB2 show good agreement with the Ohtsubo and Watanabe ring element, but PB3 again gives markedly different stress distribution.

Discussion of In-Plane Analyses Results

The displacement results presented in Table 6.6 show that the elbow element solutions are stiffer than the ANSYS Castigliano based element STIF18 solutions for in-plane bending. It should be emphasised that the ANSYS solution is not

an ideal solution and the difference may be accounted for in part, at least, by inadequacies in the flexibility analysis approach. However, high stiffness may also arise in the finite element solution due to one or more of the following factors:

a) Poor representation of elbow ovalisation.

b) Inadequacies in the beam model.

c) Beam-ovalisation interaction in the elbow elements.

In order to establish the most likely cause of the high stiffness of the elements, consider first element PB1.

The ovalisation model of PB1 is essentially the Von Karman model, whereas, in the ANSYS STIF18 element, ovalisation is accounted for by using Clark and Reissner flexibility factors. In Figure 2.6 Von Karman flexibility factors based on a three even term Fourier series solution (as used in the element formulations) are compared with Clark and Reissner values, and show very similar values for the range of λ examined in the above tests (Von Karman factors being slightly higher). Thus both models would be expected to give similar representation of ovalisation.

The beam model used in PB1 was examined in a number of beam verification tests. Good agreement was found between the PB1 beam model and STIF18 beam model (unit flexibility factor) for in-plane bending. Therefore, the two beam models could be expected to give similar representation of beam bending deformation in the elbows considered above.

However, PB1 differs significantly from STIF18 in the way in which the ovalisation and bending models are combined to form an elbow element. In STIF18 the beam stiffness matrix is factored by dividing the second moment of area of the beam bending stiffness coefficients by the Clark and Reissner flexibility factor. In PB1 the ovalisation deformation is coupled to the beam deformation through the coupling matrices defined in Chapter 5; thus there is a direct interaction between the two models. A consequence of this coupling is that the exact beam solution, based on Vlasov's curved beam theory, is no longer completely valid. From the results presented above, it appears that the shape functions derived from the exact solution may be unable to adequately represent the elbow deformation, causing the element to be over-stiff.

As the ovalisation model used in PB2 is essentially an extension of that of PB1 the above arguments apply to both elements. However, the ovalisation model of PB3 is fundamentally different from that used in the other two elements and the element performance is poorer due to a combination of (a) and (c) above.

The magnitude and distribution of stresses for bends under in-plane bending given by elements PB1 and PB2 shows good agreement with published solutions, but PB3 gives stress distribution curves of significantly different form. In particular, the circumferential stress distribution curves for bends IP1, IP3 and IP4 given by PB3 show very sharp peak stresses at around $\Theta = 90°$, and local peaks at $\theta = 0°$ and $\theta = 180°$. The best agreement between PB3 and alternative analyses was found for the higher λ bend IP2.

It should be noted that most of the theoretical solutions presented in Figures 6.17 to 6.21 are based on trigonometric interpolation of ovalisation displacement. In such cases a trigonometric form of stress distribution is expected and is seen to occur. However, the MARC Element 17 and Clark and Reissner solutions shown for bend IP3 in Figures 6.17 and 6.18 are not based on trigonometric functions, but give a stress distribution form similar to the Fourier based solutions. It would therefore appear that element PB3 is not able to represent the true stress distribution as well as elements PB1 and PB2.

In assessing the above results, it is of interest to compare the computing requirements of the various finite element solutions presented. This is most simply done by considering the number of degrees of freedom required by the alternative elements to model the bend in each case.

All four bends were modelled using three PB1, PB2 and PB3 elements, with the element matrices statically condensed to 12x12 matrices before assembly. However, it is of interest to consider the total number of degrees of freedom at element level as well as in the global stiffness matrix, as this allows a more complete comparison of element effectiveness.

In IP2 Bathe and Almeida used three ADINAP elements with two active Fourier modes to model the bend. Three ADINAP elements were also used to model IP3, but in this case three Fourier modes were active. Results were also presented for IP3 using the MARC Element 17, in which three elbow elements with sixteen axisymmetric elements around the circumference were used. Ohtsubo and Watanabe analysed bend IP4 using six ring elements with six Fourier ovalisation

modes. Thus the degrees of freedom per bend model, assuming a fully representative element with equivalent in-plane and out-of-plane freedoms, are given in Table 6.7.

Element	Bend	element level dof	global dof
PB1	All	42	24
PB2	All	48	24
PB3	All	96	24
ADINAP 2	IP2	100	100
ADINAP 3	IP3	120	120
MARC El. 17	IP3	268	30
O/W Ring	IP4	432	432

Table 6.7. In-plane bend finite element model sizes.

Considering Figures 6.13 to 6.17 and Table 6.7, it is seen that elements PB1 and PB2 give accurate stress results for significantly fewer degrees of freedom than the alternative analyses presented. The polynomial ovalisation element PB3 is less accurate in both displacement and stress evaluation and is significantly larger than PB1 and PB2 at element level. MARC Element 17 is also a polynomial based ovalisation element but is able to accurately evaluate stress distribution. However, the MARC element requires a large number of degrees of freedom at element level to achieve this. The in-plane tests therefore indicate that Fourier based elements give greater accuracy per degree of freedom than polynomial based elements. It is also noted that although the linearly varying element PB2 converges on average one element quicker than the constant ovalisation element PB1 it offers no significant advantage over PB1 in terms of degrees of freedom at element level.

6.3.3 Sample Analyses: Out-of-Plane Bending of 90^{o} Bends

In order to investigate the accuracy and applicability of the elements PB1, PB2 and PB3 under out-of-plane bending, three 90^{o} bends were analysed. The bends are identified by the names OP1, OP2 and OP3 and dimensions, material properties and applied moments used in the finite element analyses are given in Table 6.8.

Bend	E	ν	MX	R	r	t	λ	R/r
OP1	210E3 (N/mm^2)	0.3	-1E8 (Nmm)	1000 (mm)	173.01 (mm)	13.35 (mm)	0.446	5.78
OP2	210E3 (N/mm^2)	0.3	-1E8 (Nmm)	1000 (mm)	347.22 (mm)	13.02 (mm)	0.108	2.88
OP3	2.82E7 (lbf/in^2)	0.3	-1E4 (lbfin)	6.06 (in)	1.99 (in)	0.555 (in)	0.849	3.04

Table 6.8. Elbow geometry for out-of-plane bending tests.

Analysis of a bend similar to OP1 using the ADINAP element was presented by Bathe and Almeida in [3.12], where results were compared with Smith and Ford Experimental values [6.5]. Bend OP1 has the same geometry as IP2. Stress results are presented in terms of a stress factor which is the ratio of the elbow stress to the stress occurring in a straight pipe of equivalent nominal dimensions under the same moment loading. However, in the case of out-of-plane bending, the moment varies with axial position according to the equation

$$M = MX \sin\phi$$

Taking this into account, the out-of-plane stress factor as used by Bathe and Almeida is given by

$$\sigma_o^* = \sigma\left(\frac{I}{Mr_o}\right) = \sigma\left(\frac{I}{MX \sin\phi^* r_o}\right)$$

where Φ^* is the axial position at which the stress is evaluated. In all the analyses presented in this section $\phi^* = 45^o$.

A bend similar to OP2 was analysed by Ohtsubo and Watanabe in [3.5]. OP2 is, in fact, out-of-plane bending of a pipe of geometry IP4. For consistency with Ohtsubo and Watanabe, the stress factor in this case is given by the equation

$$\sigma^* = \sigma\left(\frac{I}{MX \sin\phi^* r}\right)$$

Bend OP3 is a thick-walled welding elbow (forged 90^o bend) which was investigated experimentally and theoretically by Smith and Ford in [6.1]. The stress factor for OP3 is given by the same equation as for IP2.

Convergence Requirements.

Out-of-plane convergence plots for 90^{o} bends loaded by an out-of-plane shear force are given in Figures 6.13 to 6.15. The number of elements required to give a converged solution for bend parameters of $\lambda = 0.1, 0.2, 0.3$ are summarised in Table 6.9.

Element	Number of Elements					
	$\lambda = 0.1$		$\lambda = 0.2$		$\lambda = 0.5$	
	R/r=10	R/r=3	R/r=10	R/r=3	R/r=10	R/r=3
PB1	11	16	8	12	7	8
PB2	10	14	8	8	7	7
PB3	10	14	8	12	7	8

Table 6.9. Out-of-plane convergence requirements.

The element convergence studies show that many more elements are required for out-of-plane bending than for in-plane bending. Also, further investigation has shown that there is little difference in convergence rates between out-of-plane shear force and moment loading. Considering Table 6.9, two different bend discretizations were required for the above bends. Bends OP1 and OP3 were modelled by 6 elements, and bend OP2 by 14 elements. These models and the applied force and displacement boundary conditions are shown in Figure 6.22.

Displacement Results: Comparison with ANSYS STIF18.

In order to verify the performance of elements PB1, PB2 and PB3 under out-of-plane loading, displacements calculated using elements are compared with results given by the ANSYS STIF18 element under ANSI B31.1 flexibility factors in Table 6.10. For a converged solution, elements PB1 and PB3 yield very similar results and a single row is given for both these elements. The percentage difference between the elements and ANSYS STIF18 results are also given in the tables.

Element	OP1 ROTZ x 10^2	OP2 ROTZ x 10^2	OP3 ROTZ x 10^4
PB1/PB2	2.39	1.82	4.51
PB3	2.27	0.94	4.35
STIF18	2.63	1.94	4.93
$\left(\frac{PB1}{STIF18}-1\right)\%$	-9.1%	-6.2%	-8.5%
$\left(\frac{PB3}{STIF18}-1\right)\%$	-13.7%	-51.7%	-11.8%

Table 6.10 Out-of-plane bending end rotations.

As for in-plane bending, the elbow elements give a stiffer solution than ANSYS STIF18 for out-of-plane bending. In particular, the PB3 rotation for the low λ bend OP2 is much lower than the other solutions.

Stress Distribution

Normalised stress distributions for bends OP1, OP2 and OP3 given by the elbow elements PB1, PB2 and PB3 are compared with published results in Figures 6.23 to 6.27. As for in-plane loading, elements PB1 and PB2 give virtually identical results and, for clarity, a single curve representing both is given in the Figures.

The outer surface longitudinal and circumferential stress distributions for OP1 are compared with ADINAP results and experimental results of Smith and Ford in Figure 6.23. Three ADINAP elements were used to model the bend, with two Fourier terms used in the ovalisation interpolation series.

Elements PB1 and PB2 give good agreement with the ADINAP results. There is also good agreement with experimental circumferential stresses, but longitudinal stresses are slightly low towards the intrados of the bend. The PB3 results show reasonable agreement for longitudinal stress, but the form of the curve for circumferential stress is markedly different to the other solutions.

The longitudinal and circumferential stress distributions for bend OP2 at the outside and inside surfaces respectively are compared with the experimental and theoretical results given by Ohtsubo and Watanabe in Figures 6.24 and 6.25. Six ring elements with six Fourier ovalisation modes were used in the ring element model of the bend.

The PB1 and PB2 stress distributions for bend OP2 are slightly conservative in comparison with the ring element and experimental results, but show reasonable correlation in terms of stress distribution. However, the performance of PB3 is extremely poor, especially for circumferential stress distribution, where both the magnitude and form of the stress distribution is seen to differ significantly from the other theoretical and experimental results.

The longitudinal and circumferential stress distributions at the outside and inside surfaces for bend OP3 are compared with experimental and theoretical results given by Smith and Ford in Figures 6.26 and 6.27 respectively.

PB1 and PB2 show good agreement with the Smith and Ford theoretical and experimental solution, but again the PB3 differs from the other solutions.

Discussion of out-of-plane bending results.

The results of the out-of-plane analyses emphasise the points discussed for in-plane analysis: PB1 and PB2 show reasonable agreement with published results, but PB3 gives a poorer solution. The most significant result to emerge from the out-of-plane analyses is the very poor performance of PB3 for bend OP2, which was the lowest λ bend examined for out-of-plane bending. In order to study the behaviour of PB3 with varying λ, a further series of out-of-plane analyses (of radius ratio $R/r=3$) were carried out.

Out-of-plane displacements given by PB3 are compared with PB1 results in Figure 6.28. It is seen that there is a accelerated degradation in performance of PB3 in comparison with PB1 for values of λ less than 0.4.

As in the case of the in-plane analyses, it is of interest to examine the size of the models used in the out-of-plane analyses. The model degrees of freedom at element and global level are compared with the ADINAP and ring models described in the literature in Table 6.11.

Element	Bend OP1		Bend OP2	
	Element level dof.	Global dof.	Element level dof.	Global dof.
PB1	78	42	174	90
PB2	84	42	180	90
PB3	186	42	426	90
ADINAP 2	100	100	—	—
O/W RING	—	—	432	432

Table 6.11. Out-of-plane bend model degrees of freedom.

Elements PB1 and PB2 require fewer degrees of freedom than alternative elements for comparable accuracy, although the difference is not as great as for in-plane analysis. Element PB3 required more degrees of freedom at element level than the ADINAP element to give a (poor) converged model of OP1. In the case of OP2, the converged PB3 model was similar in size to the Ohtsubo and Watanabe ring model and gave very poor results.

6.3.4 Discussion of Single Bend Analyses.

The sample analyses of single bends presented above indicate that the polynomial ovalisation element PB3 is inferior to the Fourier ovalisation elements PB1 and PB2. It would be possible to improve the performance of element PB3 by modelling the cross-section with more quintic polynomials, but Tables 6.7 and 6.11 show that it is already much "larger" than PB1 or PB2 at element level and increasing the number of polynomials would defeat the objective of formulating a simple and efficient elbow element.

Elements PB1 and PB2 have been shown to perform well in the above analyses, especially when the model sizes are taken into account. The linear interpolation of ovalisation in PB2 leads to an element which converges about one element sooner than PB1; however, this advantage is negated by the additional calculations required at element level due to the increased uncondensed element matrix size.

Considering accuracy, programming simplicity and computing requirements, the most effective of the three elbow elements proposed in Chapter 5 is element PB1. In order to investigate the accuracy, applicability and computing costs of using element PB1 in general piping analysis, sample analyses of a number of piping systems are presented in the next section.

6.4 Analysis of Piping Systems.

In order to assess the applicability and performance of element PB1 in general piping system analysis, analyses of eight piping systems denoted (SYS1 to SYS8) are presented. Finite element models of the systems were created in the ANSYS finite element package: bends were discretized by sufficient PB1 elbow elements to give convergence and straight pipes were modelled by ANSYS STIF16 elements [4.5,6.1]. The PB1/STIF16 models will be referred to as PB1 models in the remainder of this Chapter.

For conciseness, only outer surface stress results are presented in the analyses presented in this section.

In order to compare results obtained using elbow element PB1 with the simpler flexibility analysis approach, the PB1 model results for systems SYS1 to SYS4 are compared with ANSYS flexibility analysis results. The results of two flexibility analyses are presented for each system, one based on ANSI B31.1 correction factors and the other based on Clark and Reissner correction factors.

In ANSYS flexibility analysis, the STIF18 elbow flexibility factor FF and stress intensification factor SI may be input directly or default values taken. The definition of flexibility and stress intensification factors chosen for the analysis can significantly effect the analysis results. Unless specified otherwise, ANSYS STIF18 flexibility and stress intensification factors default to the ANSI B31.1 definitions discussed in Chapter 2, which are based on the work of Clark and Reissner and of Markl respectively. These factors are given by the equations :

$$FF = \frac{1.65}{\lambda}$$

$$SI = \frac{0.9}{\lambda^{\frac{2}{3}}}$$

It is emphasised that Markl's stress intensification factors are based on the results of *fatigue tests* and *do not correct the beam stress to give the elastic stress in the elbow.* In the following sections, stresses denoted as STIF18 (B31.1) are the values obtained by flexibility analysis using the default B31.1 stress intensification factor

and have not been further corrected to give elastic stress. These results are presented in order to demonstrate the importance of the definition of stress intensification used in flexibility analysis.

The Clark and Reissner stress intensification factor *is* based on elastic stress analysis and, ideally, when used in a flexibility analysis yields the maximum elastic stress in the elbow. The Clark and Reissner SI factor is given by the equation:

$$SI = \pm \frac{1.892}{\lambda^{\frac{2}{3}}} - \frac{0.480}{\lambda^{\frac{1}{3}}}$$

where the positive sign denotes the outer surface. In the following sections, stress results denoted STIF18 (C/R) are values obtained by flexibility analysis using the Clark and Reissner definition of stress intensification factor.

In systems SYS5 to SYS7 the PB1 model results are compared with ABAQUS finite element analysis, in which elbows were modelled by ELBOW 31B elements and straight beam elements modelled by beam elements B31 [3.22].

In system SYS8 the PB1 model results are compared with an ANSYS thin shell analysis, using the doubly curved isoparametric thin shell finite elements STIF93 [4.5,6.1].

System SYS9 is a straight piping run which incorporates a semi-toroidal bellows unit. Analysis of SYS9 is included to demonstrate the use of a bellows element, BEL1, in a piping system analysis.

6.4.1 SYS1

System SYS1 consists of three straight piping runs and two 90^{o} elbows in a configuration commonly referred to as a Hovgaard bend. The system geometry, material properties and constraints are shown in Figure 6.29.

The pipework is anchored (fully fixed) at end 1, and subject to a prescribed displacement simulating a thermal expansion load of UZ=0.8 in. at end 2. All other degrees of freedom at end 2 are constrained to zero. The applied displacement effectively loads bend A of the system in-plane and bend B

out-of-plane. The object of the analysis is to obtain the displacements and stresses at the mid-sections of bends A and B. Results obtained using elbow element PB1 are compared with ANSYS flexibility analysis results.

Both bends in SYS1 have bend parameter and radius ratio:

$$\lambda = 0.110 \qquad \frac{R}{r} = 2.719$$

The number of PB1 elements required for a converged solution of such a bend under general loading is obtained by considering Figure 6.13, which indicates that 14 elements are required per bend for general three-dimensional loading. For the particular load case considered here bend A is subject to in-plane loading only which, according to Figure 6.10, requires only 6 elements for convergence. However, in order to assess the computing costs for a more general load case, both bends are modelled by 14 elements.

The flexibility and stress intensification factors of the SYS1 bends according to B31.1 and the Clark and Reissner equations are:

	FF	SI
B31.1	15.0	3.920
C/R	15.0	7.239

The stiffness matrix of ANSYS element STIF18 is basically evaluated by inverting a flexibility matrix obtained by application of Castigliano's theorem. Consequently, the stiffness of a bend modelled by STIF18 elements is the same regardless of the number of elements used in the model; that is, the solution does not converge. However, in order to obtain the mid-bend stresses, a node must be located at the middle of the bends. Therefore two STIF18 elements were used for each elbow in the flexibility analysis model of the system.

Deformed geometry plots given by the ANSYS and PB1 models are shown in Figure 6.30. The displacements at the middle of the bends are given in Table 6.12.

Model	Bend	UX (in)	UY (in)	UZ (in)	ROTX (rad)	ROTY (rad)	ROTZ (rad)
ANSYS	A	1.277E-3	-5.999E-2	0.1347	3.739E-3	-1.816E-3	-3.474E-4
model	B	-2.047E-2	-4.957E-2	0.7928	3.410E-3	-1.080E-3	2.427E-4
PB1	A	1.312E-3	-6.145E-2	0.1380	3.772E-3	-1.851E-3	-3.579E-4
model	B	-2.053E-2	-5.109E-2	0.7928	3.404E-3	-1.081E-3	2.490E-4

Table 6.12. Displacement at the centre of SYS1 bends.

The bend middle section longitudinal and circumferential stress distributions given by PB1 and the maximum elastic longitudinal stress values given by STIF18 (C/R) are shown in Figure 6.31. The maximum longitudinal, circumferential and Von Mises stresses occurring in the mid-bends are given in Table 6.13.

	Element	$\sigma_{\phi max}$ (psi)	$\sigma_{\theta max}$ (psi)	$\sigma_\phi\left(\frac{C/R}{PB1}-1\right)\%$	$\sigma_{e max}$ (psi)	$\sigma_e\left(\frac{C/R}{PB1}-1\right)\%$
Bend A	STIF18 (B31.1)	4725	—	—	5141	—
	(STIF18) (C/R)	8769	—		9016	
	PB1	6150	8664	+42.6%	8382	+7%
Bend B	STIF18 (B31.1)	8217	—	—	8576	—
	STIF18 (C/R)	15241	—		15391	
	PB1	10464	13651	+45.7%	12744	+20.8%

Table 6.13. Maximum mid-bend longitudinal, circumferential and Von Mises stresses.

Table 6.13 shows that the solution based on ASME B31.1 stress intensification factors indicates much lower stress levels than the other solutions. This is because ASME B31.1 SI factors are based on *fatigue* test data, as discussed in Chapter 2, and do not correct the beam analysis to give elastic elbow stresses. The flexibility analysis stresses based on the Clark and Reissner definition of stress intensification factor are higher than the PB1 results for both bends.

The total computing times required by the analyses, given in standard VAX CP units, were:

ANSYS model, 11 VAX CP units.
PB1 model, 64 VAX CP units.

The PB1 model took approximately six times longer than the ANSYS model to run. PB1 actually took less time than STIF18 to form the element matrices but the larger number of elements required for convergence and the detailed stress pass led to a more expensive system model.

6.4.2 SYS2.

SYS2 is a second Hovgaard bend arrangement, fully fixed at both ends and subject to a uniform temperature change of 100°C. The object of the analysis is to obtain the displacements and stresses at the mid-sections of bends A and B. Results obtained using elbow element PB1 are compared with ANSYS flexibility analysis results.

The system geometry and material properties are given in Figure 6.32. Bends A and B are identical, with bend parameter and radius ratio:

$$\lambda = 0.446 \qquad \frac{R}{r} = 5.78$$

Figure 6.13 indicates that 8 PB1 elements are required per bend for general loading. The flexibility and stress intensification factors of the bends given by the B31.1 and Clark and Reissner equations are:

	FF	SI
B31.1	3.70	1.542
C/R	3.70	2.613

Deformed geometry plots given by the flexibility and PB1 model analyses of the system are shown in Figure 6.33. The displacements at the middle of bends A and B are given in Table 6.14.

Model	Bend	UX (mm)	UY (mm)	UZ (mm)	ROTX (rad)	ROTY (rad)	ROTZ (rad)
ANSYS model	A	4.112	-1.881	1.730	8.211E-4	-4.012E-4	-1.336E-4
	B	1.730	1.881	4.112	-1.336E-4	4.012E-4	8.211E-4
PB1 model	A	4.110	-1.886	1.784	8.265E-4	-4.172E-4	-1.386E-4
	B	1.776	1.876	4.109	-1.369E-4	4.128E-4	8.282E-4

Table 6.14. SYS2 mid-bend displacements.

The longitudinal and circumferential stress distributions and the STIF18 (C/R) maximum longitudinal elastic stress at the mid-points of bends A and B are shown in Figure 6.34. The maximum longitudinal, circumferential and Von Mises stresses occurring at the mid-bends are given in Table 6.15.

	Element	$\sigma_{\phi max}$ (N/mm^2)	$\sigma_{\theta max}$ (N/mm^2)	$\sigma_\phi\left(\frac{C/R}{PB1}-1\right)\%$	$\sigma_{e max}$ (N/mm^2)	$\sigma_e\left(\frac{C/R}{PB1}-1\right)\%$
Bend A	STIF18 (B31.1)	15.3	—	—	18.2	—
	STIF18 (C/R)	26.6	—		29.0	
	PB1	18.9	32.1	+41.7%	27.4	+5.8%
Bend B	STIF18 (B31.1)	15.3	—	—	18.2	—
	STIF18 (C/R)	26.6	—		29.0	
	PB1	20.1	27.7	+32.3%	26.3	+10.3%

Table 6.15. Maximum mid-bend longitudinal, circumferential and Von Mises stresses.

The reaction forces occurring at fixed ends 1 and 2 of SYS2 due to the thermal loading are given in Table 6.16. It is seen that the PB1 model reaction forces are slightly higher than those given by the ANSYS model, indicating that the PB1 model is stiffer than the ANSYS model.

Model	Bend	FX (N)	FY (N)	FZ (N)	MX (Nmm)	MY (Nmm)	MZ (Nmm)
ANSYS model	A	6469.6	11921.5	-6469.6	-7.762E6	1.773E7	2.957E7
	B	-6469.6	-11921.5	6469.6	2.957E7	1.773E7	-7.762E6
PB1 model	A	6785.1	12316.9	-6800.8	-7.975E6	1.851E7	3.012E7
	B	-6785.1	-12316.9	6800.0	3.004E7	-1.844E7	-7.994E6

Table 6.16. SYS2 end reaction forces.

The total computing times required by the two analyses were:

ANSYS model, 11 VAX CP units.
PB1 model, 42 VAX CP units.

6.4.3 SYS3

SYS3 is a 4 bend branchless piping system fixed at one end and subject to a prescribed displacement of 12mm at the other. The object of the analysis is to obtain the displacements and stresses at the mid-sections of bends A, B, C and D. Results obtained using elbow element PB1 are compared with ANSYS flexibility analysis results.

The system is defined in Figure 6.35. The bends have parameter and radius ratio:

$$\lambda = 0.446 \qquad \frac{R}{r} = 5.78$$

Figure 6.13 indicates that 8 elements are required per bend for general loading.

Flexibility and stress intensification factors of the SYS1 bends given by the B31.1 and Clark and Reissner equations are:

	FF	SI
B31.1	3.70	1.542
C/R	3.70	2.613

Deformed geometry plots given by the ANSYS and PB1 models of the system are shown in Figure 6.36. The displacements at the middle of the four bends A, B, C and D are given in Table 6.17.

Model	Bend	UX (mm)	UY (mm)	UZ (mm)	ROTX (rad)	ROTY (rad)	ROTZ (rad)
ANSYS model	A	-4.882E-2	0.2049	1.861	3.179E-3	-1.905E-2	4.225E-4
	B	-0.6376	0.7628	8.157	3.088E-3	-1.125E-3	4.811E-4
	C	0.1691	1.642	8.624	-1.544E-3	-3.332E-4	1.200E-4
	D	8.324E-2	1.001	11.734	-2.303E-3	-7.070E-6	-1.537E-4
PB1 model	A	-4.886E-2	0.2098	1.937	3.243E-3	-1.959E-3	4.265E-4
	B	-0.6373	0.7653	8.232	3.094E-3	-1.259E-3	4.839E-4
	C	0.1770	1.661	8.690	-1.515E-3	-3.321E-4	1.169E-4
	D	8.828E-2	1.032	11.736	2.301E-3	6.653E-6	-1.607E-4

Table 6.17. SYS3 mid-bend displacements.

The longitudinal and circumferential stress distributions given by PB1 and the maximum longitudinal stress given by STIF18 (C/R) at the mid-points of bends A, B, C and D are shown in Figure 6.37. The maximum longitudinal, circumferential and Von Mises stresses at the mid-bend sections are given in Table 6.18.

	Element	$\sigma_{\phi max}$ (N/mm^2)	$\sigma_{\theta max}$ (N/mm^2)	$\sigma_\phi\left(\frac{C/R}{PB1}-1\right)\%$	$\sigma_{e max}$ (N/mm^2)	$\sigma_e\left(\frac{C/R}{PB1}-1\right)\%$
Bend A	STIF18 (B31.1)	20.5	—	—	26.2	—
	STIF18 (C/R)	34.8	—		38.5	
	PB1	25.4	35.4	+37.0%	35.9	+7.2%
Bend B	STIF18 (B31.1)	31.0	—	—	38.3	—
	STIF18 (C/R)	53.1	—		55.1	
	PB1	41.5	58.2	+30.0%	49.6	+10.1
Bend C	STIF18 (B31.1)	31.8	—	—	33.5	—
	STIF18 (C/R)	55.5	—		56.2	
	PB1	42.4	57.6	+30.9%	55.0	+02%
Bend D	STIF18 (B31.1)	18.2	—	—	21.0	—
	STIF18 (C/R)	33.2	—		35.0	
	PB1	22.3	31.7	+48.9%	28.0	+25.0

Table 6.18. Maximum mid-bend longitudinal, circumferential and Von Mises stresses.

The total computing time required to analyse the system by the two methods was

ANSYS model, 16 VAX CP units.
PB1 model, 66 VAX CP units.

6.4.4 SYS4

SYS4 is a complex 7 bend branchless piping system subject to a uniform temperature loading of 200^{o}C. The system is defined in Figure 6.38. It anchored at both ends, denoted points 1 and 6, and has rigid translation supports at points 2,3,4 and 5. The object of the analysis is to obtain the displacements and maximum stresses at the mid-sections of all 7 bends. Results obtained using elbow element PB1 are compared with ANSYS flexibility analysis results.

The bends have parameter and radius ratio

$$\lambda = 0.6667 \qquad \frac{R}{r} = 6.6667$$

Figure 6.13 indicates that 6 elements are required per bend for general loading.

Flexibility and stress intensification factors of the SYS1 bends given by the B31.1 and Clark and Reissner equations are:

	FF	SI
B31.1	2.475	1.18
C/R	2.475	1.93

Deformed geometry plots given by the ANSYS and PB1 models of the system are shown in Figure 6.39. The displacements at the middle of all seven bends are given in Table 6.19.

Bend	ANSYS model			PB1 model		
	UX (mm)	UY (mm)	UZ (mm)	UX (mm)	UY (mm)	UZ (mm)
A	9.161	-9.493	-0.613	9.151	-9.54	-0.647
B	15.016	-4.646	2.620	15.034	-4.672	2.694
C	-1.405	5.579	-5.270	-1.396	5.516	-5.161
D	-10.863	-5.112	-0.517	-10.530	-5.162	-0.521
E	6.323	-11.374	-1.193	6.650	-11.533	-1.206
F	27.224	-1.695	-18.840	28.176	-1.831	-19.089
G	10.473	8.587	-9.430	11.028	8.593	-9.774

Table 6.19. SYS4 mid-bend displacements.

The maximum stresses occurring at the middle of the bends are given in Table 6.20.

	Element	$\sigma_{\phi max}$ (N/mm^2)	$\sigma_{\theta max}$ (N/mm^2)	$\sigma_\phi\left(\frac{C/R}{PB1}-1\right)\%$	$\sigma_{e max}$ (N/mm^2)	$\sigma_e\left(\frac{C/R}{PB1}-1\right)\%$
Bend A	STIF18 (B31.1)	13.8	—	—	24.2	—
	STIF18 (C/R)	23.6	—		31.9	
	PB1	20.7	28.7	+14.0%	28.0	+13.5%
Bend B	STIF18 (B31.1)	11.0	—		19.3	—
	STIF18 (C/R)	18.9	—		25.6	
	PB1	10.5	18.2	+80.0%	22.6	+13.2%
Bend C	STIF18 (B31.1)	31.3	—		37.6	—
	STIF18 (C/R)	53.3	—		58.5	
	PB1	35.5	51.2	+50.1%	47.6	+23.5%
Bend D	STIF18 (B31.1)	38.5	—		44.1	—
	STIF18 (C/R)	65.4	—		70.5	
	PB1	53.1	77.6	+23.2%	71.1	-1.6%
Bend E	STIF18 (B31.1)	20.6	—		31.8	—
	STIF18 (C/R)	36.8	—		47.4	
	PB1	17.7	32.4	+7.9%	29.4	+61.2%
Bend F	STIF18 (B31.1)	19.1	—		30.1	—
	STIF18 (C/R)	35.3	—		44.8	
	PB1	39.8	52.6	-11.3%	53.5	-16.3%
Bend G	STIF18 (B31.1)	25.8	—		37.6	—
	STIF18 (C/R)	45.1	—		56.5	
	PB1	18.0	32.0	+150.5%	36.2	+56.1%

Table 6.20. SYS4 Maximum mid-bend longitudinal, circumferential and Von Mises stresses.

Table 6.20 shows that in SYS4 the ANSYS flexibility analysis stresses based on the Clark and Reissner SI factors are not conservative for all the bends. Bend F shows lower longitudinal stress in the ANSYS (C/R) model than in the PB1 model, and both bends D and F show lower Von Mises stresses.

The SYS4 thermal reaction forces at the ends, point 1 and 6, and at internal supports are given in Table 6.21. As in SYS2, the PB1 model reaction forces are slightly higher than the ANSYS model forces, indicating that the PB1 model is stiffer.

Model	Support	FX (N)	FY (N)	FZ (N)	MX (Nmm)	MY (Nmm)	MZ (Nmm)
ANSYS model	1	8925.0	23596.0	-6163.3	-10.97E6	3.44E6	73.81E6
	2	—	-64665.6	—	—	—	—
	3	—	11329.4	4916.2	—	—	—
	4	—	115964.3	9961.8	—	—	—
	5	—	-175515.6	—	—	—	—
	6	-8925.0	89291.5	-8714.8	9.36E6	19.85E6	39.51E6
PB1 model	1	9441.8	24219.8	-6538.5	-11.49E6	3.59E6	74.92E6
	2	—	-66918.5	—	—	—	—
	3	—	11844.8	5291.0	—	—	—
	4	—	120304.0	10493.5	—	—	—
	5	—	-182096.8	—	—	—	—
	6	-9441.8	92646.6	-9246.0	9.78E6	20.88E6	41.79E6

Table 6.21. SYS4 reaction forces.

The total computing time required to analyse the system by the above methods was:

ANSYS model, 29 VAX CP units.
PB1 model, 91 VAX CP units.

6.4.5 SYS5

Piping system SYS5 is a two bend symmetric expansion loop subject to a prescribed displacement of 10mm, as shown in Figure 6.40. Two finite element analyses of the system were performed: a PB1 model in the ANSYS program and an ABAQUS model using ABAQUS B31 (straight beam) and ELBOW 31B (elbow) elements [3.22]. ABAQUS ELBOW 31B is a constant bending elbow element, discussed in Chapter 3.

The object of the analysis was to obtain the displacement at the elbow ends A1 and B1 (as shown in Figure 6.40) and the stress distributions at the middle of the bends A and B.

End 1 of the system is pinned and has an applied displacement of UX = 10mm, and symmetry boundary conditions are applied at end 2. The bends have parameter and radius ratio

$$\lambda = 0.152 \qquad \frac{R}{r} = 3.27$$

As the system lies in the X-Y plane, and loading is also planar, the elbows are subject to in-plane loading only. The number of PB1 elements required for a converged solution is therefore given by Figure 6.10, and is found to be 6. In the ABAQUS model each elbow was modelled by 3 ELBOW 31B elements.

Deformed system geometry plots given by the PB1 and ABAQUS models are shown in Figure 6.41. The displacements at positions at elbow ends A1 and B1, as indicated in Figure 6.40, are given in Table 6.22.

Model	Position	UX (mm)	UY (mm)	ROTZ (rad)
PB1 model	A1	9.999	4.123	6.763E-3
	B1	0.909	5.662	5.680E-3
ABAQUS	A1	9.999	4.114	6.748E-3
	B1	0.8977	5.664	5.675E-3

Table 6.22. SYS5 displacements at locations A1 and B1.

The longitudinal and circumferential stresses at the middle of bends A and B given by the two analyses are shown in Figure 6.42, which shows that the stress magnitudes and distribution given by the two models are very similar, particularly in the case of circumferential stress.

The reaction forces at ends 1 and 2 given by the analyses are given in Table 6.23. The higher PB1 model reaction forces indicate that the PB1 model is slightly stiffer than the ABAQUS model.

Model	End	FX (N)	FY (N)	MZ (Nmm)
PB1 model	1	550.9	0.0	—
	2	-550.9	—	-9.23E5
ABAQUS	1	531.6	0.0	—
	2	531.6	—	8.94E5

Table 6.23. SYS5 reaction forces.

It was not possible to obtain a quantitative comparison of solution time requirements as the finite element models were run on different computers. The ANSYS model took approximately 10 seconds interactive user time to run on a VAX 11/750, whilst the ABAQUS model took approximately 2 minutes to compile and run on a Hewlett Packard 350 work station. (It should be noted that the actual solution time for the ABAQUS model was much less than the total run time).

6.4.6 SYS6

Piping system SYS6 is a two bend asymmetric expansion loop subject to a prescribed displacement of 4.32 in., as shown in Figure 6.43. The system was analysed using ANSYS STIF16 and PB1 elements in the ANSYS program, and ABAQUS B31 and ELBOW 31B elements in the ABAQUS program.

End 1 of the system is fully fixed, whilst end 2 is pinned and subject to an applied displacement of -4.23 in. applied in the x direction. The bends have parameter and radius ratio

$$\lambda = 0.11 \qquad \frac{R}{r} = 2.78$$

As in the case of SYS5, bends A and B are subject to in-plane loading only. Thus the number of PB1 elements required for a converged solution is indicated by Figure 6.10 to be 6.

Deformed deformed geometry plots given by the ANSYS PB1 and ABAQUS models are shown in Figure 6.44. The displacements at elbow end positions A1, B1 and C1 as indicated in Figure 6.43 are given in Table 6.24.

Model	Position	UX (in)	UY (in)	ROTZ (rad)
PB1 model	A1	-0.911	7.64E-4	1.18E-2
	B1	-1.002	0.277	1.03E-4
	C1	-4.059	0.191	-2.70E-2
ABAQUS	A1	-0.871	6.10E-4	1.04E-2
	B1	-0.9430	5.66E-2	2.22E-3
	C1	-4.151	-5.68E-2	-2.76E-2

Table 6.24. SYS6 displacements at locations A1, B1 and C1.

The longitudinal and circumferential stresses at the middle of bends A, B and C given by the two analyses are shown in Figures 6.45 and 6.46, and the reaction forces at ends 1 and 2 given in Table 6.25.

Model	End	FX (N)	FY (N)	MZ (Nmm)
PB1 model	1	593.3	-358.1	6.20E4
	2	-593.3	358.1	—
ABAQUS	1	571.9	-286.1	-5.72E4
	2	-571.9	286.1	—

Table 6.25. SYS6 reaction forces

Again, the PB1 model is slightly stiffer than the ABAQUS model, but Figures 6.45 and 6.46 show very good agreement between the PB1 and ELBOW 31B stresses for the SYS6 bends.

6.4.7 SYS7

Piping system SYS7 is a Hovgaard bend arrangement, fixed at end 1 and subject to prescribed horizontal and vertical displacements of UY = 1 in. and UZ = -2 in. at end 2, as shown in Figure 6.47. The system was analysed using PB1 and ABAQUS models. The bends have parameter and radius ratio:

$$\lambda = 0.705 \qquad \frac{R}{r} = 10.3$$

Both bends experience three-dimensional loading. The number of elements required for convergence given by Figure 6.13 is 6.

The deformed geometry given by both the models are shown in Figure 6.48. The displacements at elbow ends A1 and B1 (as shown in Figure 6.47) are given in Table 6.26.

Model	Position	UX (in)	UY (in)	UZ (in)	ROTX (rad)	ROTY (rad)	ROTZ (rad)
PB1	A1	-0.136	7.02E-4	-0.605	-8.77E-3	7.65E-3	3.09E-3
model	B1	-0.317	0.502	-1.687	-7.78E-3	1.06E-2	7.52E-3
ABAQUS	A1	-0.136	6.78E-4	-0.600	8.68E-3	7.54E-3	3.08E-3
model	B1	-0.321	0.502	-1.688	-7.053E-3	9.046E-3	6.297E-3

Table 6.26. SYS7 displacements at locations A1 and B1.

The longitudinal and circumferential stresses at the middle of bends A and B are shown in Figure 6.49. Very good agreement is found between the solutions. The system reaction forces at ends 1 and 2 given in Table 6.27.

Model	End	FX (N)	FY (N)	FZ (N)	MX (Nmm)	MY (Nmm)	MZ (Nmm)
PB1	1	-274.5	-973.8	1076.0	1.34E5	-5.07E4	-1.17E4
model	2	274.5	973.8	-1076.0	9.73E4	8.70E4	5.39E4
ABAQUS	1	-261.8	-939.8	1050.0	1.31E5	-5.00E4	-1.22E4
	2	261.8	939.8	-1050.0	9.41E4	-8.50E4	-5.22E4

Table 6.27. SYS7 reaction forces.

6.4.8 SYS8

SYS8 is a Hovgaard bend arrangement, as detailed in Figure 6.50. The system was analysed using elbow element PB1 and ANSYS STIF16 and the results compared with a three dimensional ANSYS thin shell analysis.

The ANSYS shell model is shown in Figure 6.51. The system was modelled using the eight node isoparametric shell element STIF93 [4.5,6.1]. The system was fully fixed at both ends and a displacement of UX = 10mm applied to end 1. The bends have parameter and radius ratio:

$$\lambda = 0.194 \qquad \frac{R}{r} = 3.00$$

The number of PB1 elements required for convergence is indicated by Figure 6.13 to be 6.

The PB1 and shell model deformed geometry plots are shown in Figure 6.52. The longitudinal and circumferential stress distributions at the middle of bends A and B given by the two analyses are shown in Figures 6.53 and 6.54.

The PB1 stress distribution in bend A is very similar to that in the complex shell analysis model. In bend B a similar form stress distribution is seen in both the PB1 and shell model curves but there is a shift in location of maxima and minima between the two models. The solutions appear to be "out of phase".

The computing times required for the PB1 and shell models were:

PB1 model: 21 VAX CP units.
Shell model: 4255 VAX CP units.

6.4.9 SYS9

SYS9 demonstrates the use of the bellows element BEL1 in a piping system analysis. SYS9 is a straight piping run in which a bellows unit has been fitted to absorb expansion loads, as shown in Figure 6.55. The bellows parameter and radius ratio are:

$$\lambda = 0.995 \qquad \frac{R}{r} = 4.30$$

In ANSYS piping analysis, bellows are represented by modified straight beam elements STIF16. The user may specify a bellows flexibility factor and an axial stiffness to account for the enhanced flexibility of the unit. A stress intensification factor may also be defined, however, this modifies the beam *bending* stresses only: it has no effect on the beam axial stress. Thus no stress intensification is evaluated for a bellows subject to pure axial loading, as in SYS9.

Three analyses of system SYS9 were performed. The first was an ANSYS model in which default values were taken for the bellows element. In practice the element defaulted to a straight beam.

In the second model the bellows axial stiffness was obtained from the BEL3 element formulation presented in Chapter 4, by modifying the FORTRAN program BEL3FLEX given in Appendix 3. Stress concentration factors were obtained by modifying the BEL3 program BEL3STR.FOR of Appendix 3. The stress factors are defined as:

$$\sigma^{*}_{\phi/\theta} = \frac{\sigma_{\phi/\theta}}{\sigma_{st}}$$

where σ_{st} is direct axial stress in a straight pipe of the same nominal dimensions. The PB3 axial stiffness and stress intensification factors given by the programs:

Surface	K axial (N/mm)	$\sigma^{\cdot}_{\phi}$	$\sigma^{\cdot}_{\theta}$
inside	61924	-7.40	-6.78
outside		5.29	-4.82

In the third model the bellows unit was represented by the BEL1 ANSYS user element formulated in chapter 4. The ANSYS user element source code of BEL1 is given in Appendix 4.

The reaction Forces calculated by the three models are given in Table 6.28.

Model	F_X (KN)
ANSYS (Default)	928
ANSYS (PB3 factors)	158
Element BEL1	150

Table 6.28. SYS9 reaction forces.

The maximum longitudinal and circumferential stress at the inside and outside surfaces of the bellows are given in Table 6.29. The model 2 results have been modified by the BEL3 stress intensification factors given above.

Stress	ANSYS (Default)		ANSYS (PB3 factors)		Element BEL1	
	Outside	Inside	Outside	Inside	Outside	Inside
σ_ϕ(N/mm^2)	-120.82		108.9	-152.3	138.2	-168.0
σ_θ(N/mm^2)	—		-99.19	139.3	-90.9	-133.8

Table 6.29. SYS9 bellows maximum stress.

The BEL2 longitudinal and circumferential stress distributions are shown in Figure 6.56.

6.4.10 Discussion of System Analyses.

Analyses SYS1 to SYS8 demonstrate the applicability of element PB1 in piping system analysis.

In SYS1 to SYS4 the PB1 results were compared with ANSYS flexibility analysis results. Two flexibility analyses were performed for each system; one using ANSYS default flexibility and stress intensification factors based on ASME B31.1, and the other using Clark and Reissner factors. (The definition of flexibility factor is the same in both cases).

In all four systems, the stresses evaluated according to B31.1 were significantly lower than the stresses evaluated using the Clark and Reissner factors and the PB1 element. These results highlight the fact that the B31.1 factors do not correct for elastic stresses in elbows.

In SYS1 to SYS3, the PB1 stresses proved to be lower than the ANSYS flexibility analysis results based on the Clark and Reissner definition of stress intensification factor. However, in SYS4 the ANSYS stresses for two of the bends in the system were lower than the PB1 stresses. This does not necessarily mean that the results are not conservative, as PB1 itself is based on a number of conservative assumptions, but does bring in to question the applicability of flexibility analysis under certain conditions.

Comparisons of computing requirements for flexibility analysis and PB1 models were given for systems SYS1 to SYS4. The PB1 models proved to require more processing time, mainly due to the element's detailed stress pass and the larger number of elbow elements required for convergence for certain geometries (especially for out-of-plane bending). However, the difference was not so great as to preclude the use of the element in general piping analysis.

In analyses SYS5 to SYS7 the performance of ANSYS/PB1 analysis was compared with ABAQUS analysis using the ELBOW 31B element. Stress results showed very good agreement, but the PB1 model reaction forces were slightly higher than those given by ABAQUS (3% - 4%), indicating that PB1 is slightly stiffer than with the ABAQUS elbow element. No direct comparison of computing requirements was possible, as the ANSYS/PB1 and ABAQUS models were run on different computers. However, the use of PB1 analyses may be preferred by piping designers as ANSYS modelling is relatively simple in comparison with ABAQUS modelling. The use of ANSYS is already well established in piping design and analysis; for example, in the offshore oil industry.

In system SYS8, the PB1 model results were compared with a doubly curved thin shell finite element analysis of a Hovgaard bend. It was found that the simple PB1 analysis showed good agreement with the complex shell model for the system bend under in-plane loading. Under out-of-plane loading the PB1 stress magnitudes were in general agreement with the shell results but the stress distribution appeared to be slightly "out of phase". The shell model took approximately 200 longer to run than the PB1 model.

In SYS9 the bellows element BEL1 was used in the analysis of a straight piping run with an in-line bellows expansion joint under axial loading. The results were compared with two flexibility analyses: a standard analysis using default flexibility and stress intensification factors and an analysis using axial stiffness and stress intensification factors given by the bellows element formulation BEL3. The analysis of SYS9 highlighted two important deficiencies in the treatment of bellows in flexibility analysis:

i) The bellows flexibility factor modifies bending stiffness only.

ii) The bellows stress intensification factor modifies bending strains only.

Hence, in an analysis of an axially loaded bellows the ANSYS bellows element is effectively a straight beam with no enhanced flexibility or stress intensification. Consequently, an analysis of an axially loaded bellows using the ANSYS element gives low stresses and high reaction forces.

In the next Chapter a number of conclusions based on the results presented above are made and proposals for further work on elbow elements given.

6.5 References

[6.1] 'ANSYS'
Theoretical Manual
Swanson Analysis Systems, Inc., Houston, Pennsylvania, 15342 (USA).

[6.2] J T BOYLE and J SPENCE
"Stress Analysis of Non-Uniform Bellows," Report to Spirax-Sarco Ltd., (1980).

[6.3] M Hamada *et al*
"Design Diagrams and Formulae for U-Shaped Bellows," Int. J. Press. Vessels and Piping, Vol. 4, pp 315-328, (1976).

[6.4] L H CHEN
"Piping Flexibility Analysis by Stiffness Matrix," J. App. Mech, (Dec. 1959).

[6.5] R T SMITH and H FORD
"Experiments on Pipelines and Pipe Bends Subject to Three-Dimensional Loading," J. Mech. Eng. Sci., Vol. 9, No. 2, (1967).

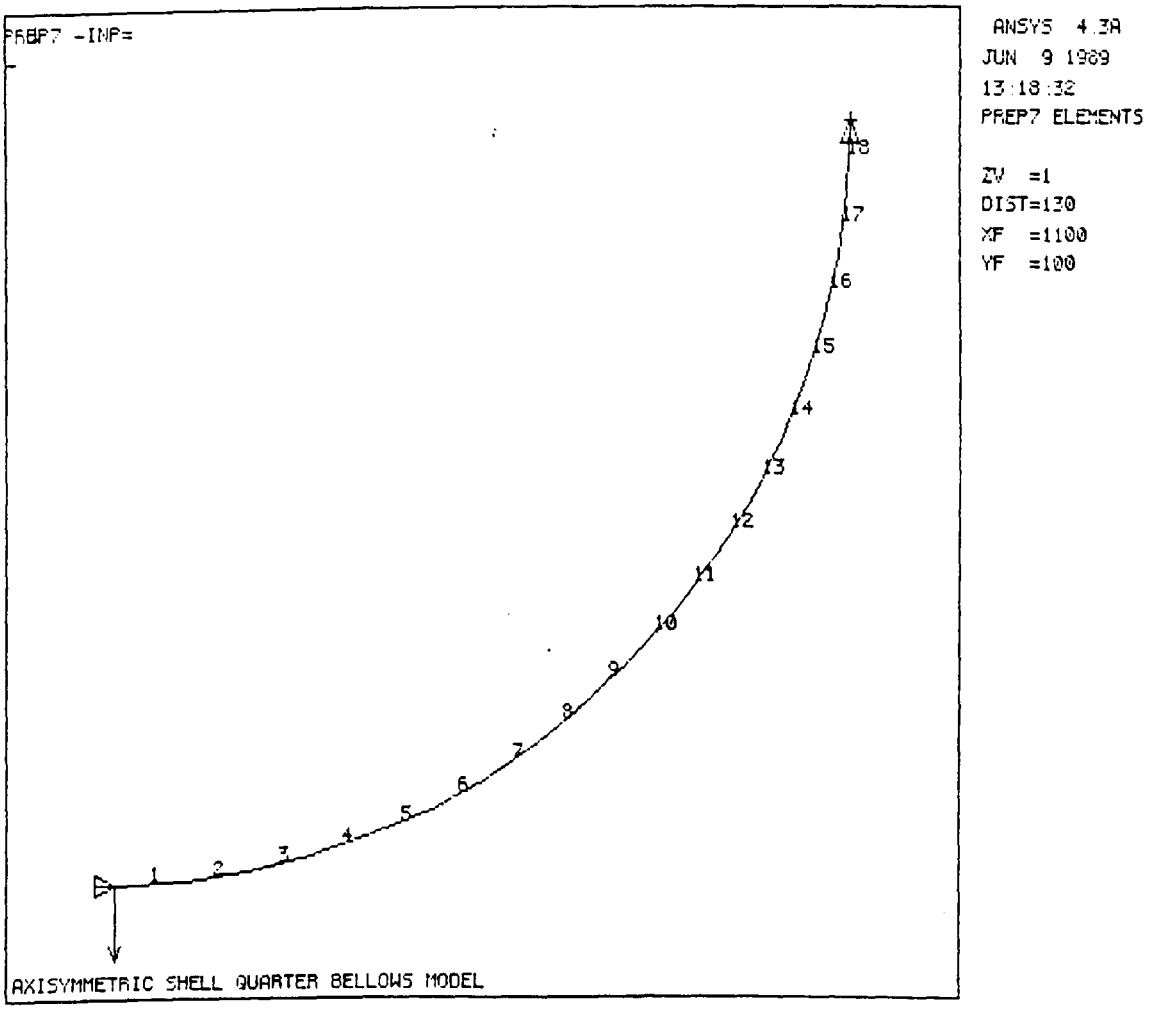


Figure 6.1 ANSYS STIF61 axisymmetric conical shell 90° bellows section finite element model plot, showing element numbers and boundary conditions.

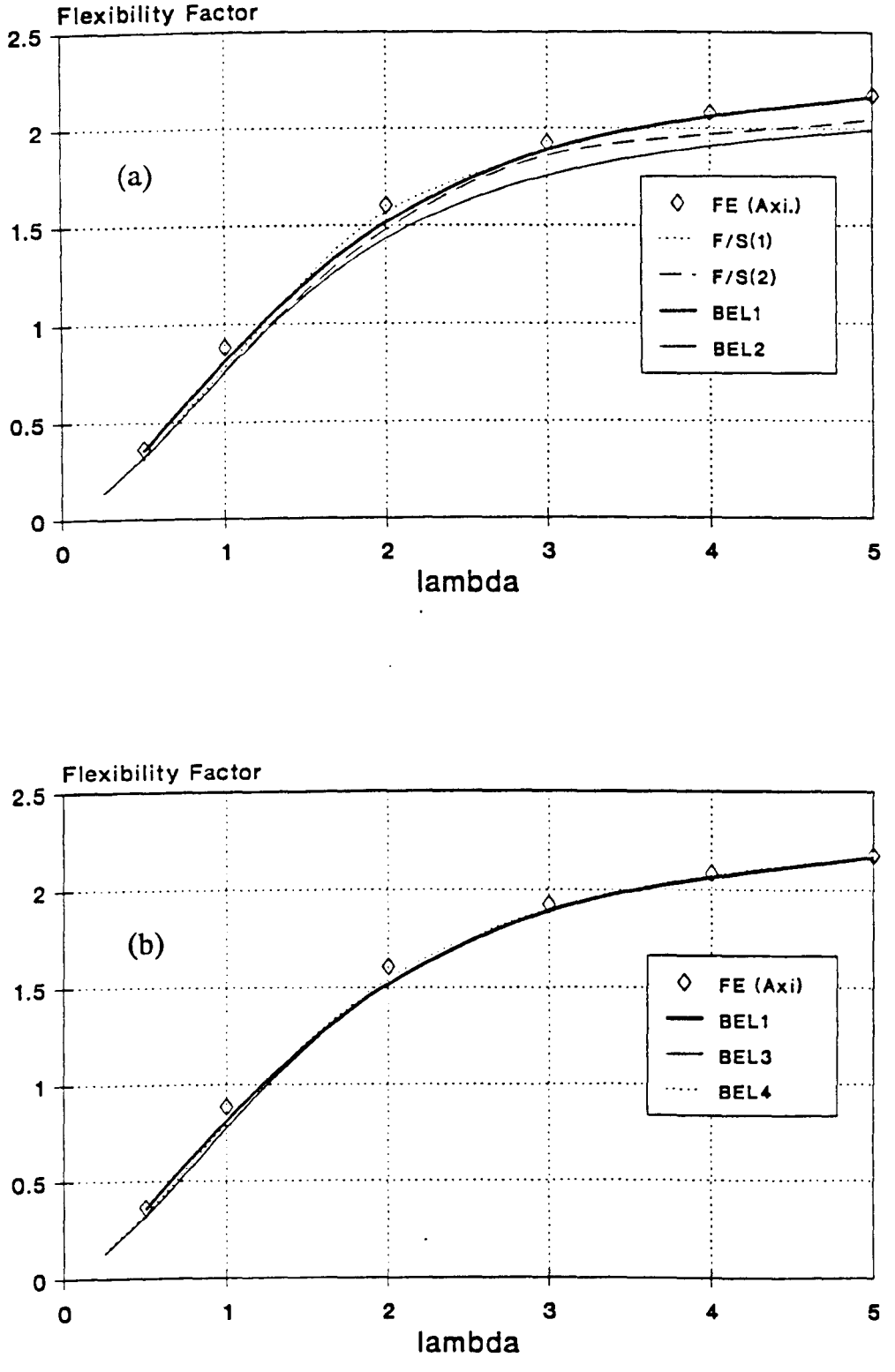


Figure 6.2 Bellows flexibility factor versus bellows parameter λ. Comparison of bellows element flexibility factors with Findlay and Spence (F/S) and axisymmetric finite element (FE(Axi)) solutions.

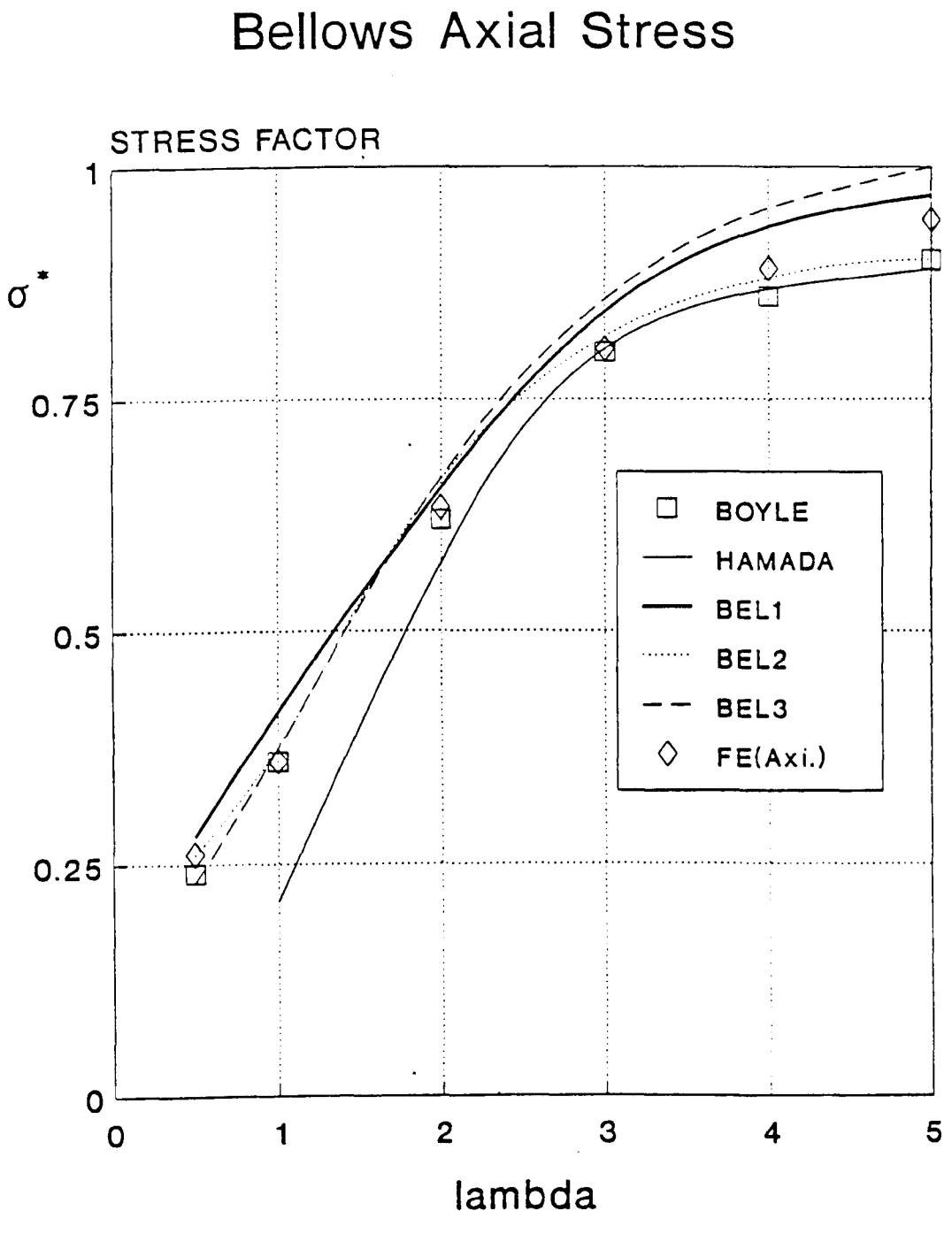


Figure 6.3 Bellows maximum axial stress versus bend parameter λ. Comparison of bellows element maximum stresses with Boyle, Hamada and axisymmetric finite element analysis (FE(Axi)) solutions.

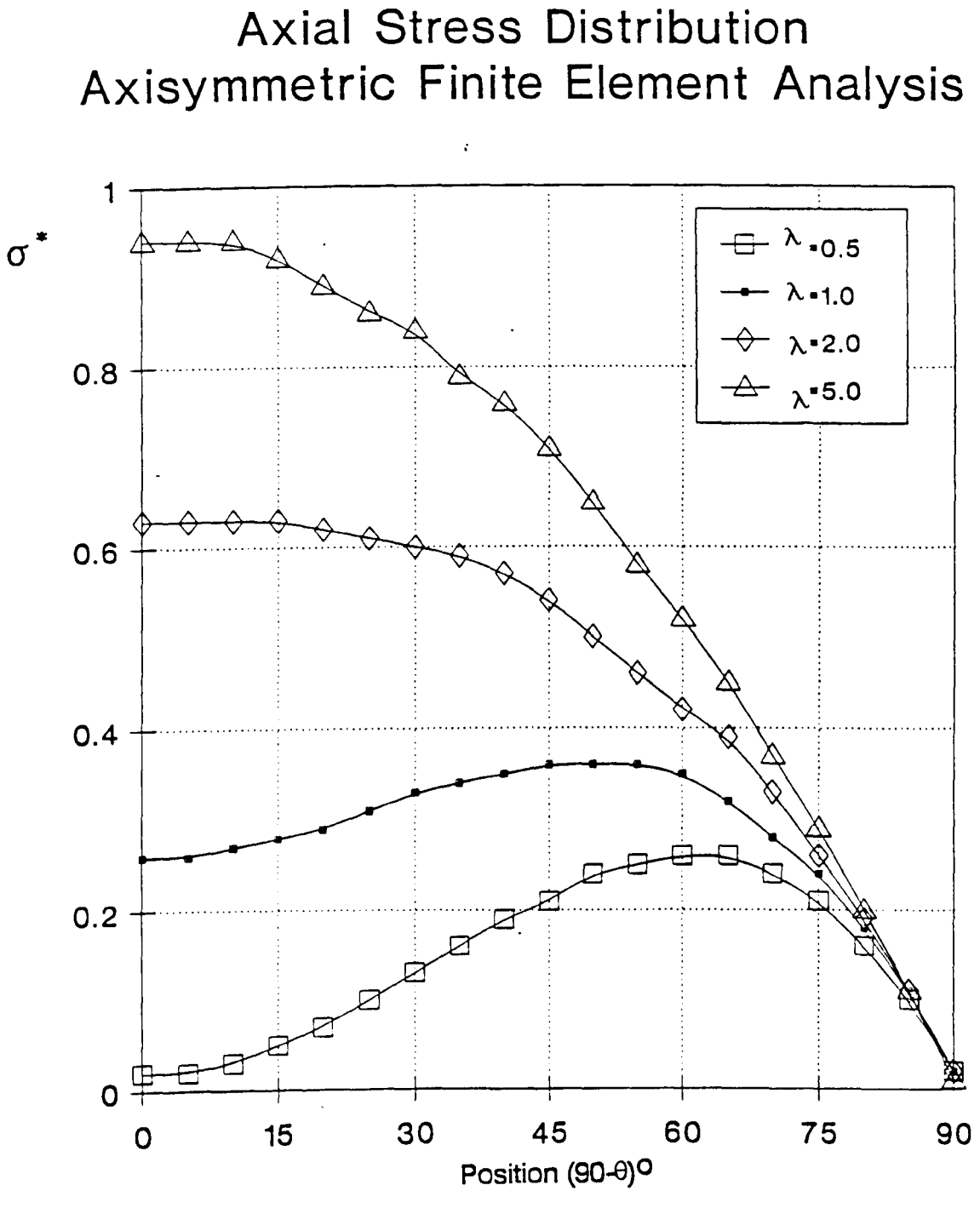


Figure 6.4 Axisymmetric shell model FE(Axi) inner surface axial stress distribution.

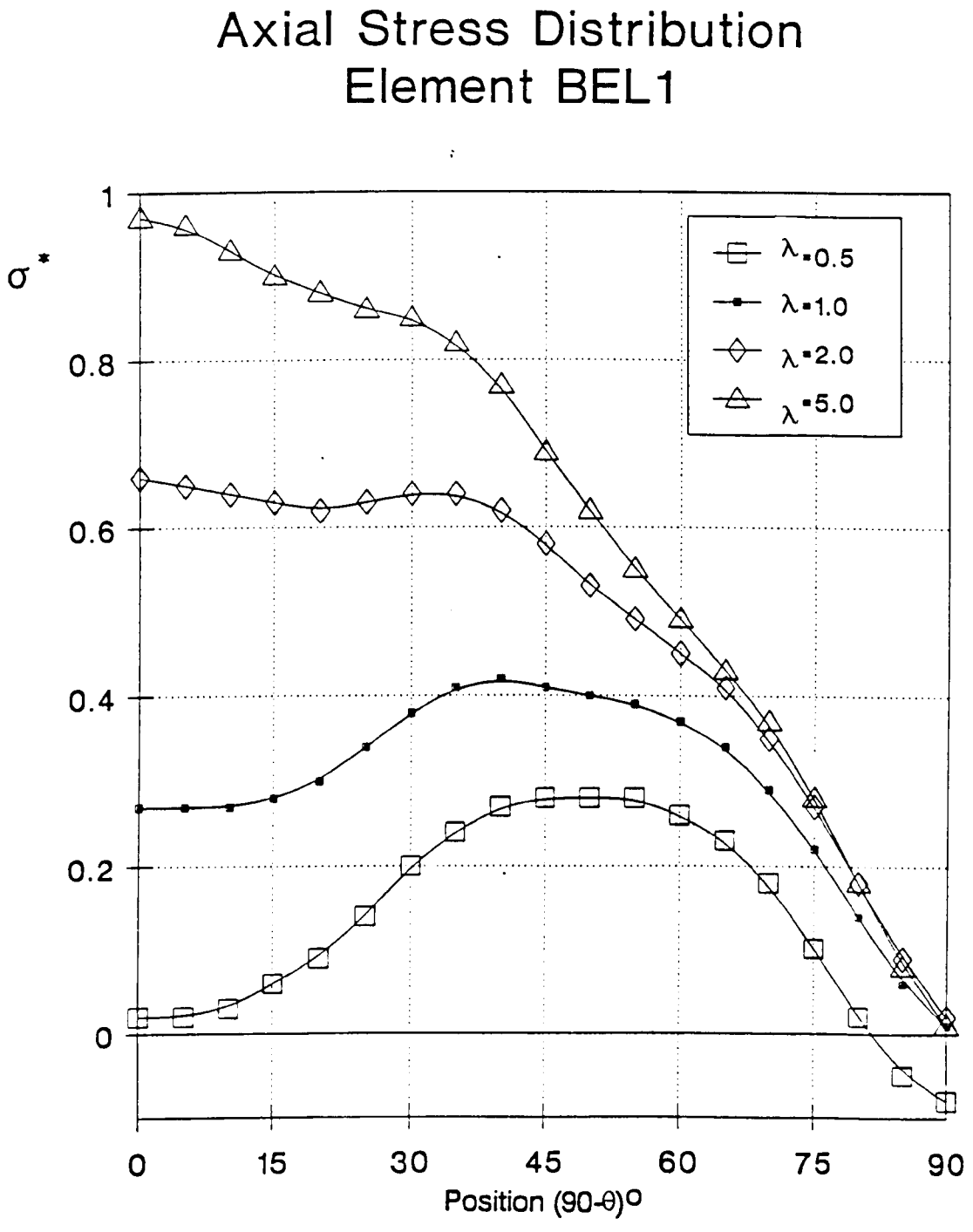


Figure 6.5 Bellows element BEL1 inner surface axial stress distribution.

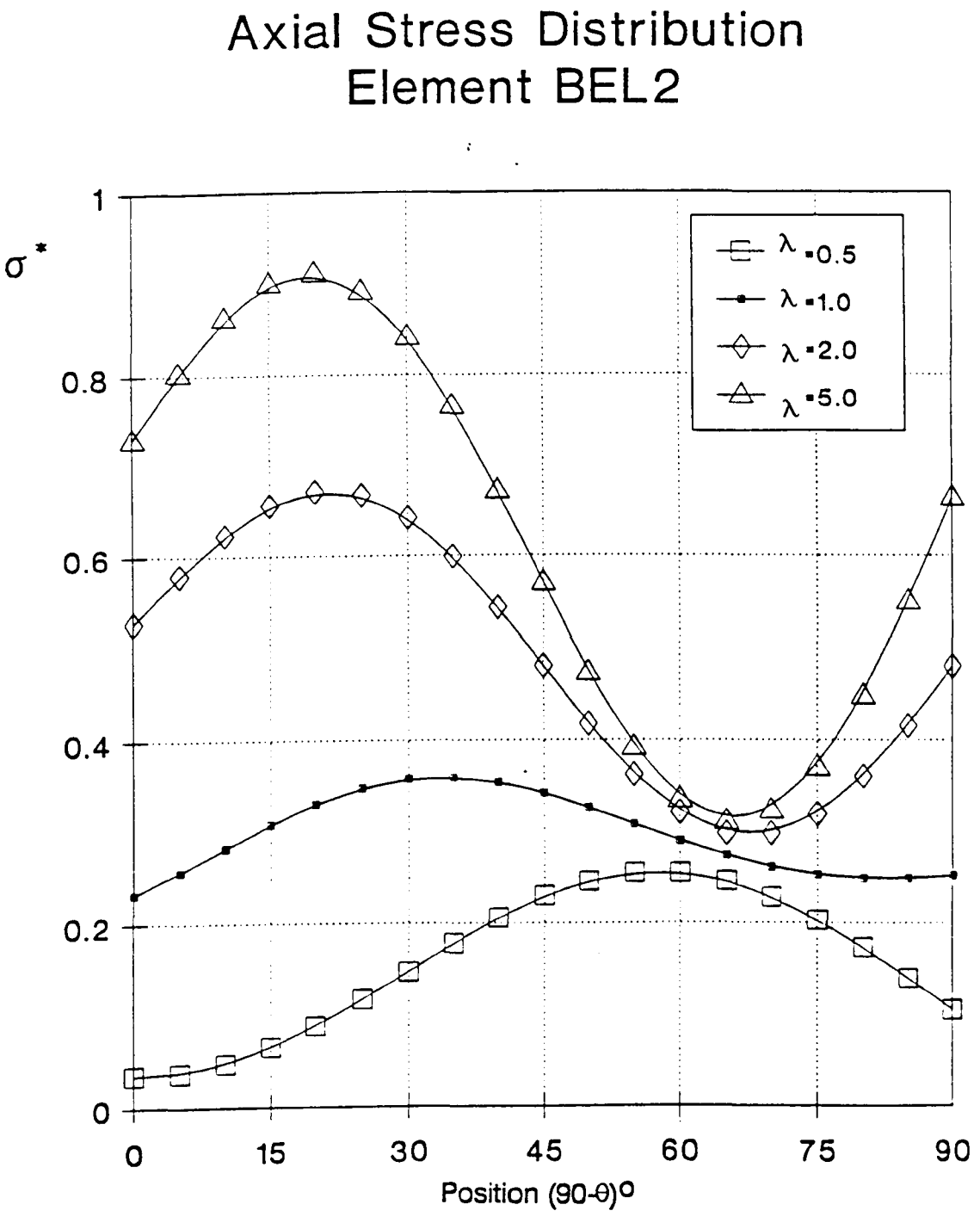


Figure 6.6 Bellows element BEL2 inner surface axial stress distribution.

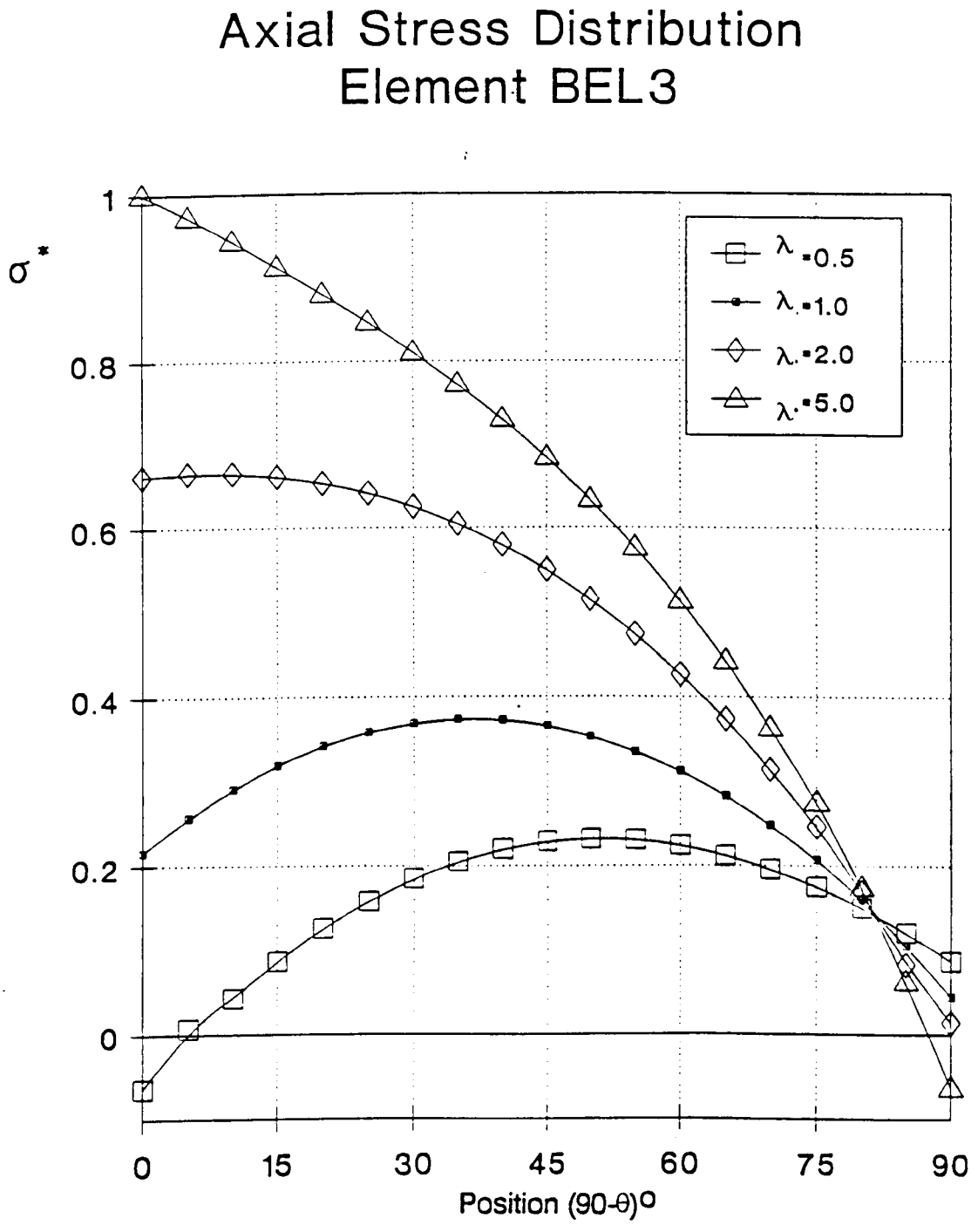


Figure 6.7 Bellows element BEL3 inner surface axial stress distribution.

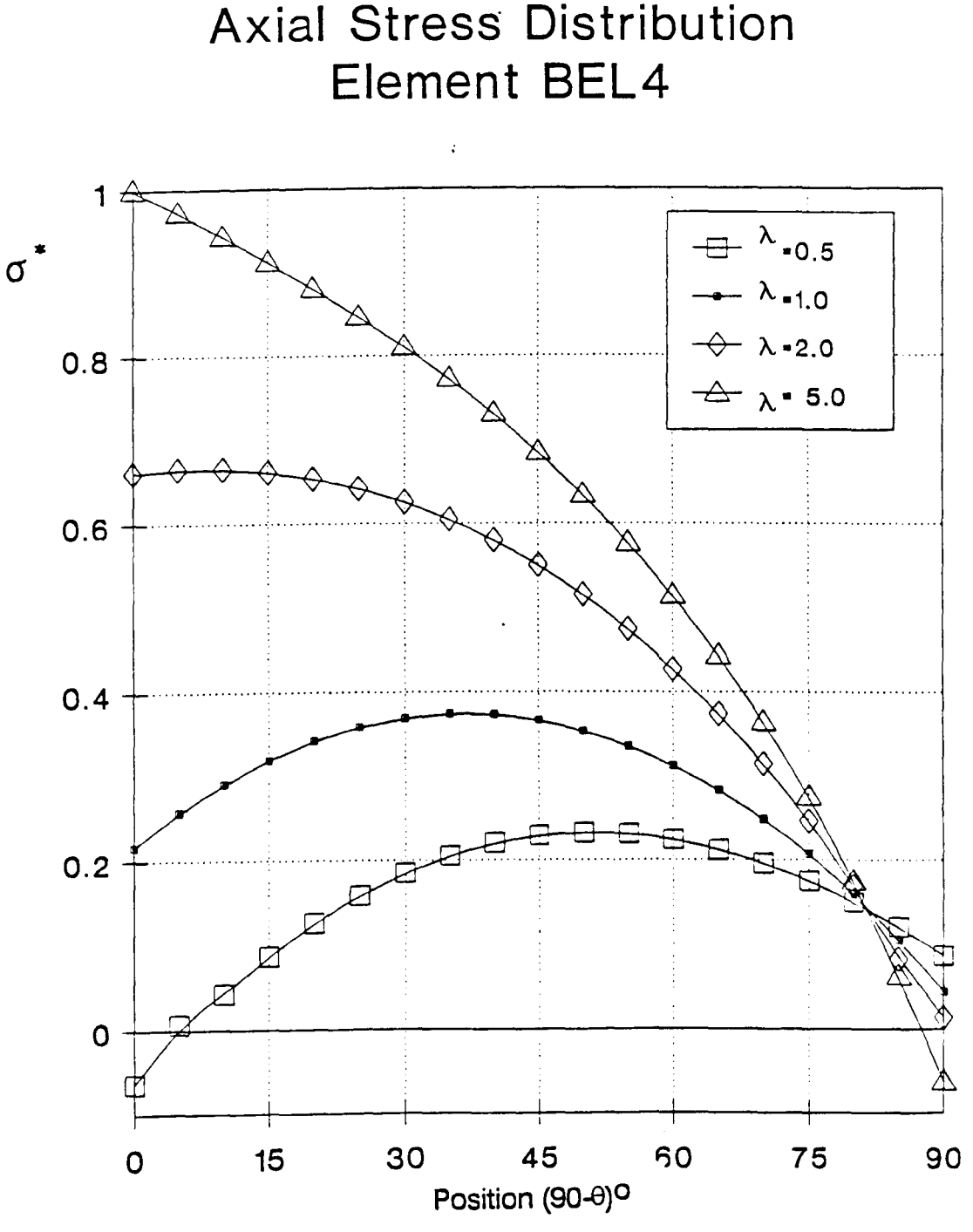
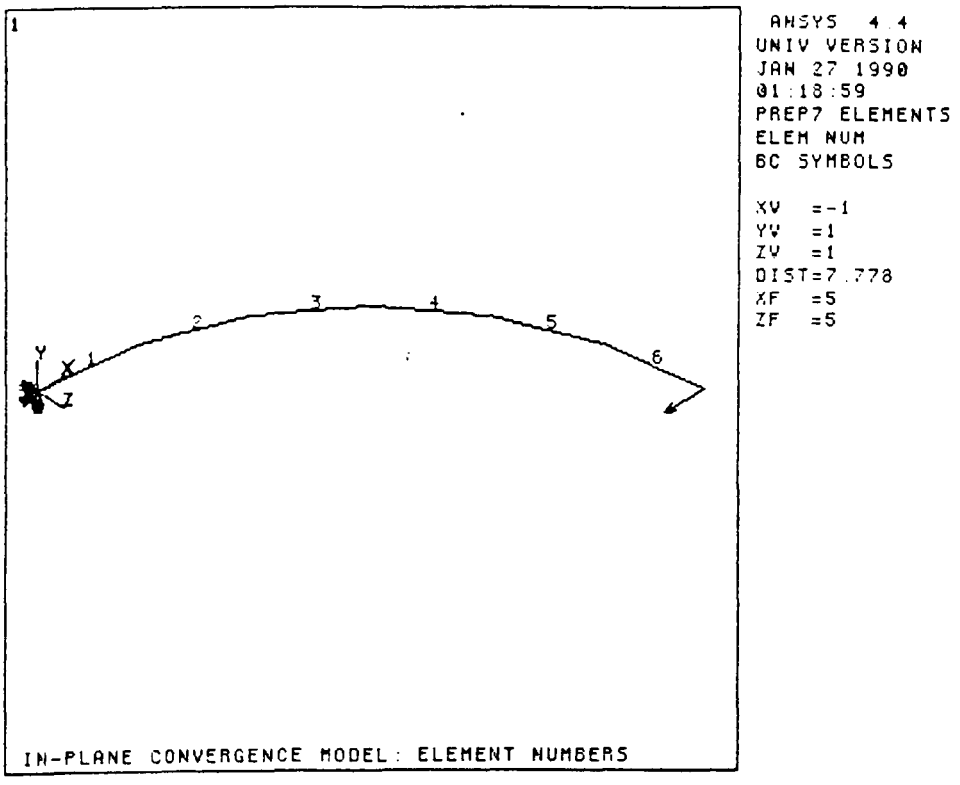


Figure 6.8 Bellows element BEL4 inner surface axial stress distribution.



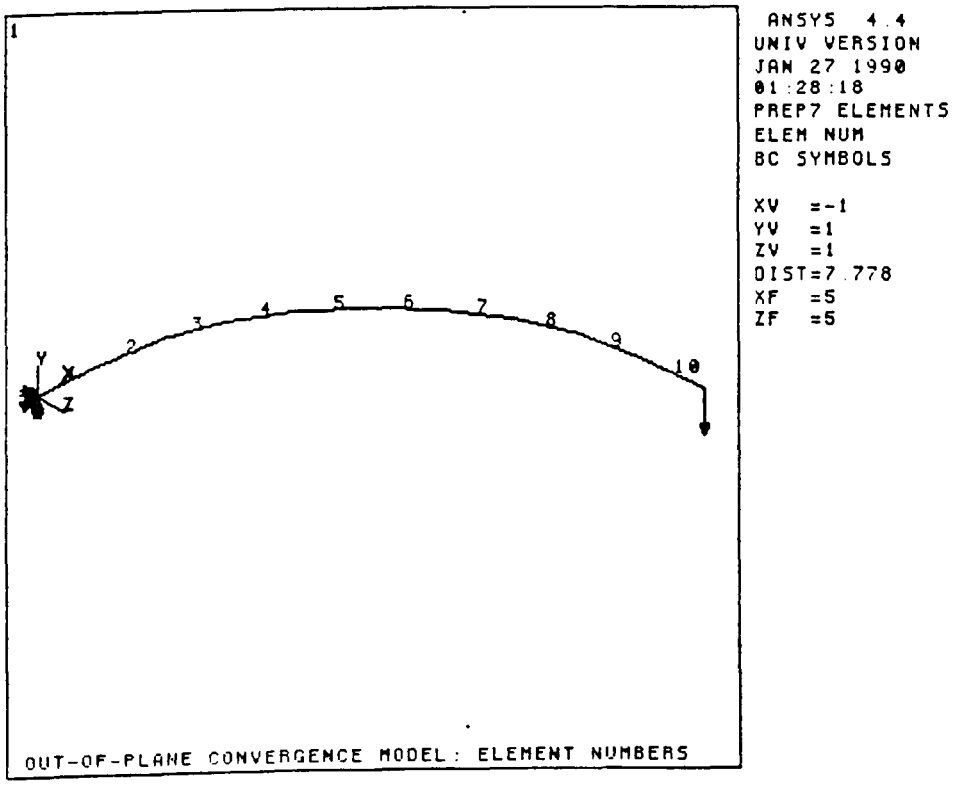
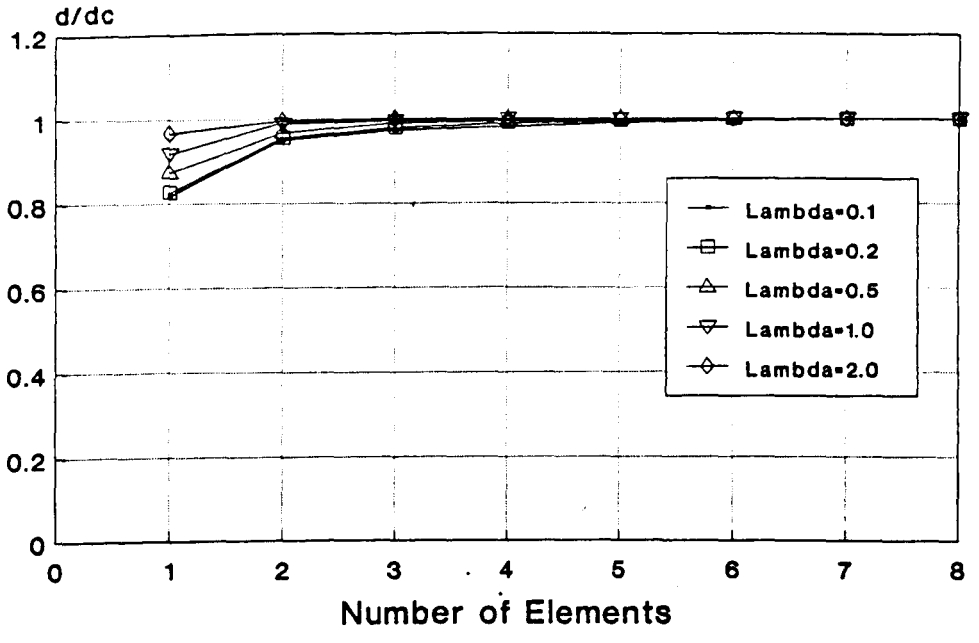


Figure 6.9 Typical in-plane and out-of-plane convergence test finite element models.



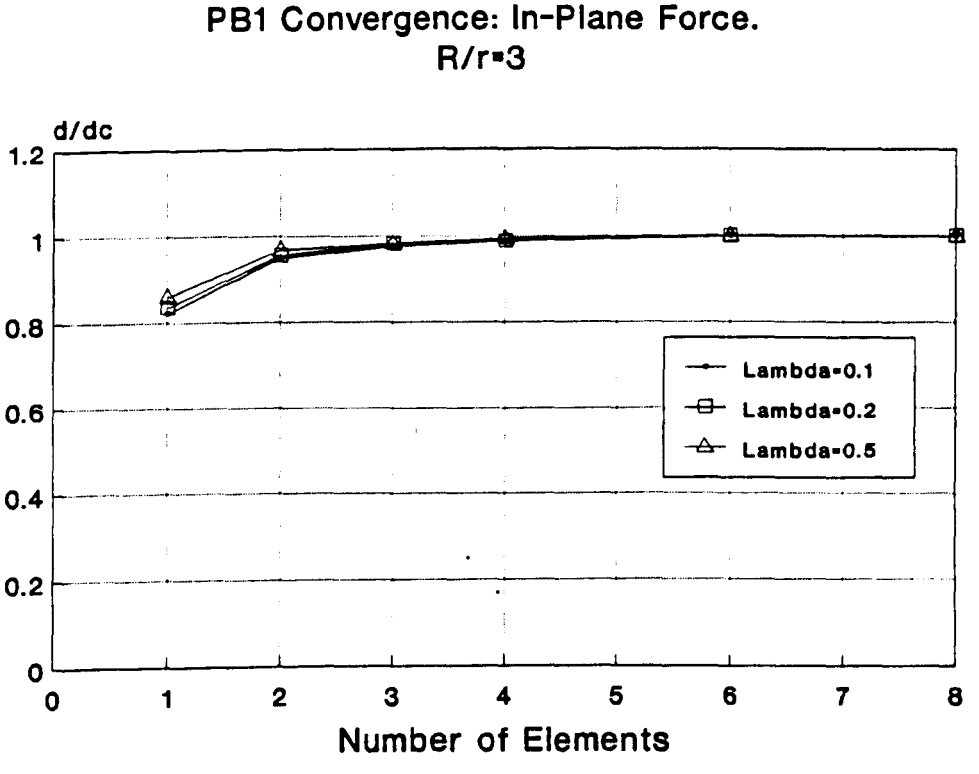
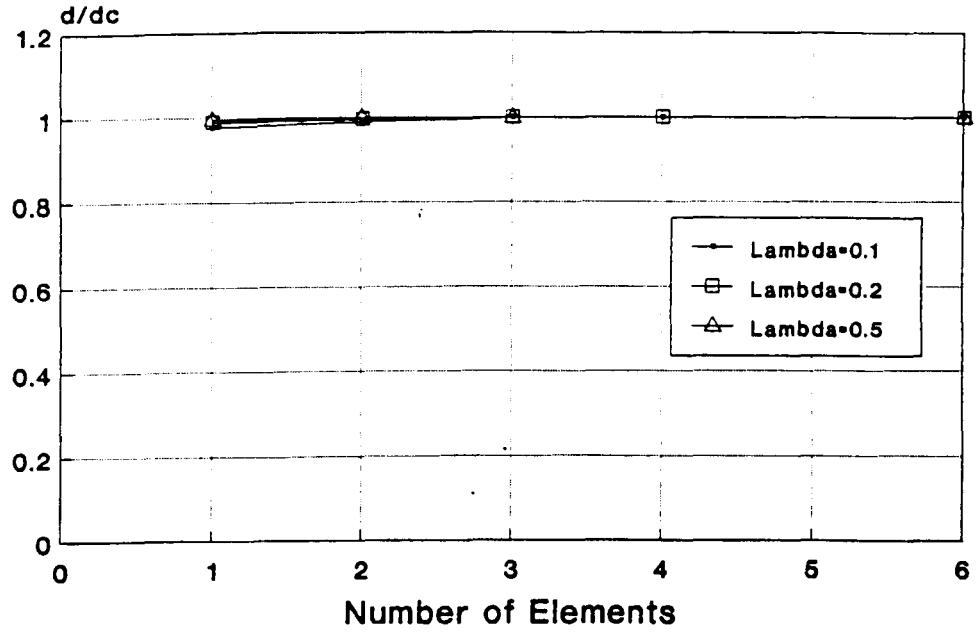


Figure 6.10 Elbow element PB1 in-plane shear force loading convergence plots.



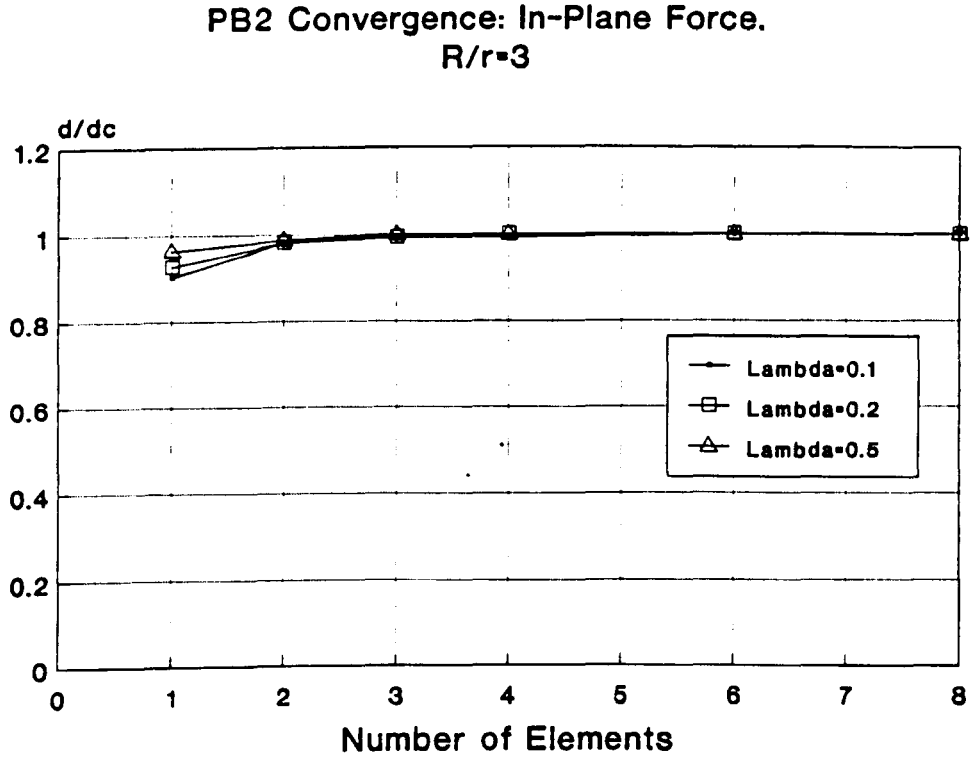


Figure 6.11 Elbow element PB2 in-plane shear force loading convergence plots.

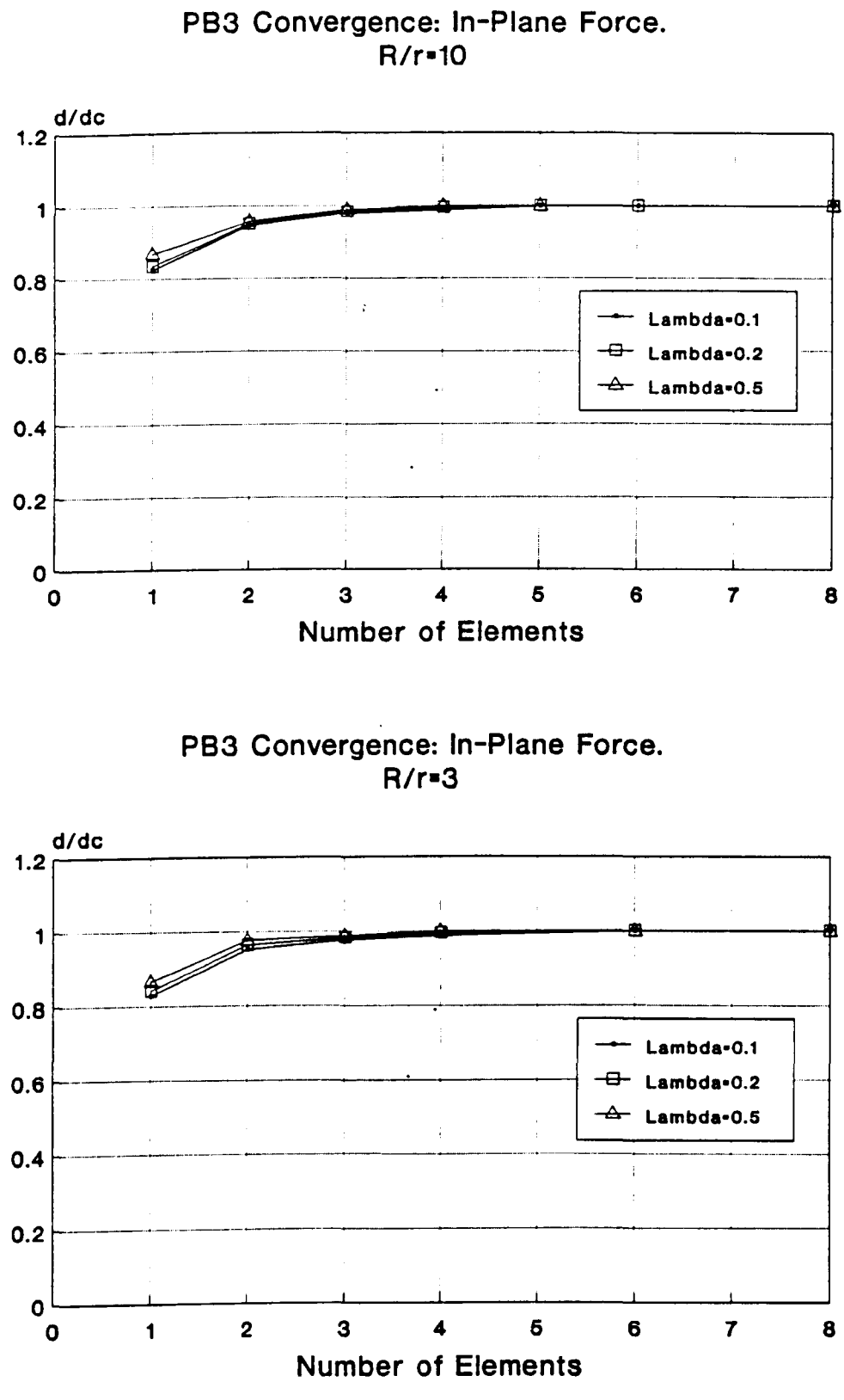
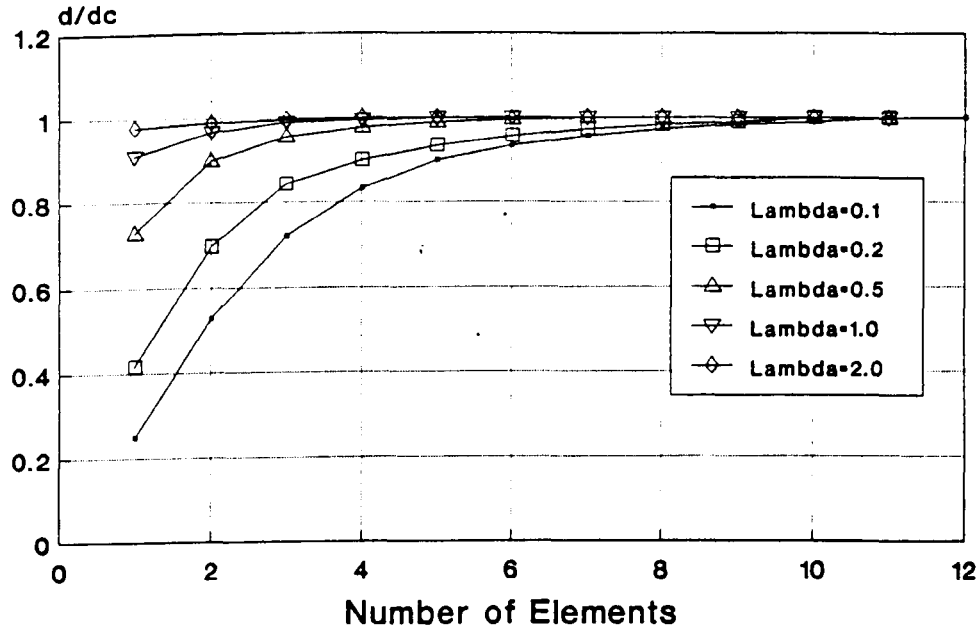


Figure 6.12 Elbow element PB3 in-plane shear force loading convergence plots.



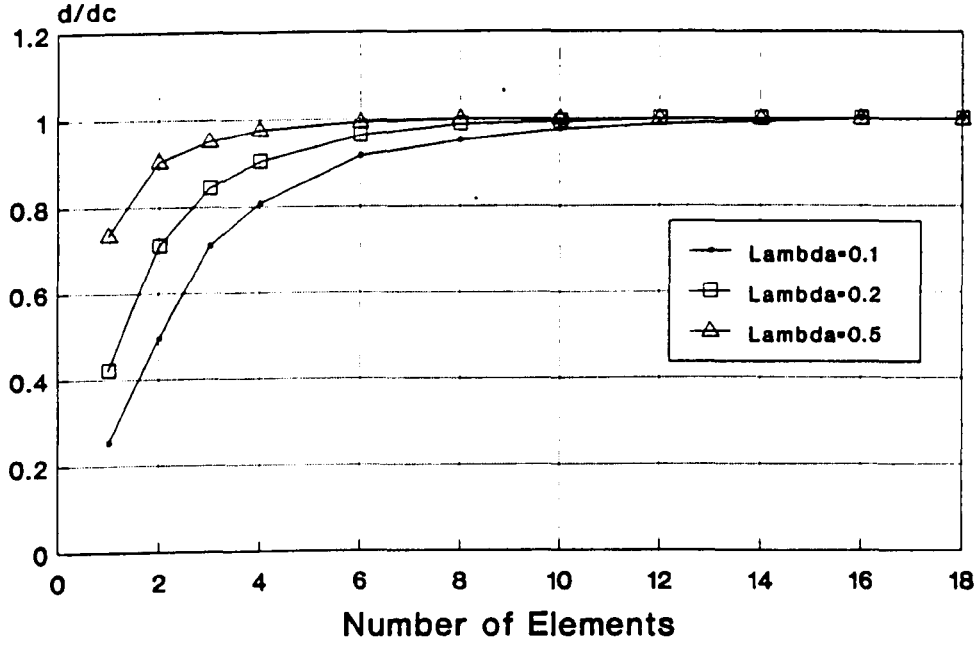
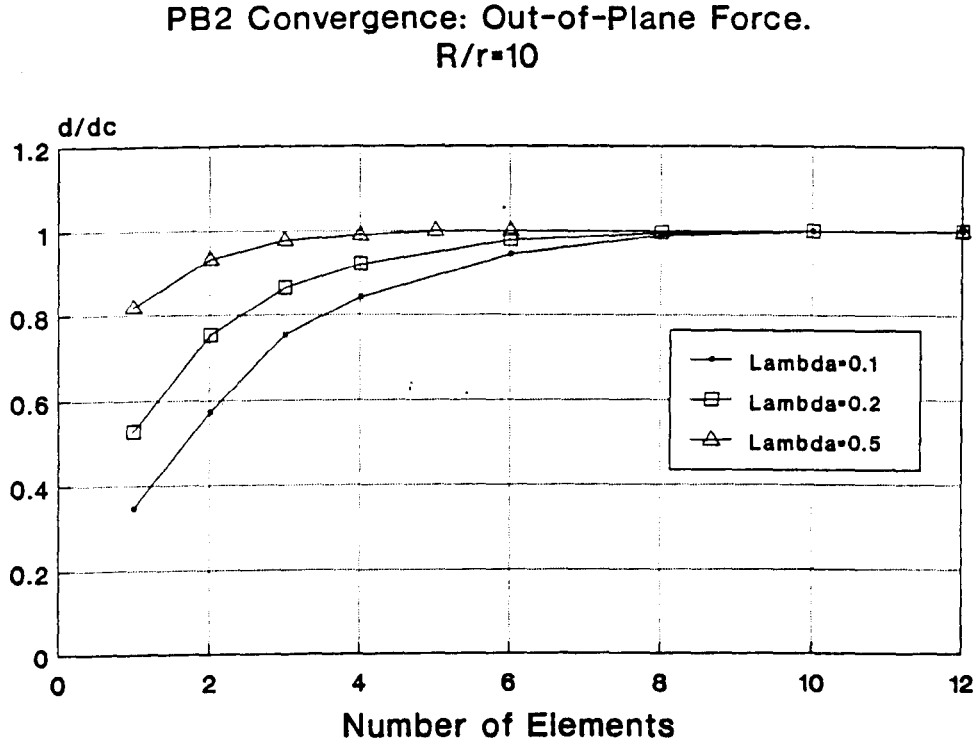


Figure 6.13 Elbow element PB1 out-of-plane shear force loading convergence plots.



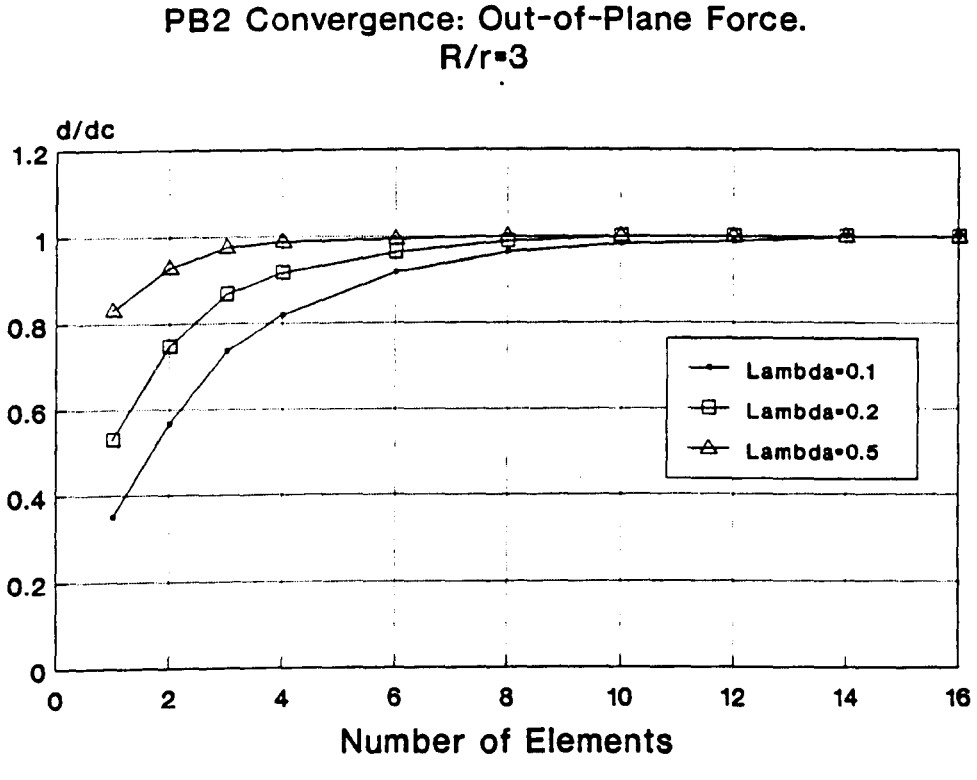
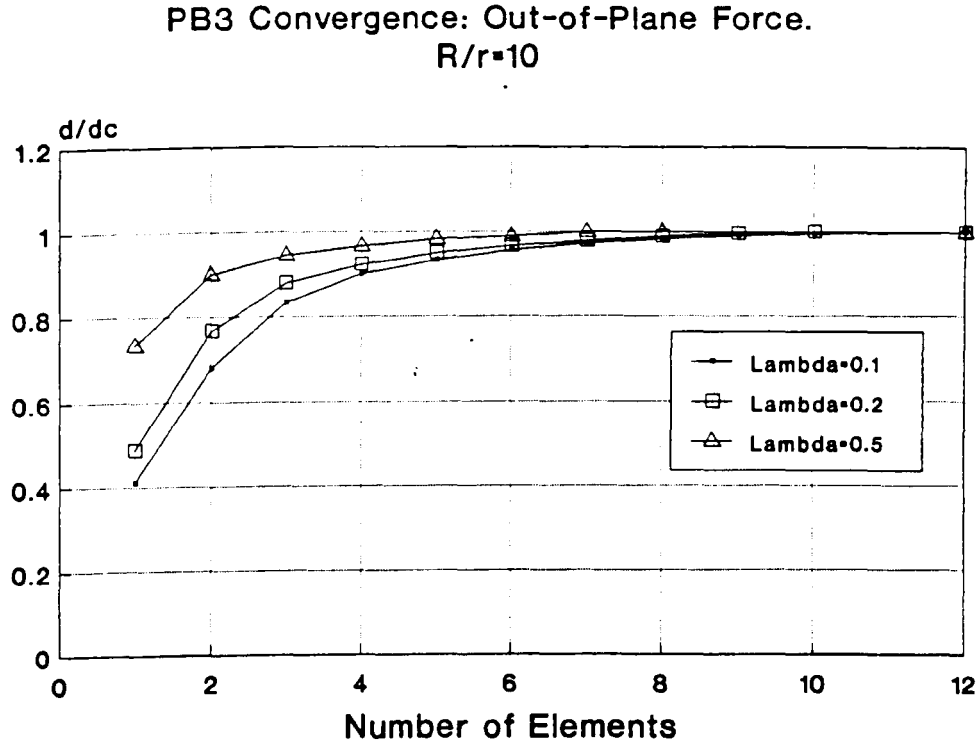


Figure 6.14 Elbow element PB2 out-of-plane shear force loading convergence plots.



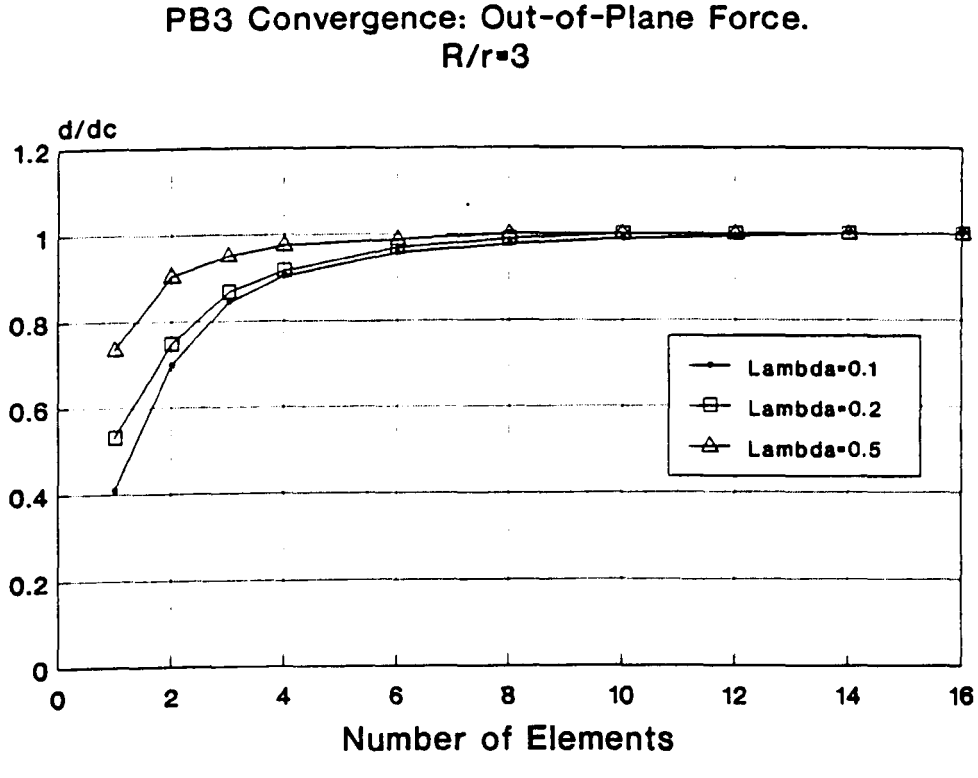


Figure 6.15 Elbow element PB3 out-of-plane shear force loading convergence plots.

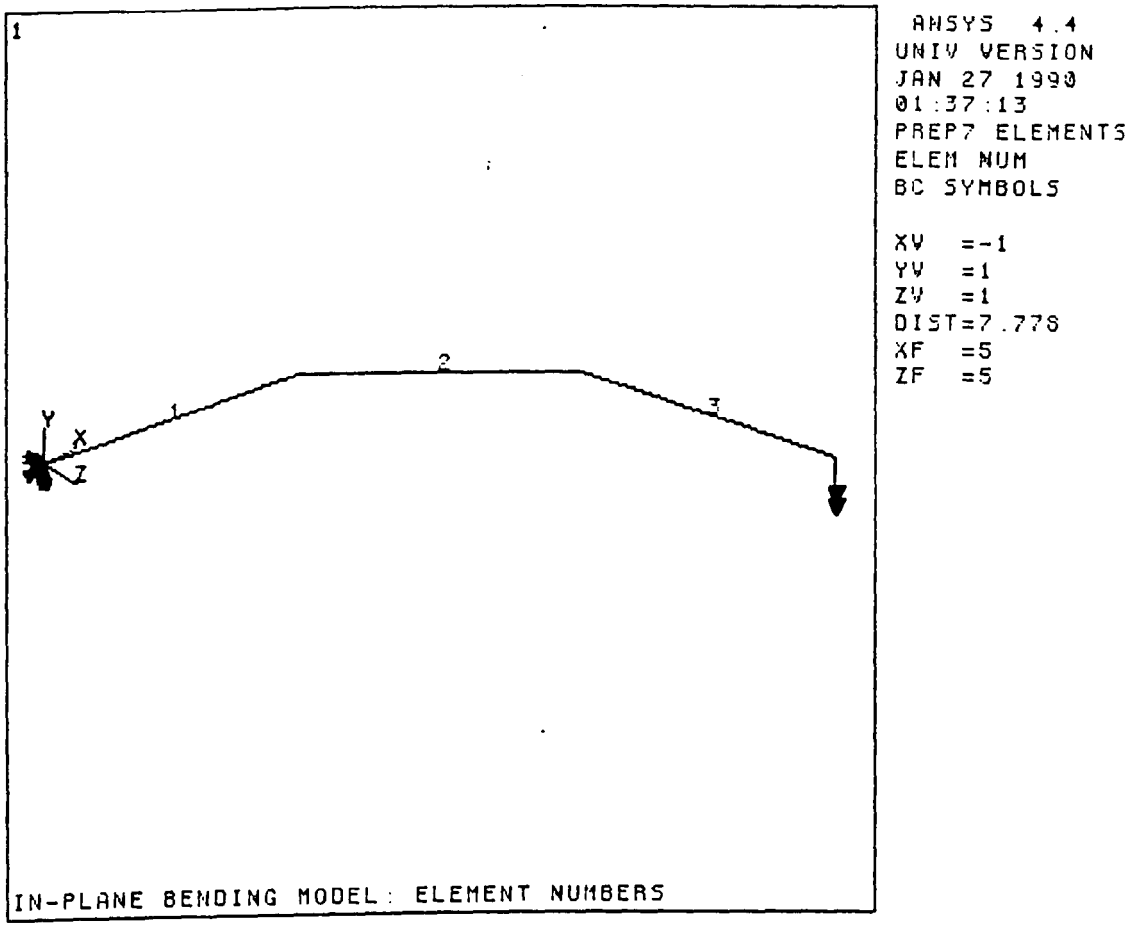
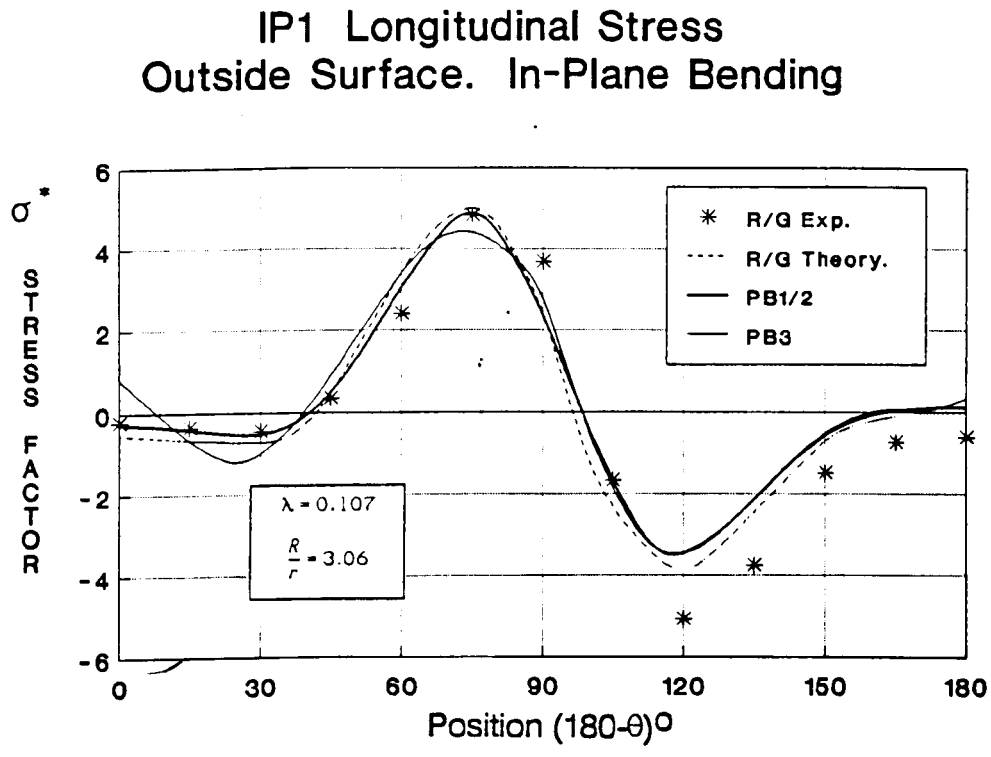


Figure 6.16 90^{o} bend, 3 element model: in-plane bending analysis.



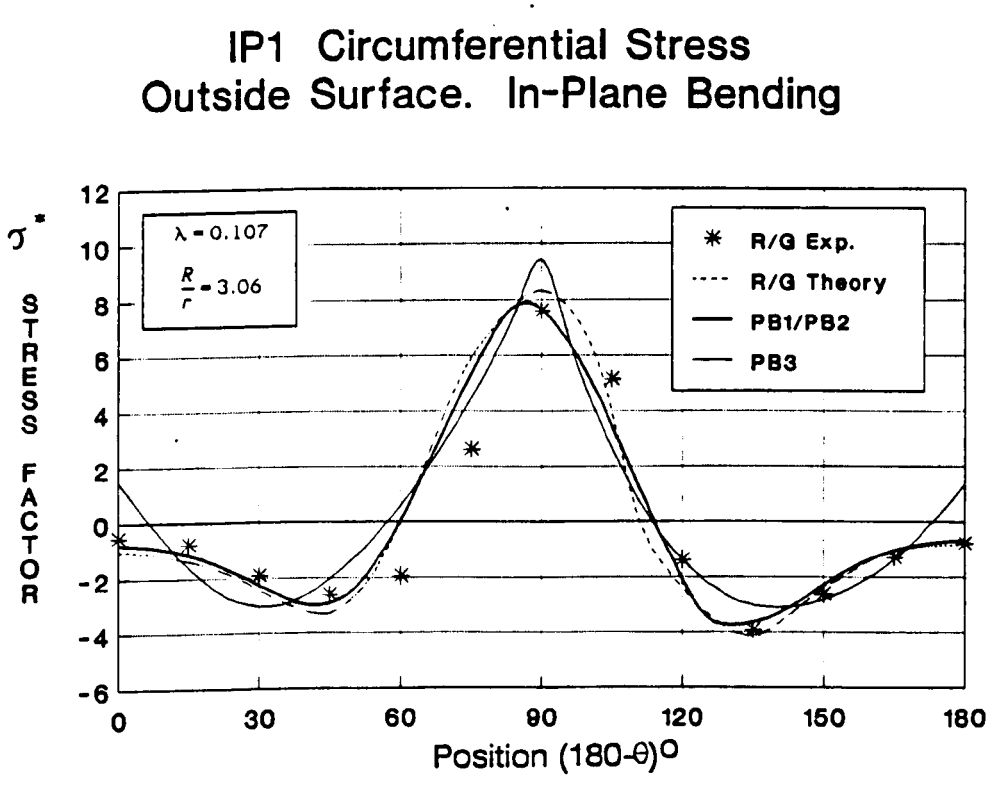
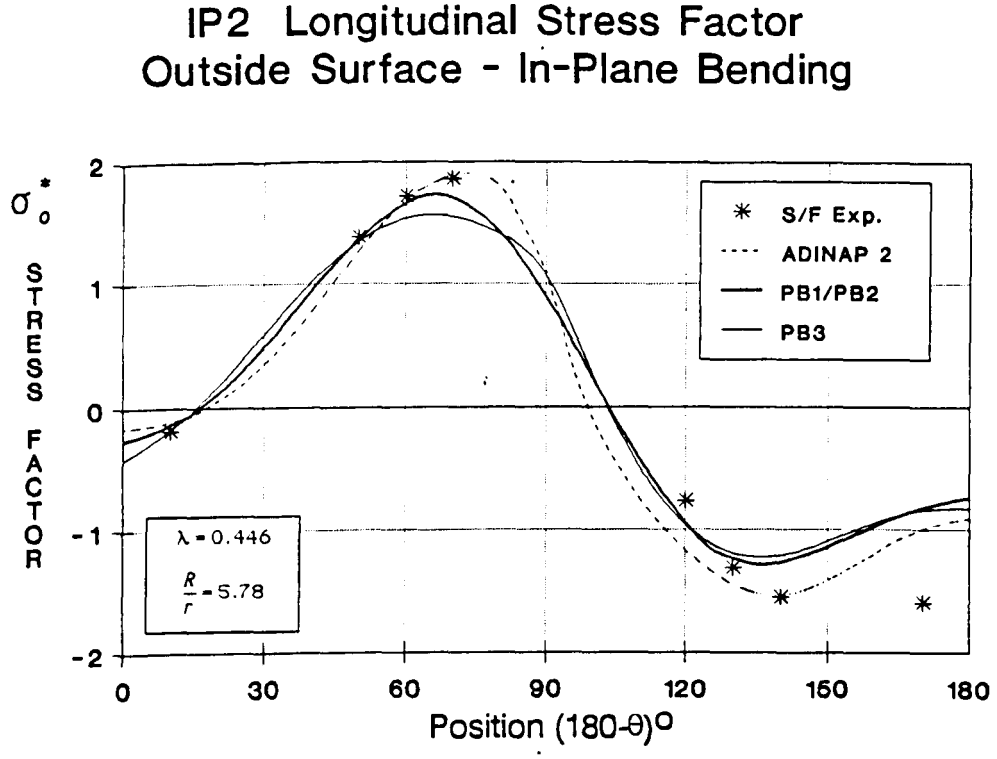


Figure 6.17 Bend IP1 stress distribution, outside surface. Comparison of elbow element stresses with Rodabaugh and George (R/G) theoretical and experimental results.



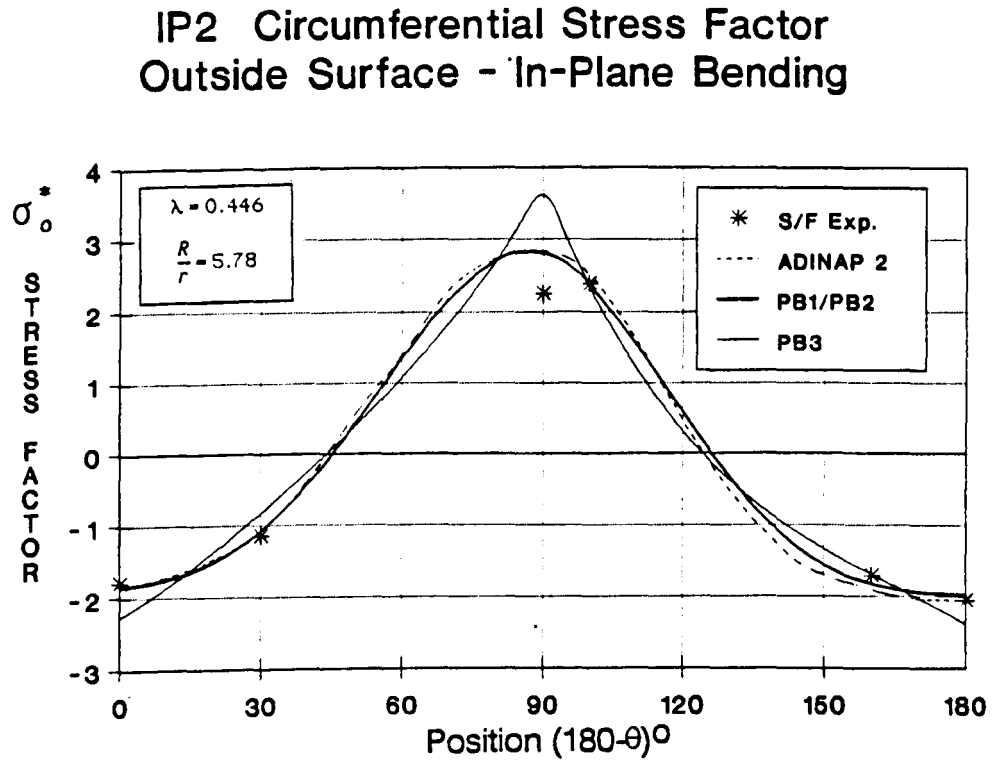
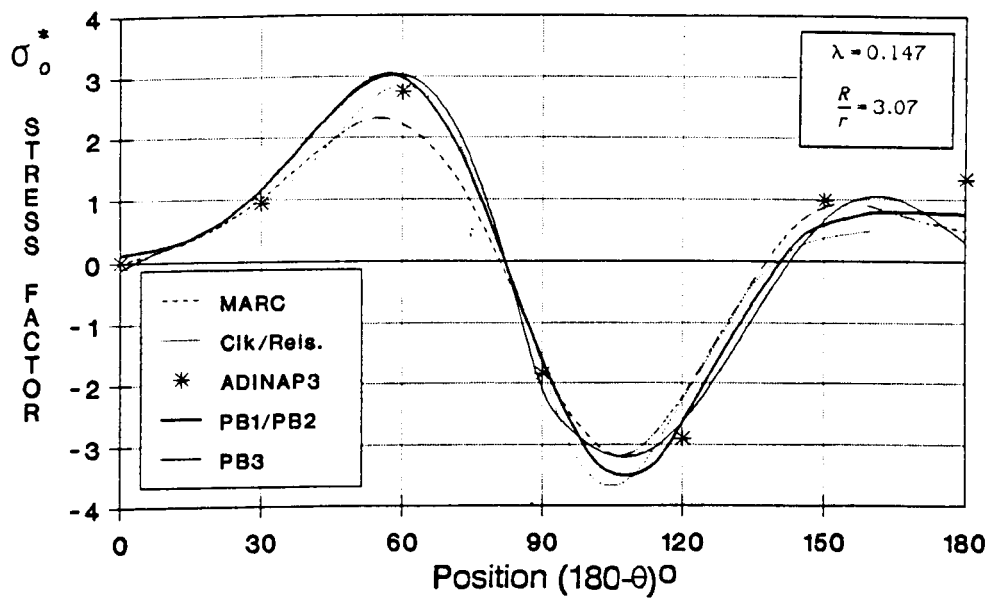


Figure 6.18 Bend IP2 stress distribution, outside surface. Comparison of elbow element stresses with ADINAP (2-Fourier mode) finite element solution and Smith and Ford (S/F) experimental results.



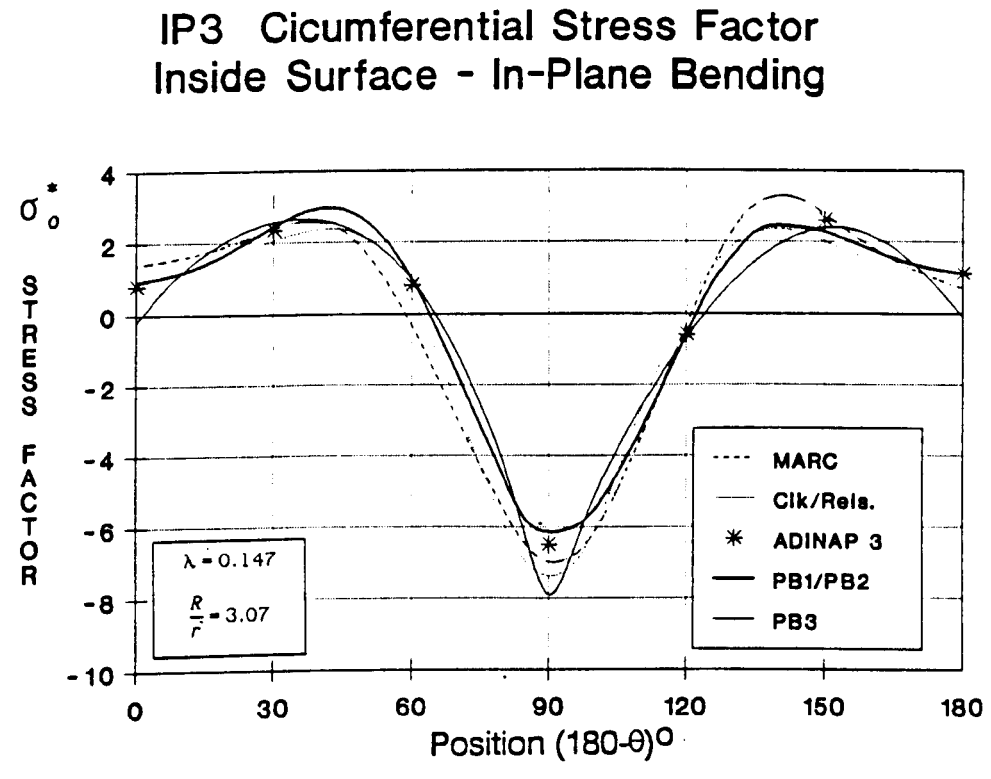


Figure 6.19 Bend IP3 stress distribution, inside surface. Comparison of elbow element stresses with MARC and ADINAP (3 Fourier mode) finite element solutions and Clark and Reissner (Clk/Reis) shell solution.

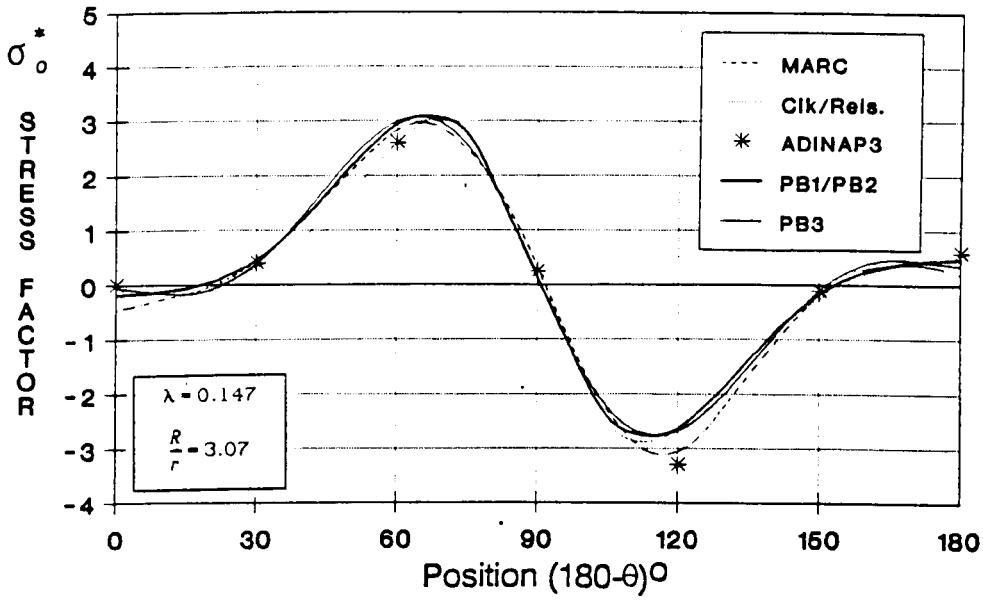
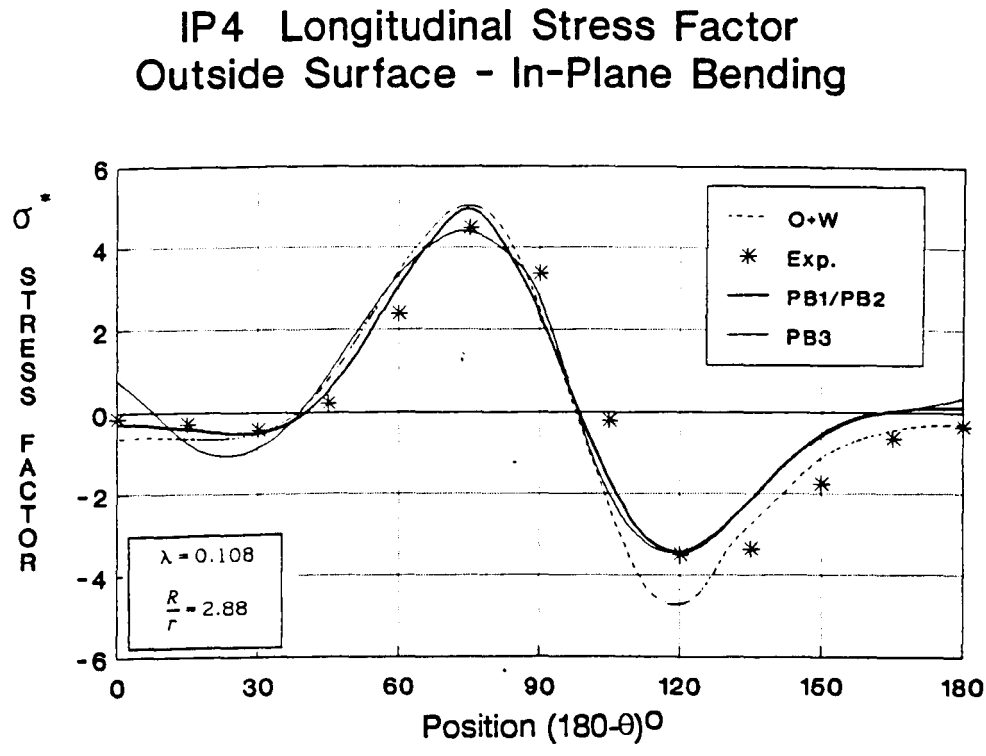


Figure 6.20 Bend IP3 longitudinal stress distribution, middle surface. Comparison of elbow element stresses with MARC and ADINAP (3 Fourier mode) finite element solutions and Clark and Reissner (Clk/Reis) shell solution.



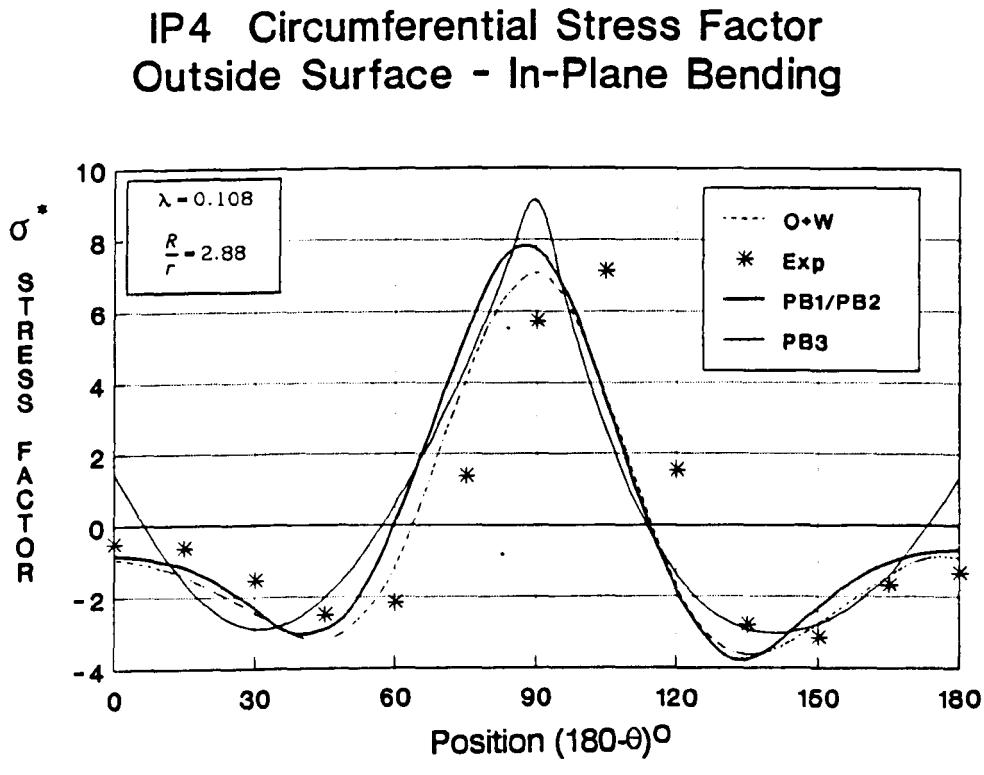


Figure 6.21 Bend IP4 stress distribution, outer surface. Comparison with Ohtsubo and Watanabe (O+W) ring element solution and Japan Welding Engineering Society experimental results.

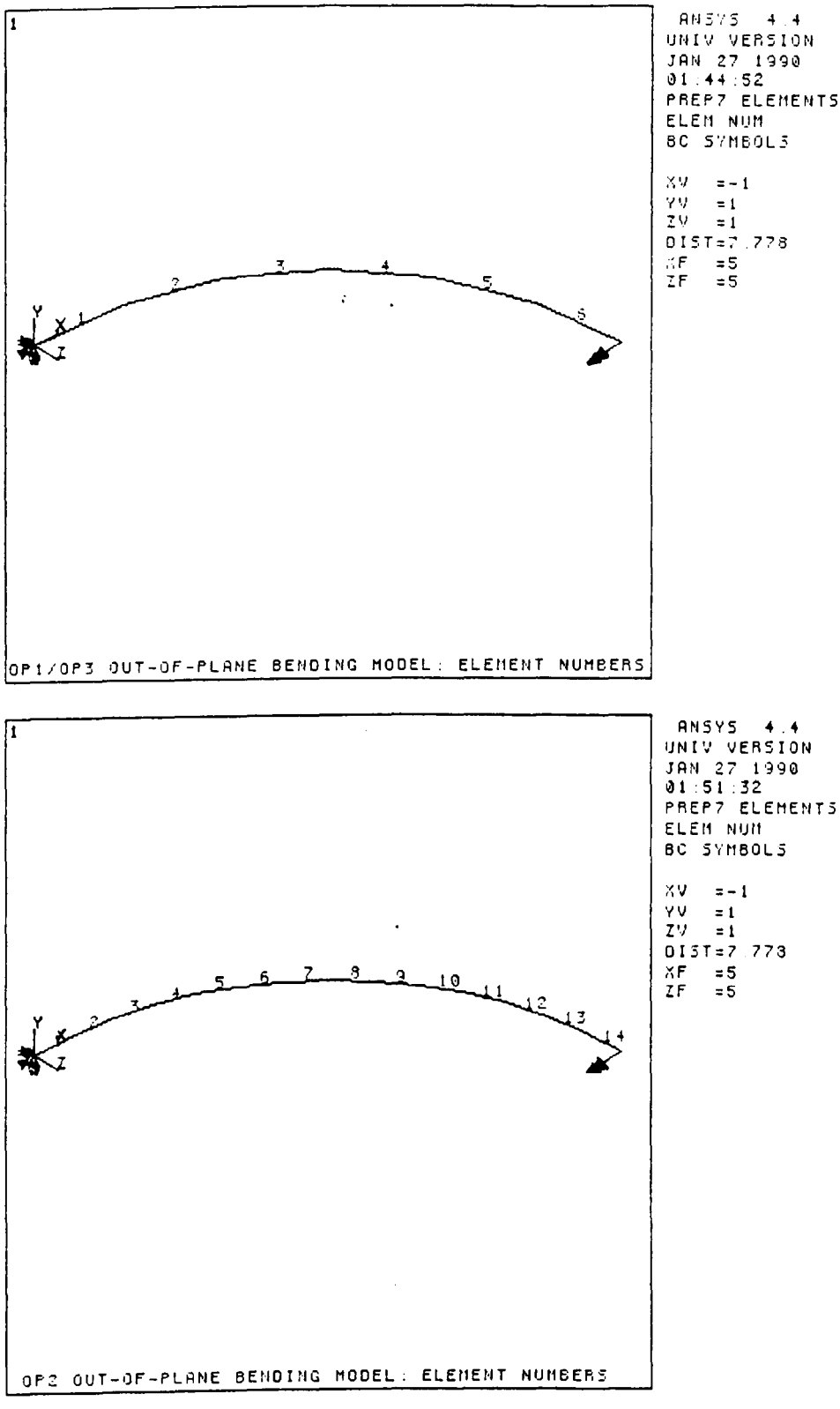


Figure 6.22 90^{o} bend, 6 and 14 element models: out-of-plane bending analysis.

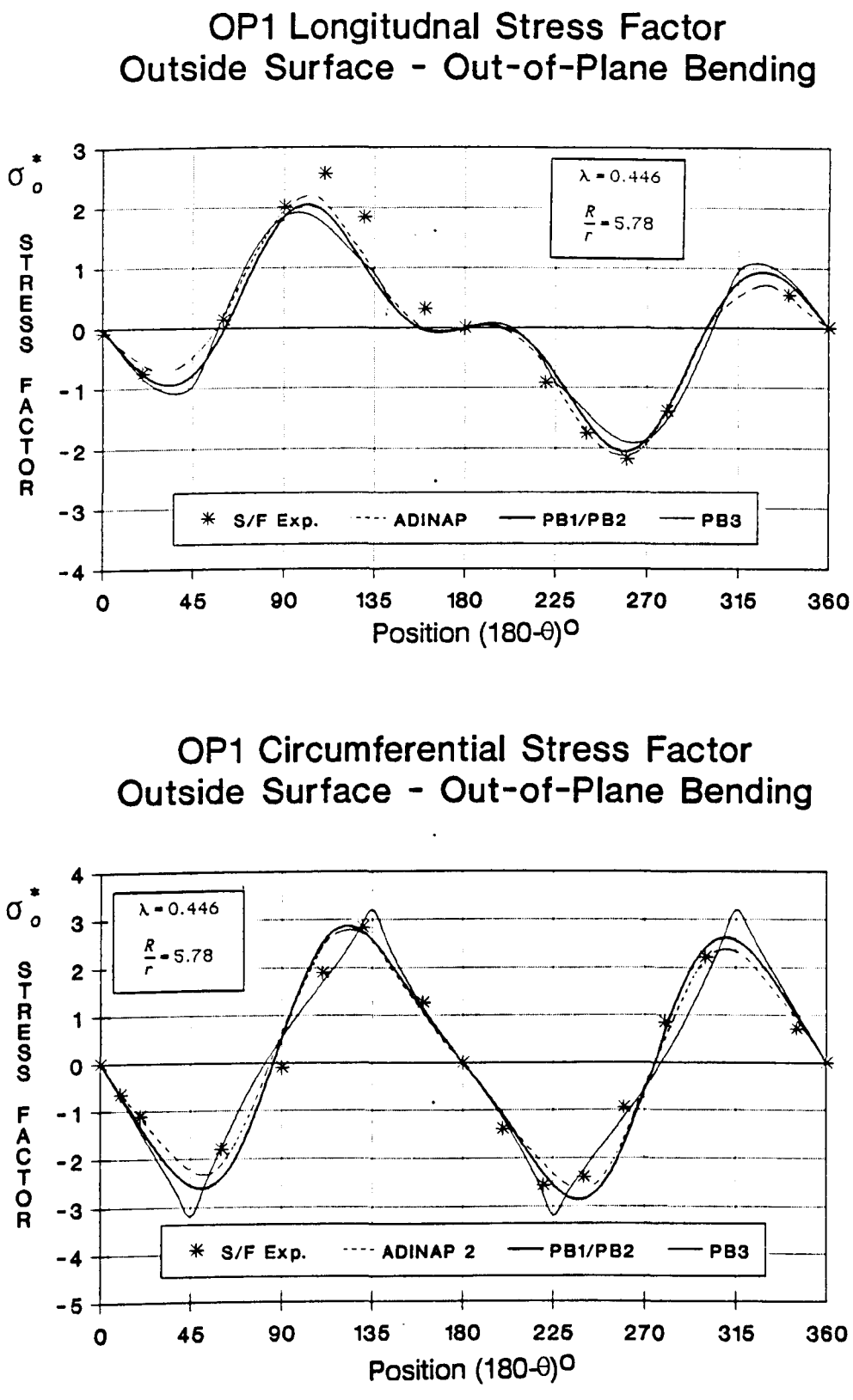


Figure 6.23 Bend OP1 stress distribution, outside surface. Comparison of elbow element stresses with ADINAP (2-Fourier mode) finite element solution and Smith and Ford (S/F) experimental results.

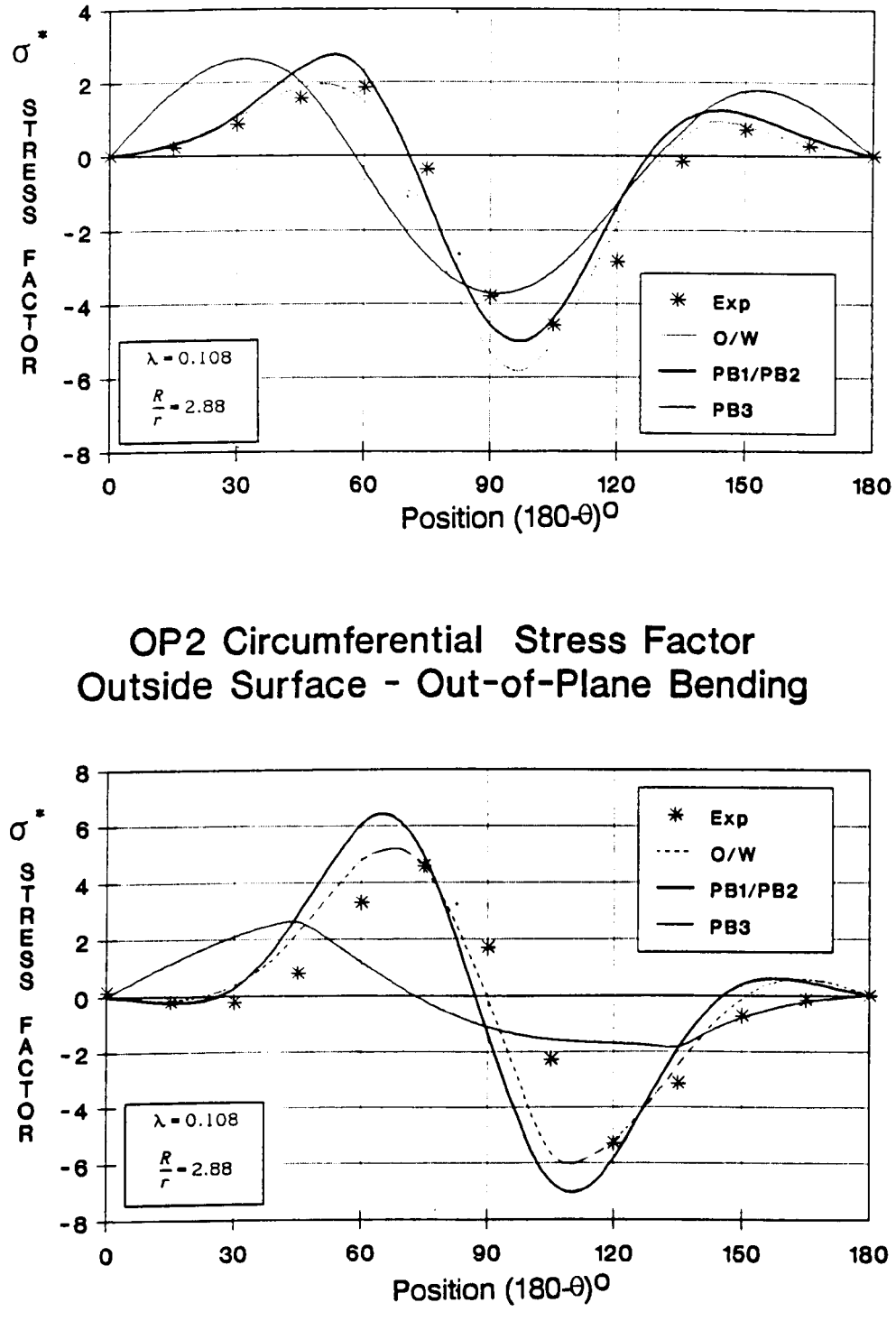
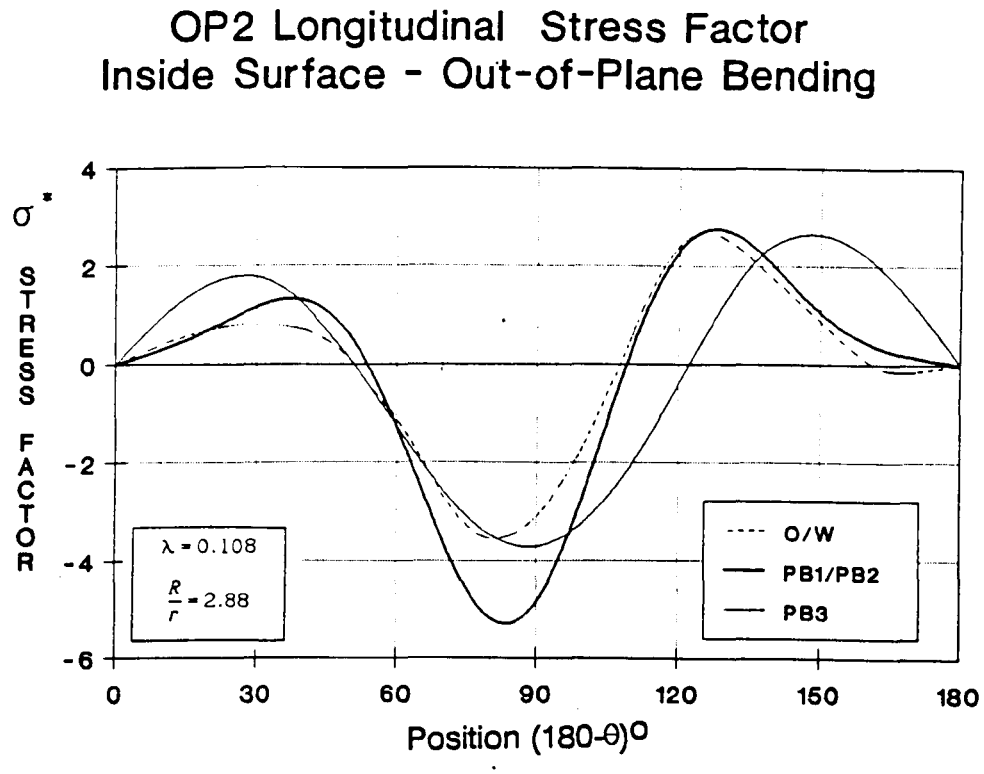


Figure 6.24 Bend OP2 stress distribution, outside surface. Comparison with Ohtsubo and Watanabe (O/W) ring element solution and Japan Welding Engineering Society experimental results.



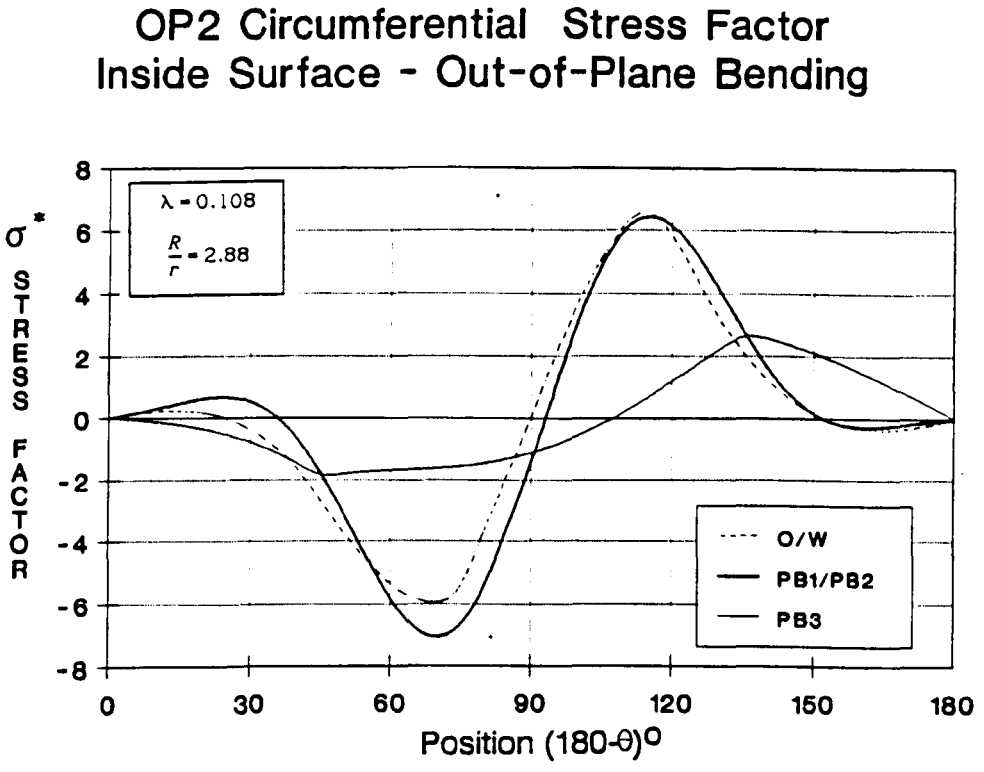


Figure 6.25 Bend OP2 stress distribution, inside surface. Comparison with Ohtsubo and Watanabe (O/W) ring element solution.

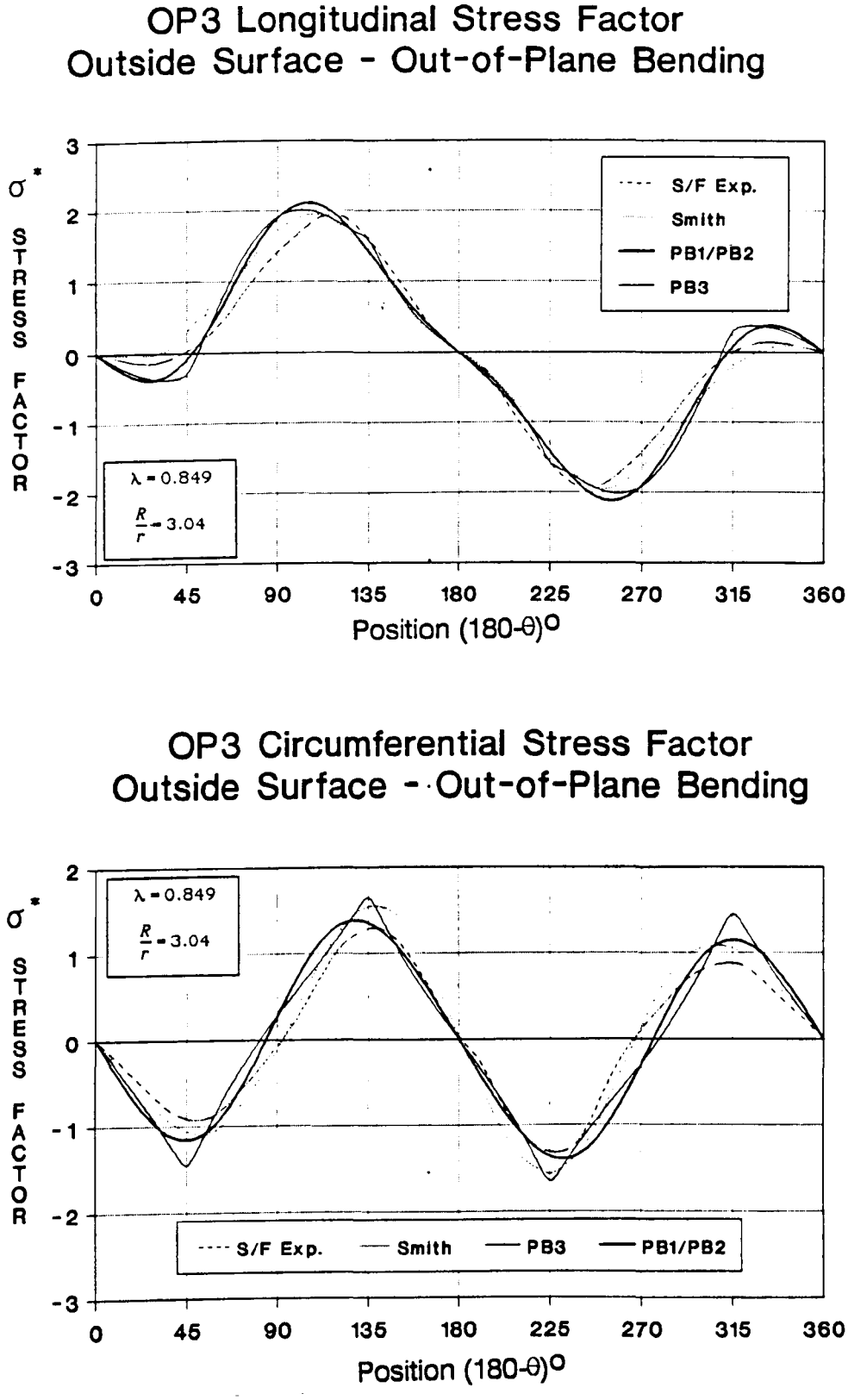


Figure 6.26 Bend OP3 stress distribution, outside surface. Comparison with Smith theoretical solution and Smith and Ford experimental results.

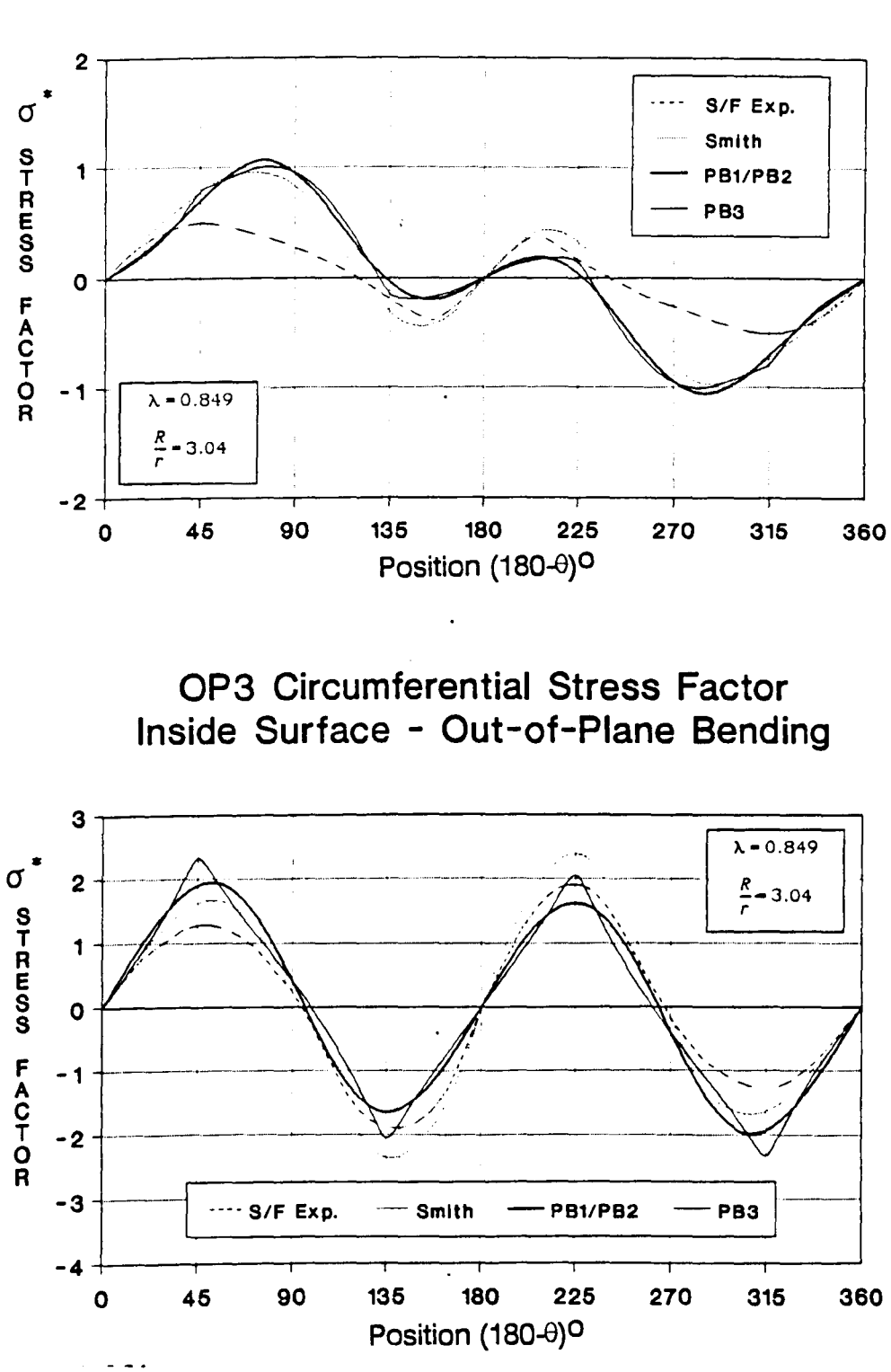


Figure 6.27 Bend OP3 stress distribution, inside surface. Comparison with Smith theoretical solution and Smith and Ford experimental results.

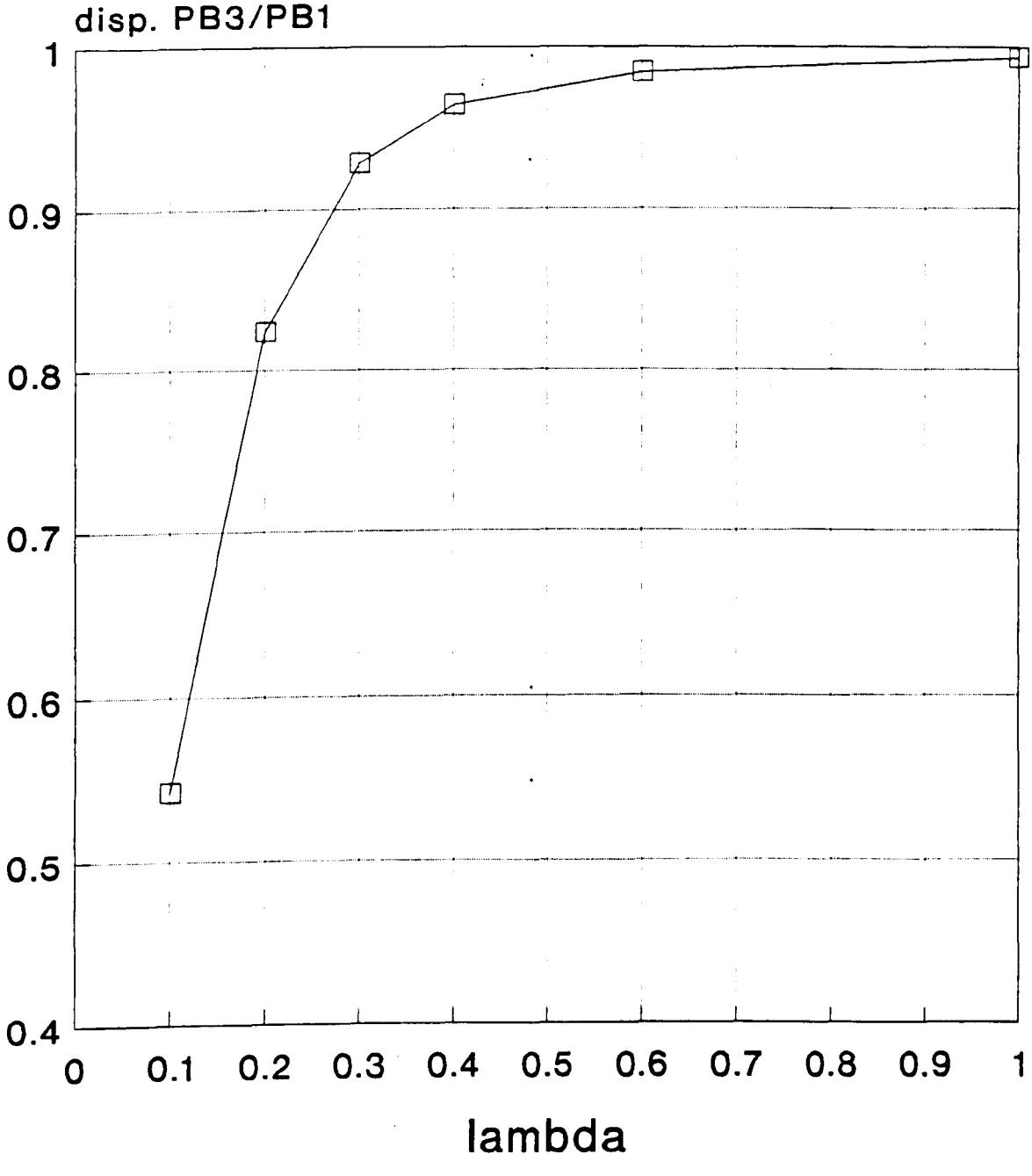
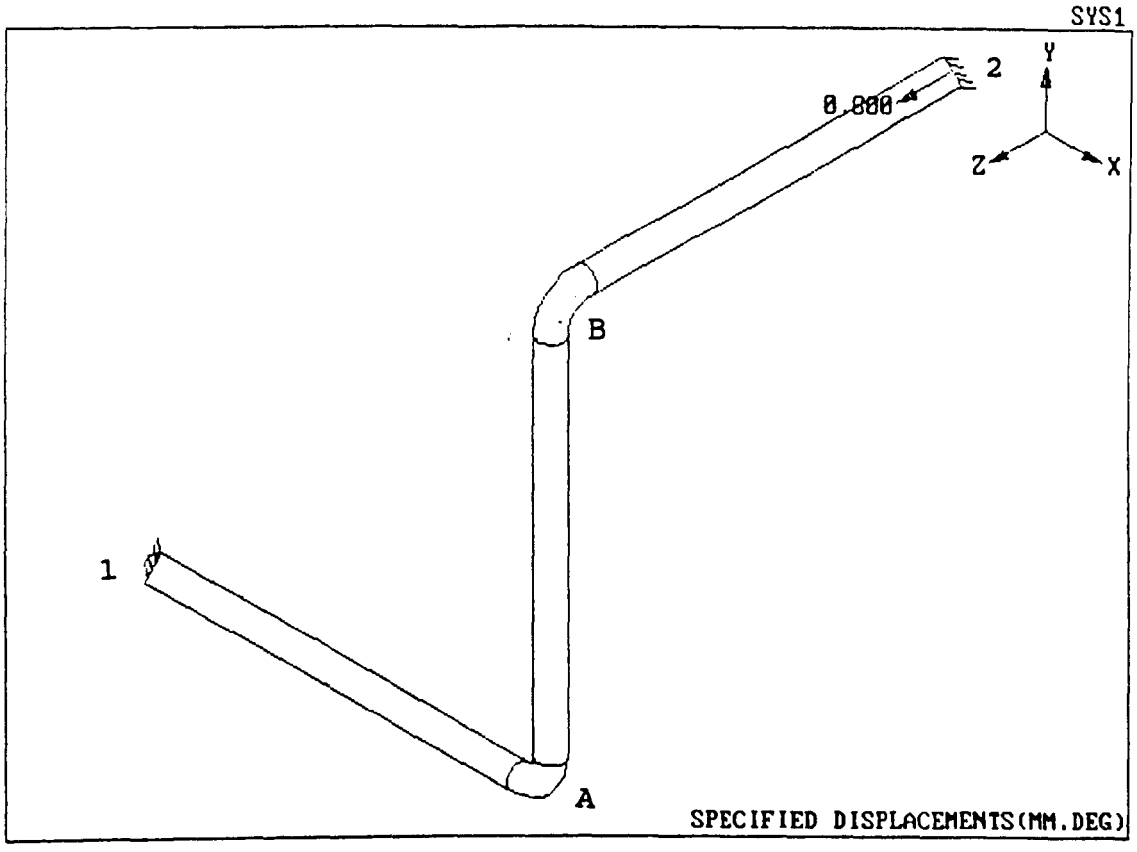


Figure 6.28 Ratio of out-of-plane displacement of element PB3 to PB1 versus bend parameter λ.



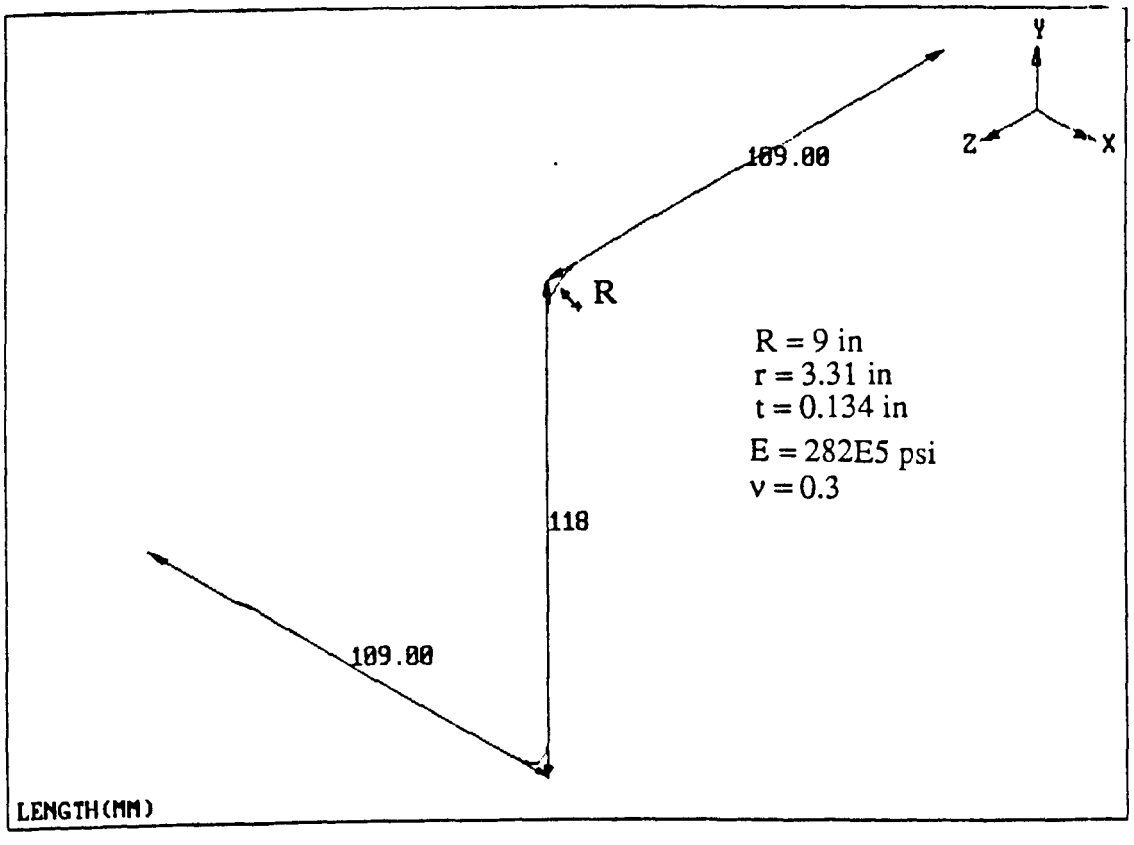


Figure 6.29 Piping system SYS1 : co-ordinate system, geometry, material properties and prescribed displacement.

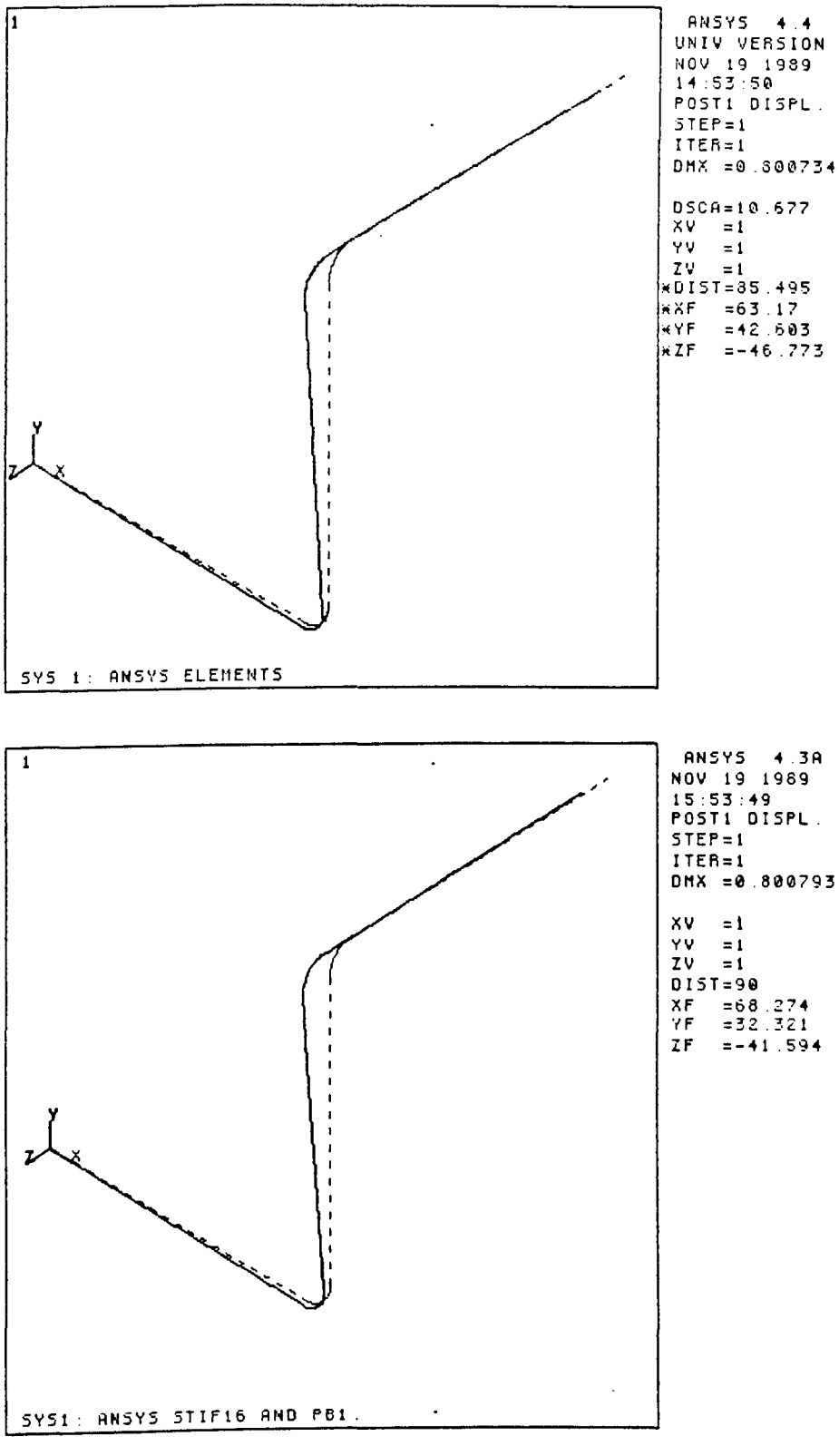
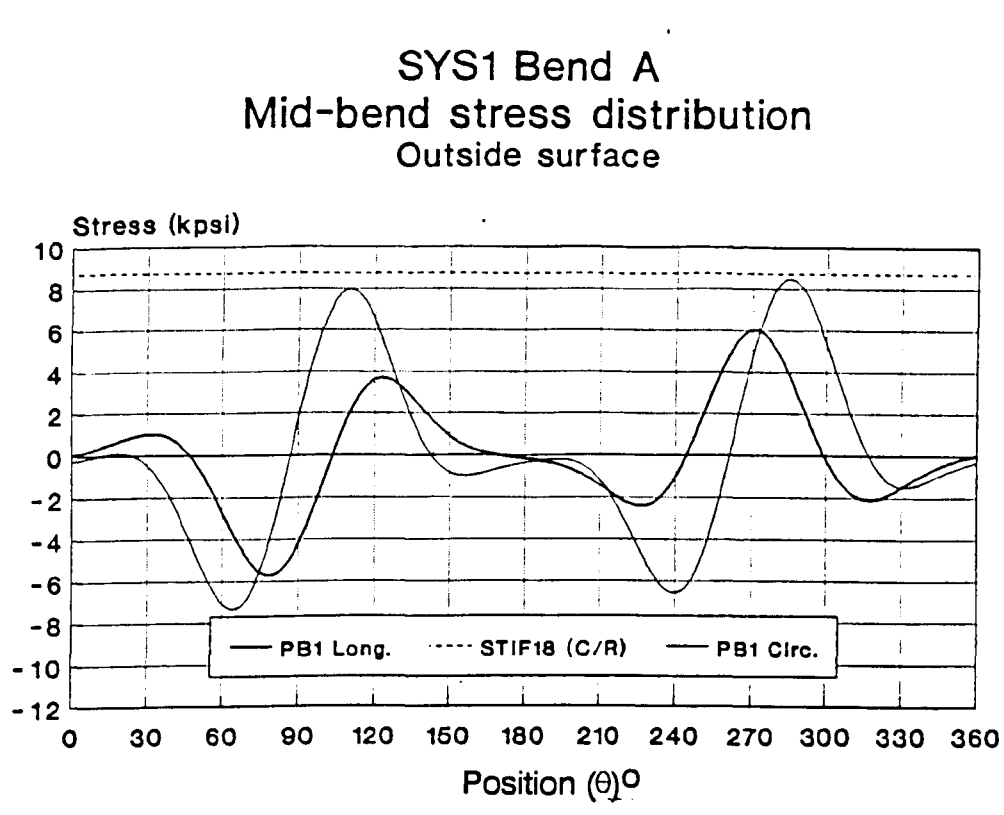


Figure 6.30 Piping system SYS1 deformed geometry plots.



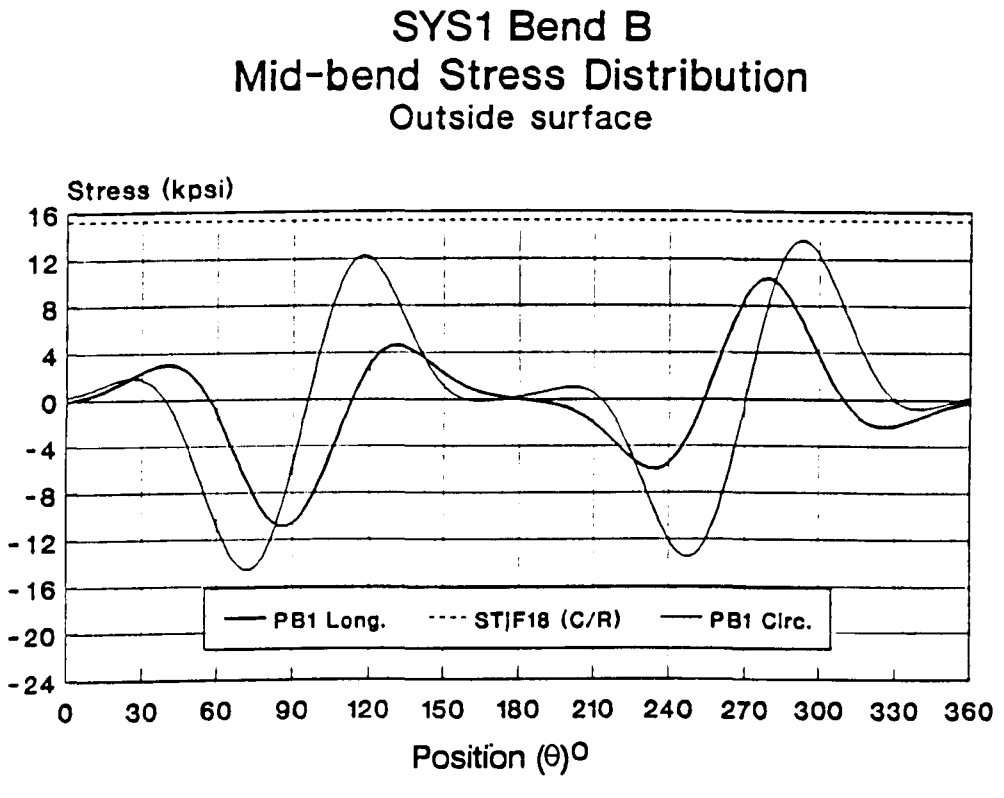


Figure 6.31 Piping system SYS1 bend A and B PB1 longitudinal and circumferential stress distribution.

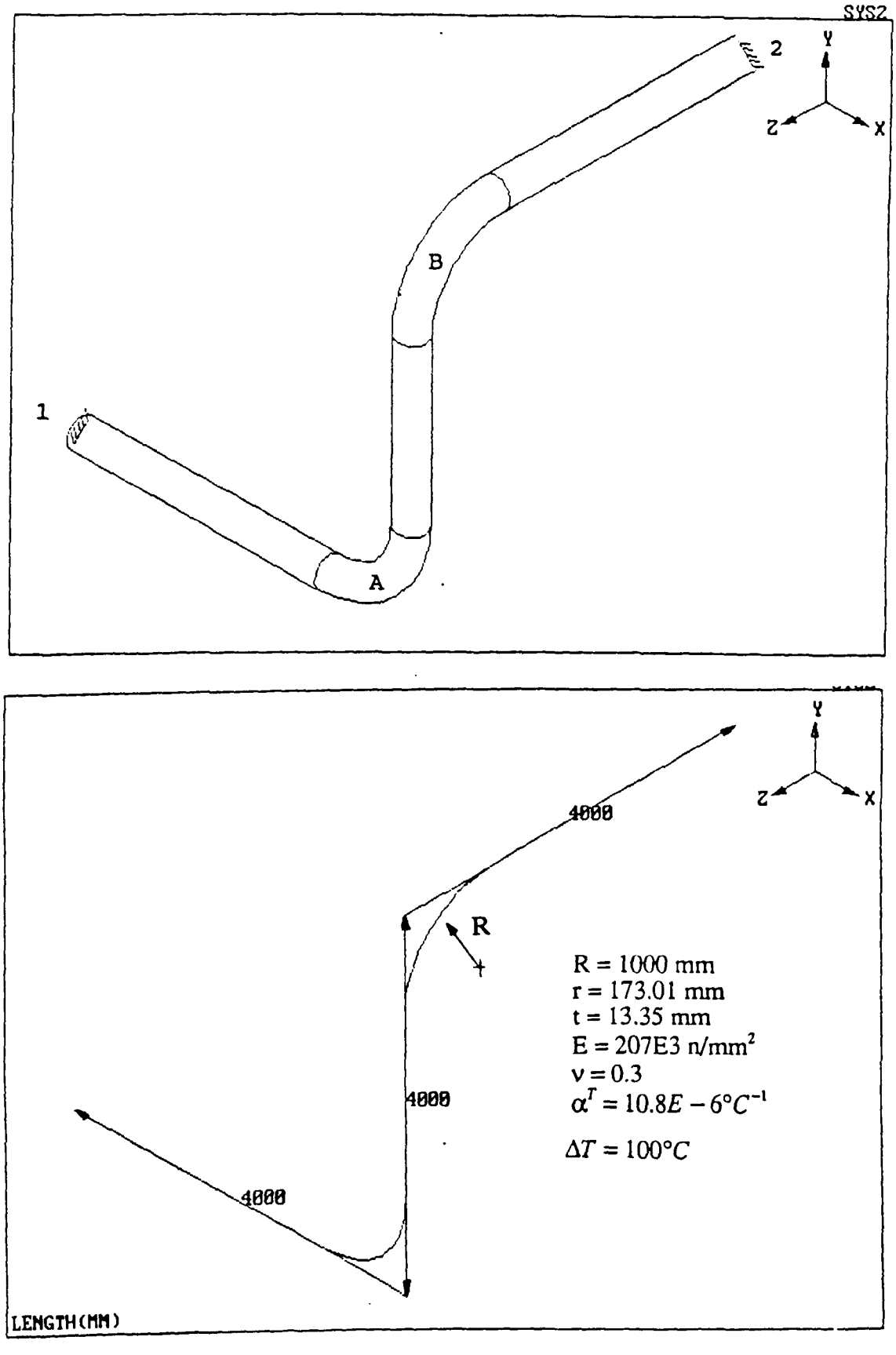


Figure 6.32 Piping system SYS2 : coordinate system, geometry, material properties and applied uniform temperature difference.

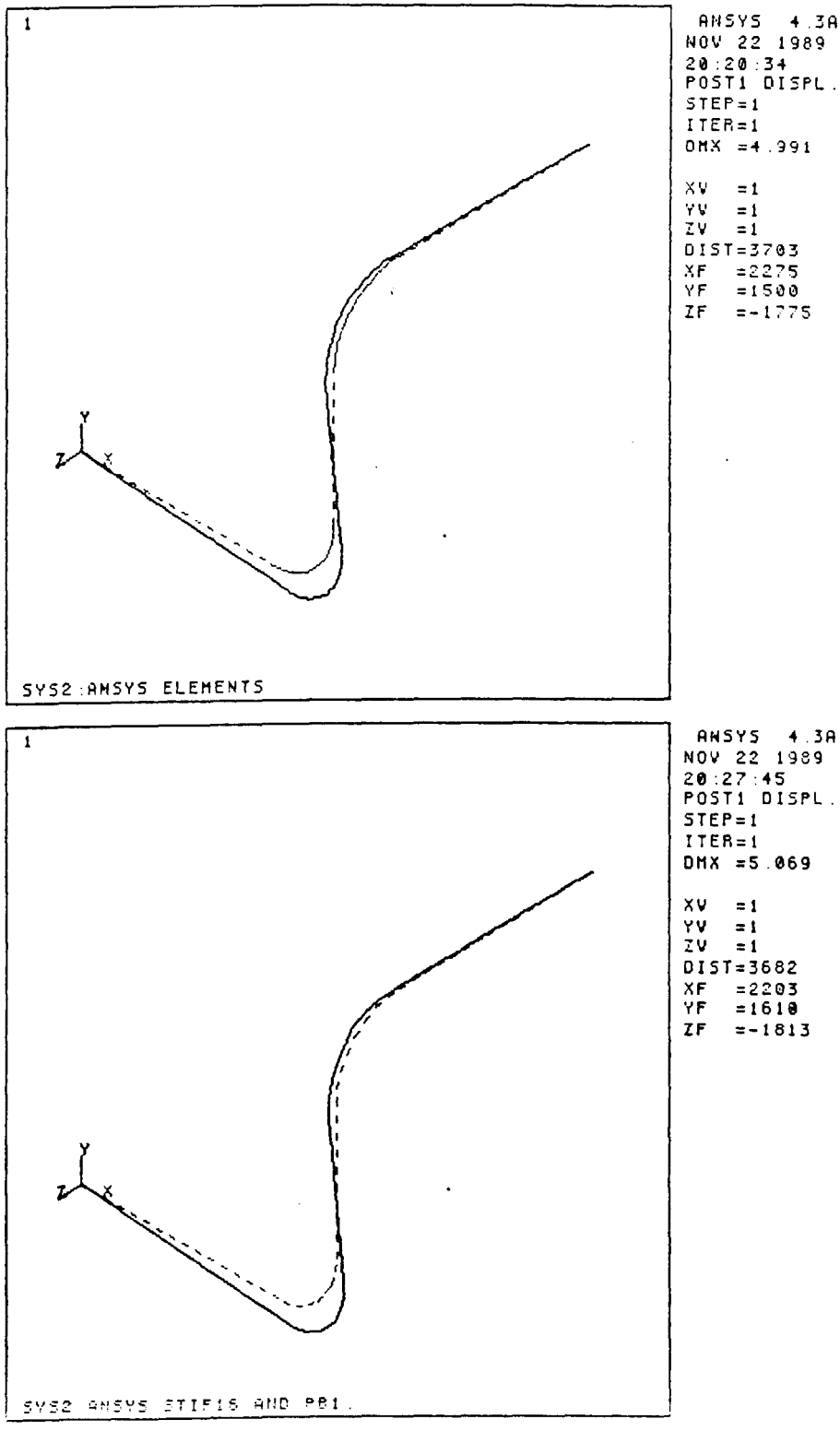
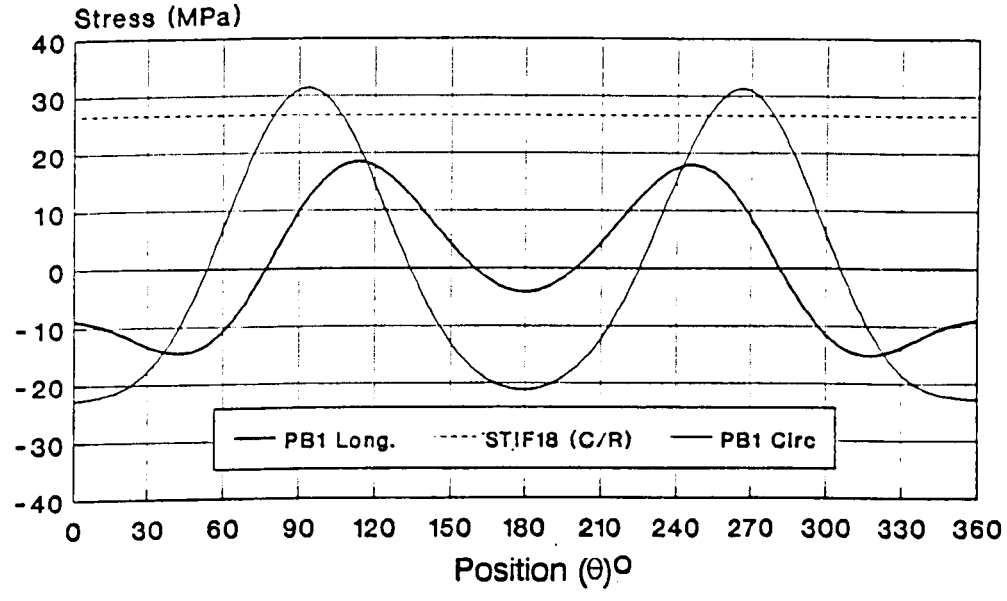


Figure 6.33 Piping system SYS2 deformed geometry plots.



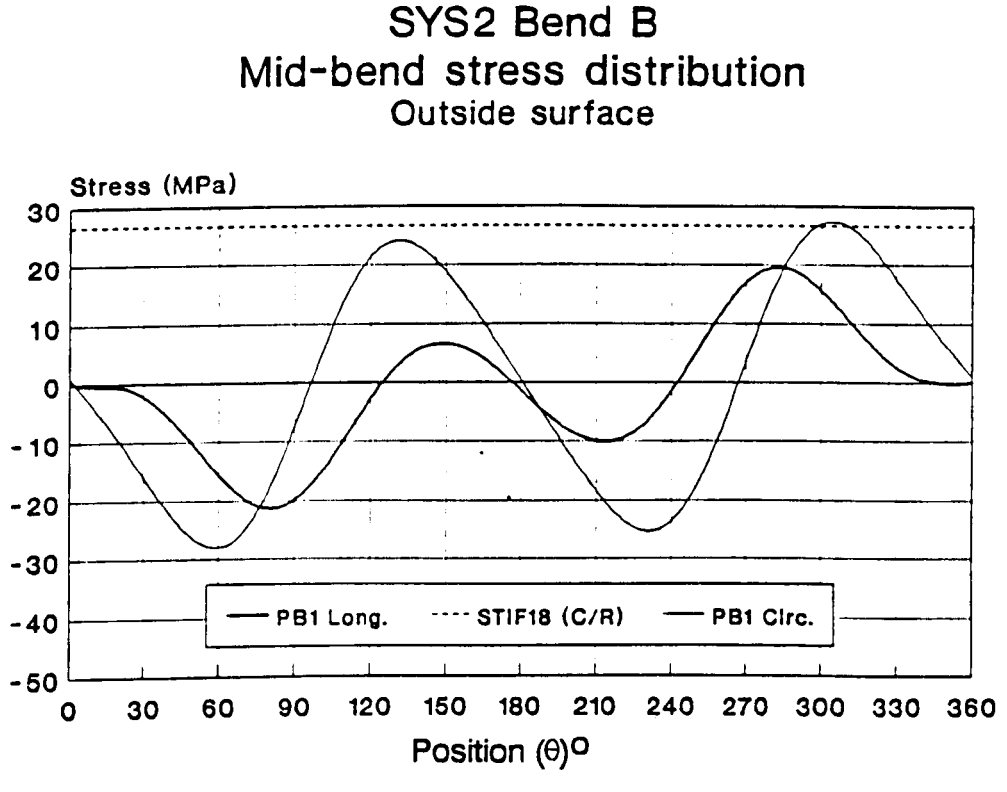
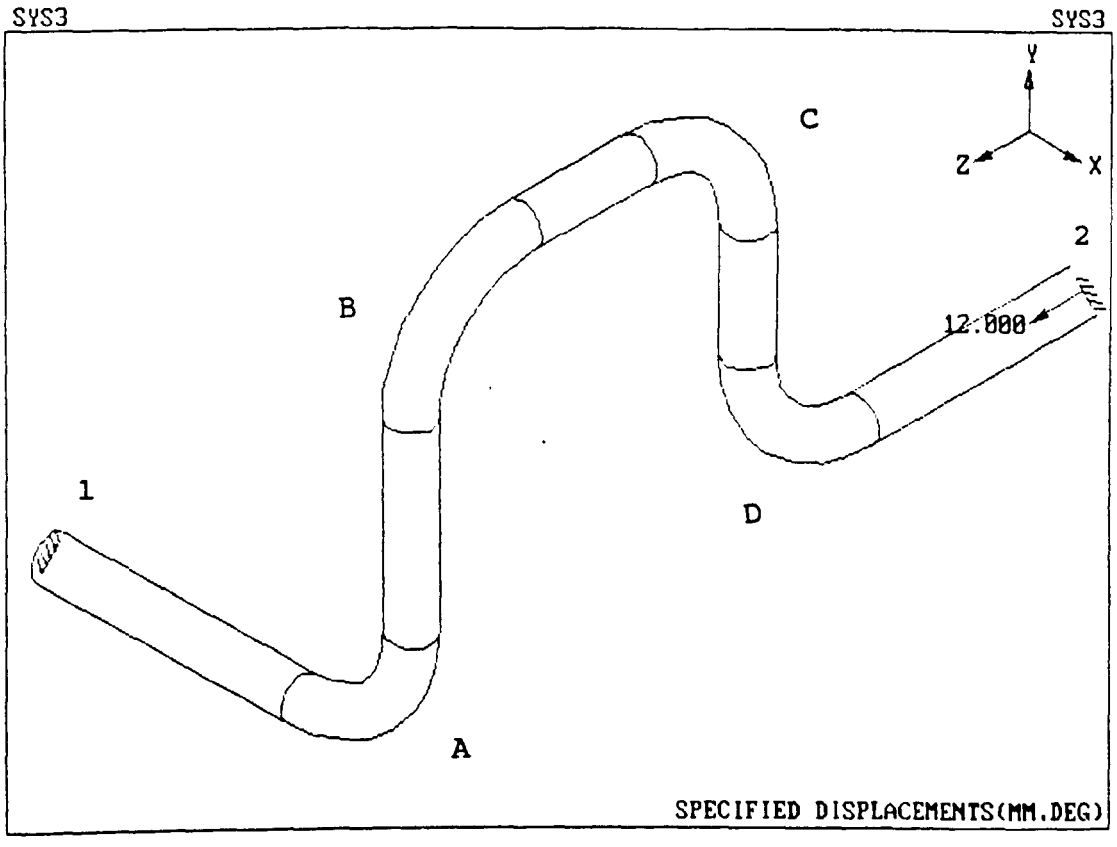


Figure 6.34 Piping system SYS2 bend A and B PB1 longitudinal and circumferential stress distribution.



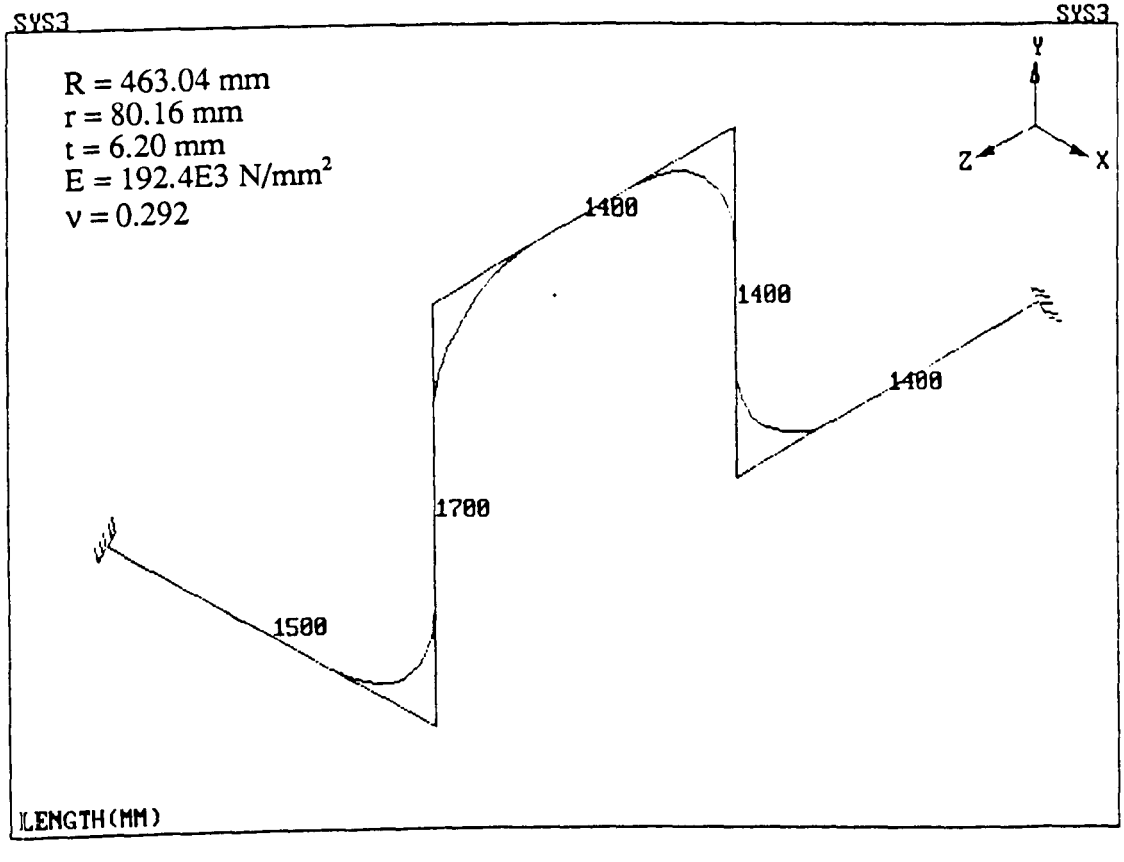


Figure 6.35 Piping system SYS3 : co-ordinate system, geometry, material properties and prescribed displacement.

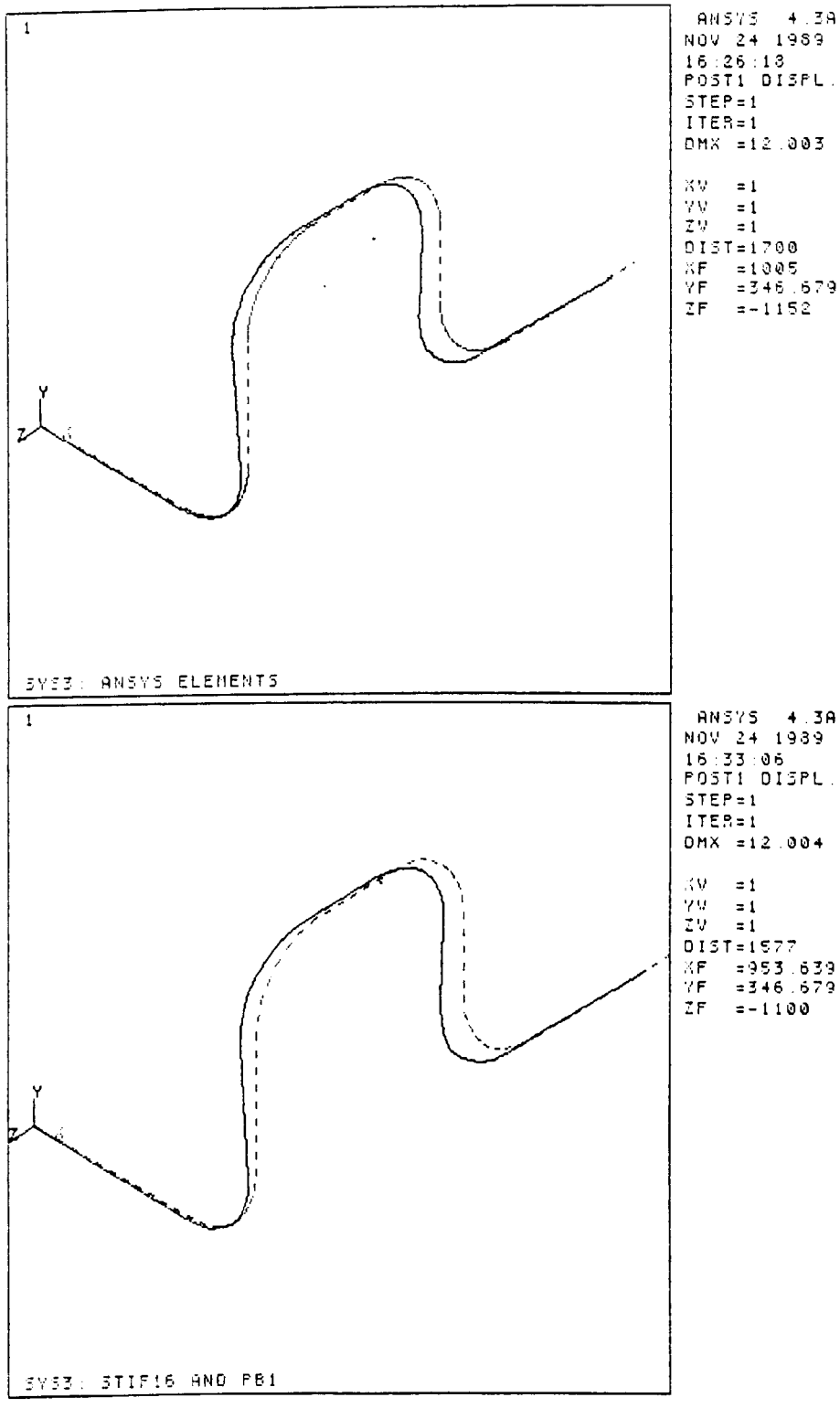
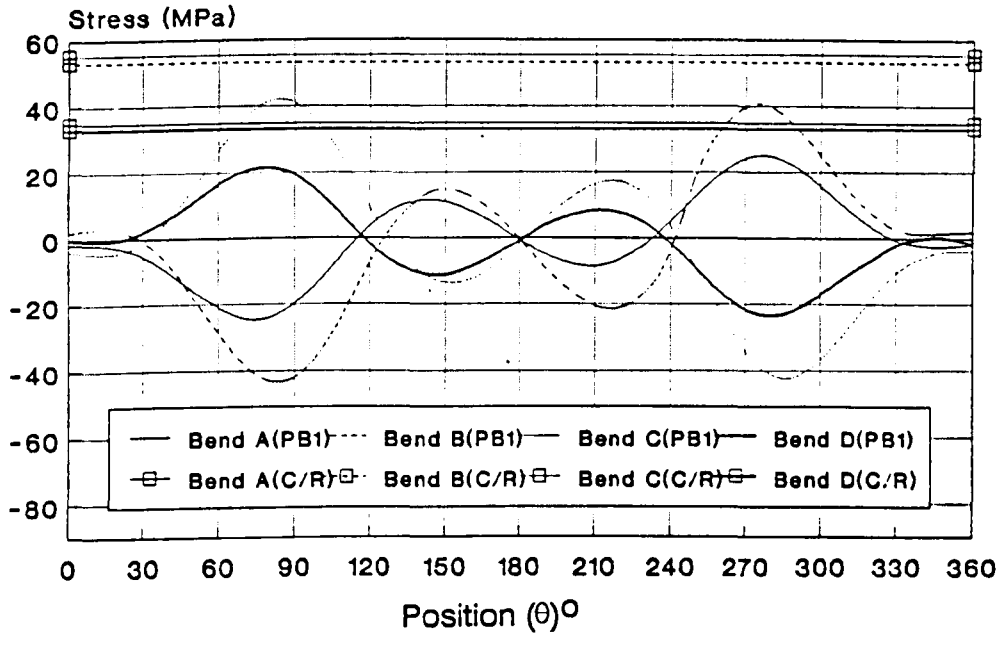


Figure 6.36 Piping system SYS3 deformed geometry plots.



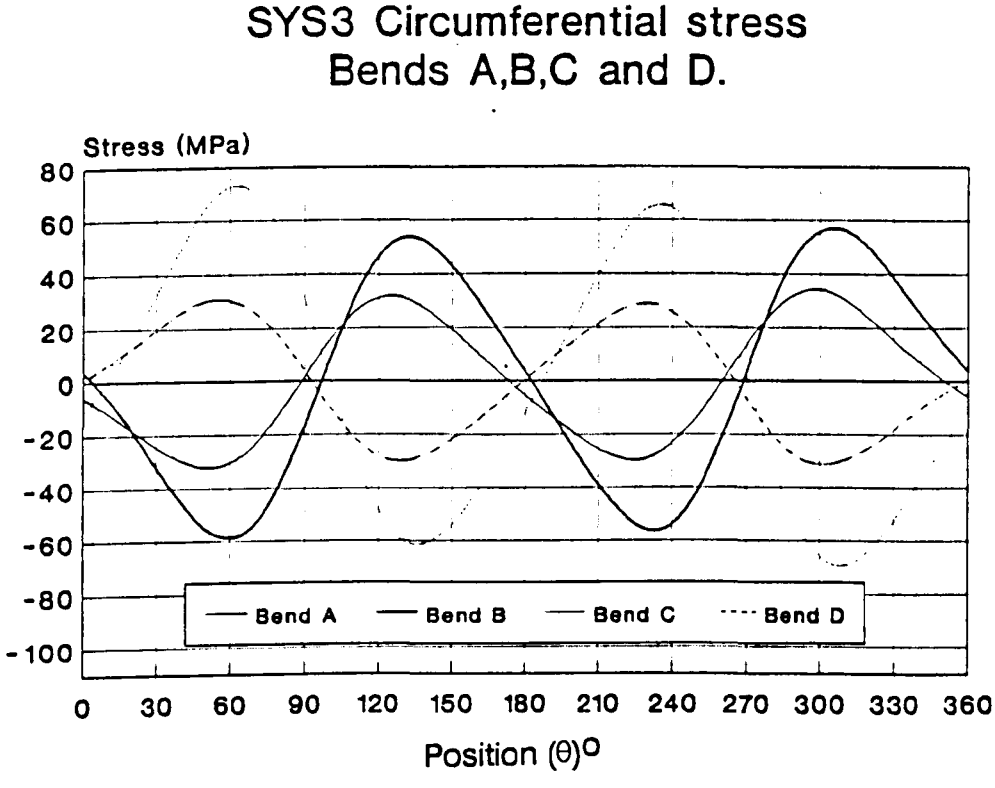


Figure 6.37 Piping system SYS3 bend A to D PB1 longitudinal and circumferential stress distribution.

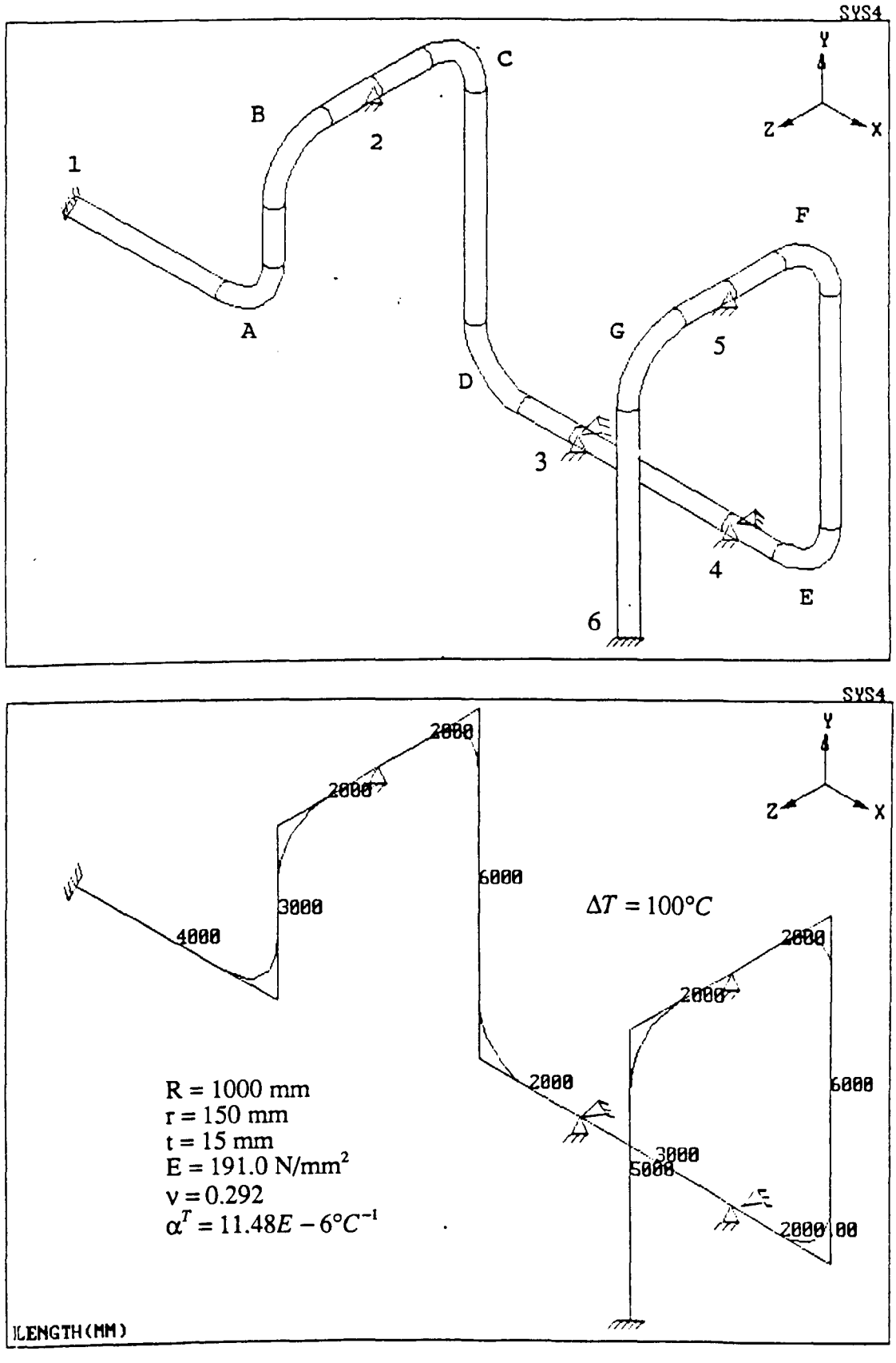
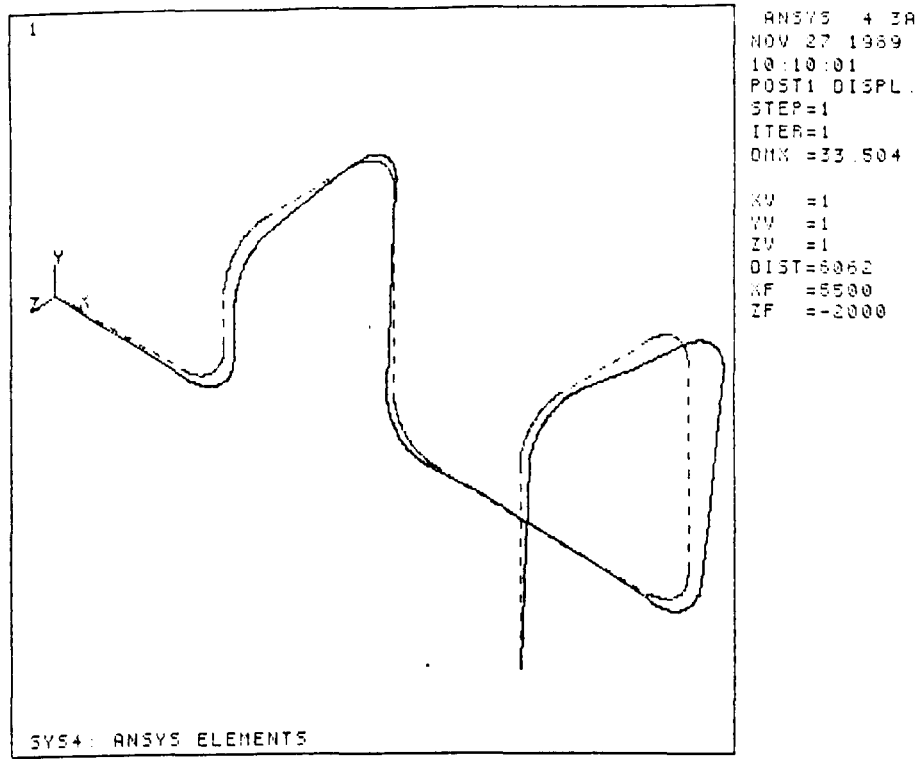


Figure 6.38 Piping system SYS4 : coordinate system, geometry, material properties and applied uniform temperature difference.



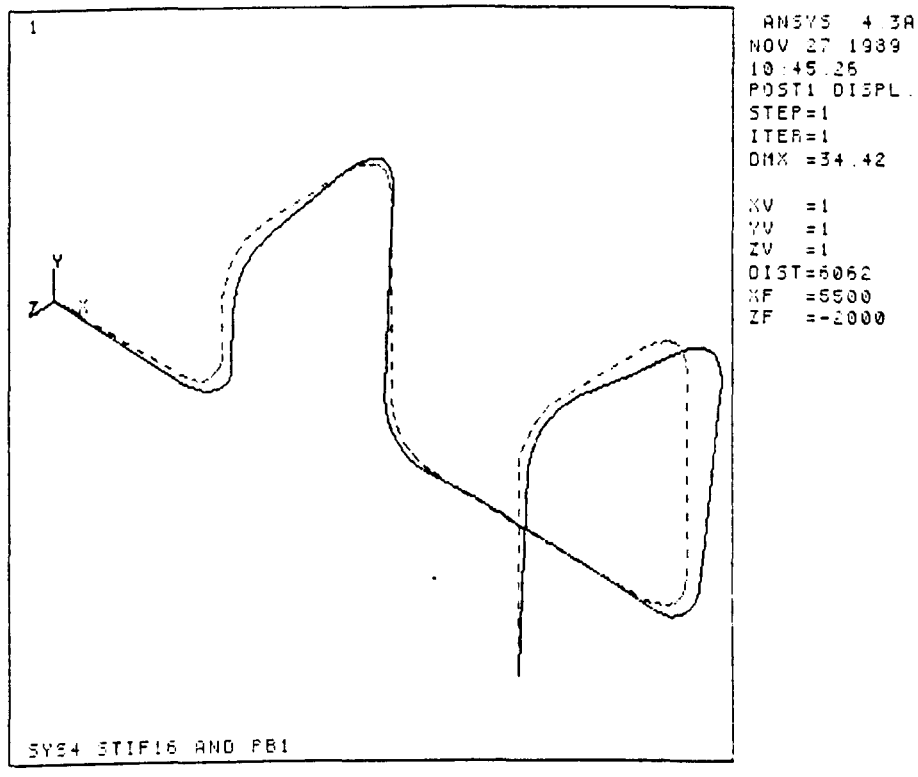
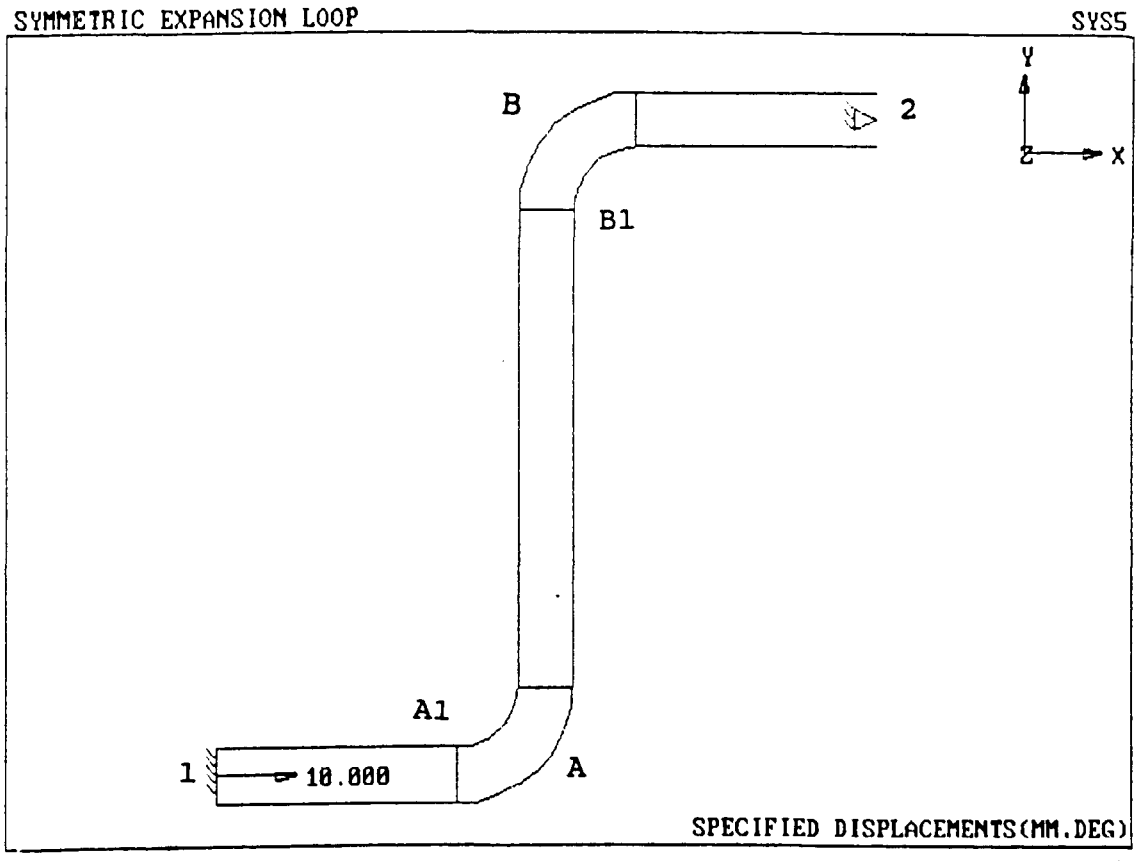


Figure 6.39 Piping system SYS4 deformed geometry plots.



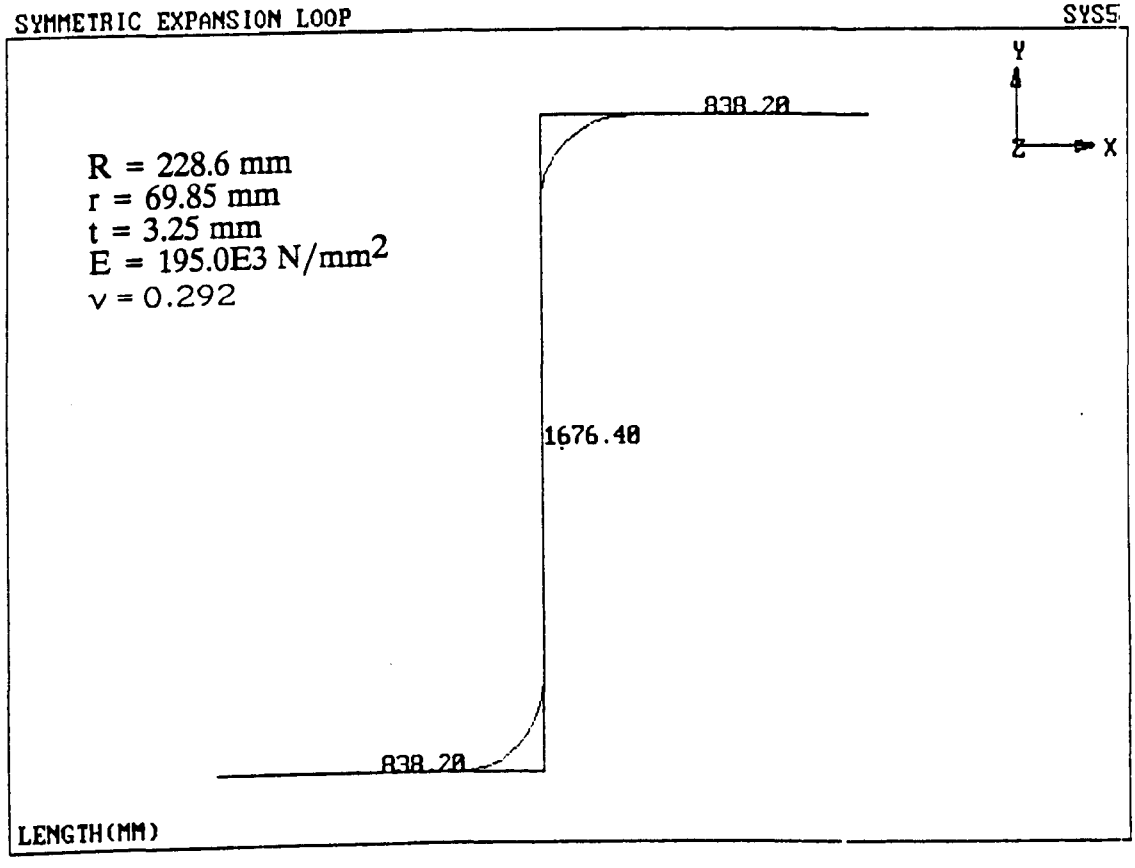


Figure 6.40 Piping system SYS5 : coordinate system, geometry, material properties and prescribed displacement.

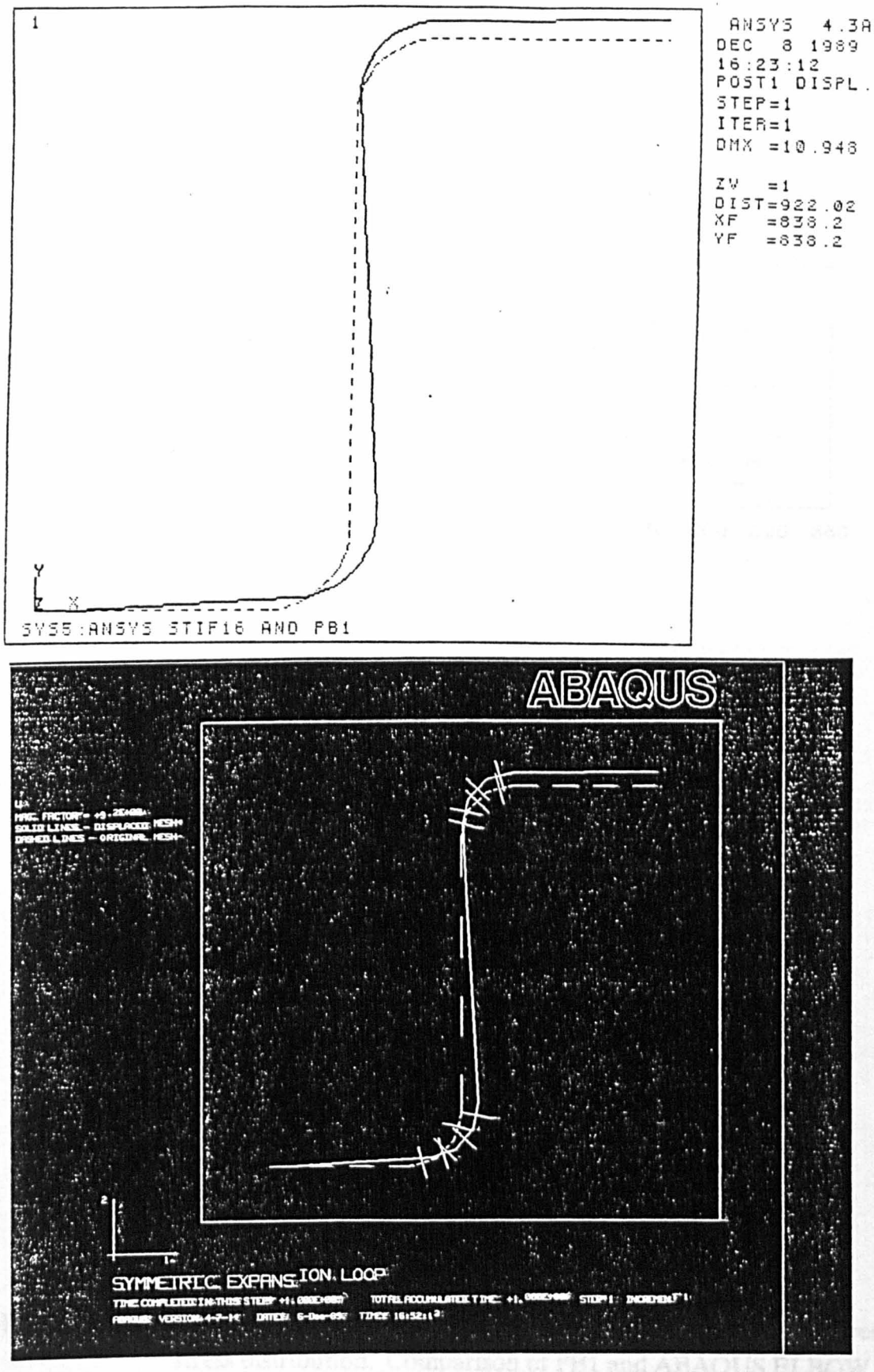
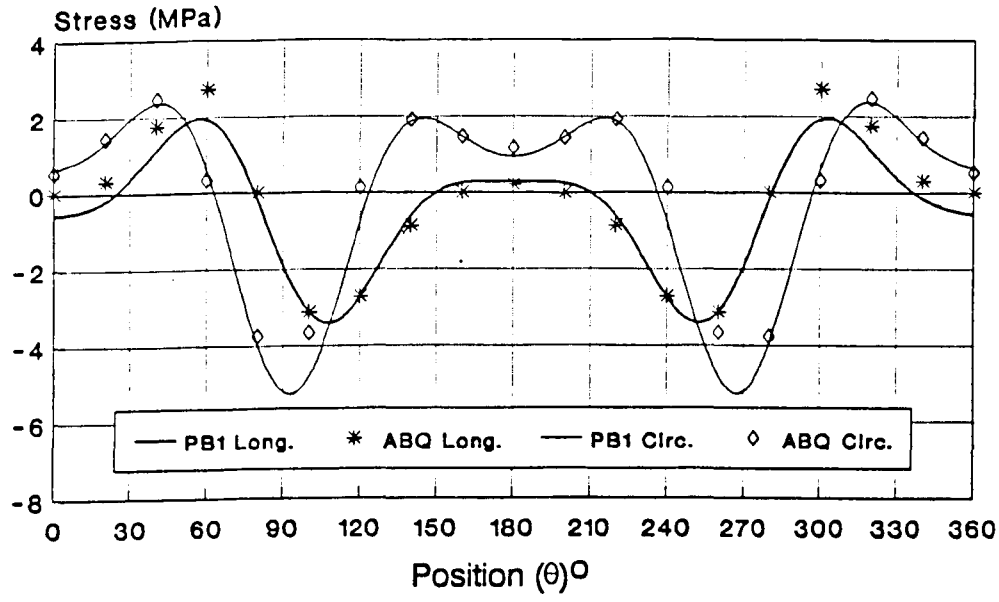


Figure 6.41 Piping system SYS5 deformed geometry plots.



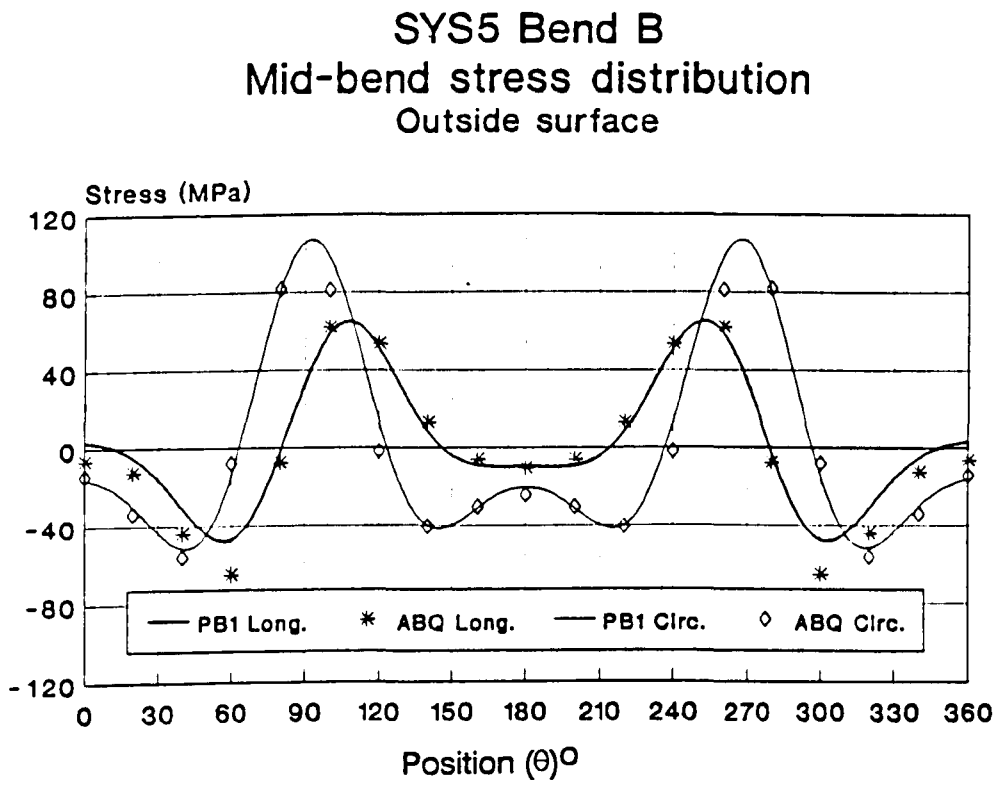
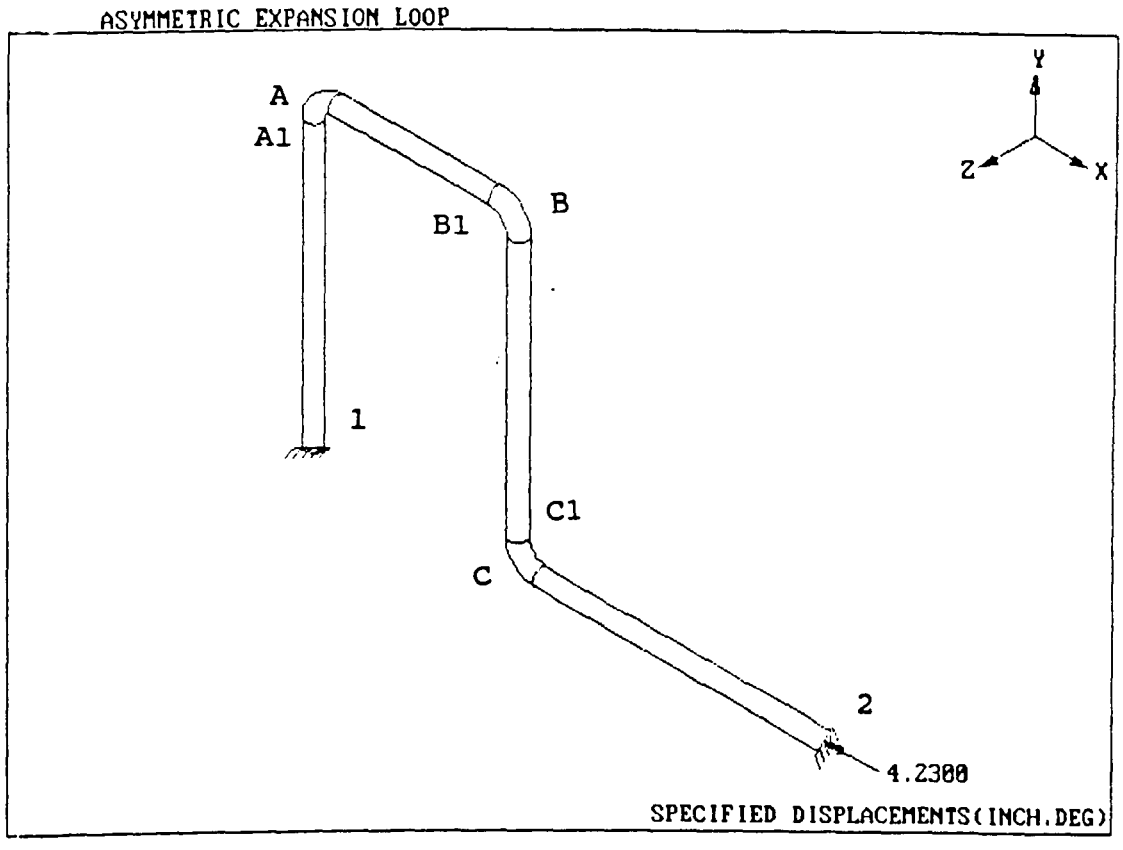


Figure 6.42 Piping system SYS5 bend A and B longitudinal and circumferential stress distribution. Comparison of PB1 and ABAQUS ELBOW 31B (ABQ) results.



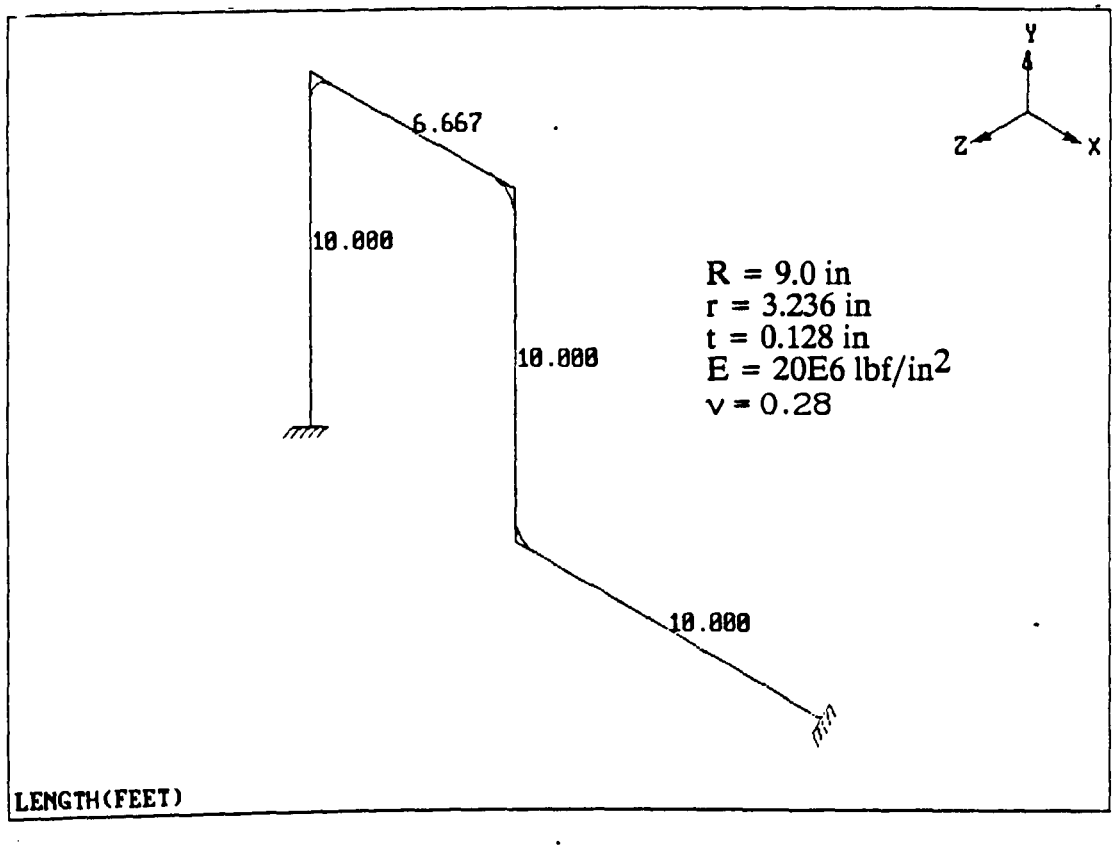
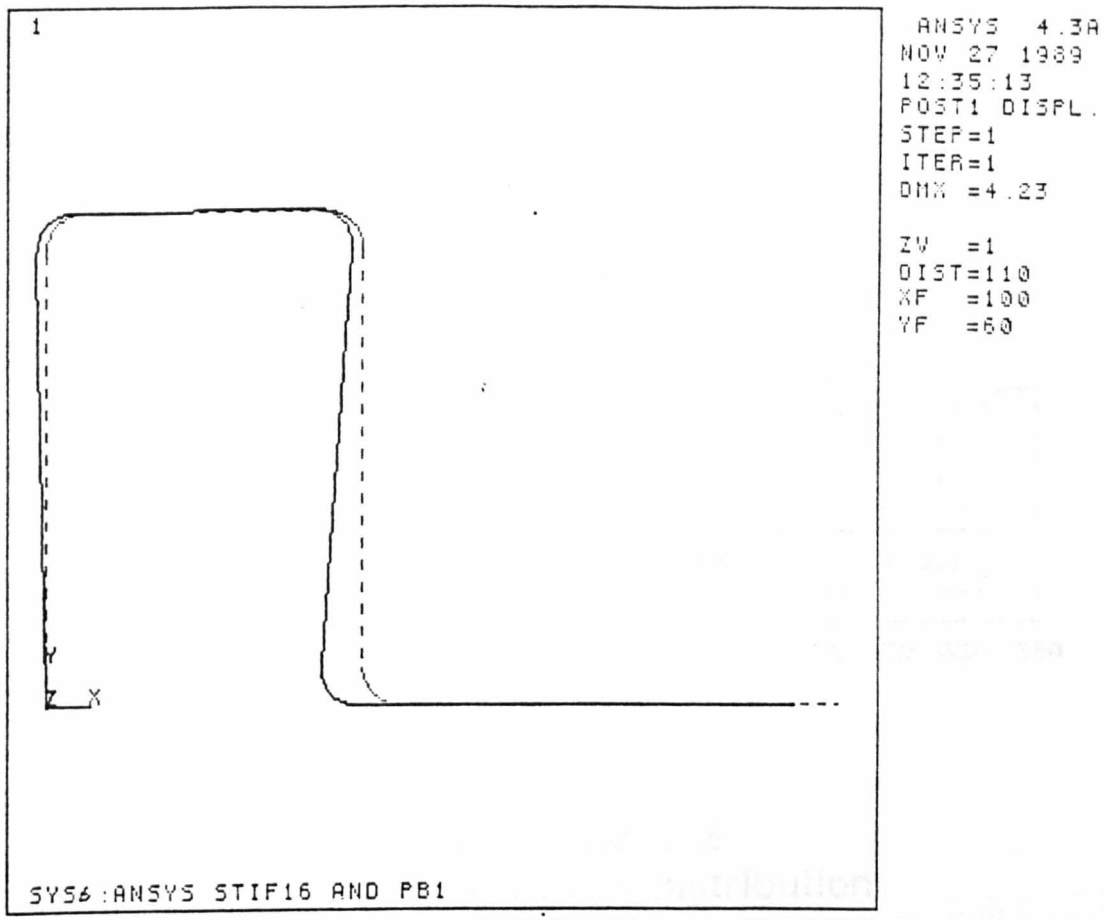


Figure 6.43 Piping system SYS6 : coordinate system, geometry, material properties and prescribed displacement.



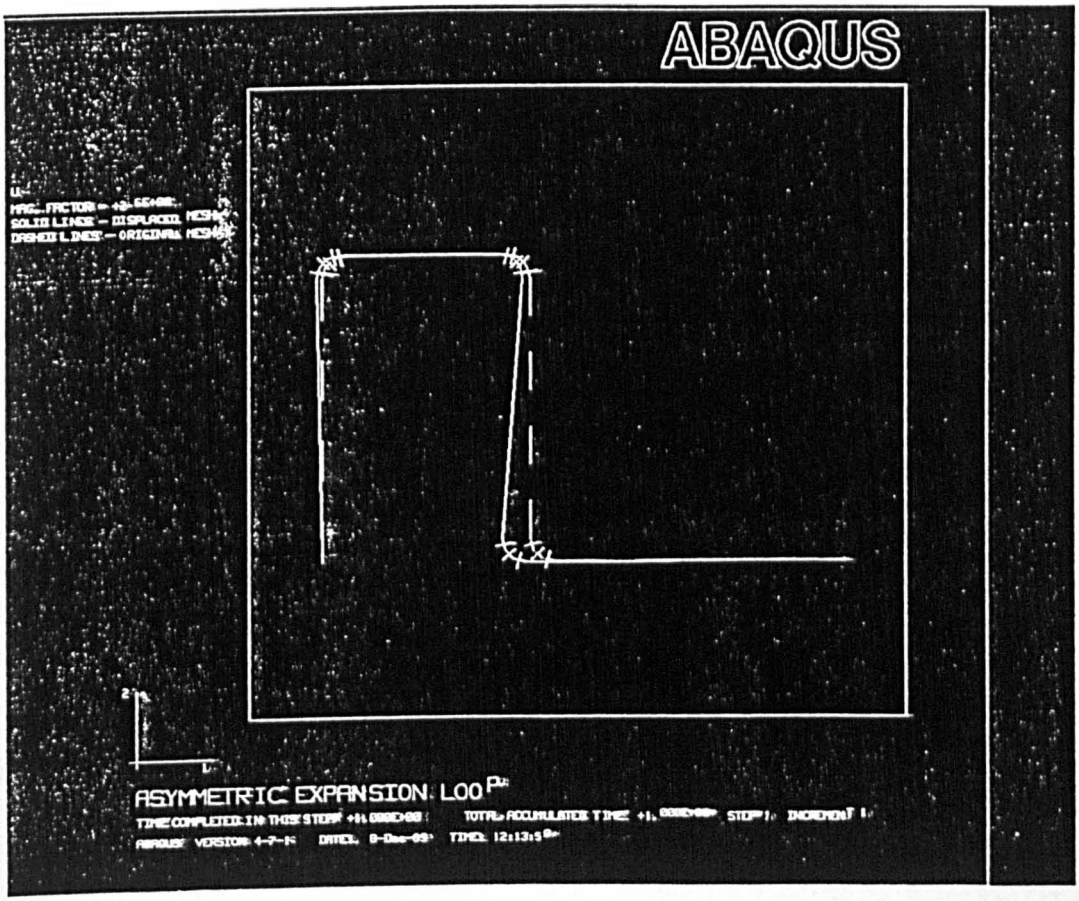
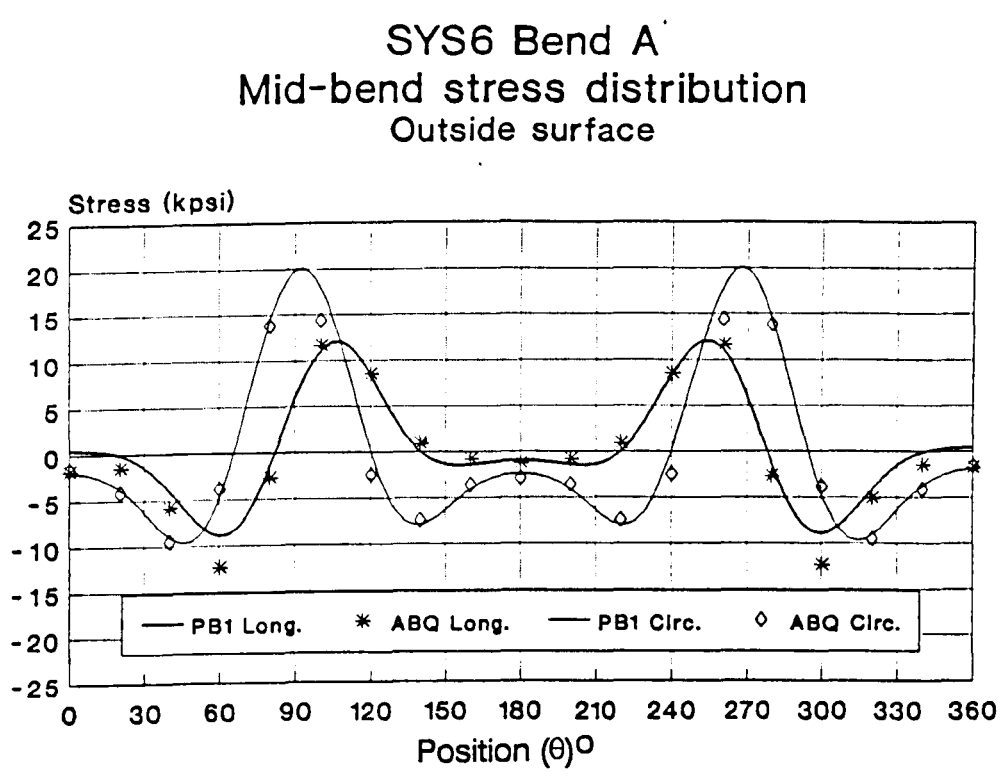


Figure 6.44 Piping system SYS6 deformed geometry plots.



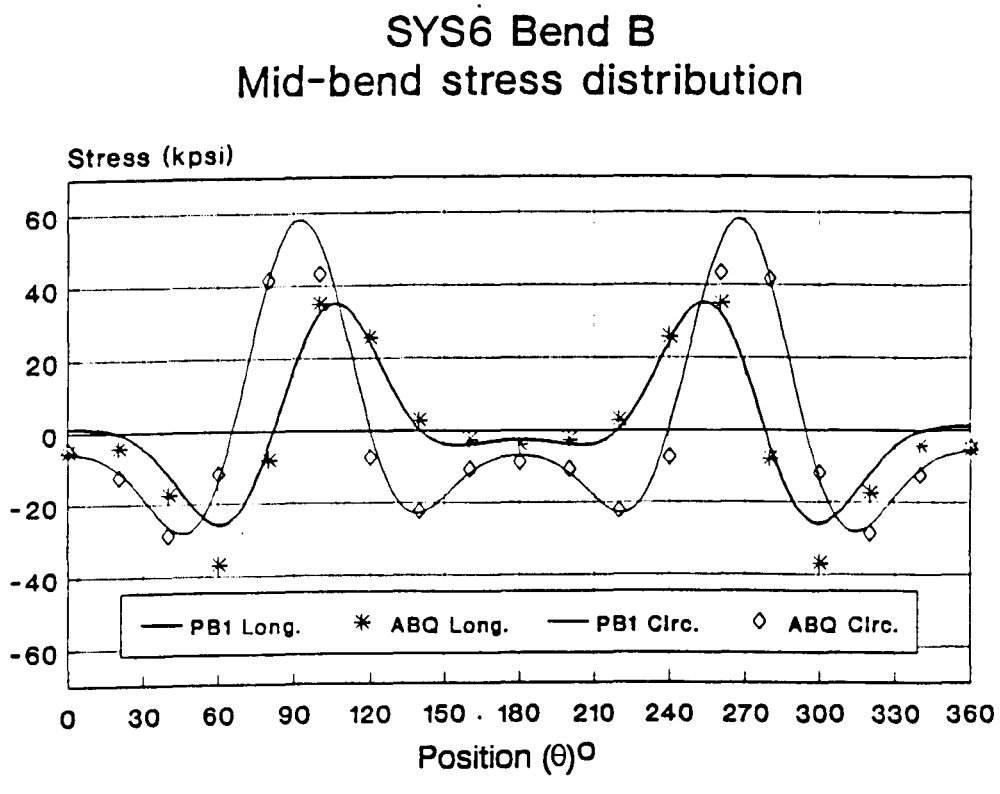


Figure 6.45 Piping system SYS6 bend A and B longitudinal and circumferential stress distribution. Comparison of PB1 and ABAQUS ELBOW 31B (ABQ) results.

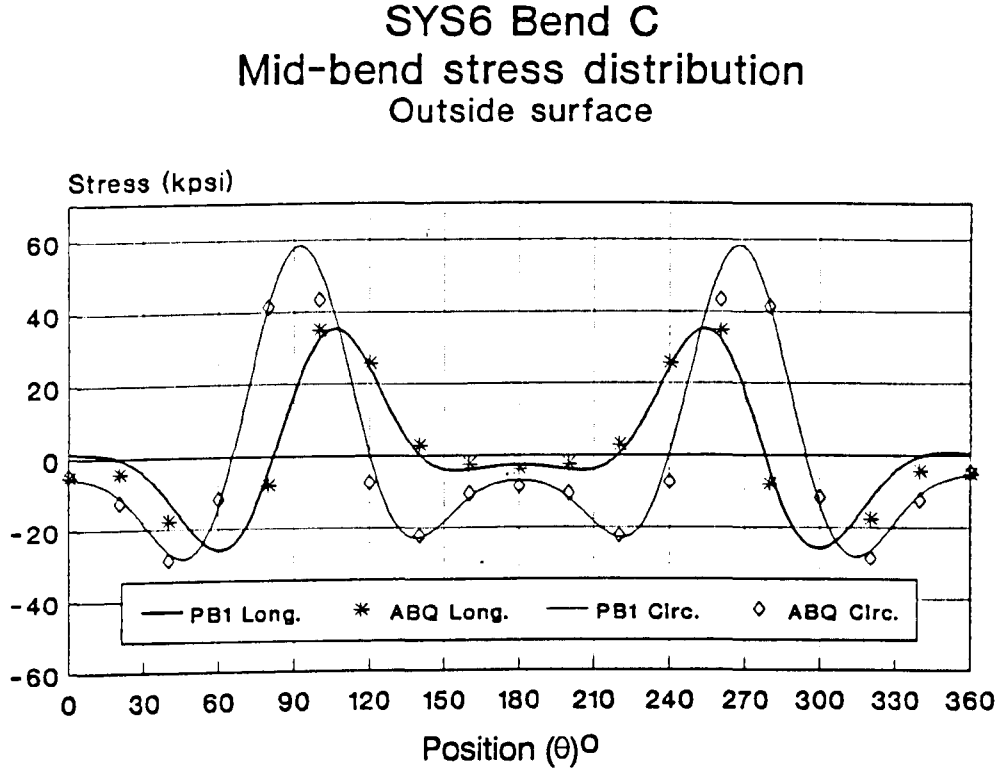
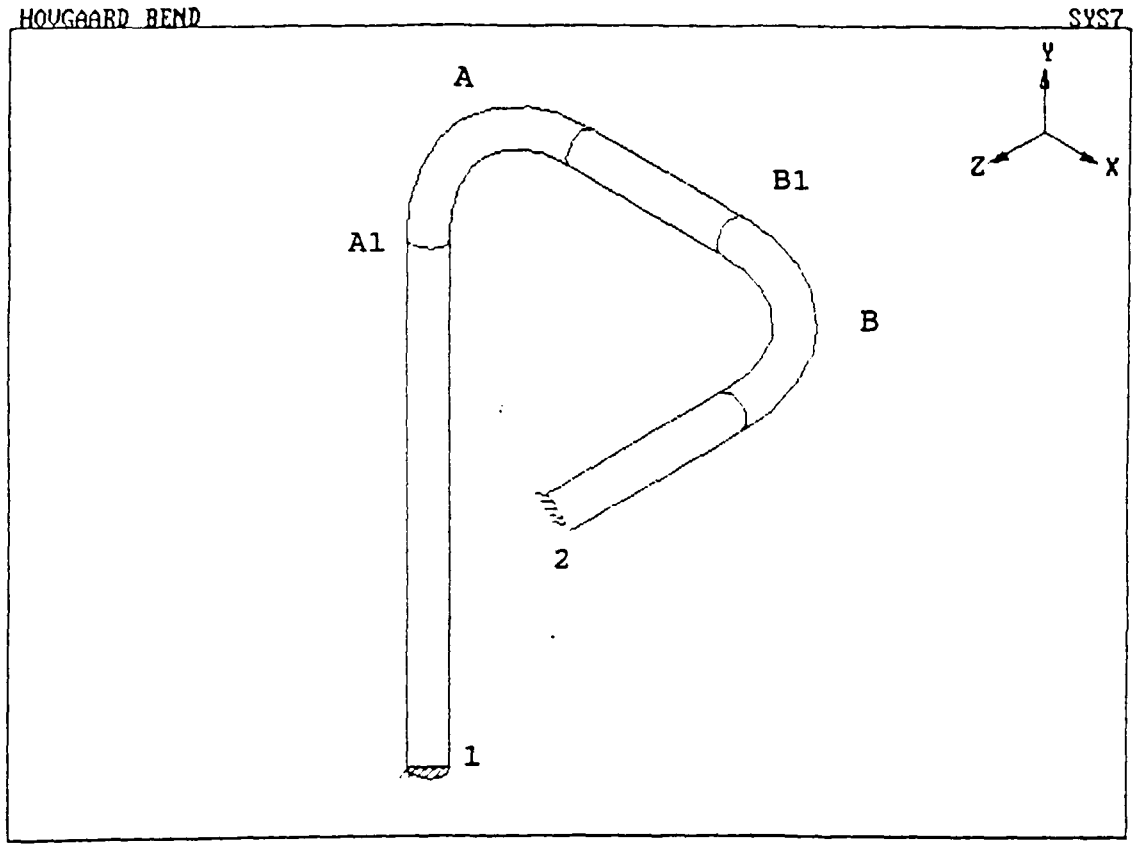


Figure 6.46 Piping system SYS6 bend C longitudinal and circumferential stress distribution. Comparison of PB1 and ABAQUS ELBOW 31B (ABQ) results.



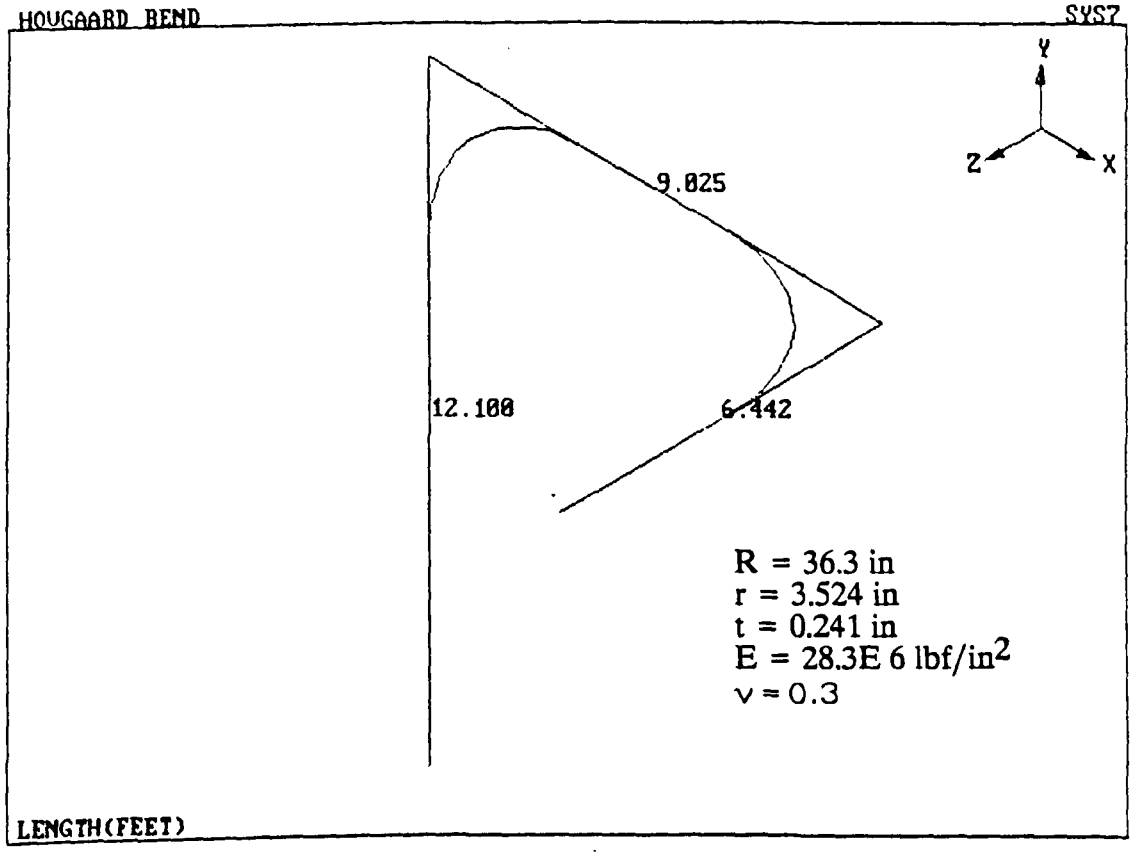


Figure 6.47 Piping system SYS7 : coordinate system, geometry, material properties and prescribed displacement.

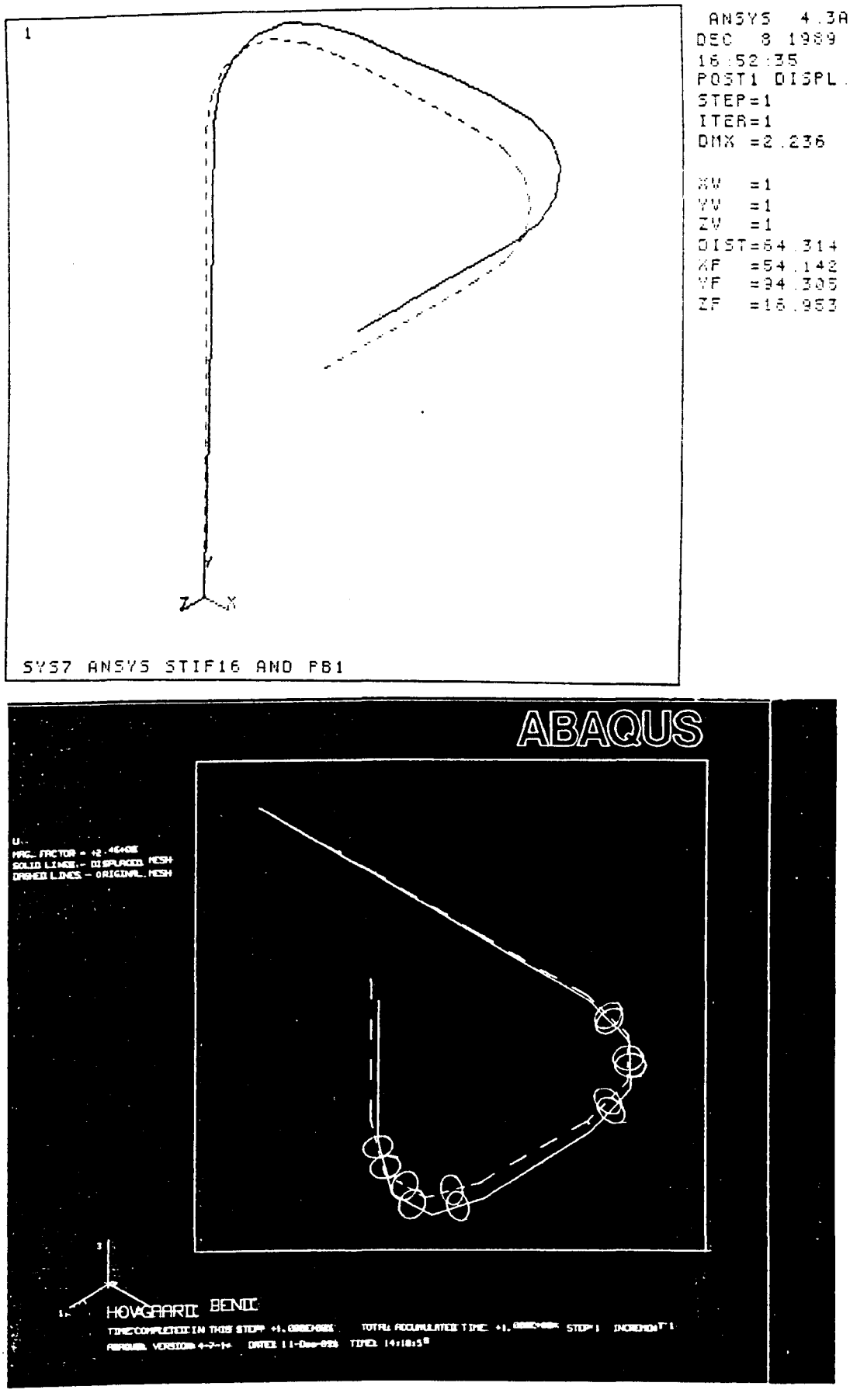
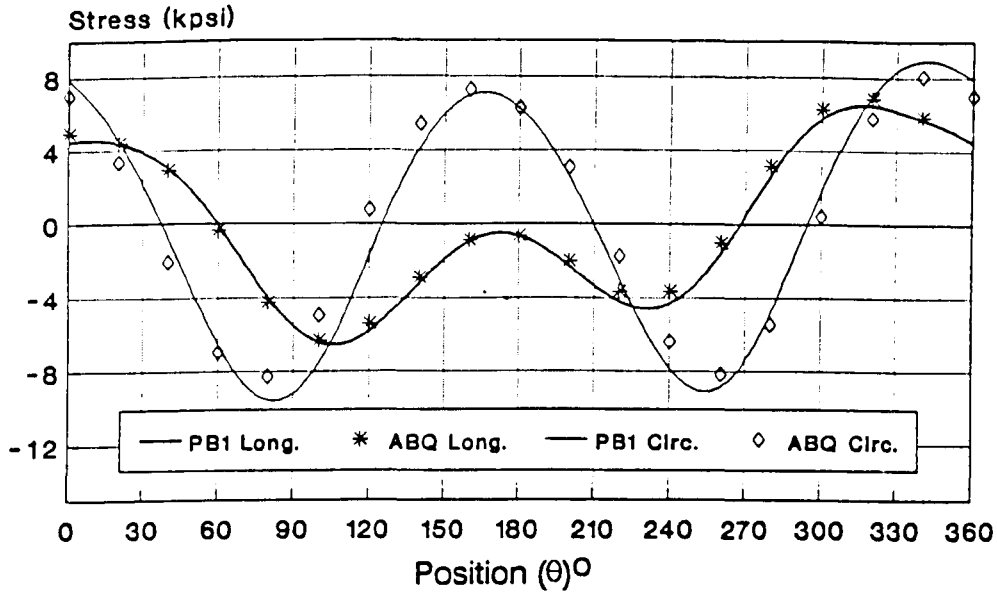


Figure 6.48 Piping system SYS7 deformed geometry plots.



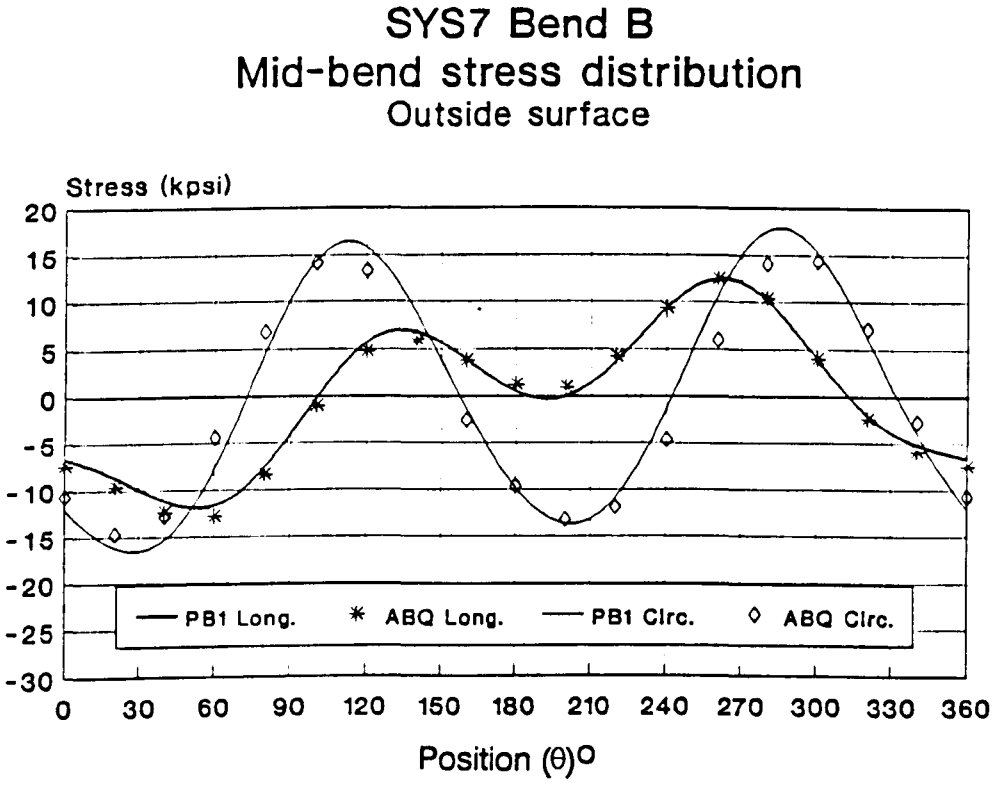
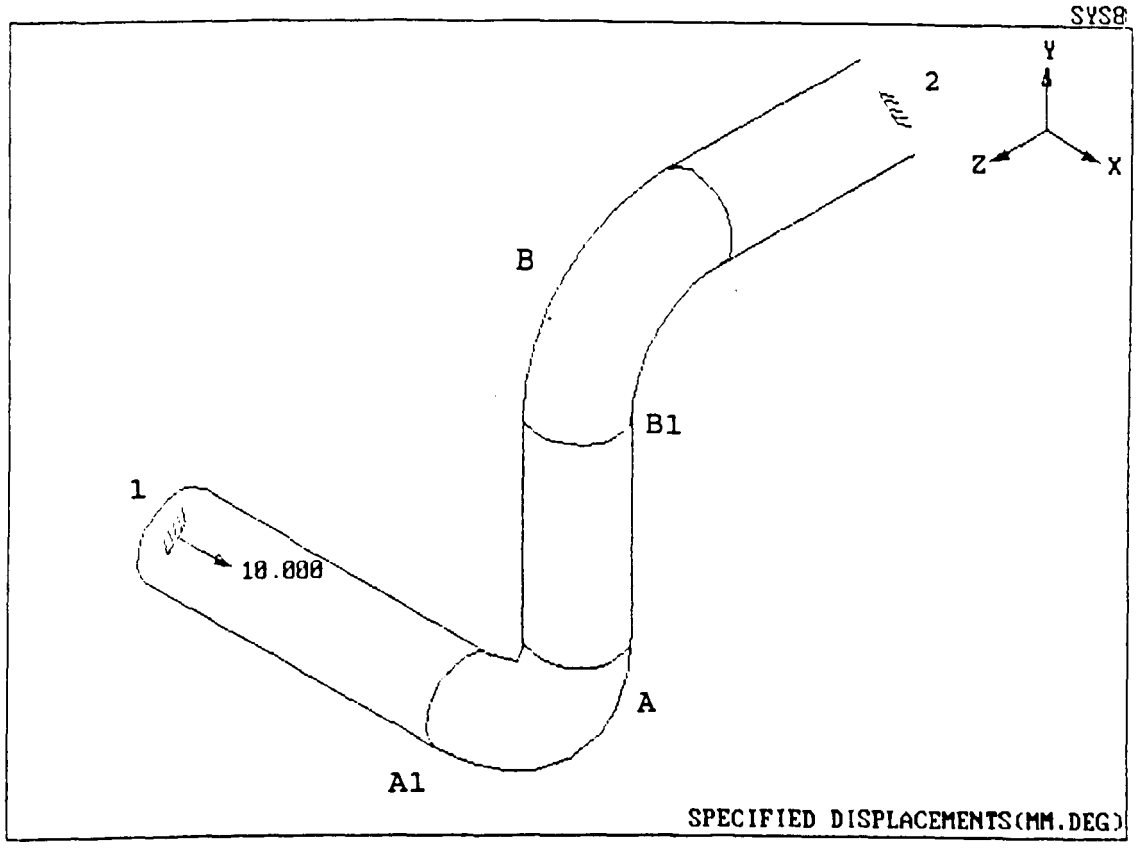


Figure 6.49 Piping system SYS7 bend A and B longitudinal and circumferential stress distribution. Comparison of PB1 and ABAQUS ELBOW 31B (ABQ) results.



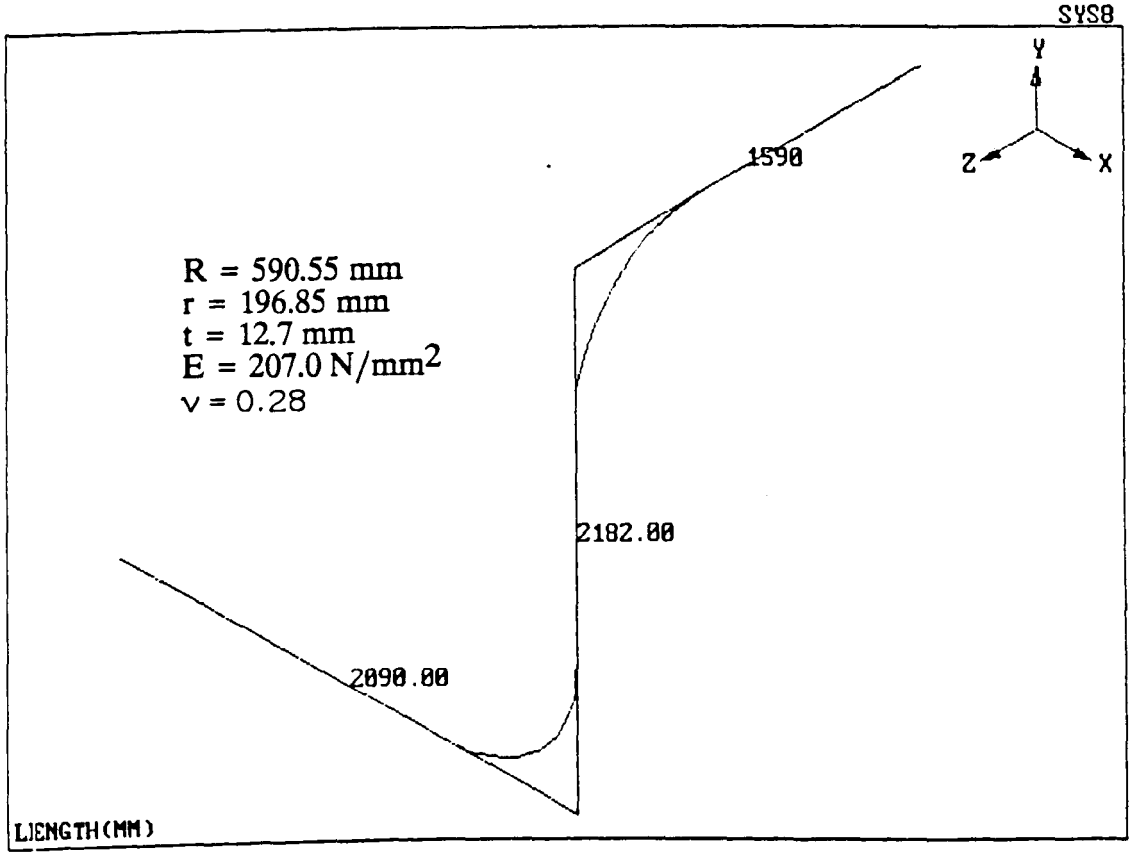


Figure 6.50 Piping system SYS8 : coordinate system geometry, material properties and prescribed displacement.

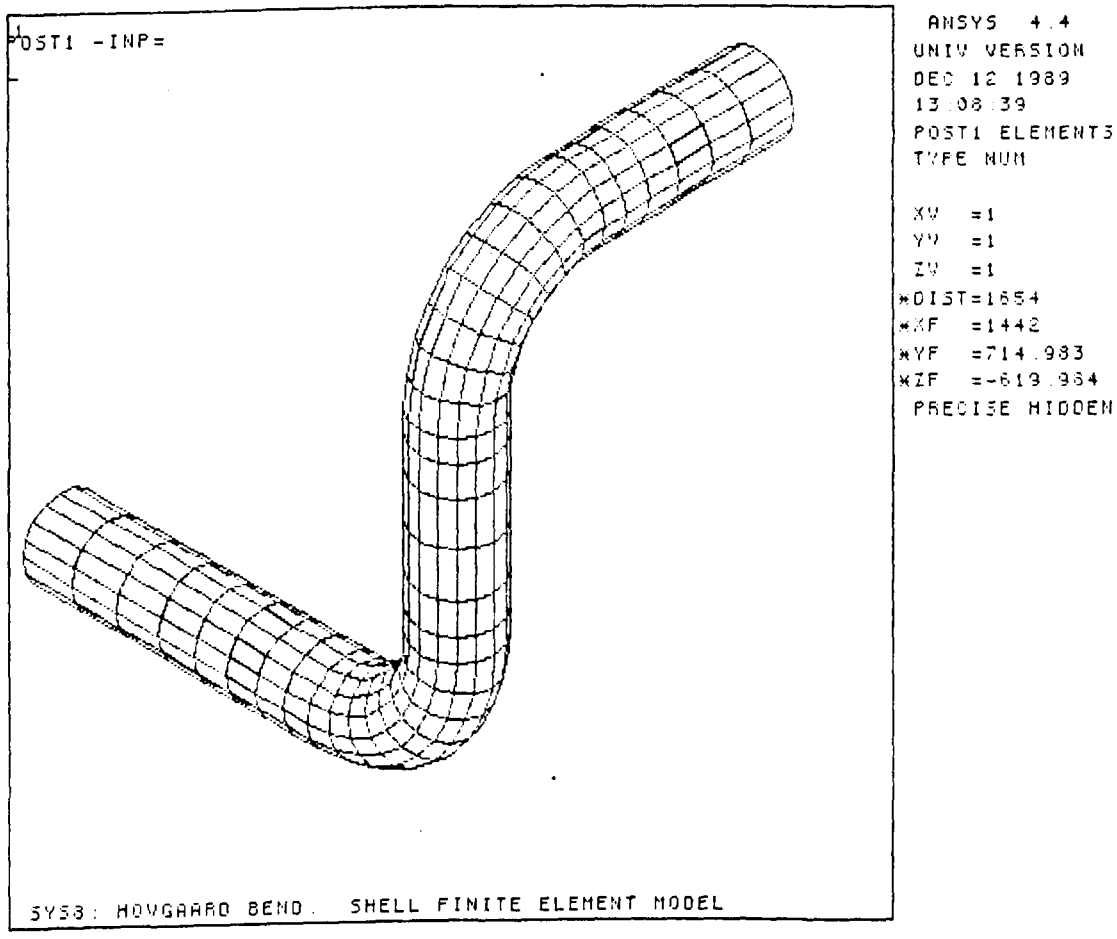
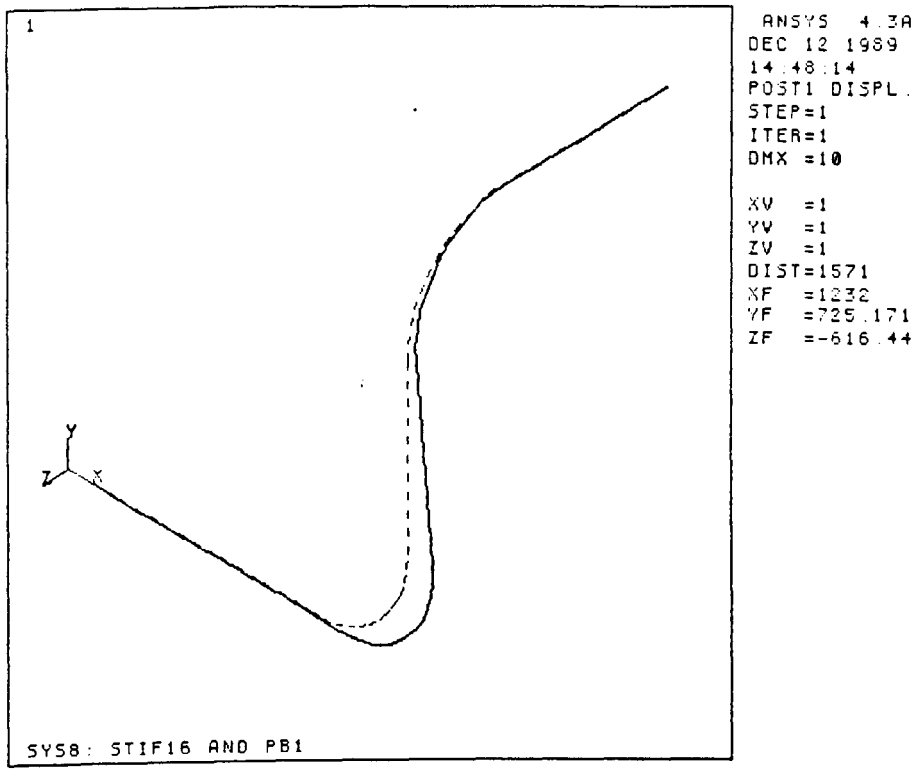


Figure 6.51 Piping system SYS8 ANSYS thin shell finite element model.



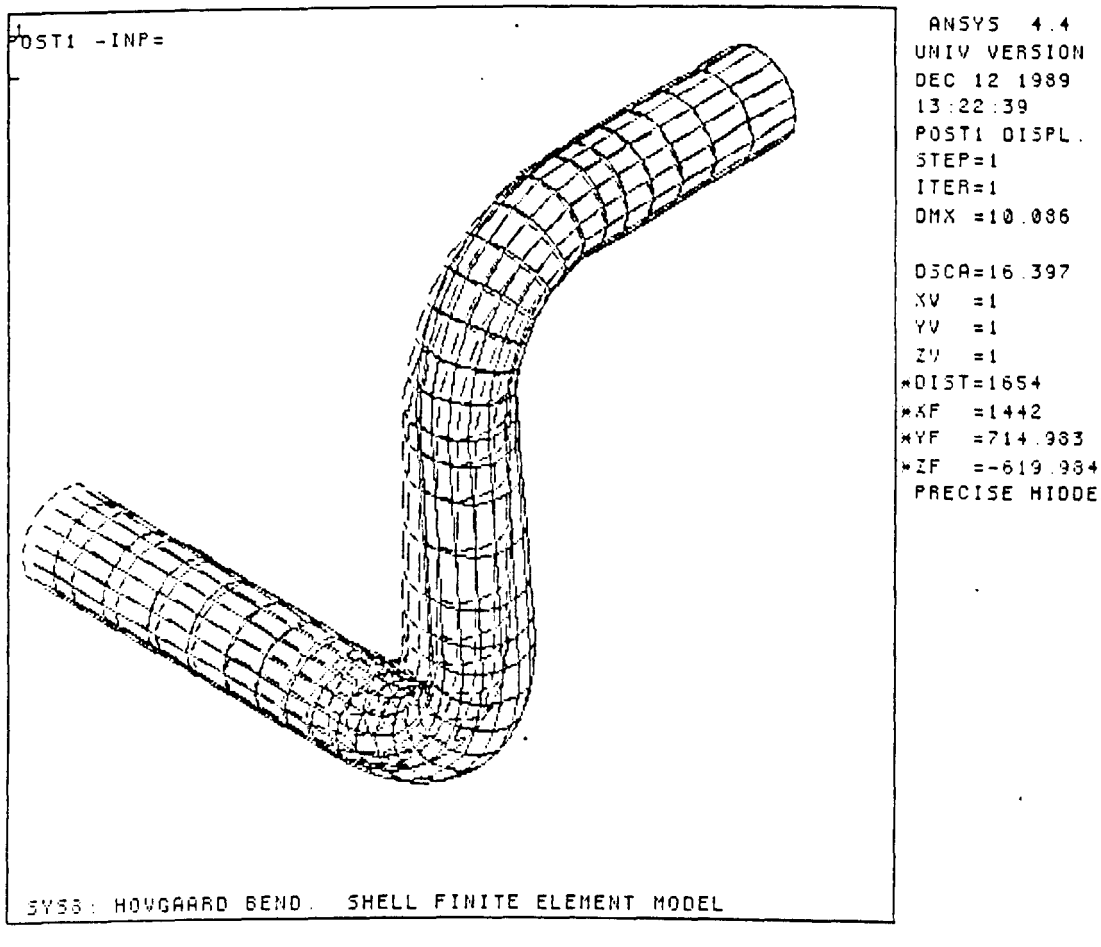
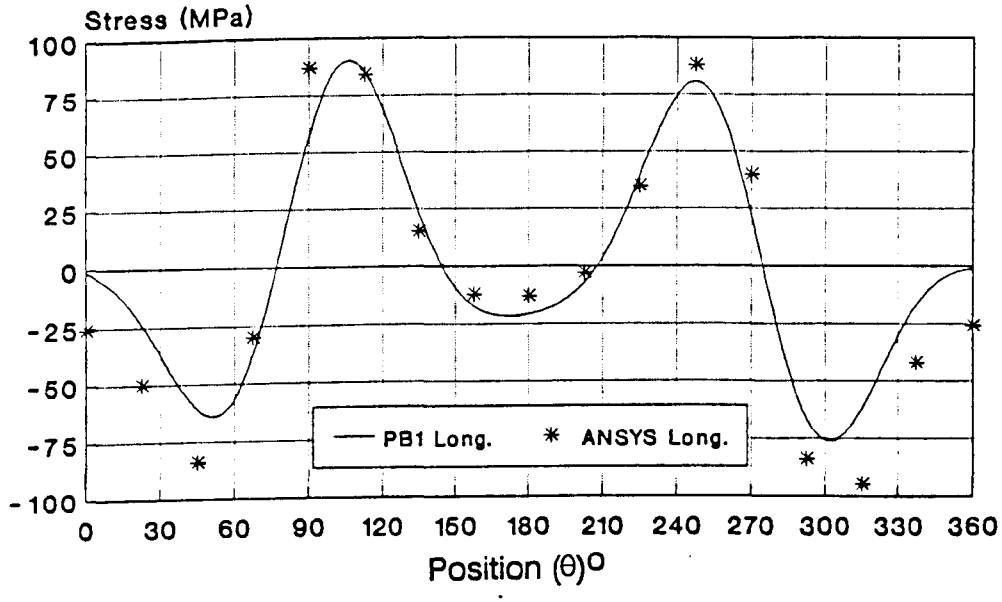


Figure 6.52 Piping system SYS8 deformed geometry plots.



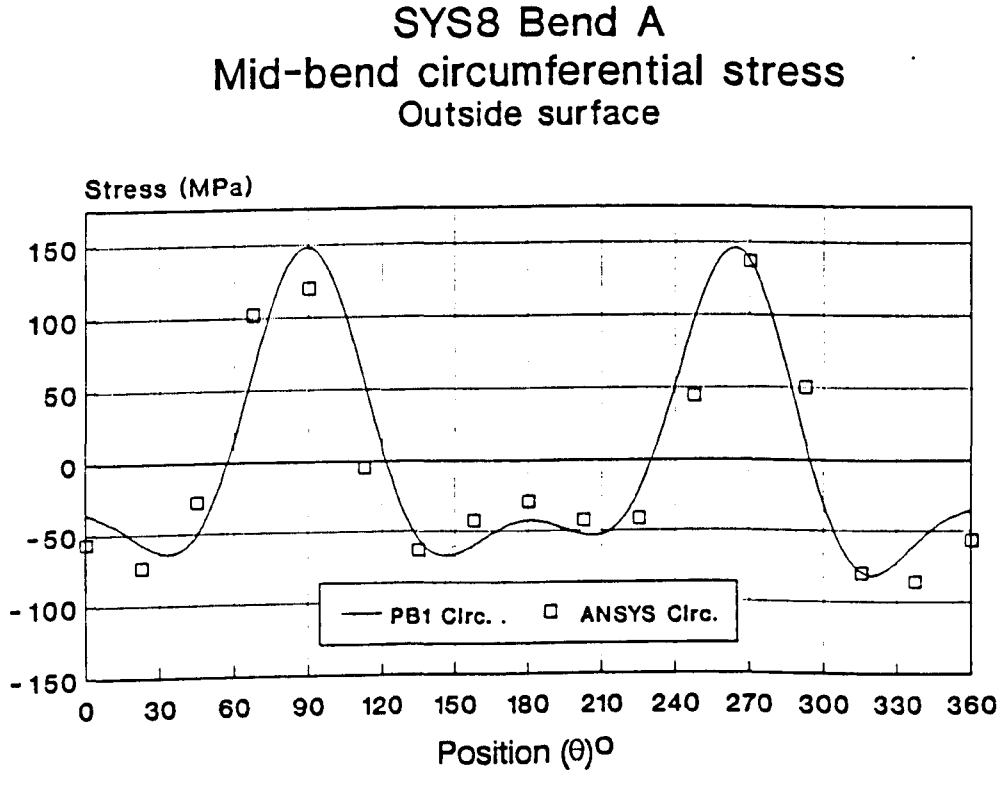
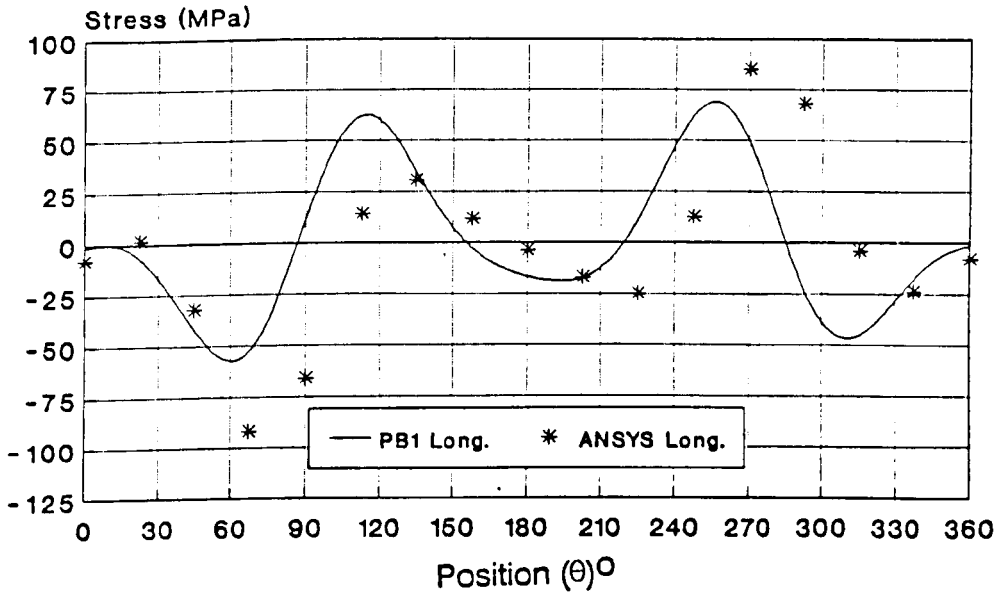


Figure 6.53 Piping system SYS8 bend A longitudinal and circumferential stress distribution. Comparison of PB1 and ANSYS STIF93 thin shell element results.



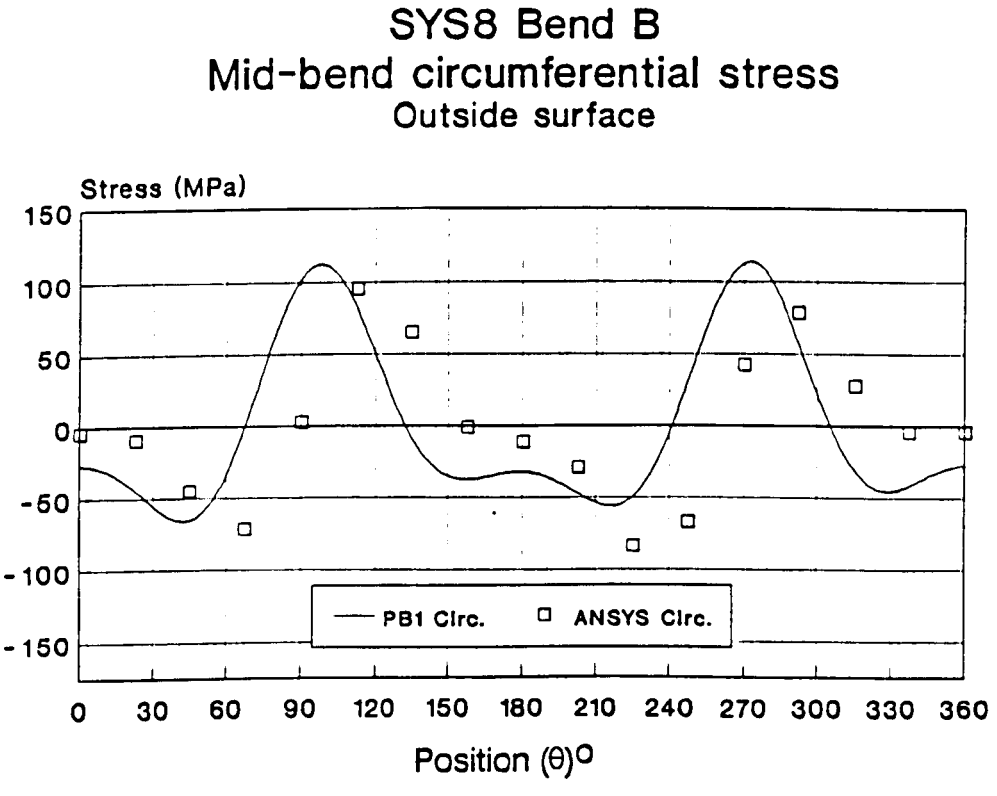


Figure 6.54 Piping system SYS8 bend B longitudinal and circumferential stress distribution. Comparison of PB1 and ANSYS STIF93 thin shell element results.

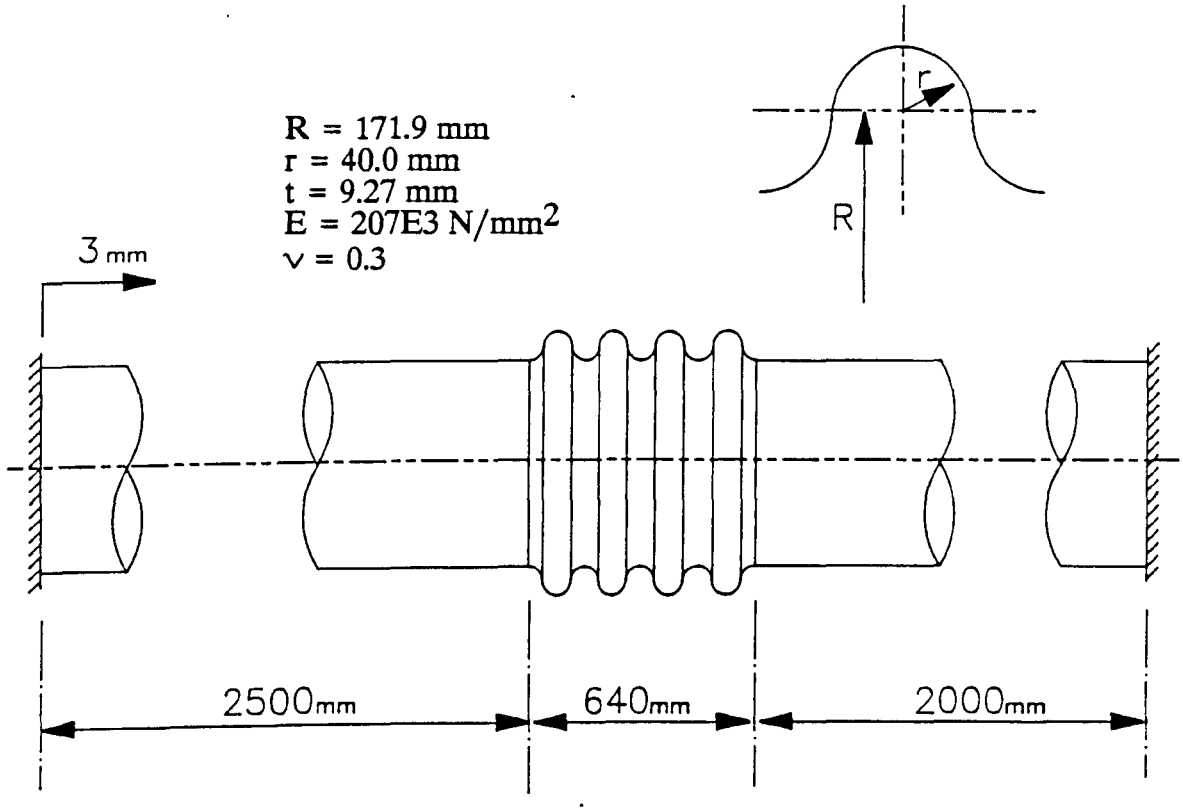


Figure 6.55 Piping system SYS9 geometry, material properties and prescribed displacement.

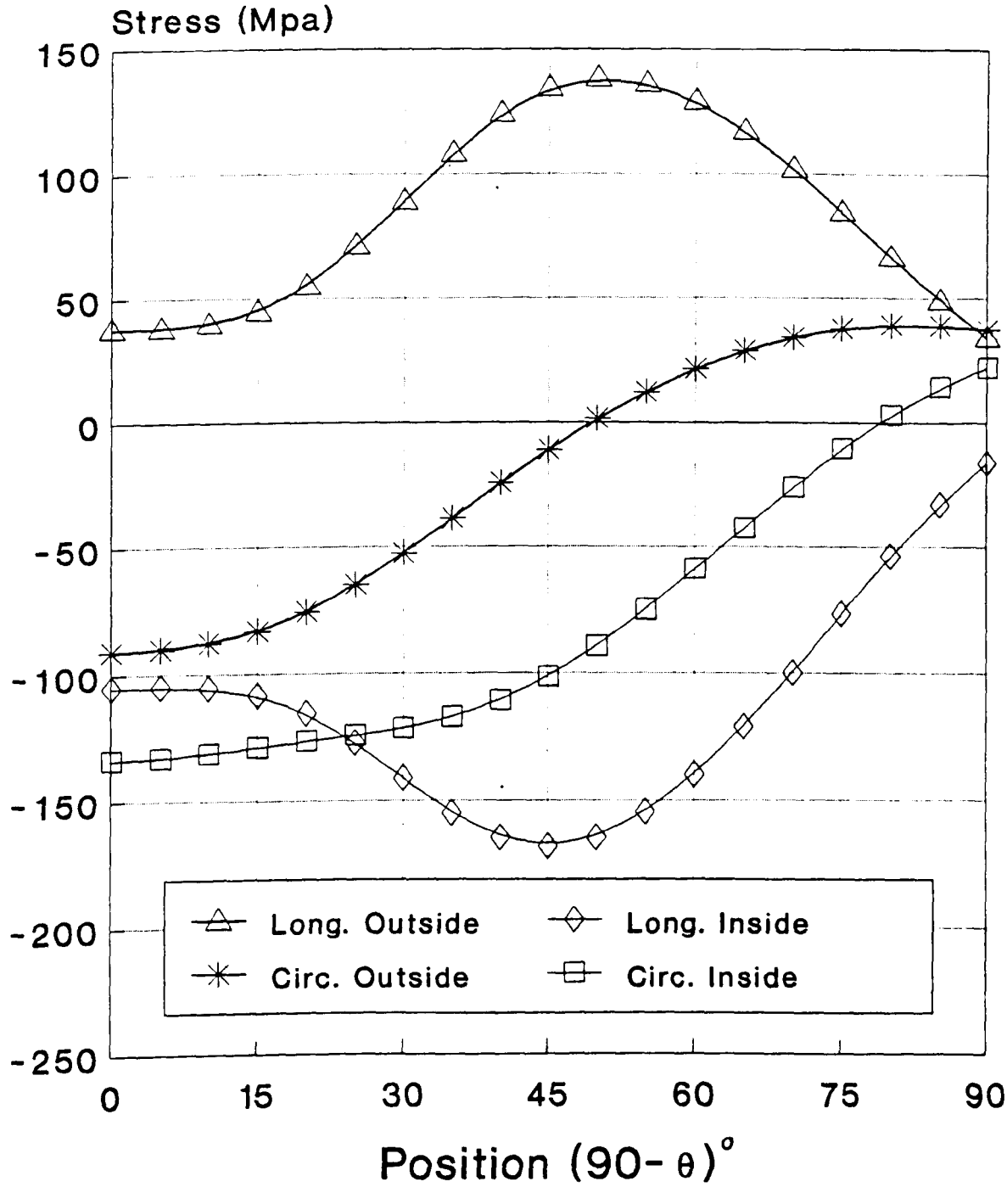


Figure 6.56 Piping system SYS9 BEL1 longitudinal and circumferential stress distribution at bellows inside and outside surfaces.

CHAPTER 7.

CONCLUSIONS.

7 CONCLUSIONS.

Three simple piping elbow elements have been formulated and their suitability for linear elastic analysis of general piping systems investigated. The elements are based on the beam-shell approach, in which a cross-section ovalisation model is superposed on a curved beam element formulation.

The elements share a common beam model, based on an exact solution of Vlasov's thin wall curved beam theory. The ovalisation models were based on two-dimensional shell theory, which precluded representation of end effects and inter-element ovalisation compatibility, but considerably simplified the formulations and reduced computing requirements.

In the first elbow element presented, element PB1, the ovalisation was assumed to be constant with respect to axial position and interpolated around the cross-section by a truncated three even term Fourier series. The second element, PB2, extended the PB1 formulation to include linear interpolation of the constant ovalisation model along the axis of the elbow. In the third element, PB3, the ovalisation was assumed to be constant with respect to axial position, as in the case of PB1. However, PB3 differed from the other elements in that it employed polynomial interpolation of ovalisation. The element cross-section was divided into four 90^{o} arcs (for both in-plane and out-of-plane ovalisation models), over which the ovalisation displacements were interpolated by quintic polynomial functions.

As the above beam and ovalisation models have uncoupled in-plane and out-of-plane responses it was possible to simplify element formulation and subsequently reduce computing calculations by formulating uncoupled in-plane and out-of-plane elbow models, which are assembled at element level to form a fully representative elbow element. The in-plane and out-of-plane models include both beam and ovalisation degrees of freedom. The ovalisation degrees of freedom are statically condensed from the elements to leave only beam degrees of freedom before the complete element is assembled. (Condensed degrees of freedom are recovered for stress calculations in the element stress pass).

The beam stiffness matrix common to all three elements and the ovalisation and bending-ovalisation coupling stiffness matrices of PB1 were integrated analytically, using a symbolic algebra computer program, removing the need for

computationally expensive numerical integration of these matrices. In elements PB2 and PB3 the ovalisation and coupling stiffness matrices are integrated numerically using Gaussian quadrature.

The elbow elements were programmed as user elements for the ANSYS finite element program. This allowed the ANSYS pre-processor to be used to create finite element models of single bends and piping systems interactively, (with straight piping runs modelled by ANSYS STIF16 straight pipe elements where required). The resulting ANSYS finite element models were solved using the ANSYS solution routines, and the ANSYS post-processor used to display the results of the analysis in graphical and numerical form. An extensive "User Element Programming Manual", detailing ANSYS user element programming requirements for linear elastic structural elements, was written specifically to accompany this thesis (Appendix 1).

By basing the elements on a common beam model, it was possible to make direct comparison between the three ovalisation models used and to determine which was the best option for a simple elbow element. Comparison between PB1 and PB3 allowed the relative performance of the Fourier and Polynomial schemes to be assessed. Comparison between PB1 and PB2 allowed the convergence characteristics of axially constant and linearly varying ovalisation to be investigated.

The convergence requirements of the elbow elements were studied for a range of bend geometries. It was found that the linear element PB2 converged most rapidly. PB1 and PB3 showed similar convergence rates, however, element PB3 required many more degrees of freedom (at uncondensed element level) than either PB1 or PB2. In all three elements, out-of-plane loading required many more elements for convergence than in-plane loading.

Although PB2 converged on average one element more quickly than PB1, no significant computational advantage was found in using it in preference to PB1 because of its greater number of degrees of freedom at uncondensed element level.

In order to assess the accuracy, applicability and effectiveness of the three elbow elements, several sample analyses of 90^{o} bends under in-plane and out-of-plane bending were presented and results compared with published and flexibility analysis results.

PB1 and PB2 were found to give almost identical stress and displacement results for a converged solution. The element stress distributions compared well with results presented in the literature, but the displacement results indicated that the elements were slightly stiffer than the ANSYS flexibility analysis elbow element. It was proposed that the higher stiffness was due, in part at least, to the use of an exact beam solution in the element formulation.

Element PB3 gave relatively poor results for both stress and displacement. In particular, it was found that an accelerated degradation of performance occurred for out-of-plane bending of elbows of parameter λ less than 0.4. It was noted that it would be possible to improve the performance of PB3 by using more polynomials to interpolate around the cross-section of the elbow, however, this refinement would increase the computing cost of the element which is already greater than PB1 and PB2.

From the single bend sample analyses results, it was concluded that the piecewise quintic polynomial interpolation scheme used in PB3 was less effective in modelling ovalisation behaviour than the Fourier interpolation schemes used in PB1 and PB2.

As PB1 and PB2 give virtually identical results for a converged solution, the choice of which was the better simple elbow element was made by considering the computing requirements of the elements. For a converged solution, there was little difference in bend model size between the elements in terms of degrees of freedom required at uncondensed element level: the linear element offered no significant advantage over the simpler constant ovalisation element. However, as all the element stiffness matrix integrals of PB1 had been obtained analytically it was more economical than PB2, which required numerical integration of the ovalisation and coupling stiffness matrices. It was therefore concluded that element PB1 was the better option for general analysis of piping systems.

In order to investigate the performance of element PB1 in the analysis of piping systems, a number of sample analyses of branchless systems of varying complexity were performed. Results obtained using elbow element PB1 were compared with ANSYS flexibility analysis, ABAQUS piping finite element analysis and ANSYS thin shell finite element analysis results. From these analyses it was observed that:

i) Flexibility analysis based on Clark and Reissner stress intensification factors is not consistently conservative in comparison with finite element analysis using element PB1.

ii) Stress results given by element PB1 compare well with results given by the ABAQUS program using the ABAQUS elbow element Elbow 31B. However, PB1 gives a slightly stiffer solution than ABAQUS, resulting in reaction forces typically of the order of 3% to 4% higher.

iii) In an analysis of a Hovgaard bend, PB1 stresses were found to compare well with thin shell finite element analysis results. This was especially true for the system bend which was primarily loaded in-plane. The second bend in the system was primarily loaded out-of-plane and in this case, although the stress magnitudes given by the analyses were similar, the stress distribution curve of PB1 appeared to "lag" the thin shell element analysis curve by approximately 15^{o}.

iv) The computing costs of element PB1 are greater than those incurred in flexibility analysis but not prohibitively so.

In conclusion, it may be said that finite element analysis of piping systems using element PB1 gives stress results comparable with more complex and expensive analyses at acceptable computing costs.

7.1 Recommendations for Future Work.

This thesis detailed a study on simple piping elbow elements for elastic analysis of piping systems. From this study, a number of areas requiring further work can be identified.

1) The elements developed in this thesis have been shown to give stiffer solutions than alternative analyses. It has been proposed that this is due, at least in part, to the exact beam solution used in the element formulation. A consequence of this high stiffness is that system reaction forces evaluated by the element are high. An improved simple element could be formulated by superposing the ovalisation model of PB1 on an alternative beam model. It would be of interest to examine other beam formulations in order to determine the most suitable for use in an elbow element.

2) The results of several sample analyses, in which results given by the elbow elements presented in this thesis were compared with alternative solutions, were presented in Chapter 6. However there is, in fact, a limited amount of published work with which new elbow elements can be compared. This is especially true for out-of-plane bending of elbows, which has been the subject of much less research effort than the simpler case of in-plane bending. It would therefore be useful to investigate bend behaviour using complex thin shell finite element models of bends and piping systems. A range of analyses could be performed and results used as bench-marks against which elbow element performance could be gauged.

3) Neglecting end effects and ovalisation continuity in elbow element formulations leads to conservative elements. Including these effects in an elbow element could help to reduce structural redundancy in the piping system but would increase the analysis computing requirements. However, more powerful computers are now becoming available to piping analysts at lower costs. It is therefore envisaged that within the next few years computing restrictions on piping analysis will decrease significantly, allowing the use of more complex analyses. In view of this it would be appropriate to investigate more complex elbow element formulations in order to assess the most suitable for general elastic analysis.

APPENDICES.

APPENDICES. PAGE

APPENDIX 1: ANSYS USER ELEMENT PROGRAMMING MANUAL.

APPENDIX 2: TRANSFORMATION OF ELEMENT CO-ORDINATE SYSTEMS.

APPENDIX 3: BELLOWS FORMULATION FORTRAN PROGRAMS.

APPENDIX 4: ANSYS USER ELEMENT CODE FOR ELEMENTS BEL1, PB1, PB2 AND PB3.

APPENDIX 1.

ANSYS USER ELEMENT PROGRAMMING MANUAL.

A1.1 APPENDIX 1: ANSYS USER ELEMENT PROGRAMMING MANUAL.

ANSYS is a powerful commercial finite element package, offering a wide range of applications and powerful preprocessing and postprocessing features with interactive graphics. The package incorporates a large number of structural finite elements, as well as elements for thermodynamic, fluid mechanics and electostatic applications. In addition to standard predefined elements, ANSYS incorporates a user-defined element capability. ANSYS user elements are defined by a number of FORTRAN subroutines, which when linked to the ANSYS program interact with the preprocessing, solution and postprocessing routines as a standard ANSYS element. (There are limitations to such interaction. Automatic mesh generation is precluded, and there are some restrictions in defining curved lines on plots etc.).

In this Appendix the ANSYS user element programming of linear elastic structural elements is described in detail. There are no ANSYS publications dealing with the User Element capability, although the ANSYS User's Manual Appendix U contains a brief summary of the user routines available [4.5]. The capability is defined for the user in an example user element program for a three dimensional spar element called USER.ROUTINES, supplied with mainframe and workstation versions of ANSYS. It is intended that USER.ROUTINES should be self-explanatory, however in practice this may not prove to be the case. The object of this Appendix is to guide the reader through the code and comments of the linear elastic user routines in order that the programming of elements developed in the thesis, as given in subsequent Appendices, may be fully understood.

A1.1.1 Overview of the User Element Capability.

The format of the user element ST100 must be such that it is able to communicate with the rest of ANSYS. The element must be programmed in FORTRAN 77, and the required code format is demonstated in an ANSYS subroutine called USER.ROUTINES.

USER.ROUTINES consists of a number of subroutines which fully define a 3-D spar type element with non-linear capabilities. The code is heavily documented, detailing how a user element must be programmed.

In this Appendix only small displacement structural analysis as required in the body of the thesis is considered. To program such elements in ANSYS four subroutines from USER.ROUTINES are required. These are:

i) USEREL, in which the parameters of the element are defined: eg number of nodes, number of degrees of freedom per node, dimensions of element matrices, types of loading etc.

ii) USERPT, in which the geometry of the element is defined for ANSYS plotting routines.

iii) ST100, in which the element matrices and vectors are generated: eg stiffness matrix, loading vector etc.

iv) SR100, in which stresses, strains, forces etc. are evaluated from the results of the stiffness analysis.

Figure A1.1 is a flowchart showing how these subroutines fit into the overall ANSYS Finite Element Analysis procedure.

The first step in an analysis is model creation, or pre-processing, in the ANSYS PREP7 module. In order to create a finite element model various element parameters are required; for example, the number and configuration of nodes, loading arrangements etc. In the case of the user element ST100 this information is defined in the USEREL and USERPT subroutines.

The second step in the procedure is element formation, in which the user element matrices and vectors are defined in the ST100 subroutine. This is often referred to as the "stiffness pass" of the procedure. Individual element matrices are then assembled to form the global stiffness equation, which is solved for the global displacement vector, U. Once the stiffness solution is complete, the results are processed in subroutine SR100, where element stress, strain, nodal forces etc. are evaluated. This is referred to as the "stress pass" of the procedure.

The final stage in the analysis is post processing in the POST1 module, where the results may be printed or plotted in a number of forms.

A1.1.2 Structure of Appendix 1.

This section describes the structure of this Appendix. The content is based on the ANSYS USER.ROUTINES code, which defines the required user element format by means of a worked example: ANSYS element STIF8, a three dimensional spar element. An edited version of USER.ROUTINES, in which most of the comments have been deleted, is given in the next section. Subsequent sections document the code in detail.

The first two USER.ROUTINE subroutines of interest are USEREL and USERPT. These are simple routines and are described simply by discussing the USER.ROUTINES code for the spar element.

Subroutines ST100 and SR100 are complicated routines, and discussion is broken into two sections. The first section examines the routines in general, whilst the second considers the specific example given in USER.ROUTINES.

Several types of text formats have been used in order to differentiate between text and source code, and between certain types of coding.

All explanatory and descriptive text is written in lower case except where convention demands capitals: FORTRAN, ANSYS etc.

```
FORTRAN CODE IS WRITTEN IN SMALL, UPPER CASE TYPE.
```

FORTRAN CODE NOT OF INTEREST TO THE USER ELEMENT PROGRAMMER (DESIGNATED "NOITEUP" IN USER.ROUTINES) *IS WRITTEN IN ITALIC UPPER CASE TYPE.*

It is acknowledged that some of the descriptions of user element facilities given in this manual is quoted verbatum from USER.ROUTINES.

A1.2 USER.ROUTINES: Edited Source Code.

In this section an edited version of the ANSYS program USER.ROUTINES is give. Only routines required for linear elastic structural analysis are included, and the bulk of the program comments have been deleted.

```
      PROGRAM ANSYS
C     ANSYS VERSION 4.3A
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      EXTERNAL MAIN,STOPER
      EXTERNAL ELSHFN
      CALL NNDIM
      CALL MAIN
      CALL STOPER
      END

      SUBROUTINE USEREL (ITYP,IPARM,KYSUB,KEY3D,KDOF,KUNSYM,KTRANS)
C ******** DEFINE PARAMETERS FOR ANSYS USER ELEMENT ***********
C
      INTEGER IPARM(20,12),KYSUB(9),ITYP,JTYPE,KEY3D,KDOF,KUNSYM,KTRANS
C
C    ****** DETERMINE TYPE OF ELEMENT AND THEN BYPASS IF NOT USER ELEMENT *****
      JTYPE = IPARM(ITYP,3)
      IF (JTYPE .NE. 100) GO TO 100
C
C          **********  SET 3-D KEY  **********
      KEY3D = 1
C
C          **********  DEFINE DOF SET AT EACH NODE  **********
      KDOF = 14
C
C          **********  SET UNSYMMETRIC MATRIX KEY  **********
      KUNSYM = 0
C
C      *****  DEFINE PATTERN FOR ELEMENT TO GLOBAL TRANSFORMATION  ***
      KTRANS = 3
C
C          **********  DEFINE NUMBER OF NODES  **********
      IPARM(ITYP,8) = 2
C
C          ***** DEFINE NUMBER OF TEMPERATURES (DELTEM,TEMPER) ******
      IPARM(ITYP,11) = 2
C
C          ***** DEFINE NUMBER OF PRESSURES (PRESS) *******
      IPARM(ITYP,6) = 3
C
C          ***** SET ZEROED VARIABLES (NOITUEP)
      IPARM(ITYP,12) = 0
C
C          ***** DEFINE NUMBER OF REAL CONSTANTS FOR ELEMENT (RVR) ********
      IPARM(ITYP,10) = 2
C
C          ***** DEFINE NUMBER OF VARIABLES TO BE SAVED (SVR) *****
      IPARM(ITYP,7) = 11
C
C          ***** DEFINE NUMBER OF ROWS IN ELEMENT MATRICES (KTIK) *****
      IPARM(ITYP,9) = 6
C
C          **********  SET KEY TO IDENTIFY NON-LINEAR ELEMENT  **********
      IPARM(ITYP,4) = 0
C
C          **********  SET KEY FOR THERMAL ELEMENT (KAN,-1)  **********
      IPARM(ITYP,1) = 0
 100  RETURN
      END
C
C
      SUBROUTINE USERPT (INODE,JTYPE,KSHAPE,NNODE)
C **********  USER SUBROUTINE FOR ANSYS PLOT SHAPE  **********
```

```
C         DEFINE ELEMENT SHAPE AND NUMBER OF NODES, FOR PLOTTING
      INTEGER INODE(20),JTYPE,KSHAPE,NNODE
C     **********BYPASS IF NOT USER ELEMENT (JTYPE = 100) *********
      IF (JTYPE .NE. 100) GO TO 100
C     ****** SELECT SHAPE TO BE PLOTTED BY SETTING KSHAPE *******
      KSHAPE = 2
C           **********  SET NUMBER OF ACTUAL NODES  **********
      NNODE = 2
  100 RETURN
      END
C
C
      SUBROUTINE ST100 (IELNUM,ITYP,KELIN,KELOUT,NR,KTIK,ZS,ZASS,DAMP,
     1    GSTIF,ZSC)
C ******* STIFFNESS PASS FOR 3-D SPAR DEMO ELEMENT ********
C
      EXTERNAL TRACK,GETELD,PUTELD,PROPEV,NONTBL,VZERO,MHTCH,USEERR
      INTEGER I,J,K,I3,J3,NSTR,NUM,KDEMO,NFKEY
C
      INTEGER  IELNUM,ITYP,KELIN(6),KELOUT(6),NR,KTIK,
     1 KEYERR,IOUT,NSTEPS,KFSTLD,ITTER,ITIME,NCUMIT,KRSTRT,ISPARE,
     2 K13,NPRPVL,MATST,K5,K16,IPROP,KCPDS,
     3 K20,KAY,MODE,ISYM,KAHD,IDEBUG,IXXX,
     4 ITYPE,MAT,IELEM,NROW,JTYPE,IPLOT,IPRINT,KTEMTP,KCONCV,KBICNV,
     5 KEYPLS,KEYCRP,KEYSWL,KYSUB(9),K21,NODES(20),   EPAR(50)
      REAL  ERRVAR(5)
      DOUBLE PRECISION                                                  DPDCLR
     1 DPZERO,DPHALF,DPONE,DPTWO,DPTEN,DTORAD,RADTOD,
     2 TREF,TUNIF,TOFSET, DELTIM,TIME,TIMOLD,TIME2,TIME3,DELT2,
     3 ACEL,OMEGA,CGOMEG,CGLOC,DXXX,
     4 ELMASS,XCENTR,YCENTR,ZCENTR,TFCP,SUBEX,  ERPAR(20),
     5 XYZEQ(20,3),X(20),Y(20),Z(20),  ELVOL
      COMMON /STCOM/ DPZERO,DPHALF,DPONE,DPTWO,DPTEN,DTORAD,RADTOD,
     1 TREF,TUNIF,TOFSET, DELTIM,TIME,TIMOLD,TIME2,TIME3,DELT2,
     2 ACEL(3),OMEGA(6),CGOMEG(6),CGLOC(3),       DXXX(16),
     3 KEYERR,IOUT,NSTEPS,KFSTLD,ITTER,ITIME,NCUMIT,KRSTRT,ISPARE,
     4 K13,NPRPVL,MATST,K5,K16,IPROP(20),KCPDS,
     5 K20,KAY(10),MODE,ISYM,KAHD,IDEBUG(10),       IXXX(41)
      EQUIVALENCE  (ITYPE,EPAR(1)), (MAT,EPAR(2)), (IELEM,EPAR(5)),
     1 (NROW,EPAR(7)), (JTYPE,EPAR(11)), (IPLOT,EPAR(12)),
     2 (IPRINT,EPAR(13)), (KTEMTP,EPAR(14)), (KCONCV,EPAR(16)),
     4 (KBICNV,EPAR(17)), (KEYPLS,EPAR(18)), (KEYCRP,EPAR(19)),
     5 (KEYSWL,EPAR(20)), (KYSUB(1),EPAR(21)), (K21,EPAR(30)),
     6 (NODES(1),EPAR(31))
      EQUIVALENCE  (ELMASS,ERPAR(1)), (XCENTR,ERPAR(2)),
     1 (YCENTR,ERPAR(3)), (ZCENTR,ERPAR(4)), (TFCP,ERPAR(5)),
     2 (SUBEX,ERPAR(6))
      EQUIVALENCE (X(1),XYZEQ(1,1)),(Y(1),XYZEQ(1,2)),(Z(1),XYZEQ(1,3))
C
      DOUBLE PRECISION                                                  *UPD*
     1 ZS(KTIK,KTIK),ZASS(KTIK,KTIK),DAMP(KTIK,KTIK),GSTIF(KTIK,KTIK),
     2                       ZSC(KTIK),
     3 AREA,EPORG,
     4 PROP(3),ALEN2,ALENG,DX,DY,DZ,AVETEM,FORCE,EPEL,
     5 EX,ALPX,DENS,
     6 DPSIX
      DOUBLE PRECISION                                                  *UPD*
     1 DELTEM(2),TEMPER(2),PRESS(3),
     2 RVR(2),SVR(11),
     3 TR(3,3),DFORL(6),CON,ALENN1,SALP1,CALP1,SALP2,CALP2,
     4 WTO6,EPTHT,TABLE(48),U(240)
      EQUIVALENCE (RVR(1),AREA), (RVR(2),EPORG)
C
      EQUIVALENCE (SVR(1),PROP(1)), (SVR(4),ALEN2),
     1 (SVR(5),ALENG), (SVR(6),DX), (SVR(7),DY), (SVR(8),DZ),
     2 (SVR(9),AVETEM), (SVR(10),FORCE), (SVR(11),EPEL)
C
      EQUIVALENCE (KDEMO,KYSUB(2))
C
      EQUIVALENCE  (PROP(1),EX), (PROP(2),ALPX), (PROP(3),DENS)
C
      DATA DPSIX /6.0D0/                                                DPCONST
C
      CALL TRACK(5,'ST100 ')                                            TRACK
C
      CALL GETELD (IELNUM,ITYP,EPAR(1),ERPAR(1),DELTEM(1),TEMPER(1),    GETELD
     1 PRESS(1),CON,RVR(1),SVR(1),XYZEQ(1,1),U(1))
C
C
C     ***** INITIALIZE VARIABLES FIRST TIME THRU IF NEEDED ****
```

```
      IF (KFSTLD .EQ. 0) GO TO 100
      FORCE = DPZERO
      EPEL = DPZERO
  100 CONTINUE
C
      AVETEM = DPHALF*(TEMPER(1) + TEMPER(2))
      CALL PROPEV (IELEM,MAT,JTYPE, 1,AVETEM,EX  ,1)                    PROPEV
      CALL PROPEV (IELEM,MAT,JTYPE, 2,AVETEM,ALPX,1)                    PROPEV
      CALL PROPEV (IELEM,MAT,JTYPE,10,AVETEM,DENS,1)                    PROPEV
C
C  ****  DEMONSTRATE ACCESS DATA FROM NON-LINEAR TABLE (NL COMMANDS)  ****
      IF (K13 .EQ. 0) GO TO 120
C
      NSTR = 1
      NUM = 48
      CALL NONTBL (MAT,TABLE(1),NSTR,NUM)                               NONTBL
  120 CONTINUE
C
C          **********  VERIFY GEOMETRY  **********
      DX = X(2) - X(1)
      DY = Y(2) - Y(1)
      DZ = Z(2) - Z(1)
      CON = DX**2 + DY**2
      ALEN2 = CON + DZ**2
      IF (ALEN2 .GT. DPZERO) GO TO 150
      WRITE (IOUT,2000) IELEM                                           WRITE
 2000 FORMAT(/' *** ERROR ***'/ '  ZERO LENGTH ELEMENT ',I5)
      KEYERR = 1
C
C***** THIS SUBROUTINE CALL IS USED TO PASS KEYERR TO COM2 FOR NORMAL ABORTS
C
      NFKEY = 1
      CALL USEERR (NFKEY)                                               USEERR
      GO TO 990
  150 ALENG = DSQRT(ALEN2)                                              DSQRT
      ALENN1 = DSQRT(CON)                                               DSQRT
C
C          *************** CALCULATE MASS AND CENTROID ********
      XCENTR = (X(1) + X(2))*DPHALF
      YCENTR = (Y(1) + Y(2))*DPHALF
      ZCENTR = (Z(1) + Z(2))*DPHALF
      ELMASS = DENS*AREA*ALENG*(DPONE - EPORG)
C          **********  RETURN IF ERROR(S) OR CHECK RUN  **********
      IF ((NSTEPS .EQ. 0) .OR. (KEYERR.EQ.1)) GO TO 990
C          ********** FORM TR MATRIX **********
C             THE TR MATRIX IS THE LOCAL TO GLOBAL CONVERSION MATRIX
      IF (ALENN1 .GT. .0001*ALENG) GO TO 200
      SALP1 = DPZERO
      CALP1 = DPONE
      GO TO 250
  200 SALP1 = DY/ALENN1
      CALP1 = DX/ALENN1
  250 SALP2 = DZ/ALENG
      CALP2 = ALENN1/ALENG
      TR(1,1) = CALP1*CALP2
      TR(2,1) = -SALP1
      TR(3,1) = -CALP1*SALP2
      TR(1,2) = SALP1*CALP2
      TR(2,2) = CALP1
      TR(3,2) = - SALP1*SALP2
      TR(1,3) = SALP2
      TR(2,3) = DPZERO
      TR(3,3) = CALP2
C       ******* STIFFNESS MATRIX ********
      IF (KELIN(1) .NE. 1) GO TO 400
C
C     SET UP STIFFNESS MATRIX AT END I IN ELEMENT COORDINATES
C
      CALL VZERO (ZS(1,1),36)                                           VZERO
C
      ZS(1,1) = EX*AREA/ALENG
C     CONVERT 3 BY 3 MATRIX FROM ELEMENT TO GLOBAL CARTESIAN COORDINATES.
C
      CALL MHTCH (TR(1,1),ZS(1,1), 3,KTIK, 3)                           MHTCH
C
C     FILL OUT THE COMPLETE 6 X 6 MATRIX FROM THE COMPUTED 3X3 MATRIX
      DO 300 I = 1, 3
         I3 = I + 3
         DO 300 J = 1, 3
            J3 = J + 3
```

```
            ZS(I3,J) = -ZS(I,J)
            ZS(I,J3) = -ZS(I,J)
            ZS(I3,J3) = ZS(I,J)
 300        CONTINUE
C     SET KEY THAT MATRIX WAS INDEED COMPUTED.
      KELOUT(1) = 1
C
C          ********* MASS MATRIX *****
 400  IF (KELIN(2) .NE. 1) GO TO 600
      IF (DENS .EQ. DPZERO) GO TO 600
      WTO6 = DENS*ALENG*(DPONE - EPORG)*AREA/DPSIX
      CON = DPTWO * WTO6
      CALL VZERO (ZASS(1,1),36)                                        VZERO
      DO 450 I = 1, 6
 450     ZASS(I,I) = CON
      DO 500 I = 1, 3
         I3 = I + 3
         ZASS(I,I3) = WTO6
 500     ZASS(I3,I) = WTO6
      KELOUT(2) = 1
C
C       ****** DAMPING MATRIX NORMALLY PUT IN HERE, BUT NOT INCLUDED IN
C                          THIS EXAMPLE ******
C
C       ******** STRESS STIFFNESS MATRIX *******
 600  IF (KELIN(4) .NE. 1) GO TO 800
      IF (KFSTLD .EQ. 1) FORCE = AREA*EX*EPORG
      IF (FORCE .EQ. DPZERO) GO TO 800
      CALL VZERO (GSTIF(1,1),36)                                       VZERO
      GSTIF(2,2) = FORCE/ALENG
      GSTIF(3,3) = GSTIF(2,2)
      CALL MHTCH (TR(1,1),GSTIF(1,1), 3,KTIK, 3)                       MHTCH
      DO 700 I = 1, 3
         I3 = I + 3
         DO 700 J = 1, 3
            J3 = J + 3
            GSTIF(I3,J) = -GSTIF(I,J)
            GSTIF(I,J3) = -GSTIF(I,J)
 700        GSTIF(I3,J3) = GSTIF(I,J)
      KELOUT(4) = 1
C
C          ******* LOAD VECTOR *******
 800  IF (KELIN(5) .NE. 1) GO TO 990
      CALL VZERO (DFORL(1),6)                                          VZERO
C     FIRST COMPUTE LOAD VECTOR DUE TO THERMAL AND PRESTRAIN EFFECTS
C     IN ELEMENT COORDINATES.
      EPTHT = ALPX*(AVETEM - TREF) - EPORG
C     TREF = REFERENCE TEMPERENCE (INPUT QUANTITY VALUE, TREF COMMAND)
      DFORL(1) = -AREA*EX*EPTHT
      DFORL(4) = -DFORL(1)
C     NEXT, COMPUTE LOAD VECTOR DUE TO LATERAL PRESSURES IN ELEMENT
C     COORDINATES.
      CON = PRESS(1)*ALENG*DPHALF
      DFORL(2) = - CON
      DFORL(5) = DFORL(2)
      CON = PRESS(2)*ALENG*DPHALF
      DFORL(3) = -CON
      DFORL(6) = DFORL(3)
C     FINALLY, CONVERT TO GLOBAL CARTESIAN COORDINATES, PUTTING
C     RESULT IN THE ZSC VECTOR.
      CALL VZERO (ZSC(1),6)                                            VZERO
      DO 900 I = 1,4,3
         DO 900 J = 1,3
            J3 = J + I - 1
            DO 900 K = 1,3
               I3 = K + I - 1
 900           ZSC(J3) = ZSC(J3) + TR(K,J)*DFORL(I3)
      KELOUT(5) = 1
 990  CALL PUTELD (IELNUM,EPAR(1),ERPAR(1),CON,SVR(1))                 PUTELD
C     PUTELD RESTORES DATA BACK TO FILE2
      CALL TRACK( 15,'ST100 ')                                         TRACK
      RETURN
      END
C
C
      SUBROUTINE SR100 (IELNUM,ITYP,KELOUT,ELVOL,KTIK,ZS,ZASS,ZSC)
C *********  STRESS PASS FOR 3-D SPAR DEMO ELEMENT  ********
C
      EXTERNAL TRACK,GETELD,PUTELD,SRPLT                               EXTERNAL
      INTEGER KDEMO,IPLTAY(6),I
```

```
C
      INTEGER  IELNUM,ITYP,KELIN(6),KELOUT(6),NR,KTIK,
     1 KEYERR,IOUT,NSTEPS,KFSTLD,ITTER,ITIME,NCUMIT,KRSTRT,ISPARE,
     2 K13,NPRPVL,MATST,K5,K16,IPROP,KCPDS,
     3 K20,KAY,MODE,ISYM,KAHD,IDEBUG,IXXX,
     4 ITYPE,MAT,IELEM,NROW,JTYPE,IPLOT,IPRINT,KTEMTP,KCONCV,KBICNV,
     5 KEYPLS,KEYCRP,KEYSWL,KYSUB(9),K21,NODES(20),   EPAR(50)
      REAL  ERRVAR(5)
      DOUBLE PRECISION                                                    DPDCLR
     1 DPZERO,DPHALF,DPONE,DPTWO,DPTEN,DTORAD,RADTOD,
     2 TREF,TUNIF,TOFSET, DELTIM,TIME,TIMOLD,TIME2,TIME3,DELT2,
     3 ACEL,OMEGA,CGOMEG,CGLOC,DXXX,
     4 ELMASS,XCENTR,YCENTR,ZCENTR,TFCP,SUBEX,  ERPAR(20),
     5 XYZEQ(20,3),X(20),Y(20),Z(20),  ELVOL
      COMMON /STCOM/ DPZERO,DPHALF,DPONE,DPTWO,DPTEN,DTORAD,RADTOD,
     1 TREF,TUNIF,TOFSET, DELTIM,TIME,TIMOLD,TIME2,TIME3,DELT2,
     2 ACEL(3),OMEGA(6),CGOMEG(6),CGLOC(3),        DXXX(16),
     3 KEYERR,IOUT,NSTEPS,KFSTLD,ITTER,ITIME,NCUMIT,KRSTRT,ISPARE,
     4 K13,NPRPVL,MATST,K5,K16,IPROP(20),KCPDS,
     5 K20,KAY(10),MODE,ISYM,KAHD,IDEBUG(10),      IXXX(41)
      EQUIVALENCE  (ITYPE,EPAR(1)), (MAT,EPAR(2)), (IELEM,EPAR(5)),
     1 (NROW,EPAR(7)), (JTYPE,EPAR(11)), (IPLOT,EPAR(12)),
     2 (IPRINT,EPAR(13)), (KTEMTP,EPAR(14)), (KCONCV,EPAR(16)),
     4 (KBICNV,EPAR(17)), (KEYPLS,EPAR(18)), (KEYCRP,EPAR(19)),
     5 (KEYSWL,EPAR(20)), (KYSUB(1),EPAR(21)), (K21,EPAR(30)),
     6 (NODES(1),EPAR(31))
      EQUIVALENCE  (ELMASS,ERPAR(1)), (XCENTR,ERPAR(2)),
     1 (YCENTR,ERPAR(3)), (ZCENTR,ERPAR(4)), (TFCP,ERPAR(5)),
     2 (SUBEX,ERPAR(6))
      EQUIVALENCE (X(1),XYZEQ(1,1)),(Y(1),XYZEQ(1,2)),(Z(1),XYZEQ(1,3))
C
      DOUBLE PRECISION                                                    *UPD*
     1 ZS(KTIK,KTIK),ZASS(KTIK,KTIK),ZSC(KTIK),
     2 AREA,EPORG,
     3 PROP(3),ALEN2,ALENG,DX,DY,DZ,AVETEM,FORCE,EPEL,
     4 EX,ALPX,DENS
C
      DOUBLE PRECISION                                                    *UPD*
     1 DELTEM(2),TEMPER(2),PRESS(3),
     2 RVR(2),SVR(11),
     3 EPTOT,EPTH,SIG,U(24),POSTD(19),CON
      EQUIVALENCE (RVR(1),AREA), (RVR(2),EPORG)
      EQUIVALENCE (SVR(1),PROP(1)), (SVR(4),ALEN2),
     1 (SVR(5),ALENG), (SVR(6),DX), (SVR(7),DY), (SVR(8),DZ),
     2 (SVR(9),AVETEM), (SVR(10),FORCE), (SVR(11),EPEL)
C
      EQUIVALENCE (KDEMO,KYSUB(2))
      EQUIVALENCE  (PROP(1),EX), (PROP(2),ALPX), (PROP(3),DENS)
C
      CALL TRACK (5,'SR100 ')                                             TRACK
      CALL GETELD (IELNUM,ITYP,EPAR(1),ERPAR(1),DELTEM(1),TEMPER(1),      GETELD
     1 PRESS(1),CON,RVR(1),SVR(1),XYZEQ(1,1),U(1))
C         *******  CALCULATE STRAINS  ********
      EPTOT = (DX*(U(4)-U(1)) + DY*(U(5)-U(2)) + DZ*(U(6)-U(3)))/ALEN2
      EPEL = EPTOT
      EPTH =DPZERO
C          ********  CHECK FOR LOAD VECTOR  ********
      IF (KELOUT(5) .EQ. 0) GO TO 100
      EPTH = ALPX*(AVETEM - TREF)
      EPEL = EPTOT - EPTH + EPORG
C
C        *****       STRESSES   *****
  100 SIG = EX*EPEL
      FORCE = SIG*AREA
C
C         ******   WRITE OUT RESULTS  *****
      IF (IPRINT .NE. 1) GO TO 200
      WRITE (IOUT,2000) IELEM,(NODES(I),I=1,2),MAT,(TEMPER(I),I=1,2),     WRITE
     1 EPEL,EPTH,SIG,FORCE
 2000 FORMAT(/4H EL=,I5, 7H NODES=,2I5,1X,4HMAT=,I2,7H TEMPS=,2F7.1,
     1 4H EP=,F9.6,6H EPTH=,F9.6,5H SIG=,G12.5,5H FOR=,G12.5,
     2 14H  3-D DEMO 100 )
C
C        ****** WRITE POSTDATA FILE *******
  200 IF (IPLOT .NE. 1) GO TO 900
C             *****  NUMBER OF FORCES (LEVEL 1)  *****
      IPLTAY(2) = 2
C             *****  NUMBER OF STRESSES (LEVEL 2)  *****
      IPLTAY(3) = 1
C            *****  NUMBER OF TOTAL SAVED (LEVELS 1, 2, AND 3)  *****
```

```
      IPLTAY(4) = 5
C               *****  SAVE GEOMETRY FOR CONTOURS (0,NO  1,YES)  *****
      IPLTAY(6) = 0
C      ****** PUT POSTDATA INFORMATION INTO POSTD ******
      POSTD(1) = -FORCE
      POSTD(2) =  FORCE
      POSTD(3) = SIG
      POSTD(4) = TEMPER(1)
      POSTD(5) = TEMPER(2)
      IF (K21 .LE. 4) GO TO 400
      IPLTAY(4) = 7
      POSTD(6) = EPEL
      POSTD(7) = EPTH
 400  CONTINUE
C
C     ***** PUT PLTARY INFORMATION ONTO FILE 12 *******
      CALL SRPLT (IELEM,ITYP,NROW,MAT,100,2,U(1),NODES(1),XYZEQ(1,1),   SRPLT
     1 IPLTAY(1),POSTD(1))
C     ***** COMPUTE VOLUME FOR OPTIMIZATION STUDIES ****
      ELVOL = ALENG*(DPONE - EPORG)*AREA
 900  CALL PUTELD (IELNUM,EPAR(1),ERPAR(1),CON,SVR(1))                  PUTELD
C     PUTELD RESTORES DATA BACK TO FILE3
      CALL TRACK( 15,'SR100 ')                                          TRACK
      RETURN
      END
```

A1.3 Subroutine USEREL.

This subroutine defines the parameters of the STIF100 user element. These include number of nodes, number of real constants, number of possible pressure loads etc.

Programming this routine is simple, as all that is required is to assign appropriate values to a number of "keys".

A1.3.1 Calling Arguments.

```
SUBROUTINE USEREL(ITYP,IPARM,KYSUB,KEY3D,KDOF,KUNSYM,KTRANS)
```

The calling arguments of the subroutine are ITYP,IPARM(20,12), KYSUB(9), JTYPE, KEY3D, KDOF, KUNSYM and KTRANS.

ITYP is the element type number assigned to a particular type of element by the user when modelling in PREP7. ITYP is defined in PREP7 by the ET command: ET,ITYPE(= 1 to 20).

IPARM is a 2-d array containing input/output information about all the elements in the model. Each ITYP has a row of integers in the array representing the number of nodes, real constants etc. for that ITYP.

KYSUB is a 1-D array containing a list of element option settings KEYOPT(n). Element options are declared in the PREP7 ET and KEYOPT commands.

KEY3D is a key specifying whether the element has 2-D OR 3-D geometry.

KDOF specifies the degrees of freedom selected at each node.

KUNSYM specifies whether the element matrices are symmetric or unsymmetric.

KTRANS specifies the type of element transformation required when converting from element to global coordinates.

These variables are declared in the following INTEGER statement:

```
INTEGER IPARM(20,12),KYSUB(9),ITYP,JTYPE,KEY3D,KDOF,KUNSYM,KTRANS
```

A1.3.2 Subroutine call verification.

The first step in the program is to verify that STIF100 is being called in the FE model. If this is not the case the rest of the routine is bypassed.

```
JTYPE = IPARM(ITYP,3)
IF (JTYPE .NE. 100) GO TO 100
```

This procedure is repeated at the start of all the other user routines.

A1.3.3 IMPCOM.CDK.

The line

```
INCLUDE IMPCOM.CDK
```

is not required for machines other than the microVAX, and should be deleted in this and the other subroutines.

A1.3.4 Set parameters for STIF100 to STIF8 (3-D spar).

In this example the element is programmed as a STIF8 3-D spar element. It has two nodes, with three translational degrees of freedom at each node. For further information see ANSYS user and theory manuals.

Set the 3-D key.

If the element geometry is defined in 2-D, KEY3D = 0
If the element geometry is defined in 3-D, KEY3D = 1

```
KEY3D = 1
```

Define the degrees of freedom at each node.

```
KDOF=0-UX,UY,UZ,ROTX,ROTY,ROTZ
```

1-UX	7-PRES	13-UZ,ROTX,ROTY
2-UY	8-TEMP	14-UX,UY,UZ
3-UZ	9-VOLT	15-PRES,TEMP
4-ROTX	10-MAG	16-ROTX,ROTY,ROTZ
5-ROTY	11-UX,UY	17-UX,UZ
6-ROTZ	12-UX,UY,ROTZ	19-TEMP,VOLT,MAG
		20-UX,UY,PRES
		21-UX,UY,UZ,TEMP,VOLT,MAG

```
KDOF = 14
```

Set the unsymmetric matrix key.

KUNSYM = 0 presumes symmetric matrices
KUNSYM = 1 presumes unsymmetric matrices

If the user element matrices are unsymmetric for, KEYOPT(8) must also be set to 1. It has been reserved for this purpose.

```
KUNSYM = 0
```

Define the pattern for element to global transformation.

All elements must be generated in the global cartesian system. However, the user may be using a nodal system which is different from the global cartesian system (eg NROTATE command). KTRANS permits the program to properly rotate the degrees of freedom.

0-NO NODE ROTATION
1-UX,UY
2-UX,UY,UZ,ROTX,ROTY,ROTZ
3-UX,UY,UZ
4-UZ,ROTX,ROTY
5-UX,UY,UZ,ROTZ
7-UX,UY,UZ,-,-,-(3 DOF NOT TRANSFORMED)

```
KTRANS = 3
```

Define the number of nodes.

```
IPARM(ITYP,8) = 2
```

Define the number of temperatures (DELTEM,TEMPER).

Fluences may be included with the temperatures.
Use maximum of either element temperatures or nodal temperatures

```
IPARM(ITYP,11) = 2
```

Define the number of pressures (PRESS).

If thermal analysis, two times number of convection surface.

```
IPARM(ITYP,6) = 3
```

Set zeroed variables (not applicable).

```
IPARM(ITYP,12) = 0
```

Define the number of real constants for the element (RVR).

The element real variables are defined in the PREP7 R command. In the case of STIF8 element there are two real variables: R,AREA,EPORG - area and initial strain respectively.

```
IPARM(ITYP,10) = 2
```

Define the number of variables to be saved (SVR).

"Saved variables" are those which will be passed out of ST100 for post processing in the stiffness pass or for results output data.

```
IPARM(ITYP,7) = 11
```

Define the number of rows in the element matrices (KTIK).

This value is determined by multiplying the number of nodes by the number of degrees of freedom per nodes (=NUMROW(ITYP)).

```
IPARM(ITYP,9) = 6
```

Set key to identify non-linear element.

This is used to identify inherently nonlinear elements, such as the gap or radiation link. For such elements the matrices will be reformed every iteration, regardless of other information.
0-LINEAR ELEMENT 1-NONLINEAR ELEMENT

```
IPARM(ITYP,4) = 0
```

Set key for thermal element (KAN,-1).

```
IPARM(ITYP,1) = 0
IPARM(ITYP,1) = 1
```

Element may only be used in a stress analysis.
Element may only be used in a thermal analysis.

Thermal analysis is defined as KAN,-1 or thermal substructure analyses. Stress analyses are defined as all other analyses.

```
IPARM(ITYP,1)=0

100  RETURN
END
```

A1.4 Subroutine USERPT.

This subroutine defines the geometry and configuration of the element for the ANSYS preprocessing and postprocessing routines. It is a short subroutine, and as with USEREL simply requires values to be set for integer variables.

A1.4.1 Subroutine call verification.

As in the USEREL routine, the first step is to check that the user element was called:

```
~~INCLUDE 'IMPCOM.CDK'~~ <delete from program>
INTEGER INODE(20),JTYPE,KSHAPE,NNODE
IF (JTYPE .NE. 100) GO TO 100
```

A1.4.2 Define the element shape and number of nodes for plotting.

The shape of the element for plotted is selected by setting the key KSHAPE, which can have the following values.

KSHAPE = 0 - NO PLOT
KSHAPE = 2 - 2 NODE LINE
KSHAPE = 3 - 3 NODE TRIANGLE
KSHAPE = 4 - 4 NODE QUADRILATERAL
KSHAPE = 5 - 8 NODE 3-D
KSHAPE = 6 - 8 NODE
KSHAPE = 7 - 20 NODE 3-D
KSHAPE = 10 - 16 NODE 3-D
KSHAPE = 11 - 4 NODE
KSHAPE = 12 - 10 NODE
KSHAPE = 13 - 6 NODE

```
KSHAPE = 2
```

Set the number of nodes.

```
NNODE = 2

100 CONTINUE
END
```

A1.5 Subroutine ST100: An Overview.

This routine generates the element matrices: stiffness, mass, damping,stress stiffening and force. There are three main steps carried out in ST100:

1) Read in the required data from ANSYS.

2) Process the data to form the element matrices and other required information

3) Output the evaluated information.

A1.5.1 Data transfer to ST100.

Data is transferred into ST100 by:

1 SUBROUTINE CALL
2 COMMON / STCOM /
3 CALL GETELD
4 CALL PROPEV
5 CALL NONTBL

The calling arguments of the ST100 subroutine accesses the element number, type of matrices required and matrice sizes.

The common block STCOM allows access to a great deal of information relevant to not only the element being generated but also the rest of the model. The information transferred includes analysis control integers, (such as analysis type, number of iterations etc), real constants and variables,(eg commonly used numbers in Double Precision, element co-ordinates, accelerations etc).

GETELD accesses element data from the ANSYS element data file FILE3.

PROPEV reads in the element material properties for linear elastic analysis.

NONTBL reads in non-linear material properties from the non linear property table. Non linear analysis is not covered in this manual.

A1.5.2 Matrix generation.

ST100 generates the following element matrices:

THE STIFFNESS MATRIX	= ZS
THE MASS MATRIX	= ZASS
THE DAMPING MATRIX	= DAMP
THE STRESS STIFFENING MATRIX	= GSTIF
THE FORCE VECTOR	= ZSC

ZS,ZASS,DAMP and GSTIF are all square matrices. If they are not required for the User Element then they may be omitted. However, if they are required then they must be completely defined, whether they are symmetric or not.

Most of the calculations carried out in ST100 are in double precision. This is declared by an IMPLICIT statement:

```
IMPLICIT DOUBLE PRECISION (A-H,O-Z)
```

In ANSYS all explicit real numbers are defined as FORTRAN variables in a few concise locations. STCOM has several such constants, as described in section 4.6.1. (Such definition facilitates conversion between machines and is not usually of interest to the user element programmer).

All matrices must be defined in the global co-ordinate system. Therefore, if they are generated in a local element system, they must be operated on by an appropriate transformation matrix:

[MATRIX]global = [TRANS]'[MATRIX]local [TRANS]

In ST100 the local to global coordinate transformation matrix is named TR.

A1.5.3 Transferring data out of ST100.

Data is transferred out of ST100 by:

1 SUBROUTINE CALL
(KELOUT,ZS,ZASS,DAMP,GSTIF,ZSC).

2 CALL PUTELD
(ERPAR,ELMASS,XCENTR,YCENTR,ZCENTR,SVR).

3 Printed error messages.

The subroutine call transfers all the matrices generated in ST100.

PUTELD outputs the element mass, centroid and other (saved) variables of interest.

Error messages such as "zero length element" are written directly to the output device.

A1.5.4 EXTERNAL subroutines.

The previous sections described the subroutines GETELD, PUTELD, PROPEV, and NONTBL. However, there are many other subroutines which can be referenced by ST100. Most of these are standard ANSYS vector and matrix handling routines which, for example, can zero, add, multiply and transpose vectors and matrices. ANSYS routines used in the user elements developed in this thesis are given below. The descriptions are brief as SASI do not release any detailed information on the algorithms used in the routines.

Subroutine VZERO

CALL VZERO (V,N) {V} = 0.0D0

Sets a vector or a full matrix to zero.

{V(J)} = O.0D0 ; J=1,N
or
[VM(I,J)] = 0.0D0 ; I=1,p , J=1,q

Where p and q are the dimensions of matrix VM. Also p * q = N

Subroutine MAXV

CALL MAXV (A,V1,V2,N1,N2)

$$\{V2\} = [A] \times \{V1\}$$
$$N \quad N1 \times N2 \quad N2$$

N1, N2 are the ranges within the vectors {V2} and {V1}, and matrix [A] must be dimensioned N1 x N2.

Multiplies matrix [A] with {V1} to get {V2}.

Subroutine MATXV

CALL MATXV (A,V1,V2,N1,N2)

$$\{V2\} = [A]^T \times \{V1\}$$
$$N \quad N1 \times N2 \quad N2$$

N1, N2 are the ranges within the vectors {V2} and {V1}, and matrix [A] must be dimensioned N1 x N2.

Multiplies matrix $[A]^T$ with {V1} to get {V2}.

Subroutine MAXB

CALL MAXB (A,B,C,NA,NB,NC,N1,N2,N3)

$$[C] = [A] \times [B]$$
$$N1xN2 \quad N1xN3 \quad N3xN2$$

NA, NB, NC are the row dimensions of matrices [A], [B] and [C].
N1, N2, N3 are the dimensions to be operated on.

Multiplies matrix [A] and [B] to get [C].

Subroutine MATXB

CALL MATXB (A,B,C,NA,NB,NC,N1,N2,N3)

$$[C] = [A]^T \times [B]$$
$$N1xN2 \quad N1xN3 \quad N3xN2$$

NA, NB, NC are the row dimensions of matrices [A], [B] and [C].
N1, N2, N3 are the dimensions to be operated on.

Multiplies matrix $[A]^T$ and [B] to get [C].

Subroutine MHTCH

CALL MHTCH (H,C,NH,NC,N)

$$[C] = [H]^T \times [C] \times [H]$$
$$NxN \quad NxN \quad NxN \quad NxN$$

[C] and [H] are square matrices of dimensions NxN.

Premultiplies matrix [C] with $[H]^T$, postmultiplies by [H] to get new [C].

A1.5.5 Calling arguments of the subroutine.

ST100 is called by:

```
SUBROUTINE ST100 (IELNUM,ITYP,KELIN,KELOUT,NR,KTIK,ZS,ZASS,DAMP,GSTIF,ZSC)
```

where:

IELNUM = Element number being processed

ITYP = ITYP, see section 2.

KELIN = Vector of keys if matrices are to be computed
= 0 do not compute
= 1 compute

KELIN has been defined before ST100 is called. ST100 should not change the settings.

KELOUT = Vector of keys if matrices have been computed
= 0 has not been computed
= 1 has been computed

KELOUT has been initialized to zero before ST100 is called.

KELIN(n) and KELOUT(n) refer to the following matrices:

n = 1 ZS
n = 2 ZASS
n = 3 DAMP
n = 4 GSTIF
n = 5 ZSC

There is in fact another n, n = 6, however that is not relevant to user element programming.

NR = Final matrix size (number of non-zero rows).
In general, NR is less than or equal to KTIK.

KTIK = Dimensioned matrix size (max = 60)
= IPARM(ITYP,9), NUMROW(ITYP) (see USEREL)

A1.5.6 The ST100 COMMON block: STCOM.

The ST100 COMMON block is called STCOM. The variables defined in this block are essential for the routine to run, but need not trouble the user element programmer as they do not require modification.

In USER.ROUTINES the line INCLUDE 'STCOM.CDK' concerns use on a microVAX and should be deleted.

The C (comment) at the beginning of each line of STCOM must also be deleted before compiling USER.ROUTINES.

The following variables are declared in STCOM:

INTEGER

IELNUM, ITYP, KELIN(6), KELOUT(6), NR, KTIK, KEYERR, IOUT, NSTEPS, KFSTLD, ITTER, ITIME, NCUMIT, KRSTRT, KNLRST, K13, NPRPVL, MATST, K5, K16, IPROP, KCPDS, K20, KAY, MODE, ISYM, KAHD, IDEBUG, IXXX, ITYPE, MAT, IELEM, NROW, JTYPE, IPLOT, IPRINT, KTEMTP, KCONCV, KBICNV, K5 KEYPLS, KEYCRP, KEYSWL, KYSUB(9), K21, NODES(20), EPAR(50).

REAL

ERRVAR(5).

DOUBLE PRECISION

DPZERO, DPHALF, DPONE, DPTWO, DPTEN, DTORAD, RADTOD, TREF, TUNIF, TOFSET, DELTIM, TIME, TIMOLD, TIME2, TIME3, DELT2, ACEL, OMEGA, CGOMEG, CGLOC, DXXX, ELMASS, XCENTR, YCENTR, ZCENTR, TFCP, SUBEX, ERPAR(20), XYZEQ(20,3), X(20), Y(20), Z(20), ELVOL.

The variables declared in STCOM fall into two categories: those which pertain to the whole ANSYS program, and those which occur in this subroutine only. They are described in the following sub-sections.

Real Numbers for the entire model.

These numbers concern the entire model and not just the ST100 element. These numbers may not all be needed for the particular element being programmed, but they are available if required. (For example, in STIF8 DPTEN, DTORAD are not used). With the exception of ERRVAR, they are all defined in double precision.

ERRVAR = ARRAY FOR PASSING LABELS TO ERROR SUBROUTINE (NOITUEP)

DPZERO = 0.0D0
DPHALF = 0.5D0
DPONE = 1.0D0
DPTWO = 2.0D0
DPTEN = 10.D0
DTORAD = 3.1415926535/180.
RADTOD = 180./3.1415926535

TREF = the model reference temperature (VALUE, TREF COMMAND)

TUNIF = the model uniform temperature (TEMP, TUNIF COMMAND)

TOFSET = the model offset temperature (VALUE, TOFSET COMMAND)

DELTIM = the time increment (= TIME - TIMOLD)

TIME = the time at the present iteration

TIMOLD = the time at the previous iteration

TIME2 = the time at the iteration before TIMOLD

TIME3 = the time at iteration before TIME2

DELT2 = the time increment to the next iteration for time step optimization during KAN,4 analysis with plasticity and time step extrapolation.

ACEL = an array of three accelerations (ACEL COMMAND)

OMEGA = an array of six values, the first three being the angular velocities about the origin, (OMEGA COMMAND) and the second second three the angular accelerations about the origin (DOMEGA COMMAND).

CGOMEG = an array of six values, the first three being the angular velocities about CGLOC, (CGOMEG COMMAND) and the second three are the angular accelerations about CGLOC (DCGOMG COMMAND).

CGLOC = an array of X, Y, and Z locations of the second axis of spin (CGLOC COMMAND)

DXXX = spares (NOITUEP)

Integer numbers used for the entire model.

As above, these integers concern the entire model, not just ST100.

KEYERR = error indication key.
= 0 all is reasonably okay, so keep going
= 1 terminate as quickly as possible, because of serious error

IOUT = output file number

NSTEPS = 0 if check run
> 0 if execution run

KFSTLD = 0 if after first cumulative iteration
= 1 if at first cumulative iteration

ITTER = the Iteration number in the current load step

ITIME = the load step number

NCUMIT = the cumulative iteration number

KRSTRT = the key for KRSTRT command (options module)

KNLRST = the nonlinear restart key (NOITUEP)

K13 = the key for the KNL command

NPRPVL = *twice the largest number of temperature points of material property tables*

MATST = the start of the material property tables

K5 = the maximum number of linear material properties per material

K16 = the maximum size of the nonlinear material property table

IPROP = an array of 20 keys to linear material property storage

KCPDS = *0 no C (material property specific heat) has a discontinuity*

= *1 a C discontinuity exists in the system*

K20 = analysis type (key on KAN command, ETYPE module)

KAY = input on KAY commands, OPTIONS module

MODE = mode on MODE command, LOPTION module

ISYM = ISYM on MODE command, LOPTION module

KAHD = extrapolation key used when K20(KAN) = 4 and plasticity are present:
= 0 no extrapolation
= 1 has extrapolation

When there is extrapolation the displacements, pressures, and temperatures are extrapolated. This extrapolation is done only in the stress pass.

IDEBUG = an array of 10 keys for the /DEBUG option as described in section 4.16.

IXXX = SPARES (NOITUEP)

Integers specifically relating to the STIF100 element.

These integers are used in the ST100 routine only.

IELNUM = defined above.
ITYP = defined above.
KELIN = defined above.
KELOUT = defined above.
NR = defined above.
KTIK = defined above.
ITYPE = ITYP

MAT = the material number for this element (MAT on MAT command)

IELEM = the element number

NROW = NR (NROW should not be referenced, except in the stress pass, only because NR is not available. Any modified value of NROW is not preserved, whereas a changed value of NR is).

JTYPE = the element type (JSTIF, ET command), (= 100 for user element)

IPLOT = 0 if no post data to be defined
= 1 if post data is to be defined

IPRINT = 0 suppress printout
= 1 permit printout

KTEMP = 0 if element temperature input (TE command)
= 1 if nodal temperature input (T command)

KCONCV = 1 concrete crack has changed status
= 2 no concrete crack has changed status

KBINCV = 0 not a gap element
= 1 gap element is not converged this iteration
= 2 gap element is converged this iteration

KEYPLS = 0 no plasticity for this material property
> 0 plasticity exists for this material property

KEYCRP = 0 no creep for this material property
> 0 creep exists for this material property

KEYSWL = 0 no swelling for this material property
> 0 swelling exists for this material property

KYSUB = array of 9 KEYOPT values (see USEREL)

K21 = level of post data to be generated
(KED on POSTR command)

NODES = array of up to 20 nodes (I-B on E command)

EPAR = array of convenience to pass integers

Real numbers specifically relating to the ST100 element.

ELMASS = the element mass, used by both check run and execution cases in ST100

XCENTR = X-coordinate of centroid

YCENTR = Y-coordinate of centroid

ZCENTR = Z-coordinate of centroid

TFCP = the time at the start of processing this element: needed for multiprocessing.

SUBEX = the substructure stress pass scale factor (default = 1.0). May be used for scaling of extra shapes.

ERPAR = array of convenience to pass other real values

XYZEQ = array of convenience to pass coordinates

X = array of up to 20 X-coordinates

Y = array of up to 20 Y-coordinates

Z = array of up to 20 Z-coordinates

ELVOL = the element volume, computed in stress pass for optimization

NOTE: ELMASS, ELVOL, X, Y, and Z are computed at element level, and can therefore change.

A1.5.7 Equivalencing STCOM variables.

Due to data transfer concepts a good deal of equivalencing of variables is carried out in USER.ROUTINES, (see section 1.4).

Integer constants in STCOM are equivalenced to the array EPAR. Real and Double Precision constants are equivalenced to ERPAR.

EPAR and ERPAR are simply arrays which are useful for passing information between ST100 and other ANSYS routines.

A1.5.8 DPCOM and STKCOM.

In the USER.ROUTINES comments for ST100 it states:
"Two storage regions, called DPCOM and STKCOM and labelled with *CALL DPCOM and *CALL STKCOM ... etc". This is followed by instructions to replace these *CALL commands with DOUBLE PRECISION statements.

In the case of VAX mainframe versions of ANSYS, this replacement has in fact already been done. As in the case of *CALL STCOM, these storage regions were required only for using the user element capability on a microVAX. They are not required for bigger machines. Consequently, instructions starting "Put all non-integer variables in STKCOM" etc. are not applicable and can be ignored.

A1.5.9 User defined DOUBLE PRECISION variables.

As stated above, DPCOM and STCOM are replaced by a DOUBLE PRECISION statement. The following variables are declared in this statement:

The element matrices and load vector.

All other variables and arrays required to program the element and not declared elsewhere. This includes any names used in EQUIVALENCE commands.

The U (displacement) vector.

The U vector is declared as an array of dimension U(240). This may be greater than necessary (depending on the actaul number of element degrees of freedom), but has been adopted by ANSYS simplicity. The order of the U vector, in the case of six degrees of freedom per node, is:

UX(1), UY(1), UZ(1), ROTX(1), ROTY(1), ROTZ(1),
UX(2), UY(2), etc.

A1.5.10 Element Real Variables: RVR.

Element Real variables, defined by the REAL or R command in PREP7, are transferred into ST100 from FILE3 by the GETELD subroutine. The argument used in GETELD is the array RVR. RVR(n) corresponds to the real constant in field n of the PREP7 R command, or RMORE if n>6.

When the real constants have been read into ST100, the programmer may wish to assign them more obvious names rather than RVR(1) etc. This can be done by use of the equivalence statement. For example:

```
EQUIVALENCE  (RVR(1),AREA)
```

allows RVR(1) to be referred to by the more obvious name AREA.

The RVR array must be declared by the DOUBLE PRECISION command described in section 4.9. Clearly, if the RVR array is equivalenced, the equivalent names must also be declared.

A1.5.11 Element Stored Variables: SVR.

Some of the variables calculated or read into ST100 are required at later stages of the Finite Element analysis: either for further calculations, (eg stress evaluation), or for printout as results of the analysis. Such variables are stored in an array called SVR and passed out of ST100 to FILE2 by the PUTELD subroutine.

The SVR variables are usually calculated or read into ST100 under other more familiar variable names, such as LENGTH etc. These must therefore be equivalenced to SVR(n) before they can be transferred by PUTELD.

A1.5.12 Accessing material properties: PROPEV and NONLTB.

Material properties are transferred into ST100 by the subroutine PROPEV.

Subroutine PROPEV.

PROPEV accesses the material properties selected, and will, if required, evaluate temperature dependent material properties. The arguments are defined as:

```
PROPEV (IELEM,MAT,JTYPE,LP,AVETEM,PROP(1),#).
```

IELEM	= the element number.
MAT	= material number (input quantity MAT, MATER module).
JTYPE	= element type (JSTIF, ET command) (= 100 for user element).
LP	= an integer key representing the required property as given in Table 1.
AVETEM	= temperature at which the materials are to be evaluated.
PROP	= Array in which the material properties are stored.
#	= The number of properties PROP being called.

MATERIAL SYMBOL	KEY NUMBER
EX	1
ALPX (RSVX if thermal)	2
NUXY	3
EY	4
ALPY (RSVY if thermal)	5
EZ	6
ALPZ (RSVZ if thermal)	7
NUYZ	8
NUXZ	9
DENS	10
MU	11
GXY (KXX if thermal)	12
GYZ (KYY if thermal)	13
GXZ (KZZ if thermal)	14
C	15
HF	16
VISC	17
DAMP (Constant stiffness matrix multiplier for material mat, dynamic analyses only)	18
NOT USED	19
EMIS	20

Table 1. Material Properties and associated LP keys

If temperature dependant properties are required AVETEM must be evaluated in ST100.

Using the above procedure PROPEV is called once for each individual property required: if four properties are four calls are made, each with a different LP key. However, it is possible to access all the required properties by a single PROPEV call.

This is done by setting up LP as an integer array. Integer values are assigned to elements of LP by a DATA statement. The keying of the array is as in Table 1.

For example:

```
INTEGER LP(4)
DATA LP / 1, 2, 3, 10/
PROPEV (IELEM,MAT,JTYPE,LP(1),AVETEM,PROP(1),4).
```

reads in four material properties, (EX, ALPX, NUXY, and DENS), with a single subroutine call.

Note the following changes to the subroutine calling arguments when using this second procedure:

LP is now a declared integer array defined in an INTEGER statement.

The final argument on the list specifies the number of properties to be read in by PROPEV. Even if the LP array has more elements than the N, say, specified here, only the first N will be selected by PROPEV.

Material property names: equivalencing.

The material properties accessed by PROPEV are called PROP(1), PROP(2) etc. The position of the properties in the PROP array corresponds with the order in which they are selected: either by multiple PROPEV calls or position of the property key in the LP array.

To allow the use of more familiar material property names than PROP(1) etc, (the PREP7 names for example), the PROP array can be equivalenced: eg EQUIVALENCE (PROP(1),EX) allows PROP(1) to be referred to as the more familiar EX.

Material properties as saved variables, SVR.

It is important to note that if the material properties are required as saved variables in the SVR array, the command:

```
EQUIVALENCE (SVR(1),PROP(1))
```

will equivalence all M, say, material properties declared to the first M elements of the SVR array.

For example, if the PROP array is declared to be of size PROP(4) in a DOUBLE PRECISION statement, the above equivalence would relate:

SVR(1) and PROP(1)
SVR(2) and PROP(2)
SVR(3) and PROP(3)
SVR(4) and PROP(4)

even if some or all of the PROP elements are unused, (ie less than four properties actually called in by PROPEV).

Elements with no material properties.

If an element does not need any material properties, and the call to PROPEV has been removed, EX (KXX if KAN = -1) should nevertheless be included in the input data.

A1.5.13 Accessing data from FILE3: GETELD call.

All the element data required for the solution phase is stored in the ANSYS file FILE3.DAT. Data is transferred into ST100 from FILE3 by the subroutine GETELD. The GETELD arguments are as follows:

```
GETELD(IELNUM, ITYP, EPAR(1), ERPAR(1), DELTEM(1), TEMPER(1), PRESS(1), CON, RVR(1), SVR(1),
XYZEQ(1,1), U(1))
```

EPAR = Vector defined by equivalences in STCOM above. It represents the integer quantities relating o this element

ERPAR = Vector defined by equivalences in STCOM above. It represents the real quantities relating to this element

DELTEM = Temperature (heat generation for thermal) changes between current iteration and previous iteration.

TEMPER = Temperature (heat generation for thermal) values for current iteration Note - temperatures may include fluence information

PRESS = Pressures for current iteration.
For thermal analysis, the pressure vector is ordered as
BULK TEMP(1), FILM COEF(1), BULK TEMP(2), FILM COEF(2),etc.

CON = *NOT USED (NOITUEP)*

RVR = Element real constants (R command)

SVR = Variables to be stored.

These variables can be defined or modified in either the stiffness pass or the stress pass, for use in any later calculation.

IF DELTEM, TEMPER, PRESS, RVR, or SVR are not used (appropriate value set to zero in USEREL), the variable should be set to CON in the call to GETELD.

A1.5.14 Programming the element matrices.

Obviously the element matrix generation programming will be different for every different element. However the following subsections highlight some useful general points.

The Transformation matrix.

For ease of calculation, element matrices are usually formulated and generated in a local element co-ordinate system. However, before the individual element matrices can be assembled to form the complete model, the element matrices must be defined in global co-ordinates.

Local element co-ordinates are transformed to global co-ordinates by operating upon the element matrix with a Transformation matrix, TR:

$$[\text{MATRIX}]_{\text{global}} = [\text{TR}]^{\text{T}} [\text{MATRIX}]_{\text{local}} [\text{TR}]$$

The User Element programmer may write his own code to perform the above matrix manipulation, however, an ANSYS subroutine called MHTCH may be used if both the matrices are square.

MHTCH is a service subroutine which pre and post multiplies a matrix by a second matrix. The pre-multiplication is done with the transpose of the multiplying matrix. Consider, for example, the transformation of the stiffness matrix, ZS:

$$[\text{ZS}]_{\text{global}} = [\text{TR}]^{\text{T}} [\text{ZS}]_{\text{local}} [\text{TR}]$$

To perform the above calculation using MHTCH, the following command would be given:

```
CALL MHTCH (TR(1,1),ZS(1,1), #1,KTIK, #2)
```

#1 is the first dimension of TR.
KTIK is the first dimension of ZS.
#2 is the size of the matrices being operated on.

Check run: element mass and centroid.

ST100 is called if a checking run of the ANSYS model is made prior to complete analysis. No element matrices are generated, but the mass and centroid are calculated.

Element geometry is also checked to ensure that there are no zero length elements etc. At this point the condition of the check run and error indication integers is examined. A RETURN to the main program is made if error(s) or a check run is indicated.

Matrices required/Generated matrices keying.

Before any element matrix calculations are performed, ST100 checks that the matrix is in fact required. This is done by examining the KELIN array keys. If KELIN(n) is 0 the matrix is generated; if it is 1 it is not.

KELIN has been defined before ST100 is called, and ST100 should not change the settings.

Once a required matrix has been generated the appropriate KELOUT key is set to 1. KELOUT has been initialized to zero before ST100 is called.

KELIN(n) and KELOUT(n) refer to the following matrices:

n = 1 - ZS
n = 2 - ZASS
n = 3 - DAMP
n = 4 - GSTIF
n = 5 - ZSC

There is also another n, n=6, but that is not of interest to the user element programmer.

Zeroing element matrices.

Before the element matrices are generated they are first of all set to zero. As in the case of MHTCH the user programmer may write his own code to do this, or he may use an ANSYS subroutine called VZERO. For example, a 12x12 stiffness matrix ZS is zeroed by the following subroutine call:

```
CALL VZERO (ZS(1,1),144)
```

VZERO zeroes out ZS starting at location (1,1) and going on for 144 items.

A1.5.15 Load vector.

The ST100 load vector contains loading generated at element level only; eg thermal loads, specified element prestrain and element pressure loads. Nodal forces and moments and body forces such as accelerations are not added at element level, but applied to thee assembled model.

As in the case of the element matrices, the load vector must be converted to global coordinates. The transformation in this case is:

$$[ZSC]global = [TR]^T [ZSC]local$$

A1.5.16 Debugging ST100.

Two ANSYS debugging tools are available to the user element programmer, called DEBUG and TRACK.

DEBUG consists of an array of 10 debug keys, of which only two are documented, (the others are not of interest to the user element programmer). These keys are actually available for all ANSYS elements, not just ST100. DEBUG is accessed by issuing the command:

```
/DEBUG,,,ID3,,ID5
(before)
 /INPUT,27
```

IDX = 0 no debug printout
IDX = 1 include debug printout

ID3 = 1 will print out all computed element matrices and the load vector.

ID5 = 1 will print out other element debug, including the KELIN values, the arguments of GETELD, and input and output of PROPEV.

TRACK is used find where the program is for running time studies and in case of aborts. It is accessed by issuing the command:

/TRACK,5,5
(before)
/INPUT,27

A1.6 Subroutine ST100: USER.ROUTINES.

This section examines in detail the USER.ROUTINES example of programming ST100 as a STIF8 type 3-D spar element.

The following FORTRAN names have been assigned to the matrices:

THE STIFFNESS MATRIX	= ZS
THE MASS MATRIX	= ZASS
THE DAMPING MATRIX	= DAMP
THE STRESS STIFFENING MATRIX	= GSTIF
THE FORCE VECTOR	= ZSC

A1.6.1 The ST100 subroutine call.

```
SUBROUTINE ST100 (IELNUM, ITYP, KELIN, KELOUT, NR, KTIK, ZS, ZASS, DAMP, GSTIF, ZSC)
```

A1.6.2 Double precision.

Most of the calculations carried out in ST100 are in double precision. This is declared in the IMPLICIT statement:

```
IMPLICIT DOUBLE PRECISION (A-H,O-Z)
```

In USER.ROUTINES this command has a comment C in the first line, which must be removed before compiling.

A1.6.3 EXTERNAL subroutines and functions.

External subroutines and functions used in ST100 are declared in the following EXTERNAL statement.

```
EXTERNAL  TRACK, GETELD, PUTELD, PROPEV, NONTBL, VZERO, MHTCH, USEERR
```

A1.6.4 Integers.

Integers used in the ST100 routine are declared in the following INTEGER statement:

```
INTEGER I,J,K,I3,J3,NSTR,NUM,KDEMO,NFKEY
```

These integers are used as loop counters, matrix element positioning integers, non-linear property specifiers,a demonstration integer variable, and an abort key.

A1.6.5 The ST100 COMMON block: STCOM.

The INCLUDE STCOM.CDK command should be deleted, as should the comment C at the start of each line of STCOM:

```
      INTEGER  IELNUM,ITYP,KELIN(6),KELOUT(6),NR,KTIK,
     1 KEYERR,IOUT,NSTEPS,KFSTLD,ITTER,ITIME,NCUMIT,KRSTRT,ISPARE,
     2 K13,NPRPVL,MATST,K5,K16,IPROP,KCPDS,
     3 K20,KAY,MODE,ISYM,KAHD,IDEBUG,IXXX,
     4 ITYPE,MAT,IELEM,NROW,JTYPE,IPLOT,IPRINT,KTEMTP,KCONCV,KBICNV,
     5 KEYPLS,KEYCRP,KEYSWL,KYSUB(9),K21,NODES(20),   EPAR(50)

      REAL  ERRVAR(5)

      DOUBLE PRECISION
     1 DPZERO,DPHALF,DPONE,DPTWO,DPTEN,DTORAD,RADTOD,
     2 TREF,TUNIF,TOFSET, DELTIM,TIME,TIMOLD,TIME2,TIME3,DELT2,
     3 ACEL,OMEGA,CGOMEG,CGLOC,DXXX,
     4 ELMASS,XCENTR,YCENTR,ZCENTR,TFCP,SUBEX,  ERPAR(20),
     5 XYZEQ(20,3),X(20),Y(20),Z(20),  ELVOL

      COMMON /STCOM/ DPZERO,DPHALF,DPONE,DPTWO,DPTEN,DTORAD,RADTOD,
     1 TREF,TUNIF,TOFSET, DELTIM,TIME,TIMOLD,TIME2,TIME3,DELT2,
     2 ACEL(3),OMEGA(6),CGOMEG(6),CGLOC(3),        DXXX(16),
     3 KEYERR,IOUT,NSTEPS,KFSTLD,ITTER,ITIME,NCUMIT,KRSTRT,ISPARE,
     4 K13,NPRPVL,MATST,K5,K16,IPROP(20),KCPDS,
     5 K20,KAY(10),MODE,ISYM,KAHD,IDEBUG(10),      IXXX(41)
```

A1.6.6 Equivalencing.

In order to pass data between ANSYS routines STCOM variables are equivalenced to the arrays EPAR for integer variables, and ERPAR for real and double precision variables.

```
      EQUIVALENCE (ITYPE,EPAR(1)), (MAT,EPAR(2)), (IELEM,EPAR(5)),
     1 (NROW,EPAR(7)), (JTYPE,EPAR(11)), (IPLOT,EPAR(12)),
     2 (IPRINT,EPAR(13)), (KTEMTP,EPAR(14)), (KCONCV,EPAR(16)),
     4 (KBICNV,EPAR(17)), (KEYPLS,EPAR(18)), (KEYCRP,EPAR(19)),
     5 (KEYSWL,EPAR(20)), (KYSUB(1),EPAR(21)), (K21,EPAR(30)),
     6 (NODES(1),EPAR(31))

      EQUIVALENCE  (ELMASS,ERPAR(1)), (XCENTR,ERPAR(2)),
     1 (YCENTR,ERPAR(3)), (ZCENTR,ERPAR(4)), (TFCP,ERPAR(5)),
     2 (SUBEX,ERPAR(6))

      EQUIVALENCE (X(1),XYZEQ(1,1)),(Y(1),XYZEQ(1,2)),(Z(1),XYZEQ(1,3))
```

A1.6.7 DOUBLE PRECISION: *CALL DPCOM and *CALL STKCOM.

The storage regions DPCOM and STKCOM referred to in USER.ROUTINES are replaced by the following DOUBLE PRECISION statements.

```
 DOUBLE PRECISION
1 ZS(KTIK,KTIK),ZASS(KTIK,KTIK),DAMP(KTIK,KTIK),
& GSTIF(KTIK,KTIK),ZSC(KTIK),
2 AREA,EPORG,
3 PROP(3),ALEN2,ALENG,DX,DY,DZ,AVETEM,FORCE,EPEL,
4 EX,ALPX,DENS,
5 DPSIX
 DOUBLE PRECISION
1 DELTEM(2),TEMPER(2),PRESS(3),
2 RVR(2),SVR(11),
3 TR(3,3),DFORL(6),CON,ALENN1,SALP1,CALP1,SALP2,CALP2,
4 WTO6,EPTHT,TABLE(48),U(240)
```

A1.6.8 RVR: equivalencing.

The 3-D spar element has two real constants defined in the PREP7 R command: area AREA, and initial strain EPORG. Values for AREA and EPORG are read into ST100 from FILE3 by the subroutine GETELD. GETELD reads in this data under the array name RVR. In order to use the more familiar names of AREA and EPORG these are equivalenced to the corresponding RVR elements as follows:

```
EQUIVALENCE (RVR(1),AREA), (RVR(2),EPORG)
```

A1.6.9 SVR: equivalencing.

Variables which will be required by other ANSYS routines are passed out of ST100 by the subroutine PUTELD. PUTELD requires this information in the form of the array SVR. Therefore the required element values are equivalenced to SVR as follows:

```
 EQUIVALENCE (SVR(1),PROP(1)), (SVR(4),ALEN2),
1 (SVR(5),ALENG), (SVR(6),DX), (SVR(7),DY), (SVR(8),DZ),
2 (SVR(9),AVETEM), (SVR(10),FORCE), (SVR(11),EPEL)
```

Note that SVR(2) and SVR(3) are implicitly equivalenced to PROP(2) and PROP(3) by the above command. (See sections 1.4 and 4.12.3).

A1.6.10 KEYOPT equivalencing.

The next equivalence demonstrates accessing KEYOPT(2) and and calling it KDEMO. This is done purely as a demonstration of equivalencing KEYOPT and is not in fact used in ST100.

```
EQUIVALENCE (KDEMO,KYSUB(2))
```

A1.6.11 Equivalencing material properties.

The next equivalence is to make the output from PROPEV (the material property evaluation subroutine) more convenient.

```
EQUIVALENCE  (PROP(1),EX), (PROP(2),ALPX), (PROP(3),DENS)
```

A1.6.12 Defining the number six in double precision.

The number six is used in generating the mass matrix. In the matrix generating procedures all calculations are done in double precision. DPSIX has been declared as a variable in the DOUBLE PRECISION statement described in section 5.7. It is assigned the value of six in the following DATA statement:

```
DATA DPSIX /6.0D0/
```

Several other useful constants are defined in STCOM: see section 4.6.1.

A1.6.13 Call the external subroutine TRACK.

Track is used to monitor the progress of the run. See section 4.16.

```
CALL TRACK(5,'ST100 ')
```

A1.6.14 Accessing data from FILE3: GETELD call.

GETELD takes needed data from FILE3 and makes it available to the element.

```
CALL GETELD (IELNUM,ITYP,EPAR(1),ERPAR(1),DELTEM(1),TEMPER(1),
    1 PRESS(1),CON,RVR(1),SVR(1),XYZEQ(1,1),U(1))
```

A1.6.15 Initialising the variables.

Initialize variables first time through if needed:

```
      IF (KFSTLD .EQ. 0) GO TO 100
       FORCE = DPZERO
       EPEL = DPZERO
100   CONTINUE
```

A1.6.16 Reading material properties into ST100.

The element material properties are read into ST100 by the subroutine PROPEV. If the element has temperature dependant properties, these are evaluated by PROPEV at the average element temperature AVETEM. AVETEM must therefore be evaluated before PROPEV is called.

```
AVETEM = DPHALF*(TEMPER(1) + TEMPER(2))
```

The arguments of the PROPEV routines are detailed in section 4.12. In this case three properties are required: EX, ALPX and DENS. Therefore, considering the keying options described in 4.12.1, LP is assigned the values 1, 2 and 10.

```
CALL PROPEV (IELEM,MAT,JTYPE,1,AVETEM,PROP(1),1)
CALL PROPEV (IELEM,MAT,JTYPE,2,AVETEM,PROP(1),1)
CALL PROPEV (IELEM,MAT,JTYPE,10,AVETEM,PROP(1),1)
```

An alternative procedure for reading in all the required properties with a single subroutine call is outlined in section 4.12.1.

The required material properties are declared in an integer array, LP. The same property keying as above is used, and the alternative code to the three calls is:

```
INTEGER LP(4)
DATA LP / 1, 2, 10/
PROPEV (IELEM,MAT,JTYPE,LP(1),AVETEM,PROP(1),3).
```

Note the final PROPEV calling argument, the number 3. This is the number of material properties to be read, and must be changed accordingly if more or less than three properties are required.

A1.6.17 Accessing data from a non-linear table (NL commands).

This entry is simply to demonstrate how data is accessed from a non-linear table (NL commands). It is not actually used in this example.

```
      IF (K13 .EQ. 0) GO TO 120
      NSTR = 1
       NUM = 48
       CALL NONTBL (MAT,TABLE(1),NSTR,NUM)
120   CONTINUE
```

A1.6.18 Verifying the element geometry.

The input element geometry is checked to ensure there are no zero length elements.

In the case of zero length elements a suitable message is written to the output device, and in the case of a fatal error the run is aborted by the subroutine USEERR.

```
      DX = X(2) - X(1)
      DY = Y(2) - Y(1)
      DZ = Z(2) - Z(1)
      CON = DX**2 + DY**2
       ALEN2 = CON + DZ**2
       IF (ALEN2 .GT. DPZERO) GO TO 150
       WRITE (IOUT,2000) IELEM
       2000 FORMAT ('ZERO LENGTH ELEMENT' ,I5)
       KEYERR = 1
      NFKEY = 1
      CALL USEERR (NFKEY)
       GO TO 990
150   ALENG = DSQRT(ALEN2)
      ALENN1 = DSQRT(CON)
```

A1.6.19 Forming the TR matrix.

The TR matrix is the element local to global coordinate conversion matrix.

```
      IF (ALENN1 .GT. .0001*ALENG) GO TO 200
      SALP1 = DPZERO
      CALP1 = DPONE
      GO TO 250
      200  SALP1 = DY/ALENN1
      CALP1 = DX/ALENN1
      250  SALP2 = DZ/ALENG
      CALP2 = ALENN1/ALENG
      TR(1,1) = CALP1*CALP2
      TR(2,1) = -SALP1
      TR(3,1) = -CALP1*SALP2
      TR(1,2) = SALP1*CALP2
      TR(2,2) = CALP1
      TR(3,2) = - SALP1*SALP2
      TR(1,3) = SALP2
      TR(2,3) = DPZERO
      TR(3,3) = CALP2
```

A1.6.20 Calculating the element mass and centroid.

```
XCENTR = (X(1) + X(2))*DPHALF
YCENTR = (Y(1) + Y(2))*DPHALF
ZCENTR = (Z(1) + Z(2))*DPHALF
ELMASS = DENS*AREA*ALENG*(DPONE - EPORG)
```

A1.6.21 Check run and error check RETURN.

At this point the condition of the check run and error indication integers is examined. A RETURN to the main program is made if error(s) or a check run is indicated.

```
IF ((NSTEPS .EQ. 0) .OR. (KEYERR.EQ.1)) GO TO 990
```

A1.6.22 Generating the stiffness matrix.

Check that the stiffness matrix is required to be calculated. If KELIN(1) is 0 it does, if 1 it does not.

```
IF (KELIN(1) .NE. 1) GO TO 400
```

Set up stiffness matrix at end I.

The first step is to generate a 3x3 stiffness matrix for end I of the element. This is done in local element co-ordinates.

```
CALL VZERO (ZS(1,1),36)
```

VZERO is a service subroutine that zeroes out the stiffness matrix ZS.

```
ZS(1,1) = EX*AREA/ALENG
```

Convert matrix to global cartesian coordinates.

The 3x3 stiffness matrix, (all elements are zero except ZS(1,1) which is AE/L), is converted from local element co-ordinates to global co-ordinates.

```
CALL MHTCH (TR(1,1),ZS(1,1), 3,KTIK, 3)
```

MHTCH is a service subroutine which pre and post multiplies ZS by TR.

Fill out the complete 6x6 matrix.

Finally, the full 6x6 matrix is generated from the 3x3.

```
      DO 300 I = 1, 3
      I3 = I + 3
      DO 300 J = 1, 3
      J3 = J + 3
      ZS(I3,J) = -ZS(I,J)
      ZS(I,J3) = -ZS(I,J)
      ZS(I3,J3) = ZS(I,J)
300   CONTINUE
```

Set key that matrix was indeed computed.

```
      KELOUT(1) = 1
```

A1.6.23 Mass matrix.

```
400   IF (KELIN(2) .NE. 1) GO TO 600
      IF (DENS .EQ. DPZERO) GO TO 600
      WTO6 = DENS*ALENG*(DPONE - EPORG)*AREA/DPSIX
      CON = DPTWO * WTO6
      CALL VZERO (ZASS(1,1),36)                          VZERO
      DO 450 I = 1, 6
450      ZASS(I,I) = CON
      DO 500 I = 1, 3
         I3 = I + 3
         ZASS(I,I3) = WTO6
500      ZASS(I3,I) = WTO6
      KELOUT(2) = 1
```

A1.6.24 Damping matrix.

The damping matrix is normally put in here, but not included in this example

A1.6.25 Stress stiffness matrix.

```
 600  IF (KELIN(4) .NE. 1) GO TO 800
      IF (KFSTLD .EQ. 1) FORCE = AREA*EX*EPORG
      IF (FORCE .EQ. DPZERO) GO TO 800
      CALL VZERO (GSTIF(1,1),36)
      GSTIF(2,2) = FORCE/ALENG
      GSTIF(3,3) = GSTIF(2,2)
      CALL MHTCH (TR(1,1),GSTIF(1,1), 3,KTIK, 3)
      DO 700 I = 1, 3
         I3 = I + 3
         DO 700 J = 1, 3
            J3 = J + 3
            GSTIF(I3,J) = -GSTIF(I,J)
            GSTIF(I,J3) = -GSTIF(I,J)
700         GSTIF(I3,J3) = GSTIF(I,J)
      KELOUT(4) = 1
```

A1.6.26 Load vector.

```
800  IF (KELIN(5) .NE. 1) GO TO 990
     CALL VZERO (DFORL(1),6)
     VZERO
```

There are three possible types of loading evaluated at element level: thermal, prestrain and pressure loading. Nodal loads, (such as forces and moments), and body forces, (such as accelerations), are added by ANSYS later in the solution procedure.

Thermal and Prestrain effects.

The first step is to compute the load vector due to thermal and prestrain effects in element coordinates.

```
     EPTHT = ALPX*(AVETEM - TREF) - EPORG
```

TREF = Reference temperature (input quantity value, TREF command)

```
     DFORL(1) = -AREA*EX*EPTHT
     DFORL(4) = -DFORL(1)
```

Lateral pressures.

Next, the load vector due to lateral pressures is computed in element coordinates.

```
     CON = PRESS(1)*ALENG*DPHALF
     DFORL(2) = - CON
     DFORL(5) = DFORL(2)
     CON = PRESS(2)*ALENG*DPHALF
     DFORL(3) = -CON
     DFORL(6) = DFORL(3)
```

ZSC vector.

The final step is to convert the above to the global cartesian coordinate system and put the result in the ZSC vector.

```
     CALL VZERO (ZSC(1),6)
     DO 900 I = 1,4,3
        DO 900 J = 1,3
           J3 = J + I - 1
           DO 900 K = 1,3
              I3 = K + I - 1
900           ZSC(J3) = ZSC(J3) + TR(K,J)*DFORL(I3)
     KELOUT(5) = 1
990  CALL PUTELD (IELNUM,EPAR(1),ERPAR(1),CON,SVR(1))
```

Call subroutine PUTELD

PUTELD Restores data back to FILE2

```
CALL TRACK( 15,'ST100 ')
RETURN
END
```

A1.7 Subroutine SR100: An Overview.

The subroutine SR100 performs the stress pass for the STIF100 element. It takes the displacement results from the stiffness pass and evaluates element strains, stresses, nodal forces etc. There are three main steps in the SR100 routine:

1) Read in the required data from ANSYS.
2) Process that data to evaluate stresses etc.
3) Output the required information.

SR100 is similar in structure to ST100. The following topics are common to both subroutines, and have been described in section 4:

Data transfer
PUTELD
EXTERNAL subroutines and functions
COMMON block STCOM
EQUIVALENCING of variables
DPCOM and STKCOM
RVR and SVR
TRACK
GETELD.

To avoid repetition these topics are not covered in this section.

A1.7.1 Results evaluation.

SR100 processes the results of the stiffness analysis - the U vector - to evaluate element stresses and strains, nodal forces etc.

As in ST100, the calculations are performed in double precision.

A1.7.2 Data is transferred out of SR100 by:

1 CALL PUTELD (SVR only).
2 Printed output.
3 Writing a Post data file: Subroutine SRPLT.

A1.7.3 Calling arguments of the SR100 routine.

The SR100 calling arguments are similar to those of ST100:

```
SUBROUTINE SR100 (IELNUM,ITYP,KELOUT,ELVOL,KTIK,ZS,ZASS,ZSC)
```

The calling arguments are similar to those of ST100, (see section 4.5), with the following differences:

KELIN is not a calling argument here as SR100 does not evaluate matrices. ELVOL, the element volume, is added to the list of arguments.

A1.7.4 User defined DOUBLE PRECISION variables.

As stated above, DPCOM and STKCOM are replaced by a DOUBLE PRECISION statement. The following variables are declared:

i) Any element matrices required. Most element stress passes do not use the element matrices, but they are made available if needed.

ii) The element displacement vector U.

iii) All other variables and arrays required to calculate the element stresses etc. This includes any names used in EQUIVALENCE statements and not declared elsewhere.

iv) The POSTD array. This is added for plot file item numbering. There is a one to one correspondence between the item number on the plot file and the position in the array. The size is determined by the number of items put on the plot file.

A1.7.5 Material Properties.

Material properties required in SR100 are read in by GETELD as saved variables SVR. In the USER.ROUTINES program the appropriate SVR() are equivalenced to PROP() then further equivalenced to their familiar ANSYS names; EX,NUXY etc.

A1.7.6 Calculating stresses etc.

Element stresses etc are evaluated from the element nodal displacement vector, U. The standard finite element approach is:

i) Calculate the strains from U: Strain-Displacement relationships.

ii) Calculate the stresses from the strains: Stress-Strain relationship.

iii) Calculate the nodal forces and moments from the element stresses.

This method does not require the element matrices in order to evaluate stresses, but others do. For example, in traditional engineering matrix-methods the U vector is operated on by the stiffness matrix to give the reaction forces, from which nodal stresses are evaluated.

In order to facilitate such an approach, the element matrices formed in ST100 are made available in SR100. (Via the subroutine call).

A1.7.7 Writing out the results.

The results of the stress run are written to the output device by a WRITE command. Results written out in this way include element type number, node numbers, material type, nodal temperatures, mechanical strain, thermal strain, stresses, nodal forces etc. The format of the output is defined by the user element programmer. Element printout is suppressed during interactive runs.

A1.7.8 The post data file: FILE12.

Element data for post processing is stored in the post data file, FILE12. The information is stored in FILE12 in six different "levels", as follows:

Level 1. Force components.

Level 2. Basic stress components: ie centroidal and nodal stresses.

Level 3. Principal stresses (centroidal and nodal), temperatures etc.

Level 4. Additional surface data.

Level 5. Nonlinear centroidal data.

Level 6. Linear integration point data.

The first three levels are evaluated and stored by default. Further levels must be specified by the KYPOST or the POSTR commands in PREP7.

The post data file, FILE12, is set up according to the elements in the integer array IPLTARY. IPLTARY defines the number of data to be stored in each level of FILE12. The array elements contain the following information:

IPLTAY(1)

IPLTAY(2) number of forces in level 1.

IPLTAY(3) number of basic stresses in level 2.

IPLTAY(4) total number of items stored in the first three (default) levels. This of course implies the number of principal stresses etc in level 3.

IPLTAY(5) Element type.

IPLTAY(6) Key for saving geometry for elements requiring contour plots of results.
= 1, save geometry.
= 0 do not save geometry.

The double precision array POSTD is used to pass the results of the analysis to the post data file. The size of POSTD is determined by the number of items to be output to FILE12, calculated as follows:

11 + Number of nodes in element + number of results to be stored.

A1.7.9 Using other ANSYS element PLOT file formats.

It is possible to use existing ANSYS element graphics capabilities by "fooling" ANSYS into thinking it is dealing with a standard element.

For example, if it is desired to put the plot file in the format of another element type(eg STIF(45), JELTYP(ITYP) must be temporarily reset from 100 to 45.

This variable is not in the labelled common region STCOM. Rather it is in the labelled common region COM1. therefore, the following statements would need to be put at the beginning of this subroutine(SR100):

```
INTEGER          IUXX,JELTYP
REAL             RUXX
DOUBLE PRECISION DPUXX

COMMON / COM1 / IUXX(975),RUXX(51),DPUXX(124),JELTYP(20)
```

The following statement would need to be placed just before the call to SRPLT:

```
JELTYP(ITYP) = 45
```

Finally, the following statement would need to be placed just after the call to SRPLT:

```
JELTYP(ITYP) = 100
```

A1.8 Subroutine SR100: USER.ROUTINES.

This section examines in detail the USER.ROUTINES example of programming SR100 for a STIF8 type 3-D spar element.

A1.8.1 The SR100 subroutine call.

```
SUBROUTINE SR100 (IELNUM,ITYP,KELOUT,ELVOL,KTIK,ZS,ZASS,ZSC)
```

A1.8.2 Double precision.

The SR100 calculations are performed in double precision. This is implied by the following statement:

```
IMPLICIT DOUBLE PRECISION (A-H,O-Z)
```

USER.ROUTINES has a comment C in the first line of this command, which should be removed.

A1.8.3 External subroutines

The external functions required by SR100 in this (STIF8) application are:

```
EXTERNAL TRACK,GETELD,PUTELD,SRPLT
```

A1.8.4 Integers

```
INTEGER KDEMO,IPLTAY(6),I
```

A1.8.5 STCOM storage.

A full description of the STCOM region is given in section 4.6. Again it should be noted that the line:

```
INCLUDE 'STCOM.CDK'
```

should be removed, along with the C in column 1, from this point down to the end of the STCOM block:

```
 INTEGER  IELNUM,ITYP,KELIN(6),KELOUT(6),NR,KTIK,
1 KEYERR,IOUT,NSTEPS,KFSTLD,ITTER,ITIME,NCUMIT,KRSTRT,KNLRST,
2 K13,NPRPVL,MATST,K5,K16,IPROP,KCPDS,
3 K20,KAY,MODE,ISYM,KAHD,IDEBUG,IXXX,
4 ITYPE,MAT,IELEM,NROW,JTYPE,IPLOT,IPRINT,KTEMTP,KCONCV,KBICNV,
5 KEYPLS,KEYCRP,KEYSWL,KYSUB(9),K21,NODES(20),   EPAR(50)
 REAL  ERRVAR(5)
 DOUBLE PRECISION
```

```
1 DPZERO,DPHALF,DPONE,DPTWO,DPTEN,DTORAD,RADTOD,
2 TREF,TUNIF,TOFSET, DELTIM,TIME,TIMOLD,TIME2,TIME3,DELT2,
3 ACEL,OMEGA,CGOMEG,CGLOC,DXXX,
4 ELMASS,XCENTR,YCENTR,ZCENTR,TFCP,SUBEX,  ERPAR(20),
5 XYZEQ(20,3),X(20),Y(20),Z(20),  ELVOL
 COMMON /STCOM/ DPZERO,DPHALF,DPONE,DPTWO,DPTEN,DTORAD,RADTOD,
1 TREF,TUNIF,TOFSET, DELTIM,TIME,TIMOLD,TIME2,TIME3,DELT2,
2 ACEL(3),OMEGA(6),CGOMEG(6),CGLOC(3),        DXXX(16),
3 KEYERR,IOUT,NSTEPS,KFSTLD,ITTER,ITIME,NCUMIT,KRSTRT,KNLRST,
4 K13,NPRPVL,MATST,K5,K16,IPROP(20),KCPDS,
5 K20,KAY(10),MODE,ISYM,KAHD,IDEBUG(10),IXXX(41)
```

A1.8.6 STCOM equivalencing.

For parameter passing, STCOM variables are equivalenced to the EPAR (integer) and ERPAR (double precision) arrays.

```
  EQUIVALENCE  (ITYPE,EPAR(1)), (MAT,EPAR(2)), (IELEM,EPAR(5)),
1 (NROW,EPAR(7)), (JTYPE,EPAR(11)), (IPLOT,EPAR(12)),
2 (IPRINT,EPAR(13)), (KTEMTP,EPAR(14)), (KCONCV,EPAR(16)),
4 (KBICNV,EPAR(17)), (KEYPLS,EPAR(18)), (KEYCRP,EPAR(19)),
5 (KEYSWL,EPAR(20)), (KYSUB(1),EPAR(21)), (K21,EPAR(30)),
6 (NODES(1),EPAR(31))

 EQUIVALENCE  (ELMASS,ERPAR(1)), (XCENTR,ERPAR(2)),
1 (YCENTR,ERPAR(3)), (ZCENTR,ERPAR(4)), (TFCP,ERPAR(5)),
2 (SUBEX,ERPAR(6))

 EQUIVALENCE (X(1),XYZEQ(1,1)),(Y(1),XYZEQ(1,2)),(Z(1),XYZEQ(1,3))
```

A1.8.7 DOUBLE PRECISION: *CALL DPCOM and *CALL STKCOM.

As in ST100, DPCOM and STKCOM have been replaced by DOUBLE PRECISION statements.

```
 DOUBLE PRECISION
1 ZS(KTIK,1),ZASS(KTIK,1),ZSC(KTIK),
2 AREA,EPORG,
3 PROP(3),ALEN2,ALENG,DX,DY,DZ,AVETEM,FORCE,EPEL,
4 EX,ALPX,DENS

 DOUBLE PRECISION
1 DELTEM(2),TEMPER(2),PRESS(3),
2 RVR(2),SVR(11),
3 EPTOT,EPTH,SIG,U(24),POST(19),CON
```

The POSTD array is added for plot file item numbering.

A1.8.8 Real and saved variable equivalencing.

The equivalent variable names used for RVR and SVR must agree with those used in ST100. See sections 5.8 and 5.9.

```
 EQUIVALENCE (RVR(1),AREA), (RVR(2),EPORG)

 EQUIVALENCE (SVR(1),PROP(1)), (SVR(4),ALEN2),
1 (SVR(5),ALENG), (SVR(6),DX), (SVR(7),DY), (SVR(8),DZ),
2 (SVR(9),AVETEM), (SVR(10),FORCE), (SVR(11),EPEL)
```

A1.8.9 KEYOPT equivalencing.

This equivalence is for demonstration purposes only.

```
EQUIVALENCE (KDEMO,KYSUB(2))
```

A1.8.10 Material property equivalencing.

The material properties are read in as saved variables SVR and equivalenced to the PROP array as described in section 7.8 above. They are now equivalenced to their more familiar ANSYS names.

```
EQUIVALENCE  (PROP(1),EX), (PROP(2),ALPX), (PROP(3),DENS)
```

A1.8.11 Calling the external subroutine TRACK.

TRACK is used to monitor the progress of the run. See section 4.16.

```
CALL TRACK (5,'SR100 ')
```

A1.8.12 Accessing data from FILE3: GETELD call.

GETELD takes the required element information from FILE3 and makes it available to the element. See section 4.13.

```
CALL GETELD (IELNUM, ITYP, EPAR(1), ERPAR(1), DELTEM(1), TEMPER(1), PRESS(1), CON, RVR(1),
                                                                    SVR(1), XYZEQ(1,1), U(1))
```

A1.8.13 Calculating the strains.

EPTOT is the total element strain .
EPEL is the mechanical strain.
EPTH is the thermal strain.
EPORG is the initial strain, and is a PREP7 Real variable.

If there is no load vector then there will be no strain due to thermal and prestrain effects. If that is the case the mechanical strain EPEL will be equal to the total strain EPTOT. If there is a load vector then the thermal strain must be evaluated, if not go on to the stress calculations.

```
EPTOT = (DX*(U(4)-U(1)) + DY*(U(5)-U(2)) + DZ*(U(6)-U(3)))/ALEN2
EPEL = EPTOT
EPTH = DPZERO
```

```
      IF (KELOUT(5) .EQ. 0) GO TO 100
      EPTH = ALPX*(AVETEM - TREF)
      EPEL = EPTOT - EPTH + EPORG
```

A1.8.14 Calculating the stresses.

```
100   SIG = EX*EPEL
      FORCE = SIG*AREA
```

A1.8.15 Writing out the results.

```
      IF (IPRINT .NE. 1) GO TO 200

      WRITE (IOUT,2000) IELEM, (NODES(I),I=1,2), MAT, (TEMPER(I),I=1,2), EPEL, EPTH, SIG, FORCE

      2000 FORMAT(/4H EL=,I5, 7H NODES=,2I5,1X,4HMAT=,I2,7H TEMPS=,2F7.1,
      1 4H EP=,F9.6,6H EPTH=,F9.6,5H SIG=,G12.5,5H FOR=,G12.5,
      2 14H  3-D DEMO 100 )
```

A1.8.16 Writing a postdata file.

The STIF8 post data file is written on FILE12 as shown below.

1. FX(I) level 1
2. FX(J)

---

3. SIG level 2

---

4. TEMP(I) level 3
5. TEMP(J)

---

6. EP
7. EPTH
8. EPPL
9. EPOR
10. EPCR
11. SIGEPL
12. EPSW
13. FLS(I)
14. FLS(J) level 5

```
200   IF (IPLOT .NE. 1) GO TO 900
```

Number of forces (LEVEL 1)

```
      IPLTAY(2) = 2
```

Number of stresses (LEVEL 2)

```
      IPLTAY(3) = 1
```

Number of total saved (LEVELS 1, 2, AND 3)

```
IPLTAY(4) = 5
```

Save geometry for contours (0,NO 1,YES)

```
IPLTAY(6) = 0
```

Put postdata information into POSTD

```
POSTD(1) = -FORCE
POSTD(2) =  FORCE
POSTD(3) = SIG
POSTD(4) = TEMPER(1)
POSTD(5) = TEMPER(2)
```

If more than the default level of postdata information is desired, it should be added here. KED(K21) was set > 3. IPLTAY(4) must be set to the total of all levels.

```
    IF (K21 .LE. 4) GO TO 400
    IPLTAY(4) = 7
    POSTD(6) = EPEL
    POSTD(7) = EPTH
400 CONTINUE
```

A1.8.17 Writing results to FILE12: subroutine SRPLT

Put PLTARY information onto FILE 12

```
CALL SRPLT (IELEM, ITYP, NROW, MAT, U(1), NODES(1), XYZEQ(1,1), IPLTAY(1), PLTARY(1))
```

A1.8.18 Compute volume for optimization studies

The element volume is evaluated for possible use in an optimization analysis .

```
    ELVOL = ALENG*(DPONE - EPORG)*AREA
900 CALL PUTELD (IELNUM,EPAR(1),ERPAR(1),CON,SVR(1))
```

A1.8.19 Restoring the data to FILE3: PUTELD.

PUTELD restores data back to FILE3.

```
CALL TRACK( 15,'SR100 ')
RETURN
END
```

A1.9 Example: Straight Cylindrical Beam (Pipe) User Element.

This section details the programming of an ANSYS straight cylindrical beam user element. The beam stiffness matrix is a simplified version of the ANSYS straight beam element STIF4 [6.1].

The element is a 2 noded line element, with 6 degrees of freedom per node: three translations and three rotations. No mass or stress stiffening effects are considered.

A1.9.1 Input information.

Real constants.

The beam formulation requires the following input values: elastic modulus, Poisson's ratio, mid-wall radius and wall thickness.

From these it evaluates

i) Modulus of rigidity $G = \frac{E}{1+\nu}$

ii) Second moment of area $I = \pi r^3 h$

iii) Polar moment of area $J = 2\pi r^3 h = 2i$

iv) Cross sectional area $A = 2\pi r h$

Variable names

The following variable names are used in ST100 and SR100:

RAD	mid surface radius
THICK	wall thickness
AREA	cross sectional area
RAD3	radius cubed
SECMOM	second moment of area
POLMOM	polar moment of area
EI	elastic mod. x SECMOM

EA	" " x AREA
ALEN2	rigidity mod x POLMOM
ALENG3	element length cubed

A1.9.2 ANSYS USER.FOR FORTRAN source code.

The element source code is given in the following sections. Additional comments are written in lower case large text.

```
C     ANSYS USER ELEMENT CODE FOR STRAIGHT CYLINDRICAL BEAM ELEMENT
C     ------------------------------------------------------------
C
C                          DONALD MACKENZIE  OCT/NOV 1987
C
C     ANSYS VERSION 4.3A

      PROGRAM ANSYS
C     ANSYS VERSION 4.3A
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      EXTERNAL MAIN,STOPER
      EXTERNAL ELSHFN
      CALL NNDIM
      CALL MAIN
      CALL STOPER
      END
```

Subroutine USEREL.

```
SUBROUTINE USEREL (ITYP,IPARM,KYSUB,KEY3D,KDOF,KUNSYM,KTRANS)

C
      INTEGER IPARM(20,12),KYSUB(9),ITYP,JTYPE,KEY3D,KDOF,KUNSYM,KTRANS
C
C     ****** DETERMINE TYPE OF ELEMENT AND THEN BYPASS IF NOT USER ELEMENT
      JTYPE = IPARM(ITYP,3)
      IF (JTYPE .NE. 100) GO TO 100
C          **********  SET 3-D KEY  **********
      KEY3D = 1
C
C          **********  DEFINE DOF SET AT EACH NODE  **********
      KDOF = 0
C
C          **********  SET UNSYMMETRIC MATRIX KEY  **********
      KUNSYM = 0
C
C      *****  DEFINE PATTERN FOR ELEMENT TO GLOBAL TRANSFORMATION
      KTRANS = 2
C
C          **********  DEFINE NUMBER OF NODES  **********
      IPARM(ITYP,8) = 2
C
C          *****  DEFINE NUMBER OF TEMPERATURES (DELTEM,TEMPER)
      IPARM(ITYP,11) = 0
C
C          *****  DEFINE NUMBER OF PRESSURES (PRESS) *******
C        IF THERMAL ANALYSIS, TWO TIMES NUMBER OF CONVECTION SURFACES
      IPARM(ITYP,6) = 0
C
C          *****  SET ZEROED VARIABLES (NOITUEP)
      IPARM(ITYP,12) = 0
C
C          *****  DEFINE NUMBER OF REAL CONSTANTS FOR ELEMENT (RVR)
      IPARM(ITYP,10) = 2
```

```
C
C          ***** DEFINE NUMBER OF VARIABLES TO BE SAVED (SVR)
      IPARM(ITYP,7) = 24
C
C          ***** DEFINE NUMBER OF ROWS IN ELEMENT MATRICES (KTIK)
      IPARM(ITYP,9) = 12
C
C          **********  SET KEY TO IDENTIFY NON-LINEAR ELEMENT
      IPARM(ITYP,4) = 0
C
C          **********  SET KEY FOR THERMAL ELEMENT (KAN,-1)
      IPARM(ITYP,1) = 0
C
 100  RETURN
      END
```

Subroutine USERPT.

```
SUBROUTINE USERPT (INODE,JTYPE,KSHAPE,NNODE)

C
C **********  USER SUBROUTINE FOR ANSYS PLOT SHAPE  **********
C
C     DEFINE ELEMENT SHAPE AND NUMBER OF NODES, FOR PLOTTING
C
      INTEGER INODE(20),JTYPE,KSHAPE,NNODE
C    *********BYPASS IF NOT USER ELEMENT (JTYPE = 100) *********
      IF (JTYPE .NE. 100) GO TO 100
C    ****** SELECT SHAPE TO BE PLOTTED BY SETTING KSHAPE *******
      KSHAPE = 2
C          **********  SET NUMBER OF ACTUAL NODES  **********
      NNODE = 2
 100  RETURN
      END
C
```

Subroutine ST100.

```
SUBROUTINE ST100 (IELNUM,ITYP,KELIN,KELOUT,NR,KTIK,ZS,ZASS,DAMP,
     1    GSTIF,ZSC)
```

Subroutine ST100 evaluates the element stiffness matrix, ZS. No mass, stress stiffening or damping matrices are evaluated, nor is a load vector.

```
C ******* STIFFNESS PASS FOR 3-D CYL BEAM ELEMENT ********
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      EXTERNAL TRACK,GETELD,PUTELD,PROPEV,NONTBL,VZERO,MHTCH,USEERR
      INTEGER I,J,K,I3,J3,I6,J6,I9,J9,LP(4),NSTR,NUM,KDEMO,NFKEY
C

C    ***** START STCOM STORAGE ********
      INTEGER  IELNUM,ITYP,KELIN(6),KELOUT(6),NR,KTIK,
     1 KEYERR,IOUT,NSTEPS,KFSTLD,ITTER,ITIME,NCUMIT,KRSTRT,ISPARE,
     2 K13,NPRPVL,MATST,K5,K16,IPROP,KCPDS,
     3 K20,KAY,MODE,ISYM,KAHD,IDEBUG,IXXX,
     4 ITYPE,MAT,IELEM,NROW,JTYPE,IPLOT,IPRINT,KTEMTP,KCONCV,KBICNV,
     5 KEYPLS,KEYCRP,KEYSWL,KYSUB(9),K21,NODES(20),   EPAR(50)
      REAL  ERRVAR(5)
C
      DOUBLE PRECISION
     1 DPZERO,DPHALF,DPONE,DPTWO,DPTEN,DTORAD,RADTOD,
     2 TREF,TUNIF,TOFSET, DELTIM,TIME,TIMOLD,TIME2,TIME3,DELT2,
     3 ACEL,OMEGA,CGOMEG,CGLOC,DXXX,
     4 ELMASS,XCENTR,YCENTR,ZCENTR,TFCP,SUBEX,  ERPAR(20),
     5 XYZEQ(20,3),X(20),Y(20),Z(20),  ELVOL
C
      COMMON /STCOM/ DPZERO,DPHALF,DPONE,DPTWO,DPTEN,DTORAD,RADTOD,
     1 TREF,TUNIF,TOFSET, DELTIM,TIME,TIMOLD,TIME2,TIME3,DELT2,
     2 ACEL(3),OMEGA(6),CGOMEG(6),CGLOC(3),        DXXX(16),
```

```
     3 KEYERR,IOUT,NSTEPS,KFSTLD,ITTER,ITIME,NCUMIT,KRSTRT,ISPARE,
     4 K13,NPRPVL,MATST,K5,K16,IPROP(20),KCPDS,
     5 K20,KAY(10),MODE,ISYM,KAHD,IDEBUG(10),      IXXX(41)
C

      EQUIVALENCE  (ITYPE,EPAR(1)), (MAT,EPAR(2)), (IELEM,EPAR(5)),
     1 (NROW,EPAR(7)), (JTYPE,EPAR(11)), (IPLOT,EPAR(12)),
     2 (IPRINT,EPAR(13)), (KTEMTP,EPAR(14)), (KCONCV,EPAR(16)),
     4 (KBICNV,EPAR(17)), (KEYPLS,EPAR(18)), (KEYCRP,EPAR(19)),
     5 (KEYSWL,EPAR(20)), (KYSUB(1),EPAR(21)), (K21,EPAR(30)),
     6 (NODES(1),EPAR(31))

C
      EQUIVALENCE  (ELMASS,ERPAR(1)), (XCENTR,ERPAR(2)),
     1 (YCENTR,ERPAR(3)), (ZCENTR,ERPAR(4)), (TFCP,ERPAR(5)),
     2 (SUBEX,ERPAR(6))
C
      EQUIVALENCE (X(1),XYZEQ(1,1)),(Y(1),XYZEQ(1,2)),(Z(1),XYZEQ(1,3))
C
```

The following DOUBLE PRECISION statements replace the *CALL DPCOM and STKCOM commands in USER.ROUTINES.

```
      DOUBLE PRECISION
     1 ZS(KTIK,KTIK),
     2 RAD,THICK,AREA,SECMOM,POLMOM,EI,EA,GJ,ALENG3,
     3 PROP(4),ALEN2,ALENG,DX,DY,DZ,AVETEM,FORCE,EPEL,
     4 EX,ALPX,NUXY,DENS,
     5 DPPI
C
      DOUBLE PRECISION
     1 RVR(2),SVR(24),
     2 TR(12,12),DFORL(6),CON,ALENN1,SALP1,CALP1,SALP2,CALP2,
     3 U(240)
C

      EQUIVALENCE (RVR(1),RAD), (RVR(2),THICK)
C
      EQUIVALENCE (SVR(1),PROP(1)), (SVR(5),ALEN2),
     1 (SVR(6),ALENG), (SVR(7),DX), (SVR(8),DY), (SVR(9),DZ),
     2 (SVR(10),AVETEM), (SVR(11),FORCE), (SVR(12),EPEL),
     3 (SVR(13),AREA), (SVR(14),SECMOM), (SVR(15),POLMOM)
C    A FURTHER 9 SAVED VARIABLES SVR ARE DEFINED LATER AS THE
C    3x3 TRANSFORMATION MATRIX.
C
      EQUIVALENCE  (PROP(1),EX), (PROP(2),ALPX), (PROP(3),NUXY),
     1 (PROP(4),DENS)

C
      DATA DPPI / 3.14159265358979300 /
C     SET UP INTEGER ARRAY FOR ACCESSING MATERIAL PROPERTIES
      DATA LP / 1, 2, 3,10/
C
      CALL TRACK(5,'ST100 ')
C
C ***** NOTE CHANGE OF  DELTEM(1) ETC TO CON AS THEY ARE SET TO
C       ZERO IN USEREL
C
```

Element data is accessed from FILE3 by GETELD. Note the CON arguments replacing the unused pressure and temperature arguments from USER.ROUTINES.

```
      CALL GETELD (IELNUM,ITYP,EPAR(1),ERPAR(1),CON,CON,
     1 CON,CON,RVR(1),SVR(1),XYZEQ(1,1),U(1))
```

```
C     ***** INITIALIZE VARIABLES FIRST TIME THROUGH IF NEEDED ****
      IF (KFSTLD .EQ. 0) GO TO 100
      FORCE = DPZERO
      EPEL = DPZERO
  100 CONTINUE
```

Material properties are accessed by PROPEV. A single subroutine call is made, utilising a LP array, rather than one call per property required.

```
      CALL PROPEV (IELEM,MAT,JTYPE,LP(1),AVETEM,PROP(1),4)

C           ********** VERIFY GEOMETRY **********
      DX = X(2) - X(1)
      DY = Y(2) - Y(1)
      DZ = Z(2) - Z(1)
      CON = DX**2 + DY**2
      ALEN2 = CON + DZ**2
      IF (ALEN2 .GT. DPZERO) GO TO 150
      WRITE (IOUT,2000) IELEM
 2000 FORMAT ('ZERO LENGTH ELEMENT' ,I5)
      KEYERR = 1
      NFKEY = 1
      CALL USEERR (NFKEY)
      GO TO 990
  150 ALENG = DSQRT(ALEN2)
      ALENN1 = DSQRT(CON)
```

As the cylindrical beam is symmetric about its longitudinal axis the transformation matrix is identical to the USER.ROUTINES spar TR matrix. For non-circular beams the orientation of the beams 'depth' axis would have to be accounted for.

```
C           ********** FORM TR MATRIX **********
C         THE TR MATRIX IS THE LOCAL TO GLOBAL CONVERSION MATRIX
C   ZERO TR MATRIX
      CALL VZERO(TR(1,1),144)
      IF (ALENN1 .GT. .0001*ALENG) GO TO 200
      SALP1 = DPZERO
      CALP1 = DPONE
      GO TO 250
  200 SALP1 = DY/ALENN1
      CALP1 = DX/ALENN1
  250 SALP2 = DZ/ALENG
      CALP2 = ALENN1/ALENG
      TR(1,1) = CALP1*CALP2
      TR(2,1) = -SALP1
      TR(3,1) = -CALP1*SALP2
      TR(1,2) = SALP1*CALP2
      TR(2,2) = CALP1
      TR(3,2) = -SALP1*SALP2
      TR(1,3) = SALP2
      TR(2,3) = DPZERO
      TR(3,3) = CALP2
```

In order to re-define the TR matrix in SR100, the 3x3 TR matrix elements are stored as saved variables and thus passed out of ST100 by PUTELD.

```
C     THE 3x3 TRANSFORMATION MATRIX IS STORED IN THE SAVED
C      VARIABLES ARRAY IT IS THUS PASSED TO SR100,
C     WHERE IT IS REQUIRED FOR STRESS EVALUATION.
      SVR(16)=TR(1,1)
      SVR(17)=TR(1,2)
      SVR(18)=TR(1,3)
      SVR(19)=TR(2,1)
      SVR(20)=TR(2,2)
      SVR(21)=TR(2,3)
      SVR(22)=TR(3,1)
      SVR(23)=TR(3,2)
      SVR(24)=TR(3,3)
```

As the stiffness matrix is to be fully defined as a 12x12 matrix, so must the TR matrix.

```
C FILL OUT 12x12 MATRIX FROM THE 3x3
C
      DO 270  I=1,3
         I3=I+3
         I6=I+6
         I9=I+9
         DO 270 J=1,3
            J3=J+3
            J6=J+6
            J9=J+9
            TR(I3,J3) = TR(I,J)
            TR(I6,J6) = TR(I,J)
            TR(I9,J9) = TR(I,J)
 270   CONTINUE

C          *************** CALCULATE MASS AND CENTROID ********
      XCENTR = (X(1) + X(2))*DPHALF
      YCENTR = (Y(1) + Y(2))*DPHALF
      ZCENTR = (Z(1) + Z(2))*DPHALF

C          **********  RETURN IF ERROR(S) OR CHECK RUN  *********
      IF ((NSTEPS .EQ. 0) .OR. (KEYERR.EQ.1)) GO TO 990

C       ******* STIFFNESS MATRIX ********
      IF (KELIN(1) .NE. 1) GO TO 400
```

The stiffness matrix is essentially the ANSYS STIF4 matrix, with simplifications as the beam is axisymmetric.

```
C     STRAIGHT CYLINDRICAL THIN WALLED BEAM ELEMENT STIFFNESS MATRIX
C     BASED ON ANSYS STIF4 WITH SIMPLIFICATIONS FOR SYMMETRY ETC.
```

Initially, constants useful in the matrix calculations are evaluated.

```
C     EVALUATE SECMOM (ie I), POLMOM (ie J)
C
      AREA = 2*DPPI*RAD*THICK
      SECMOM = DPPI*RAD**3*THICK
      POLMOM = SECMOM*2
C
C    EVALUATE USEFUL COMBINATIONS OF THE ABOVE
C
      EI = EX*SECMOM
      EA = EX*AREA
      GJ = (EX/(1+NUXY))*POLMOM
C     LENGTH CUBED:
      ALENG3 = ALEN2*ALENG

C     ZERO THE STIFFNESS MATRIX
      CALL VZERO(ZS(1,1),144)

C     EVALUATE ELEMENTS OF STIFFNESS MATRIX
C
      ZS(1,1) = EA/ALENG
      ZS(1,7) = -ZS(1,1)
      ZS(7,1) = -ZS(1,1)
      ZS(7,7) = ZS(1,1)
      ZS(2,2) = 12*EI/ALENG3
      ZS(2,8) = -ZS(2,2)
      ZS(8,2) = ZS(2,8)
      ZS(3,3) = ZS(2,2)
      ZS(3,9) = ZS(2,8)
      ZS(9,3) = ZS(2,8)
      ZS(8,8) = ZS(2,2)
      ZS(9,9) = ZS(2,2)
      ZS(2,6) = 6*EI/ALEN2
```

```
      ZS(6,2) = ZS(2,6)
      ZS(2,12) = ZS(2,6)
      ZS(12,2) = ZS(2,6)
      ZS(3,5) = -ZS(2,6)
      ZS(5,3) = ZS(3,5)
      ZS(3,11) = ZS(3,5)
      ZS(11,3) = ZS(3,5)
      ZS(5,9) = ZS(2,6)
      ZS(9,5) = ZS(2,6)
      ZS(6,8) = ZS(3,5)
      ZS(8,6) = ZS(3,5)
      ZS(8,12) = ZS(3,5)
      ZS(12,8) = ZS(3,5)
      ZS(9,11) = ZS(2,6)
      ZS(11,9) = ZS(2,6)
      ZS(4,4) = GJ/ALENG
      ZS(4,10) = -ZS(4,4)
      ZS(10,4) = -ZS(4,4)
      ZS(10,10) = ZS(4,4)
      ZS(5,5) = 4*EI/ALENG
      ZS(6,6) = ZS(5,5)
      ZS(11,11) = ZS(5,5)
      ZS(5,11) = ZS(5,5)/2
      ZS(11,5) = ZS(5,11)
      ZS(6,12) = ZS(5,11)
      ZS(12,6) = ZS(5,11)
      ZS(12,12) = ZS(5,5)
```

The element stiffness matrix is transformed to global co-ordinates by evaluating:

$$[ZS]_{global} = [TR]^T [ZS]_{local} [TR]$$

This is done by the ANSYS routine MHTCH.

```
C     CONVERT TO GLOBAL COORDINATES
      CALL MHTCH(TR(1,1),ZS(1,1),12,KTIK,12)

C     SET KEY THAT MATRIX WAS INDEED COMPUTED.
      KELOUT(1) = 1
 400   CONTINUE

 990  CALL PUTELD (IELNUM,EPAR(1),ERPAR(1),CON,SVR(1))
C     PUTELD RESTORES DATA BACK TO FILE2
      CALL TRACK( 15,'ST100 ')
      RETURN
      END
```

Subroutine SR100.

```
SUBROUTINE SR100 (IELNUM,ITYP,KELOUT,ELVOL,KTIK,ZS,ZASS,ZSC)
```

SR100 evaluates direct and bending stresses at the nodes, and combines these to give maximum and minimum nodal stresses.

Stresses are evaluated from the nodal forces and moments:

$$\sigma_{dir} = \frac{F_x}{A} \qquad \sigma_{bendz} = \frac{M_y r_o}{I_y} \qquad \sigma_{bendy} = \frac{M_z r_o}{I_z}$$

The nodal forces and moments are evaluated by a matrix-displacement method. The displacement vector **d** from the stiffness pass is back substituted into the force-displacement relationship to yield nodal forces.

Forces are evaluated in the local element co-ordinate system. Thus:

$$\mathbf{F}_l = \mathbf{K}_l \, \mathbf{d}_l$$

where

$$\mathbf{K}_l = \mathbf{TR}\, \mathbf{K}_g\, \mathbf{TR}^T \qquad \mathbf{d}_l = \mathbf{TR}\, \mathbf{d}_g$$

Thus

$$\mathbf{F}_l = \mathbf{TR}\, \mathbf{K}_g\, \mathbf{D}_g$$

```
C *********  STRESS PASS FOR 3-D CYL BEAM  ELEMENT  ********
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      EXTERNAL TRACK,GETELD,PUTELD,SRPLT,MAXV,VZERO
      INTEGER KDEMO,IPLTAY(6),I,J,I3,J3,I6,J6,I9,J9
C

C   ***** START STCOM STORAGE ********
      INTEGER  IELNUM,ITYP,KELIN(6),KELOUT(6),NR,KTIK,
     1 KEYERR,IOUT,NSTEPS,KFSTLD,ITTER,ITIME,NCUMIT,KRSTRT,ISPARE,
     2 K13,NPRPVL,MATST,K5,K16,IPROP,KCPDS,
     3 K20,KAY,MODE,ISYM,KAHD,IDEBUG,IXXX,
     4 ITYPE,MAT,IELEM,NROW,JTYPE,IPLOT,IPRINT,KTEMTP,KCONCV,KBICNV,
     5 KEYPLS,KEYCRP,KEYSWL,KYSUB(9),K21,NODES(20),   EPAR(50)
      REAL  ERRVAR(5)
C
      DOUBLE PRECISION
     1 DPZERO,DPHALF,DPONE,DPTWO,DPTEN,DTORAD,RADTOD,
     2 TREF,TUNIF,TOFSET, DELTIM,TIME,TIMOLD,TIME2,TIME3,DELT2,
     3 ACEL,OMEGA,CGOMEG,CGLOC,DXXX,
     4 ELMASS,XCENTR,YCENTR,ZCENTR,TFCP,SUBEX,  ERPAR(20),
     5 XYZEQ(20,3),X(20),Y(20),Z(20),  ELVOL
C
      COMMON /STCOM/ DPZERO,DPHALF,DPONE,DPTWO,DPTEN,DTORAD,RADTOD,
     1 TREF,TUNIF,TOFSET, DELTIM,TIME,TIMOLD,TIME2,TIME3,DELT2,
     2 ACEL(3),OMEGA(6),CGOMEG(6),CGLOC(3),        DXXX(16),
     3 KEYERR,IOUT,NSTEPS,KFSTLD,ITTER,ITIME,NCUMIT,KRSTRT,ISPARE,
     4 K13,NPRPVL,MATST,K5,K16,IPROP(20),KCPDS,
     5 K20,KAY(10),MODE,ISYM,KAHD,IDEBUG(10),      IXXX(41)
C

      EQUIVALENCE  (ITYPE,EPAR(1)), (MAT,EPAR(2)), (IELEM,EPAR(5)),
     1 (NROW,EPAR(7)), (JTYPE,EPAR(11)), (IPLOT,EPAR(12)),
     2 (IPRINT,EPAR(13)), (KTEMTP,EPAR(14)), (KCONCV,EPAR(16)),
     4 (KBICNV,EPAR(17)), (KEYPLS,EPAR(18)), (KEYCRP,EPAR(19)),
     5 (KEYSWL,EPAR(20)), (KYSUB(1),EPAR(21)), (K21,EPAR(30)),
     6 (NODES(1),EPAR(31))
C
      EQUIVALENCE  (ELMASS,ERPAR(1)), (XCENTR,ERPAR(2)),
     1 (YCENTR,ERPAR(3)), (ZCENTR,ERPAR(4)), (TFCP,ERPAR(5)),
     2 (SUBEX,ERPAR(6))
      EQUIVALENCE (X(1),XYZEQ(1,1)),(Y(1),XYZEQ(1,2)),(Z(1),XYZEQ(1,3))
C
```

```
      DOUBLE PRECISION
     1 ZS(KTIK,KTIK),
     2 RAD,THICK,AREA,SECMOM,POLMOM,
     3 PROP(4),ALEN2,ALENG,DX,DY,DZ,AVETEM,FORCE,EPEL,
     4 EX,ALPX,NUXY,DENS,ALENN1,
     5 TR(12,12),SALP1,CALP1,SALP2,CALP2,
     6 FELEM(12),DUMMY(12),SDIRI,SBENDI,SDIRJ,SBENDJ,
     7 SMAXI,SMINI,SMAXJ,SMINJ
C
      DOUBLE PRECISION
     1 RVR(2),SVR(24),
     2 EPTOT,EPTH,SIG,U(24),POSTD(20),CON
C

      EQUIVALENCE (RVR(1),RAD), (RVR(2),THICK)
C
      EQUIVALENCE (SVR(1),PROP(1)), (SVR(5),ALEN2),
     1 (SVR(5),ALENG), (SVR(7),DX), (SVR(8),DY), (SVR(9),DZ),
     2 (SVR(10),AVETEM), (SVR(11),FORCE), (SVR(12),EPEL),
     3 (SVR(13),AREA), (SVR(14),SECMOM), (SVR(15),POLMOM)
C    THE 3x3 TR MATRIX IS DEFINED FROM THE OTHER  9 SVR FURTHER ON.

C
      EQUIVALENCE (KDEMO,KYSUB(2))
      EQUIVALENCE  (PROP(1),EX), (PROP(2),ALPX), (PROP(3),NUXY),
     1 (PROP(4),DENS)

C
      CALL TRACK (5,'SR100 ')
      CALL GETELD (IELNUM,ITYP,EPAR(1),ERPAR(1),CON,CON,
     1 CON,CON,RVR(1),SVR(1),XYZEQ(1,1),U(1))
C
```

The full 12x12 TR matrix is required in order to evaluate the nodal forces. The 3x3 matrix passed out from ST100 as saved variables is redefined and expanded to 12x12.

```
C    RE-EVALUATE TRANSFORMATION MATRIX: REQD. FOR EVALUATION OF
C    NODAL FORCES.
      TR(1,1)= SVR(16)
      TR(1,2)= SVR(17)
      TR(1,3)= SVR(18)
      TR(2,1)= SVR(19)
      TR(2,2)= SVR(20)
      TR(2,3)= SVR(21)
      TR(3,1)= SVR(22)
      TR(3,2)= SVR(23)
      TR(3,3)= SVR(24)
C
C FILL OUT 12X12 MATRIX FROM THE 3x3
C
      DO 80  I=1,3
         I3=I+3
         I6=I+6
         I9=I+9
         DO 80 J=1,3
            J3=J+3
            J6=J+6
            J9=J+9
            TR(I3,J3) = TR(I,J)
            TR(I6,J6) = TR(I,J)
            TR(I9,J9) = TR(I,J)
 80    CONTINUE

C      ******** EVALUATE NODAL FORCES ********
```

The nodal force evaluation procedure is described at the beginning of this section.

```
C     THE NODAL FORCES IN ELEMENT CO-ORDS ARE EVALUATED FROM THE EQUATION
C     {FELEM} = [TR] [ZS] {U}
C     THE MATRIX/VECTOR MULTIPLICATION IS PERFORMED BY THE ANSYS ROUTINE
C     MAXV, WHICH MULTS. MATRIX A AND VECTOR V.
C
      CALL MAXV(ZS(1,1),U(1),DUMMY(1),KTIK,KTIK)
      CALL MAXV(TR(1,1),DUMMY(1),FELEM(1),KTIK,KTIK)
C

C     EVALUATE NODAL STRESSES.
```

The nodal direct and bending stresses are calculated from the equations given at the beginning of this section. Note that the direct stress at node i must be negated. This is a consequence of the force evaluation procedure.

The bending stress evaluated is due to the combined y and z moments: ie

$$\sigma_b = \sqrt{\sigma_{bz}^2 + \sigma_{by}^2}$$

The maximum stress is the direct plus bending, and the minimum stress the direct minus bending.

```
C     THE STRESSES ARE EVALUATED AT THE NODES ONLY.
C
      AREA=2*3.14159*RAD*THICK
      SDIRI = -FELEM(1)/AREA
      SBENDI = ((RAD+THICK/2)/SECMOM)*DSQRT(FELEM(5)**2+FELEM(6)**2)
      SMAXI = SDIRI+SBENDI
      SMINI = SDIRI-SBENDI
C
      SDIRJ = FELEM(7)/AREA
      SBENDJ = ((RAD+THICK/2)/SECMOM)*DSQRT(FELEM(11)**2+FELEM(12)**2)
      SMAXJ = SDIRJ+SBENDJ
      SMINJ = SDIRJ-SBENDJ
C
```

The results of the stress run are written to FILE12 by SRPLT. In order to do this the 'structure' of FILE12 must be defined, and the required results set equal to elements of the POSTD array.

```
C        ****** WRITE POSTDATA FILE *******
 200  IF (IPLOT .NE. 1) GO TO 900
C              *****  NUMBER OF FORCES (LEVEL 1)  *****
      IPLTAY(2) = 12
C              *****  NUMBER OF STRESSES (LEVEL 2)  *****
      IPLTAY(3) = 4
C             *****  NUMBER OF TOTAL SAVED (LEVELS 1, 2, AND 3)
      IPLTAY(4) = 20
C              *****  SAVE GEOMETRY FOR CONTOURS (0,NO  1,YES)
      IPLTAY(6) = 0
C

C      ****** PUT POSTDATA INFORMATION INTO POSTD ******
      DO 395 I=1,12
      POSTD(I) = FELEM(I)
 395  CONTINUE
      POSTD(13) = SDIRI
      POSTD(14) = SBENDI
      POSTD(15) = SDIRJ
      POSTD(16) = SBENDJ
      POSTD(17) = SMAXI
      POSTD(18) = SMINI
      POSTD(19) = SMAXJ
```

```
      POSTD(20) = SMINJ
C
 400  CONTINUE

C
C     ***** PUT PLTARY INFORMATION ONTO FILE 12 *******
      CALL SRPLT (IELEM,ITYP,NROW,MAT,100,2,U(1),NODES(1),XYZEQ(1,1),
     1 IPLTAY(1),POSTD(1))

 900  CALL PUTELD (IELNUM,EPAR(1),ERPAR(1),CON,SVR(1))
C     PUTELD RESTORES DATA BACK TO FILE3
      CALL TRACK( 15,'SR100 ')
      RETURN
      END
```

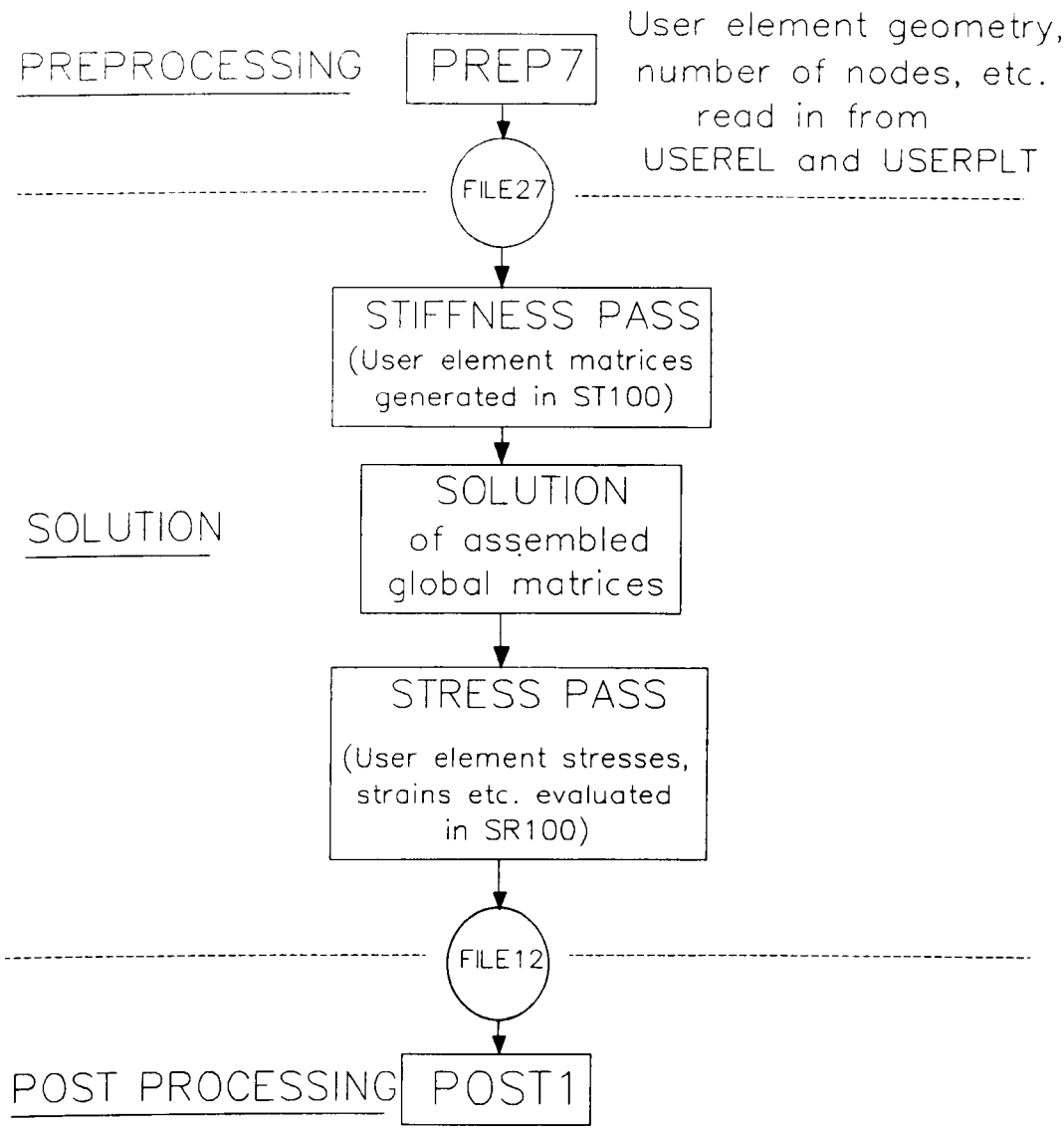


Figure A1.1. ANSYS finite element analysis procedure.

APPENDIX 2.

TRANSFORMATION OF ELEMENT CO-ORDINATE SYSTEMS.

A2.1 TRANSFORMATION OF ELEMENT CO-ORDINATE SYSTEMS.

Finite element matrices may be defined in terms of a local element co-ordinate system, however prior to assembly the individual element matrices and load vectors are required in a single global co-ordinate system. This is achieved by converting the element matrices from the local to the global system by use of a transformation matrix.

The transformation matrix is a matrix of direction cosines relating the axes of the local and global co-ordinate systems. Each type of finite element - spar, beam, plate etc. - has its own specific form of matrix, depending on the geometry and degrees of freedom of the element. The transformation matrix is defined in general by the equation:

$$\{\text{local vector}\} = TR\{\text{global vector}\} \qquad \text{(A2.1a)}$$

or conversely

$$\{\text{global vector}\} = TR^{-1}\{\text{local vector}\} \qquad \text{(A2.1b)}$$

Thus the local displacement vector is related to the global system by:

$$\{d_l\} = [TR]\{d_g\} \qquad \text{(A2.2)}$$

and similarly the local force vector by:

$$\{F_l\} = [TR]\{F_g\} \qquad \text{(A2.3)}$$

Local element matrices are also transformed into a global system by the use of the same transformation matrix. Consider, for example, the stiffness matrix. The element and global stiffness equations are

$$\{F_l\} = [K_l]\{d_l\} \qquad \text{(A2.4)}$$

$$\{F_g\} = [K_g]\{d_g\} \qquad \text{(A2.5)}$$

Inverting (A2.2), the global force vector in (A2.5) may be expressed as:

$$\{F_g\} = [TR]^{-1}\{F_l\}$$

Substituting for F_l from (A2.2) gives:

$$\{F_g\} = [TR]^{-1}[K_l]\{d_l\}$$

and equating with (A2.5):

$$[K_g]\{d_g\} = [TR]^{-1}[K_l]\{d_l\}$$

Substituting for d_l from (A2.2):

$$[K_g]\{d_g\} = [TR]^{-1}[K_l][TR]\{d_g\}$$

and equating coefficients of d_g, the global stiffness matrix is defined in terms of the local stiffness by:

$$[K_g] = [TR]^{-1}[K_l][TR]$$

This argument may be extended to include the other element and global matrices. Thus, in general:

$$[\text{global matrix}] = [TR]^{-1}[\text{element matrix}][TR] \qquad \text{(A2.6)}$$

A2.1.1 The General Rotation Matrix.

The transformation matrix is defined by examining the relationship between two general arbitrarily orientated co-ordinate systems.

Consider the systems, X,Y,Z and X′,Y′,Z′, which are initially coincident. Now let X′,Y′,Z′ be rotated about the origin O into a new orientation. The relationship between the systems may be defined in terms of the angles through which system X′,Y′,Z′ was rotated, but in practice it is simpler to use the direction cosines of the angles.

Let the first system have unit direction vectors **i,j,k,** and the second system unit direction vectors **u,v,w,** as shown in Figure A2.1. A typical direction cosine λ_{vi} is defined as the cosine of the angle between **v** and **i** (ϕ_{vi} in Figure A2.1).

It can be shown that the unit vectors **i,j,k,** have components with respect to **u,v,w** given by the equations:

$$i = \lambda_{ui}\,u + \lambda_{vi}\,v + \lambda_{wi}\,w \qquad \text{(A2.7a)}$$

$$j = \lambda_{uj}\,u + \lambda_{vj}\,v + \lambda_{wj}\,w \qquad \text{(A2.7b)}$$

$$k = \lambda_{uk}\,u + \lambda_{vk}\,v + \lambda_{wk}\,w \qquad \text{(A2.7c)}$$

These equations can be used to establish the general relationship between the two systems by considering the co-ordinates of a point P in space.

Let P have co-ordinates (x,y,z) in the original system, and (x′,y′,z′) in the rotated system. The position vector $\overrightarrow{OP}$ is thus

$\overrightarrow{OP} = x\,i + y\,j + z\,k$ In system XYZ

$\overrightarrow{OP} = x'\,u + y'\,v + z'\,w$ In system X′,Y′,Z′.

Equating these definition of $\overrightarrow{OP}$

gives:

$$x'\,u + y'\,v + z'\,w = x\,i + y\,j + z\,k \qquad \text{(A2.8)}$$

Substituting the equations for i, j and k given in (A2.7) into (A2.8) and rearranging gives:

$$\begin{aligned} x'\,u + y'\,v + z'\,w = &(\lambda_{ui}x + \lambda_{uj}y + \lambda_{uk}z)\,u \\ &+ (\lambda_{vi}x + \lambda_{vj}y + \lambda_{vk}z)\,v \\ &+ (\lambda_{wi}x + \lambda_{wj}y + \lambda_{wk}z)\,w \end{aligned}$$

Thus equating the coefficients of u , v and w defines the relationship between the two co-ordinate systems in the equations:

$$x' = \lambda_{ui}x + \lambda_{uj}y + \lambda_{uk}z \qquad \text{(A2.9a)}$$

$$y' = \lambda_{vi}x + \lambda_{jv}y + \lambda_{jw}z \qquad \text{(A2.9b)}$$

$$z' = \lambda_{ku}x + \lambda_{kv}y + \lambda_{kw}z \qquad \text{(A2.9c)}$$

This may be written in matrix form:

$$\{X'\} = [T]\{X\} \qquad \text{(A2.10)}$$

where:

$$X' = \begin{Bmatrix} x' \\ y' \\ z' \end{Bmatrix} \qquad X = \begin{Bmatrix} x \\ y \\ z \end{Bmatrix} \qquad [T] = \begin{bmatrix} \lambda_{ui} & \lambda_{uj} & \lambda_{uk} \\ \lambda_{vi} & \lambda_{vj} & \lambda_{vk} \\ \lambda_{wi} & \lambda_{wj} & \lambda_{wk} \end{bmatrix} \qquad \text{(A2.11)}$$

This relationship is valid for any vector quantity, and thus [T] is defined as the general vector rotation matrix between co-ordinate systems X′ and X.

For clarity the direction cosines are henceforth defined by axes labels x,y and z, such that, typically, λ_{xy} is the direction vector of the local x axis to the global y axis etc.

Thus the general rotation matrix is defined:

$$[T] = \begin{bmatrix} \lambda_{xx} & \lambda_{xy} & \lambda_{xz} \\ \lambda_{yx} & \lambda_{yy} & \lambda_{yz} \\ \lambda_{zx} & \lambda_{zy} & \lambda_{zz} \end{bmatrix} \qquad \text{(A2.12)}$$

A2.1.2 Orthogonality of Transformation Matrices.

The vector rotation matrix has the very useful property of orthogonality: its transpose is also its inverse. This is due to the relationship between the direction cosines of the mutually perpendicular axes. Typically:

$$\lambda_{xx}^2 + \lambda_{xy}^2 + \lambda_{xz}^2 = 1 \qquad \text{and} \qquad \lambda_{xx}\lambda_{yx} + \lambda_{xy}\lambda_{yy} + \lambda_{xz}\lambda_{yz} = 0$$

It can be shown that

$$[T][T]^T = [I]$$

and thus, by definition,

$$[T]^T = [T]^{-1}$$

Similarly, as the general element transformation matrix [TR] relates vector quantities, it may also be shown to be orthogonal. Thus:

$$[TR]^T = [TR]^{-1} \qquad \text{(A2.13)}$$

A2.1.3 Element to Global Transformation Equations.

The orthogonality of the general transformation matrix simplifies the general element-global vector and matrix relationships given in equations (A2.1) and (A2.6). From these equations and (A2.13) above we may define the following relationships:

$$\{d_g\} = [T]^T \{d_l\} \qquad \text{(A2.14a)}$$

$$\{F_g\} = [T]^T \{F_l\} \qquad \text{(A2.14b)}$$

$$[K_g] = [T]^T [K_l][T] \qquad \text{(A2.14c)}$$

$$[M_g] = [T]^T [M_l][T] \qquad \text{(A2.14d)}$$

Where d = displacement
F = Force
K = Stiffness
M = Mass

A2.1.4 Cylindrical Straight Beam Transformation Matrix.

Consider a 2-node cylindrical straight beam element, as shown in Figure A2.2. Each node has 6 degrees of freedom: 3 translations and 3 rotations.

The general rotation matrix [T] for a vector in three dimensional space is defined in (A2.12). In this case each point in space has 6 degrees of freedom. However, as the translations and rotations are independent orthogonal vector quantities, the 6 degree of freedom transformation matrix for an arbitrary point, or node, *i* is simply:

$$[T_i] = \begin{bmatrix} [T] & [0] \\ [0] & [T] \end{bmatrix}$$

Applying the same argument to node *j*, the full 12x12 element transformation is:

$$[TR] = \begin{bmatrix} [T_i] & [0] \\ [0] & [T_j] \end{bmatrix}$$

In the case of a straight beam the orientation of nodes *i* and *j* are the same. Thus the transformation matrix for node *j* is the same as that for *i*.

Therefore [TR] becomes:

$$[TR] = \begin{bmatrix} [T_l] & [0] \\ [0] & [T_l] \end{bmatrix} \qquad \text{(A2.15)}$$

Hence, in order to fully define [TR] only the 3x3 [T] matrix is required.

To define [T] nine direction cosines, as defined in (A2.12), are required. The simplest way to evaluate these is to consider the element originating from the origin of a system parallel to the global axes, as shown in Figure A2.3a.

The local x axis, x_l, is along the length of the element from node *i* to node *j*. Thus the direction cosines relating it to the global axes are:

$$\lambda_{xx} = \cos\alpha \qquad \lambda_{xy} = \cos\beta \qquad \lambda_{xz} = \cos\gamma$$

The length of the element is

$$L = \sqrt{(dx^2 + dy^2 + dz^2)} \qquad \text{where} \quad dx = X_{ig} - X_{jg}$$

$$dy = Y_{ig} - Y_{jg}$$

$$dz = Z_{ig} - Z_{jg}$$

and therefore:

$$\lambda_{xx} = \cos\alpha = \frac{dx}{L} \qquad \lambda_{xy} = \cos\beta = \frac{dy}{L} \qquad \lambda_{xz} = \cos\gamma = \frac{dz}{L} \qquad \text{(A2.16)}$$

Thus the orientation of the local x axis with respect to the global system is fully defined. Now consider the local y axis.

As the element is cylindrical the angle at which it is orientated about its x axis does not affect its behaviour. Therefore the local y axis, (or z if preferred), can be assigned an arbitrary orientation. To simplify the calculations let y_l lie parallel to the global X-Y plane. Thus at node *i* y is perpendicular to Z_g. Considering Figure A2.3(b):

$$\lambda_{yx} = \cos\alpha' \qquad \lambda_{yy} = \cos\beta'$$

$$\lambda_{yz} = \cos\gamma' = 0 \qquad \text{as } \gamma' = \frac{\pi}{2}$$

Viewing along the global Z axis shows

$$\cos\alpha' = \cos\left(\frac{\pi}{2} + \beta'\right) = -\sin\beta$$

Therefore

$$\lambda_{yx} = -\sin\beta' \qquad \lambda_{yy} = \cos\beta' \qquad \lambda_{yz} = 0$$

As dz=0, the projected length of the element in the X_g-Y_g plane is

$$Q = \sqrt{(dx^2 + dy^2)}$$

Thus the direction cosines of the local y axis can be defined:

$$\lambda_{yx} = \frac{-dy}{Q} \qquad \lambda_{yy} = \frac{dx}{Q} \qquad \lambda_{xz} = 0 \qquad \text{(A2.17)}$$

Finally, as the 3 local axes are orthogonal, the direction cosines of the local z axis may be defined by considering the vector product of the local x and y axes direction cosines. Thus:

$$\lambda_{zx} = \lambda_{xy}\lambda_{yz} - \lambda_{xz}\lambda_{yy} = -\frac{dxdz}{LQ}$$

$$\lambda_{zy} = \lambda_{xz}\lambda_{yx} - \lambda_{xx}\lambda_{yz} = -\frac{dydz}{LQ}$$

$$\lambda_{zz} = \lambda_{xz}\lambda_{yy} - \lambda_{xy}\lambda_{yx} = \frac{dx^2 + dz^2}{LQ} = \frac{Q}{L} \qquad \text{as } dx^2 + dz^2 = Q^2$$

Substituting the direction cosines into (A2.12), the circular beam 3x3 transformation equation is:

$$[T] = \begin{bmatrix} \frac{dx}{L} & \frac{dy}{L} & \frac{dz}{L} \\ \frac{-dy}{Q} & \frac{dx}{Q} & 0 \\ \frac{-dxdz}{LQ} & \frac{-dydz}{LQ} & \frac{Q}{L} \end{bmatrix} \qquad \text{(A2.18)}$$

A2.1.5 General Straight Beam Transformation Matrix

In the case of the circular beam the orientation of the local y axis to the global system was assigned an arbitrary value. However in the general case of a non axisymmetric beam, the beam properties are dependant on the orientation of the beam's y_1 axis. Therefore orientation of y_1 must be accounted for in the transformation of element matrices. Thus the transformation matrix must rotate the local y onto the global Y.

Consider the transformation of node *i* of a general beam as shown in Figure A2.4a.

The transformation matrix defined in equation (A2.18) rotates the local system onto the global system as shown in Figure A2.4b, (viewing from along the -X axis). The local x axis has been rotated onto the global X, but y_1 and z_1 lie at an angle ϕ to their global counterparts. Therefore, in order to align these axes, a further rotation is required. This is performed by a second rotation matrix $[\phi]$ defined from the general matrix (A2.12):

$$[\phi] = \begin{bmatrix} \lambda_{xx} & \lambda_{xy} & \lambda_{xz} \\ \lambda_{yx} & \lambda_{yy} & \lambda_{yz} \\ \lambda_{zx} & \lambda_{zy} & \lambda_{zz} \end{bmatrix} \qquad \text{(A2.19)}$$

where:

$$\lambda_{xx} = \cos 0 = 1 \qquad \lambda_{xy} = \lambda_{xz} = \lambda_{yx} = \cos\frac{\pi}{2} = 0$$

$$\lambda_{yy} = \cos\phi \qquad \lambda_{yz} = \cos\left(\frac{\pi}{2} - \phi\right) = \sin\phi$$

$$\lambda_{zx} = \lambda_{xy}\lambda_{yz} - \lambda_{xz}\lambda_{yy} = 0$$

$$\lambda_{zy} = \lambda_{yz}\lambda_{yx} - \lambda_{xx}\lambda_{yz} = \sin\phi$$

$$\lambda_{zz} = \lambda_{xz}\lambda_{yy} - \lambda_{xy}\lambda_{yx} = \cos\phi$$

The general straight beam transformation matrix is obtained by applying ϕ to the cylindrical beam transformation matrix:

$$[T] = [\phi][T_{\text{cyl beam}}]$$

Thus the 3x3 general straight beam equation is found to be:

$$[T] = \begin{bmatrix} \frac{dx}{L} & \frac{dy}{L} & \frac{dz}{L} \\ \left(\frac{-dy}{Q}\cos\phi - \frac{dxdz}{LQ}\sin\phi\right) & \left(\frac{dx}{Q}\cos\phi - \frac{dydz}{LQ}\sin\phi\right) & \left(\frac{Q}{L}\sin\phi\right) \\ \left(\frac{dy}{Q}\sin\phi - \frac{dxdz}{LQ}\cos\phi\right) & \left(\frac{-dx}{Q}\sin\phi - \frac{dydz}{LQ}\cos\phi\right) & \left(\frac{Q}{L}\cos\phi\right) \end{bmatrix} \quad \text{(A2.20)}$$

A2.1.6 Curved Beam Transformation Matrix.

Unlike a straight beam, the end nodes of a curved member do not share a common orientation with respect to the global axes. Therefore the general form of a curved beam transformation matrix is:

$$[TR] = \begin{bmatrix} [T_i] & [0] \\ [0] & [T_j] \end{bmatrix} \qquad where \quad [T_j] \neq [T_i]$$

The nodal transformation matrices $[T_i]$ and $[T_j]$ may be derived from the straight beam equations by considering the end nodes to be joined by a general straight beam.

Consider node *i* of the curved beam shown in Figure A2.5.

The bend lies in the local x_l z_l plane. Due to the curvature of the beam, its local x and z axes lie at an angle $\frac{\theta}{2}$ to the local axes of an imaginary straight beam joining *i* to *j*. By applying a rotation matrix $[\Theta]$ to the real local axes, they can be aligned with the imaginary system. Consequently the transformation from local to global co-ordinates is completed by applying the general straight beam transformation matrix to the Θ rotated matrix.

Applying the general rotation matrix equation (A2.12) to node *i*, the node *i* curved to straight beam rotation matrix Θ_i is found to be:

$$[\Theta_i] = \begin{bmatrix} \lambda_{xx} & \lambda_{xy} & \lambda_{xz} \\ \lambda_{yx} & \lambda_{yy} & \lambda_{yz} \\ \lambda_{zx} & \lambda_{zy} & \lambda_{zz} \end{bmatrix}$$

where:

$$\lambda_{xx} = \cos\frac{\theta}{2} \qquad \lambda_{xy} = \cos\frac{\pi}{2} = 0$$

$$\lambda_{xz} = \cos\left(\frac{\pi}{2} + \frac{\theta}{2}\right) = -\sin\frac{\theta}{2} \qquad \lambda_{yx} = \cos\frac{\pi}{2} = 0$$

$$\lambda_{yy} = \cos 0 = 1 \qquad \lambda_{yz} = \cos\frac{\pi}{2} = 0$$

$$\lambda_{zx} = \lambda_{xy}\lambda_{yz} - \lambda_{xz}\lambda_{yy} = \sin\frac{\theta}{2} \qquad \lambda_{zy} = \lambda_{yz}\lambda_{yx} - \lambda_{xx}\lambda_{yz} = 0$$

$$\lambda_{zz} = \lambda_{xz}\lambda_{yy} - \lambda_{xy}\lambda_{yx} = \cos\frac{\theta}{2}$$

Thus:

$$[\Theta_i] = \begin{bmatrix} \left(\cos\frac{\theta}{2}\right) & 0 & \left(-\sin\frac{\theta}{2}\right) \\ 0 & 1 & 0 \\ \left(\sin\frac{\theta}{2}\right) & 0 & \left(\cos\frac{\theta}{2}\right) \end{bmatrix}$$

As the beam is circularly curved, repeating the procedure for node *j* shows:

$$[\Theta_j] = [\Theta_i]^T$$

Therefore the nodal transformation equations are:

$$[T_i] = [\Theta][T_{is}] \qquad \text{(A2.21a)}$$

$$[T_j] = [\Theta]^T[T_{is}] \qquad \text{(A2.21b)}$$

Where

$[T_{is}]$ = Straight beam node i 6x6 TR matrix

and

$$[\Theta] = \begin{bmatrix} [\Theta_i] & [0] \\ [0] & [\Theta_i] \end{bmatrix}$$

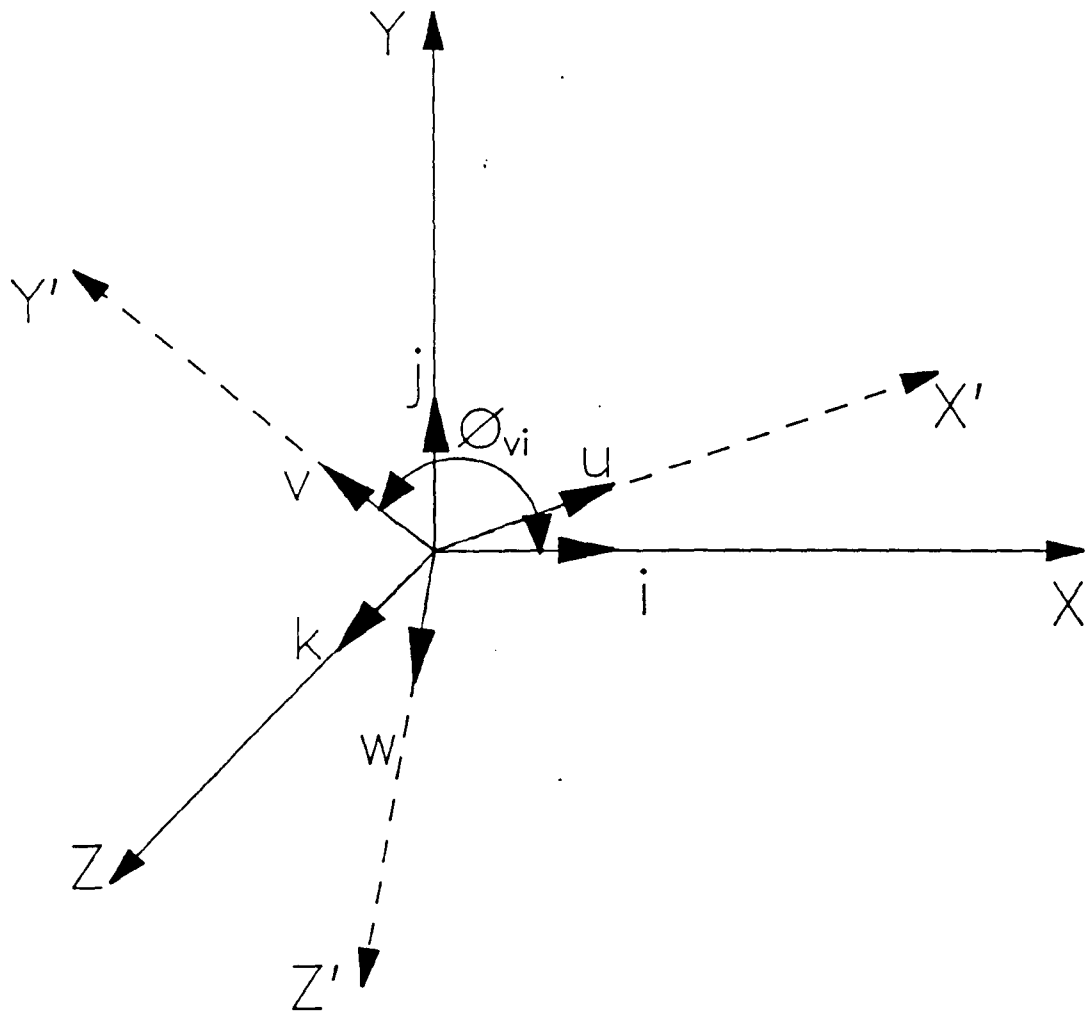


Figure A2.1 Coordinate systems X,Y,Z and X',Y',Z'.

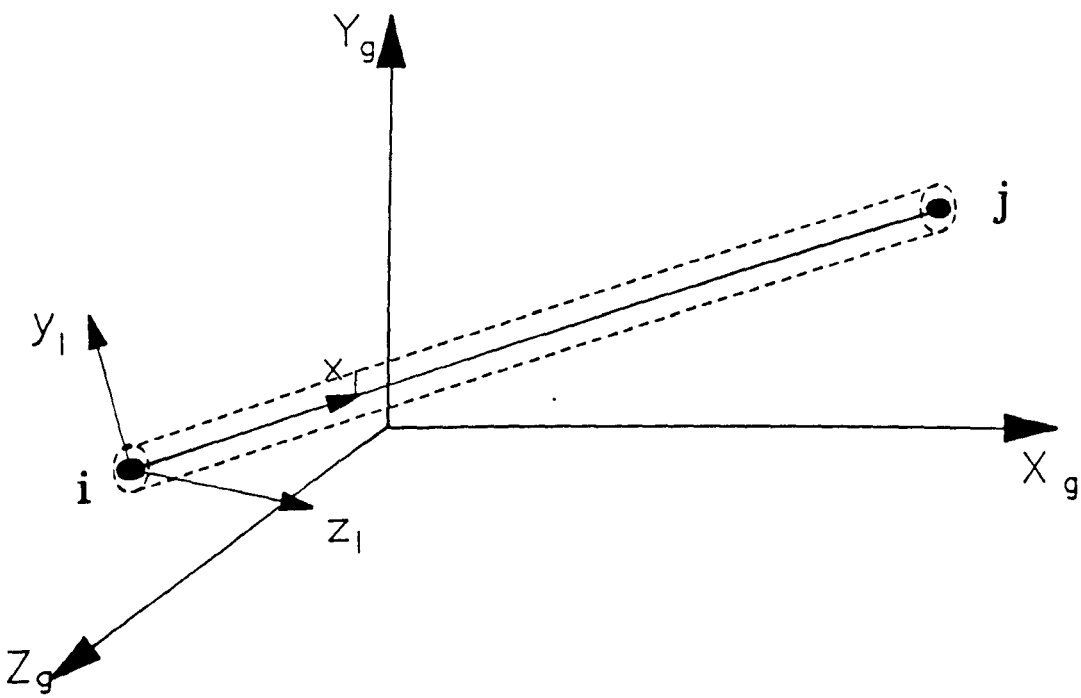


Figure A2.2 Cylindrical straight beam element coordinate system.

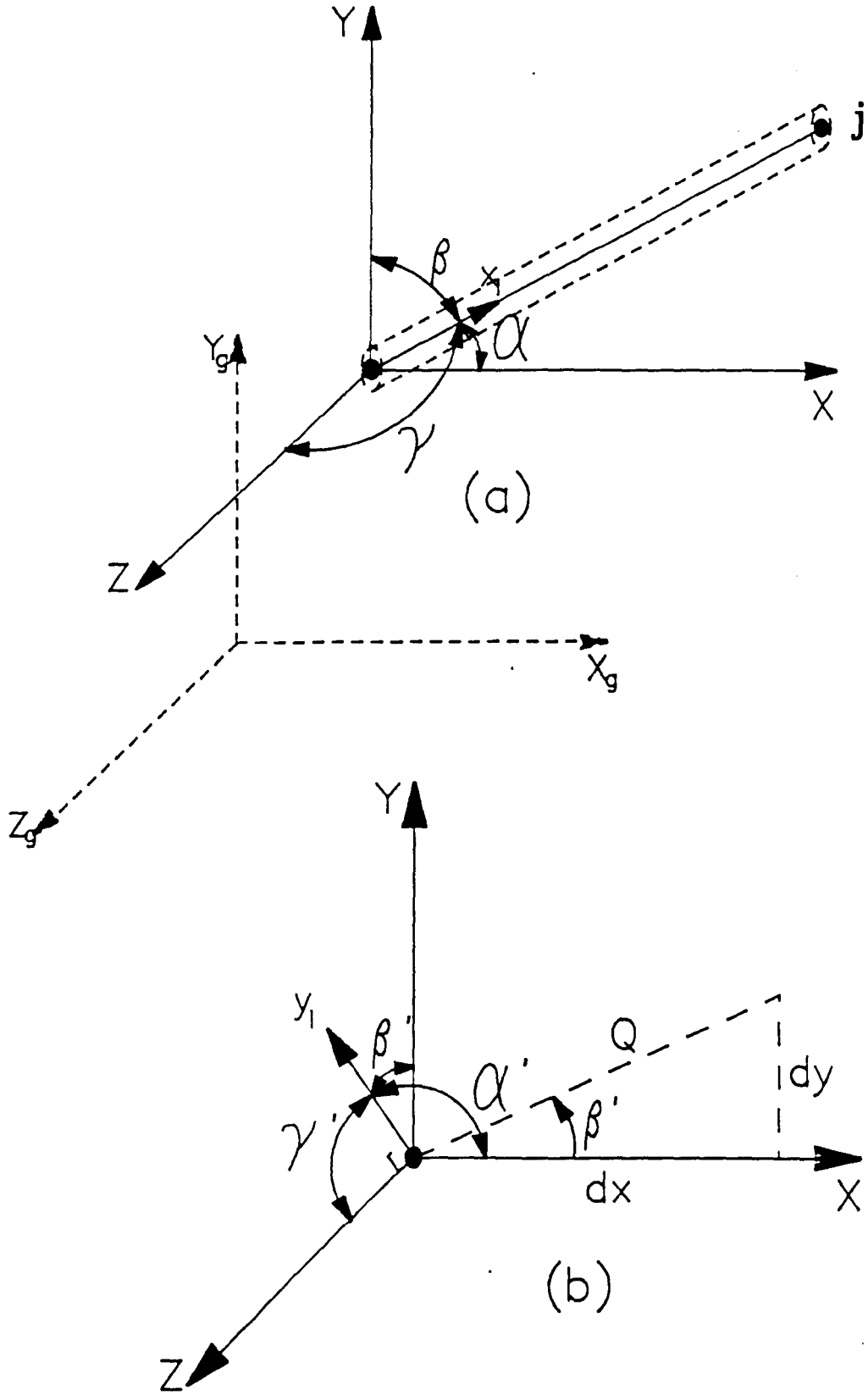


Figure A2.3 Cylindrical straight beam element orientation with respect to the global system.

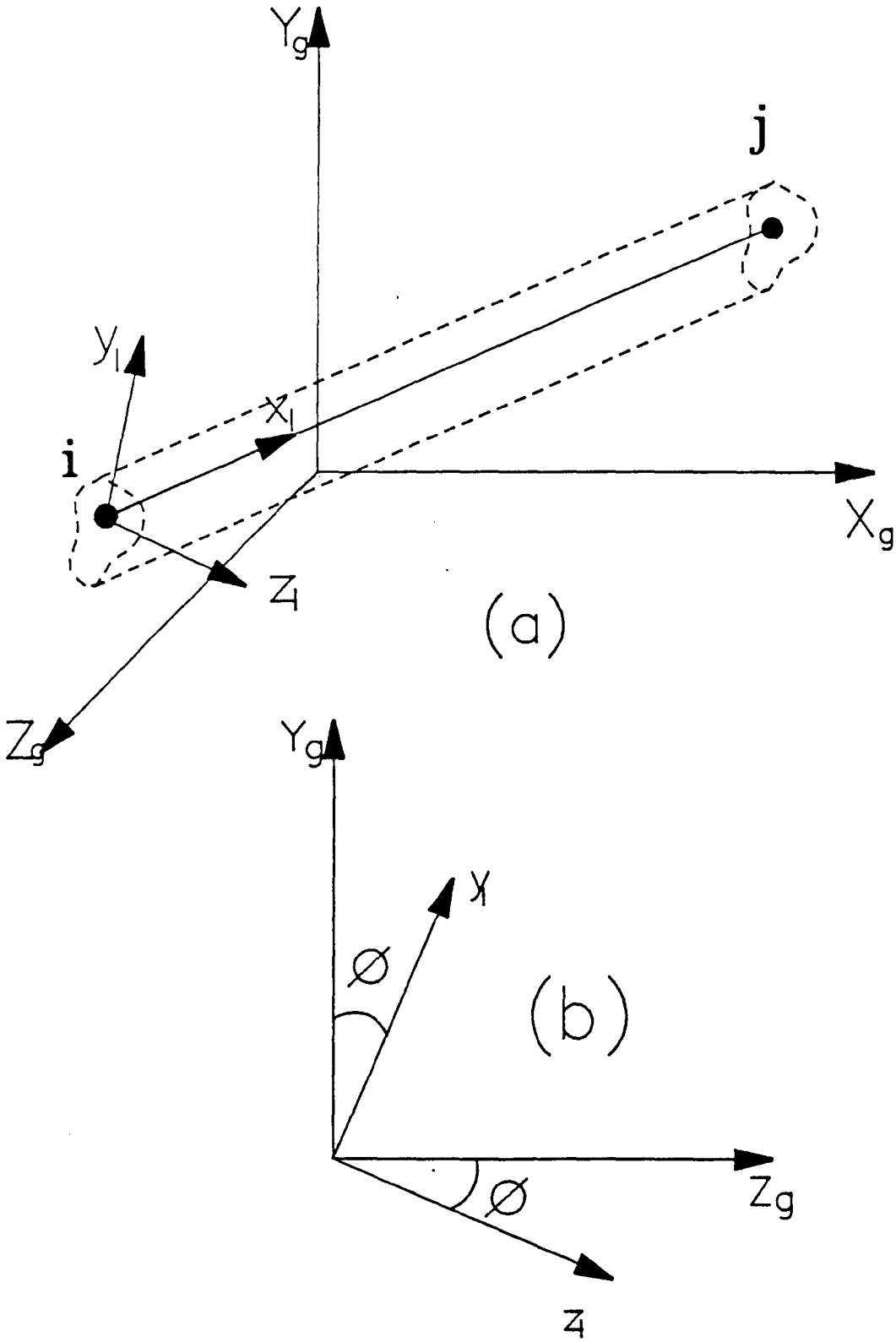


Figure A2.4 General straight beam coordinate system.

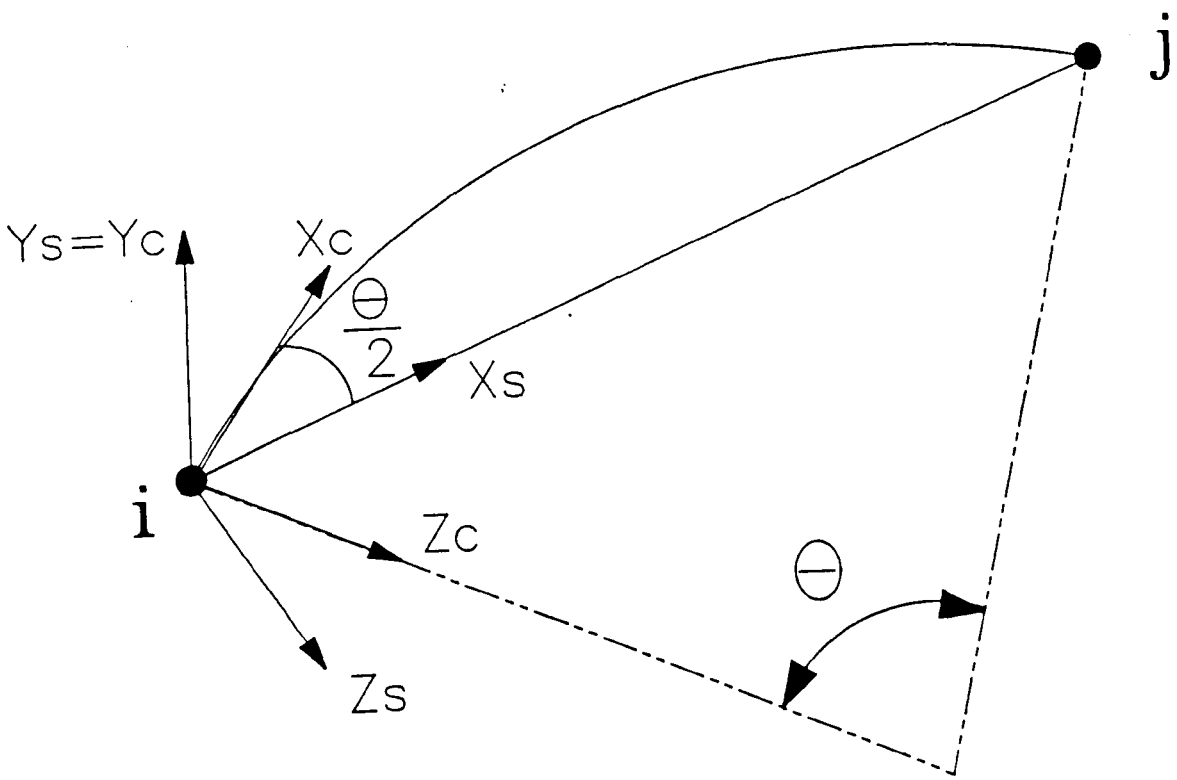


Figure A2.5 Curved beam to straight beam coordinate system transformation.

A3.1 PROGRAM BEL1PROG.FOR

```
      PROGRAM BEL1PROG
C
C       FLEXIBILITY AND STRESS FACTOR PROGRAM FOR
C             SEMI-TOROIDAL BELLOWS
C
C     WRITTEN BY    DONALD MACKENZIE, MARCH 1989
C
C     This program evaluates the flexibility and stress factors of
C     semi-toroidal bellows based on the Laupa and Weil/Findlay
C     and Spence analysis.  Six Fourier terms used in the shape
C     function.
C
      INTEGER I,NEQ,J,K,L,K1,NEQ1,LOOP
      DOUBLE PRECISION AMAT(6,6),BVEC(6),CVEC(6),CVAL,U(6),
     & LAMBDA,LS4,FLEX,DPPI,BMAT(2,6),D(2,2),SIG(2),EP(2),NU
C
C     Input routine: Interactive.
C
      PRINT *,'INPUT PIPE RADIUS'
      READ (6,*) TORR
      PRINT *,'PIPE RAD. = ',TORR
      PRINT *,'INPUT CONVOLUTION RADIUS'
      READ (6,*) R
      PRINT *,'CONV. RAD = ', R
      PRINT *,'INPUT THICKNESS'
      READ (6,*) THICK
      PRINT *,' THICK. = ',THICK
      PRINT *,'INPUT ELASTIC MOD.'
      READ (6,*) EX
      PRINT *,' E  = ',EX
      PRINT *,'INPUT POISSON RATIO'
      READ (6,*) NU
      PRINT *,'NU  = ',NU
C
      ZETA = THICK/2
      LAMBDA=TORR*THICK/R**2
      LS4=(LAMBDA**2.0)*4.0
C
      PRINT *,'LAMBDA  = ',LAMBDA
C
      AMAT(1,1) = 0.1309D0+(1.178097D0/LS4)
      AMAT(1,2) = 0.111111D0+(1.659685D0/LS4)
      AMAT(1,3) = -0.044444D0+(0.25418D0/LS4)
      AMAT(1,4) = 0.028571D0+(0.001814D0/LS4)
      AMAT(1,5) = -0.021164D0-(0.000641D0/LS4)
      AMAT(1,6) = -0.016835D0+(0.000283D0/LS4)
      AMAT(2,2) = 0.523599D0+(2.871682D0/LS4)
      AMAT(2,3) = 0.213333D0/LS4
      AMAT(2,4) = 0.016327D0/LS4
      AMAT(2,5) = 0.014109D0/LS4
      AMAT(2,6) = -0.012489D0/LS4
      AMAT(3,3) = 2.094395D0+(0.173873D0/LS4)
      AMAT(3,4) = 0.059259D0/LS4
      AMAT(3,5) = 0.017316D0/LS4
      AMAT(3,6) = 0.0D0
      AMAT(4,4) = 4.712389D0+(0.070646D0/LS4)
      AMAT(4,5) = 0.028132D0/LS4
      AMAT(4,6) = 0.011005D0/LS4
      AMAT(5,5) = 8.37758+(0.38751/LS4)
      AMAT(5,6) = 0.016433D0/LS4
      AMAT(6,6) = 13.089969D0+(0.024577D0/LS4)
C
      DPPI=3.14159265358979 3D0
      BVEC(1) = DPPI/4
      BVEC(2) = 2.0D0/3.0D0
      BVEC(3) = -4.0D0/15.0D0
      BVEC(4) = 6.0D0/35.0D0
      BVEC(5) = -8.0D0/63.0D0
      BVEC(6) = 10.0D0/99.0D0
C
C     SOLVE EQN {A} {C} = {B} FOR {C}
C     SOLUTION BY THE GAUSS ELIM. METHOD.
C
C     INITIALLY EQUATE BVEC TO CVEC
C
      DO 1 I=1,6
    1 CVEC(I)=BVEC(I)
C
      NEQ=6
C
      NEQ1=NEQ-1
      DO 100 K=1,NEQ1
      CVAL=AMAT(K,K)
      K1=K+1
      DO 11 J=K1,NEQ
   11 AMAT(J,K)=AMAT(K,J)
      IF (ABS(CVAL)-1E-6)4,4,7
    4 WRITE (6,5) K
    5 FORMAT (' **** SINGULARITY IN ROW',I5)
      GO TO 300
C
C     DIV ROW BY DIAG COEFF
C
    7 DO 8 J=K1,NEQ
    8 AMAT(K,J)=AMAT(K,J)/CVAL
      CVEC(K)=CVEC(K)/CVAL
C
C     ELIMINATE ROW UNKNOWN X(K) FROM ROW I
C
      DO 10 I=K1,NEQ
      CVAL=AMAT(I,K)
      DO 9 J=1,NEQ
    9 AMAT(I,J)=AMAT(I,J)-CVAL*AMAT(K,J)
   10 CVEC(I)=CVEC(I)-CVAL*CVEC(K)
  100 CONTINUE
C
C     COMPUTE LAST UK
C
      IF(ABS(AMAT(NEQ,NEQ))-1E-6)1,1,101
  101 CVEC(NEQ)=CVEC(NEQ)/AMAT(NEQ,NEQ)
C
C     APPLY BACKSUB TO COMPUTE REMAINING UKS
C
      DO 200 L=1,NEQ1
      K=NEQ-L
      K1=K+1
      DO 200 J=K1,NEQ
  200 CVEC(K)=CVEC(K)-AMAT(K,J)*CVEC(J)
C
C     EVALUATE FLEXIBILITY FACTOR
C
      FLEX=  (BVEC(1)*CVEC(1)+BVEC(2)*CVEC(2)+BVEC(3)*CVEC(3)
     1 +BVEC(4)*CVEC(4)+BVEC(5)*CVEC(5)+BVEC(6)*CVEC(6))/2
C
      PRINT *,'FLEX  = ',FLEX
```

APPENDIX 3.

BELLOWS FORMULATION FORTRAN PROGRAMS.

```
      PRINT *,'OUTER SURFACE'
      PRINT *,' POSN.   SIG AXIAL   SIG HOOP'
C
      POS = -5.0
      DO 250 ILOC = 1,19
      POS = POS + 5.0
      POS = POS*3.1415927/180.
      PC1 = COS(POS)
      PC2 = COS(POS*2.)
      PC3 = COS(POS*3.)
      PC4 = COS(POS*4.)
      PC5 = COS(POS*5.)
      PC6 = COS(POS*6.)
      PC7 = COS(POS*7.)
      PC8 = COS(POS*8.)
      PC9 = COS(POS*9.)
      PC10 = COS(POS*10.)
      PC11 = COS(POS*11.)
C
      BMAT(1,1) = -ZETA*PC1/R
      BMAT(1,2) = -ZETA*2.*PC2/R
      BMAT(1,3) = -ZETA*4.*PC4/R
      BMAT(1,4) = -ZETA*6.*PC6/R
      BMAT(1,5) = -ZETA*8.*PC8/R
      BMAT(1,6) = -ZETA*10.*PC10/R
      BMAT(2,1) = -PC1*PC1*R*.5/TORR
      BMAT(2,2) = -(PC3/3.+1.)*R*.5/TORR
      BMAT(2,3) = -(PC5/5.+PC2/3.)*R*.5/TORR
      BMAT(2,4) = -(PC7/7.+PC4/5.)*R*.5/TORR
      BMAT(2,5) = -(PC9/9.+PC6/7.)*R*.5/TORR
      BMAT(2,6) = -(PC11/11.+PC8/9.)*R*.5/TORR
C
      DCON = EX/(1.-NU**2)
      D(1,1) = DCON
      D(1,2) = NU*DCON
      D(2,1) = NU*DCON
      D(2,2) = DCON
C
C     LET ARBITRARY AXIAL FORCE = 1000
      DO 240 I=1,6
      U(I) = CVEC(I)*1000.*(1.-NU**2)*R**2/
     & (EX*3.14159*TORR*THICK**3)
  240 CONTINUE
C     {EP} = [B] {C}
      EP(1) = BMAT(1,1)*U(1)+BMAT(1,2)*U(2)+BMAT(1,3)*U(3)
     1 +BMAT(1,4)*U(4)+BMAT(1,5)*U(5)+BMAT(1,6)*U(6)
      EP(2) = BMAT(2,1)*U(1)+BMAT(2,2)*U(2)+BMAT(2,3)*U(3)
     1 +BMAT(2,4)*U(4)+BMAT(2,5)*U(5)+BMAT(2,6)*U(6)
C
C     EVALUATE STRESSES {SIG} = [D] {EP}
      SIG(1) = (D(1,1)*EP(1)+D(1,2)*EP(2))
      SIG(2) = (D(2,1)*EP(1)+D(2,2)*EP(2))
C
C     NORMALISE STRESSES BY DIVIDING BY SIGBO
      SIGBO = 3.0*R*1000.0/(3.1415927*TORR*THICK**2)
      SIG(1) = SIG(1)/SIGBO
      SIG(2) = SIG(2)/SIGBO
      POS = POS * 180/3.1415729
      WRITE (6,'(9F10.2)') POS,SIG(1),SIG(2)
  250 CONTINUE
C
  300 CONTINUE
      END
```


A3.2 PROGRAM BEL2FLEX.FOR

```
      PROGRAM BEL2FLEX
C
C     This program evaluates the flexibility factor of a
C     semi-toroidal bellows using a finite element
C     approach.  Based on Trigonometric (Fourier) interpolation
C     of radial displacement.
C     The bellows stiffness matrix is evaluated using gaussian
C     quadrature, 3 points through thickness and 5 along the arc.
C
C     Donald Mackenzie, March 1989.
C
C     NOTE. In this program:
C          THETA = MERIDIONAL (AXIAL) DIRECTION
C          PHI = CIRCUMFERENTIAL (AXISYMM.) DIRECTION.
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DOUBLE PRECISION WI(3),AI(3),WJ(5),AJ(5),A(2,6),
     1 P(3,3),STIFF(3,3),NUXY,G(6,3),DET1,
     2 STBEL(1,1),BMAT(2,3),BT(3,2),D(2,2),DB(2,3)
C
C     INPUT ROUTINE: Interactive.
C
C     MATERIAL PROPERTIES: Young's modulus, Poisson's ratio.
C
C     BELLOWS GEOMETRY: R = convolution radius, TORR = mean
C     radius from symmetric axis, THICK = wall thickness.
C
      PRINT *,'INPUT PIPE RADIUS'
      READ (6,*) TORR
      PRINT *,'PIPE RAD. =',TORR
      PRINT *,'INPUT CONVOLUTION RADIUS'
      READ (6,*) R
      PRINT *,'CONV. RAD =', R
      PRINT *,'INPUT THICKNESS'
      READ (6,*) THICK
      PRINT *,' THICK. =',THICK
      PRINT *,'INPUT ELASTIC MOD.'
      READ (6,*) EX
      PRINT *,' E  =',EX
      PRINT *,'INPUT POISSON RATIO'
      READ (6,*) NUXY
      PRINT *,'NU =',NUXY
C
C     STIFFNESS MATRIX CONSTANT
      P4 = 3.1415927D0/4.0D0
      CONST = 2.0D0*EX*P4*4.0D0*R*TORR/(1-NUXY**2)
C
C     CONSTITUTIVE MATRIX [D]*(1-NUXY**2)/EX
      D(1,1) = 1.0
      D(1,2) = NUXY
      D(2,1) = NUXY
      D(2,2) = 1.0
C
C     NUMERICAL INTEGRATION PARAMETERS.
C     WEIGHTS
      W1 = 0.55555555555556D0
      W2 = 0.88888888888889D0
      W3 = W1
C     ABSCISSA
      A3 = 0.774596669241483D0
      A2 = 0.0D0
```

```
      A1 = -A3
C
      WT1 = 0.236926885056189
      WT2 = 0.478628670499366
      WT3 = 0.568888888888889
      WT4 = WT2
      WT5 = WT1
C
      B1 = -0.906179845938664
      B2 = -0.538469310105683
      B3 = 0.0D0
      B4 = -B2
      B5 = -B1
C     APPLY ACTUAL INTEGRATION LIMITS:
C     SUBSCRIPT I REFERS TO H, J TO THETA
      T = THICK/2
      WI(1) = T*W1
      WI(2) = T*W2
      WI(3) = WI(1)
      AI(1) = T*A1
      AI(2) = T*A2
      AI(3) = T*A3
C
      WJ(1) = P4*WT1
      WJ(2) = P4*WT2
      WJ(3) = P4*WT3
      WJ(4) = WJ(2)
      WJ(5) = WJ(1)
      AJ(1) = P4*(1.0D0+B1)
      AJ(2) = P4*(1.0D0+B2)
      AJ(3) = P4*(1.0D0+B3)
      AJ(4) = P4*(1.0D0+B4)
      AJ(5) = P4*(1.0D0+B5)
C
C
C     ZERO STIFFNESS AND P (ATDA) MATRICES
      DO 30 I=1,6
      DO 30 J=1,6
      STIFF(I,J)=0.0D0
      P(I,J) = 0.0D0
 30   CONTINUE
C
C     START OF NUMERICAL INTEGRATION (DOUBLE) LOOP
      DO 100 I=1,3
      DO 100 J=1,5
C
C     EVALUATE B MATRIX
C
C     MATRIX [A]: Strain-displacement "shape".
C     EVALUATE ELEMENTS OF [A]
      HR = AI(I)/(R**2)
      THETA = AJ(J)
      ST = SIN(THETA)
      CT = COS(THETA)
      ST2 = SIN(2.*THETA)
      ST3 = SIN(3.*THETA)
      ST4 = SIN(4.*THETA)
      ST5 = SIN(5.*THETA)
      CT2 = COS(2.*THETA)
      CT3 = COS(3.*THETA)
      CT4 = COS(4.*THETA)
      CT5 = COS(5.*THETA)
C
      A(1,1) = -HR
      A(1,2) = 0.
      A(1,3) = -3.0*HR*ST2
      A(1,4) = -8.0*HR*ST3
      A(1,5) = -15.0*HR*ST4
      A(1,6) = 0.
      A(2,1) = (ST-THETA*CT)/TORR
      A(2,2) = (-ST*ST-CT*CT)/TORR
      A(2,3) = -(ST2*ST + .5*CT2*CT)/TORR
      A(2,4) = -(ST3*ST + .3333333333334*CT3*CT)/TORR
      A(2,5) = -(ST4*ST + .25*CT4*CT)/TORR
      A(2,6) = CT/TORR
C
C     MATRIX [G]: Strain displacement "magnitude", with actual
C     boundary conditions applied.
C
      G(1,1) = 1.0
      G(1,2) = 0.0
      G(1,3) = 0.0
      G(2,1) = 3.46238898
      G(2,2) = 3.0/4.0*R
      G(2,3) = 5.0/4.0
      G(3,1) = -2.71238898
      G(3,2) = -R
      G(3,3) = -2.0
      G(4,1) = 2.46238898
      G(4,2) = 3.0/4.0*R
      G(4,3) = 9.0/4.0
      G(5,1) = -1.35619449
      G(5,2) = -1.0/2.0*R
      G(5,3) = -1.0
      G(6,1) = 2.587942194
      G(6,2) = 3.0/8.0*R
      G(6,3) = 3.0/4.0
C
C     EVALUATE [BMAT] = [A] [G]
      DO 40 IT=1,2
      DO 40 JT=1,3
      BMAT(IT,JT) = 0.0D0
      DO 40 KT=1,6
 40   BMAT(IT,JT) = BMAT(IT,JT) + A(IT,KT)*G(KT,JT)
C
C     EVALUATE [BT] = TRANSPOSE OF [B]
      DO 47 IT=1,2
      DO 47 JT=1,3
 47   BT(JT,IT)=BMAT(IT,JT)
C
C     EVALUATE [P] = [B]T [D] [B] MATRIX
C
      DO 50 IT=1,2
      DO 50 JT=1,3
      DB(IT,JT)=0.0
      DO 50 KT=1,2
 50   DB(IT,JT) = DB(IT,JT)+D(IT,KT)*BMAT(KT,JT)
C
      DO 60 IT=1,3
      DO 60 JT=1,3
      P(IT,JT)=0.0
      DO 60 KT=1,2
 60   P(IT,JT) = P(IT,JT)+BT(IT,KT)*DB(KT,JT)
C
C     EVALUATE 3x3 STIFFNESS MATRIX.
C     [STIFF] = [STIFF] PREVIOUS + CONST*WEIGHTS * [P]
      DO 70 K=1,3
      DO 70 L=1,3
```

```
 70  STIFF(K,L) = STIFF(K,L)+CONST*WI(I)*WJ(J)*P(K,L)
C
 100 CONTINUE
C    END OF NUMERICAL INTEGRATION LOOP
C
C    APPLY STATIC CONDENSATION: Evaluate the bellows axial stiffness.
C
     CALL REDUCE (STIFF,STBEL)
C
C    EVALUATE FLEXIBILITY FACTOR
C
C    DISPLACEMENT/ UNIT FORCE OF EQUIVALENT STRAIGHT PIPE.
     DELTA = R/(2*3.1415937*TORR*THICK*EX)
     DELTA = DELTA * (R/(THICK/2))**2
     DELTA = DELTA * (1.0 - NUXY**2)
C    FLEXIBILITY FACTOR = FLEX
     FLEX = 1/(STBEL(1,1)*DELTA)
C    WRITE RESULTS TO SCREEN
     PRINT *,'LAMBDA =',TORR*THICK/R**2
     PRINT *,'STIFF =',STBEL(1,1)
     PRINT *,'FLEX =',FLEX
C
     END
C
```

A3.2.1 SUBROUTINE REDUCE

```
     SUBROUTINE REDUCE (SE,RED)
C
C    This routine applies the static condensation procedure
C    to reduce the 4x4 bellows stiffness matrix to the single
C    value of axial stiffness.
C
C    FULL, SYMMETRIC STIFFNESS MATRIX SE.
C    CONDENSATION OPERATIONS ON LOWER TRIANGLE OF SE.
C    SIZE OF FULL MATRIX = NSIZE
C    NUMBER OF DOF TO BE REDUCED = NUM.  DOF TO BE REDUCED
C    STORED IN LAST NUM DOF.
C
     IMPLICIT DOUBLE PRECISION (A-H,O-Z)
     DOUBLE PRECISION SE(3,3),RED(1,1)
C
     NSIZE = 3
     NUM = 2
C    CONDENSATION OF LOWER TRIANGLE OF SE.
     DO 30 K=1,NUM
     LL = NSIZE - K
     KK = LL + 1
     DO 20 L=1,LL
     IF (SE(KK,L) .EQ. 0.0) GO TO 20
     DUM = SE(KK,L)/SE(KK,KK)
     DO 10 M=1,LL
 10  SE(L,M) = SE(L,M) - SE(KK,M)*DUM
 20  CONTINUE
 30  CONTINUE
C    FILL IN THE UPPER TRIANGLE BY SYMMETRY.
     DO 40 K=1,LL
     DO 40 L=1,K
 40  SE(L,K) = SE(K,L)
C
C    EQUATE FIRST SE(1,1) TO RED
C    RED IS THE REDUCED MATRIX
     RED(1,1) = SE(1,1)
     RETURN
     END
C
```

A3.3 PROGRAM BEL2STR.FOR

```
     PROGRAM BEL2STR
C
C    This program evaluates the normalised stress distribution
C    in a semi-toroidal bellows using a finite element approach.
C    Radial adisplacement interpolated by trig series.
C    The bellows stiffness matrix is evaluated using gaussian
C    quadrature, 3 points through thickness and 5 along the arc.
C
C    Written by Donald Mackenzie, March 1989.
C
C    NOTE. In this program:
C         THETA = MERIDIONAL (AXIAL) DIRECTION
C         PHI = CIRCUMFERENTIAL (AXISYMM.) DIRECTION.

     IMPLICIT DOUBLE PRECISION (A-H,O-Z)
     DOUBLE PRECISION WI(3),AI(3),WJ(5),AJ(5),A(2,6),
    1 P(6,6),STIFF(6,6),NUXY,C(6,12),G(6,3),DET1,ST3(3,3),
    2 SG(6,3),GT(3,6),STBEL(1,1),U(3),BMAT(2,3),EP(2),SIG(2)
C
C    Input routine: Interactive.
C
C    MATERIAL PROPERTIES: Young's modulus, Poisson's ratio.
C    BELLOWS GEOMETRY: R = convolution radius, TORR = mean
C    radius from symmetric axis, THICK = wall thickness.
C    Here set values of geometry parameters are given.
C    Alternatively an input procedure or loop could be used.
C
     PRINT *,'INPUT PIPE RADIUS'
     READ (6,*) TORR
     PRINT *,'PIPE RAD. =',TORR
     PRINT *,'INPUT CONVOLUTION RADIUS'
     READ (6,*) R
     PRINT *,'CONV. RAD =', R
     PRINT *,'INPUT THICKNESS'
     READ (6,*) THICK
     PRINT *,' THICK. =',THICK
     PRINT *,'INPUT ELASTIC MOD.'
     READ (6,*) EX
     PRINT *,' E  =',EX
     PRINT *,'INPUT POISSON RATIO'
     READ (6,*) NUXY
     PRINT *,'NU =',NUXY
C
C    MAT PROPS
     EX = 210.0E3
     NUXY = 0.3D0
     V = NUXY
C    WEIGHTS
     W1 = 0.555555555555556D0
     W2 = 0.888888888888889D0
     W3 = W1
C    ABSCISSA
     A3 = 0.774596669241483D0
     A2 = 0.0D0
     A1 = -A3
C
```

```
      WT1 = 0.236926885056189
      WT2 = 0.478628670499366
      WT3 = 0.568888888888889
      WT4 = WT2
      WT5 = WT1
C
      B1 = -0.906179845938664
      B2 = -0.538469310105683
      B3 = 0.0D0
      B4 = -B2
      B5 = -B1
C     ACTUAL INTEGRATION LIMITS:
C     THRO' THICKNESS H = -THICK/2 TO THICK/2.
C     ALONG MERIDIAN  THETA = 0 TO PI/4
C     SUBSCRIPT I REFERS TO H, J TO THETA
      T = THICK/2
      WI(1) = T*W1
      WI(2) = T*W2
      WI(3) = WI(1)
      AI(1) = T*A1
      AI(2) = T*A2
      AI(3) = T*A3
C
      P4 = 3.1415927D0/4.0D0
      WJ(1) = P4*WT1
      WJ(2) = P4*WT2
      WJ(3) = P4*WT3
      WJ(4) = WJ(2)
      WJ(5) = WJ(1)
      AJ(1) = P4*(1.0D0+B1)
      AJ(2) = P4*(1.0D0+B2)
      AJ(3) = P4*(1.0D0+B3)
      AJ(4) = P4*(1.0D0+B4)
      AJ(5) = P4*(1.0D0+B5)
C
C     STIFFNESS MATRIX CONSTANT
      CONST = 2.0D0*EX*P4*4.0D0*R*TORR/(1-NUXY**2)
C
C     ZERO STIFFNESS AND P (ATDA) MATRICES
      DO 30 I=1,6
      DO 30 J=1,6
      STIFF(I,J)=0.0D0
      P(I,J) = 0.0D0
 30   CONTINUE
C     SET UP INTEGRATION DOUBLE LOOP
      DO 100 I=1,3
      DO 100 J=1,5
C     EVALUATE ELEMENTS OF [A]
      HR = AI(I)/(R**2)
      THETA = AJ(J)
      ST = SIN(THETA)
      CT = COS(THETA)
      ST2 = SIN(2.*THETA)
      ST3 = SIN(3.*THETA)
      ST4 = SIN(4.*THETA)
      ST5 = SIN(5.*THETA)
      CT2 = COS(2.*THETA)
      CT3 = COS(3.*THETA)
      CT4 = COS(4.*THETA)
      CT5 = COS(5.*THETA)
C
      A(1,1) = -HR
      A(1,2) = 0.
      A(1,3) = -3.0*HR*ST2
      A(1,4) = -8.0*HR*ST3
      A(1,5) = -15.0*HR*ST4
      A(1,6) = 0.
      A(2,1) = (ST-THETA*CT)/TORR
      A(2,2) = (-ST*ST-CT*CT)/TORR
      A(2,3) = -(ST2*ST+.5*CT2*CT)/TORR
      A(2,4) = -(ST3*ST+.3333333333334*CT3*CT)/TORR
      A(2,5) = -(ST4*ST+.25*CT4*CT)/TORR
      A(2,6) = CT/TORR
C
C     EVALUATE UPPER TRIANGHLE OF INTEGRAL MATRIX
C     WITHOUT MULTIPLYING BY WEIGHTS
      P(1,1) = A(1,1)*(A(1,1)+V*A(2,1))+A(2,1)*(A(2,1)+V*A(1,1))
      P(1,2) = A(1,1)*(A(1,2)+V*A(2,2))+A(2,1)*(A(2,2)+V*A(1,2))
      P(1,3) = A(1,1)*(A(1,3)+V*A(2,3))+A(2,1)*(A(2,3)+V*A(1,3))
      P(1,4) = A(1,1)*(A(1,4)+V*A(2,4))+A(2,1)*(A(2,4)+V*A(1,4))
      P(1,5) = A(1,1)*(A(1,5)+V*A(2,5))+A(2,1)*(A(2,5)+V*A(1,5))
      P(1,6) = A(1,1)*(A(1,6)+V*A(2,6))+A(2,1)*(A(2,6)+V*A(1,6))
      P(2,2) = A(1,2)*(A(1,2)+V*A(2,2))+A(2,2)*(A(2,2)+V*A(1,2))
      P(2,3) = A(1,2)*(A(1,3)+V*A(2,3))+A(2,2)*(A(2,3)+V*A(1,3))
      P(2,4) = A(1,2)*(A(1,4)+V*A(2,4))+A(2,2)*(A(2,4)+V*A(1,4))
      P(2,5) = A(1,2)*(A(1,5)+V*A(2,5))+A(2,2)*(A(2,5)+V*A(1,5))
      P(2,6) = A(1,2)*(A(1,6)+V*A(2,6))+A(2,2)*(A(2,6)+V*A(1,6))
      P(3,3) = A(1,3)*(A(1,3)+V*A(2,3))+A(2,3)*(A(2,3)+V*A(1,3))
      P(3,4) = A(1,3)*(A(1,4)+V*A(2,4))+A(2,3)*(A(2,4)+V*A(1,4))
      P(3,5) = A(1,3)*(A(1,5)+V*A(2,5))+A(2,3)*(A(2,5)+V*A(1,5))
      P(3,6) = A(1,3)*(A(1,6)+V*A(2,6))+A(2,3)*(A(2,6)+V*A(1,6))
      P(4,4) = A(1,4)*(A(1,4)+V*A(2,4))+A(2,4)*(A(2,4)+V*A(1,4))
      P(4,5) = A(1,4)*(A(1,5)+V*A(2,5))+A(2,4)*(A(2,5)+V*A(1,5))
      P(4,6) = A(1,4)*(A(1,6)+V*A(2,6))+A(2,4)*(A(2,6)+V*A(1,6))
      P(5,5) = A(1,5)*(A(1,5)+V*A(2,5))+A(2,5)*(A(2,5)+V*A(1,5))
      P(5,6) = A(1,5)*(A(1,6)+V*A(2,6))+A(2,5)*(A(2,6)+V*A(1,6))
      P(6,6) = A(1,6)*(A(1,6)+V*A(2,6))+A(2,6)*(A(2,6)+V*A(1,6))
C
C     EVALUATE UPPER TRIANGLE STIFFNESS MATRIX.
C     SET [STIFF] = [STIFF] PREVIOUS + CONST*WEIGHTS * [P]
      DO 50 K=1,6
      DO 50 L=K,6
      STIFF(K,L) = STIFF(K,L)+CONST*WI(I)*WJ(J)*P(K,L)
 50   CONTINUE
100   CONTINUE
C
C     FILL IN LOWER TRIANGLE BY SYMMETRY
      DO 300 I=1,6
      J = 6-I
      K = 7-I
      DO 300 L=1,J
      STIFF(K,L) = STIFF(L,K)
300   CONTINUE
C
C     G MATRIX
      G(1,1) = 1.0
      G(1,2) = 0.0
      G(1,3) = 0.0
      G(2,1) = 3.46238898
      G(2,2) = 3.0/4.0*R
      G(2,3) = 5.0/4.0
      G(3,1) = -2.71238898
      G(3,2) = -R
      G(3,3) = -2.0
      G(4,1) = 2.46238898
      G(4,2) = 3.0/4.0*R
      G(4,3) = 9.0/4.0
      G(5,1) = -1.35619449
```

```
      G(5,2) = -1.0/2.0*R
      G(5,3) = -1.0
      G(6,1) = 2.587942194
      G(6,2) = 3.0/8.0*R
      G(6,3) = 3.0/4.0
C
C     [SG] = [STIFF] [G]
      DO 450 I = 1,6
      DO 450 J = 1,3
      SG(I,J) = 0.0D0
      DO 450 K = 1,6
  450 SG(I,J) = SG(I,J) + STIFF(I,K)*G(K,J)
C
C     EVAL [G]T
      DO 470 I = 1,3
      DO 470 J = 1,6
  470 GT(I,J) = G(J,I)
C
C     [ST3] = [GT] [SG]
      DO 500 I = 1,3
      DO 500 J = 1,3
      ST3(I,J) = 0.0D0
      DO 500 K = 1,6
  500 ST3(I,J) = ST3(I,J) + GT(I,K)*SG(K,J)
C     EVALUATE THE DISPLACEMENTS
C     SOLVE THE EQUATION {F} = [K] {U} FOR U
C     STORE FORCES IN U ARRAY
C     LET F = 1000 FOR NOW
      U(1) = 1000.0
      U(2) = 0.0
      U(3) = 0.0
C
      CALL SOLVE (ST3,U)
      PRINT *,'--------------------'
      PRINT *,'LAMBDA = ',TORR*THICK/R**2
C       PRINT *,'DISPLACEMENTS = ',U(1),U(2),U(3)
C
C     EVALUATE THE STRESSES
C     FORM THE B MATRIX AT POSITION ANGLE PTHETA
C     OUTER SURFACE
      H = 1.5*THICK
      DO 1000 IPOS = 1,2
      H = H-THICK
      PRINT *,'SURFACE = ',H
      PTHETA = -5.0
      DO 900 ITHETA = 1,19
      PTHETA = PTHETA + 5.0
      PTHETA = PTHETA*3.1415927/180.
      PST = SIN(PTHETA)
      PCT = COS(PTHETA)
      HR = H/(R**2)
      PST2 = SIN(2.*PTHETA)
      PST3 = SIN(3.*PTHETA)
      PST4 = SIN(4.*PTHETA)
      PST5 = SIN(5.*PTHETA)
      PCT2 = COS(2.*PTHETA)
      PCT3 = COS(3.*PTHETA)
      PCT4 = COS(4.*PTHETA)
      PCT5 = COS(5.*PTHETA)
C
      A(1,1) = -HR
      A(1,2) = 0.
      A(1,3) = -3.0*HR*PST2
      A(1,4) = -8.0*HR*PST3
      A(1,5) = -15.0*HR*PST4
      A(1,6) = 0.
      A(2,1) = (PST-PTHETA*PCT)/TORR
      A(2,2) = (-PST*PST-PCT*PCT)/TORR
      A(2,3) = -(PST2*PST+.5*PCT2*PCT)/TORR
      A(2,4) = -(PST3*PST+.3333333333334*PCT3*PCT)/TORR
      A(2,5) = -(PST4*PST+.25*PCT4*PCT)/TORR
      A(2,6) = PCT/TORR
C     [B] = [A] [G]
      DO 640 IT = 1,2
      DO 640 JT = 1,3
      BMAT(IT,JT) = 0.0D0
      DO 640 KT = 1,6
  640 BMAT(IT,JT) = BMAT(IT,JT) + A(IT,KT)*G(KT,JT)
C
C     EVALUATE STRAINS {EP} = [B][U]
      EP(1) = BMAT(1,1)*U(1)+BMAT(1,2)*U(2)+BMAT(1,3)*U(3)
      EP(2) = BMAT(2,1)*U(1)+BMAT(2,2)*U(2)+BMAT(2,3)*U(3)
C
C     EVALUATE STRESSES {SIG} = [D] {EP}
      SCON = EX/(1.0-NUXY**2)
      SIG(1) = SCON*(EP(1)+NUXY*EP(2))
      SIG(2) = SCON*(EP(1)*NUXY+EP(2))
C
C     NORMALISE STRESSES BY DIVIDING BY SIGBO
      SIGBO = 3.0*R*1000.0/(3.1415927*TORR*THICK**2)
      SIG(1) = SIG(1)/SIGBO
      SIG(2) = SIG(2)/SIGBO
      PTHETA = PTHETA * 180/3.1415927
      POS = 90.0 - PTHETA
      WRITE (6,'(9F10.3)') POS,SIG(1),SIG(2)
C
  900 CONTINUE
 1000 CONTINUE
      END
C
```

A3.3.1 SUBROUTINE SOLVE

```
      SUBROUTINE SOLVE (AMAT,CVEC)
C     PROG TO SOLVE SYMM MATRICES BY THE GAUSS ELIM. METHOD.
C     REF.  BREBBIA/FERRANTE PROG13
C
C     SOLVES EQUN.  [A] {C} = {B} FOR {C}
C
C
      INTEGER I,NEQ,J,K,L,K1,NEQ1,LOOP
      DOUBLE PRECISION AMAT(3,3),CVEC(3)
C
      NEQ = 3
      NEQ1 = NEQ-1
      DO 100 K = 1,NEQ1
      CVAL = AMAT(K,K)
      K1 = K+1
      DO 11 J = K1,NEQ
   11 AMAT(J,K) = AMAT(K,J)
      IF (ABS(CVAL)-1E-6)4,4,7
    4 WRITE (6,5) K
    5 FORMAT ('**** SINGULARITY IN ROW',I5)
      GO TO 300
C
C     DIV ROW BY DIAG COEFF
C
```

```
 7   DO 8 J = K1,NEQ
 8   AMAT(K,J) = AMAT(K,J)/CVAL
     CVEC(K) = CVEC(K)/CVAL
C
C    ELIMINATE ROW UNKNOWN X(K) FROM ROW I
C
     DO 10 I = K1,NEQ
     CVAL = AMAT(I,K)
     DO 9 J = I,NEQ
 9   AMAT(I,J) = AMAT(I,J)-CVAL*AMAT(K,J)
 10  CVEC(I) = CVEC(I)-CVAL*CVEC(K)
 100 CONTINUE
C
C     COMPUTE LAST UK
C
     IF(ABS(AMAT(NEQ,NEQ))-1E-6)4,4,101
 101 CVEC(NEQ) = CVEC(NEQ)/AMAT(NEQ,NEQ)
C
C     APPLY BACKSUB TO COMPUTE REMAINING UKS
C
     DO 200 L = 1,NEQ1
     K = NEQ-L
     K1 = K+1
     DO 200 J = K1,NEQ
 200 CVEC(K) = CVEC(K)-AMAT(K,J)*CVEC(J)
C
 300 RETURN
     END
```

A3.4 PROGRAM BEL3FLEX.

```
     PROGRAM BEL3FLEX
C
C    This program evaluates the flexibility factor of a
C    semi-toroidal
C    bellows using a finite element approach.
C    Tangential displacement interpolated by quintic polynomial.
C    The bellows stiffness matrix is evaluated using gaussian
C    quadrature, 3 points through thickness and 5 along the arc.
C
C    Written by Donald Mackenzie, March 1989.
C
C    NOTE. In this program:
C        THETA = MERIDIONAL (AXIAL) DIRECTION
C        PHI = CIRCUMFERENTIAL (AXISYMM.) DIRECTION.

     IMPLICIT DOUBLE PRECISION (A-H,O-Z)
     DOUBLE PRECISION WI(3),AI(3),WJ(5),AJ(5),A(2,6),
    1 P(3,3),STIFF(3,3),NUXY,G(6,3),DET1,
    2 STBEL(1,1),BMAT(2,3),BT(3,2),D(2,2),DB(2,3)
C
C    Input routine: Interactive.
C
C    MATERIAL PROPERTIES: Young's modulus, Poisson's ratio.
C    BELLOWS GEOMETRY: R = convolution radius, TORR = mean
C    radius from symmetric axis, THICK = wall thickness.
C
     PRINT *,'INPUT PIPE RADIUS'
     READ (6,*) TORR
     PRINT *,'PIPE RAD. = ',TORR
     PRINT *,'INPUT CONVOLUTION RADIUS'
     READ (6,*) R
     PRINT *,'CONV. RAD = ', R
     PRINT *,'INPUT THICKNESS'
     READ (6,*) THICK
     PRINT *,' THICK. = ',THICK
     PRINT *,'INPUT ELASTIC MOD.'
     READ (6,*) EX
     PRINT *,' E  = ',EX
     PRINT *,'INPUT POISSON RATIO'
     READ (6,*) NUXY
     PRINT *,'NU = ',NUXY
C
C    STIFFNESS MATRIX CONSTANT
     P4 = 3.1415927D0/4.0D0
     CONST = 2.0D0*EX*P4*4.0D0*R*TORR/(1-NUXY**2)
C
C    CONSTITUTIVE MATRIX [D]*(1-NUXY**2)/E
     D(1,1) = 1.0
     D(1,2) = NUXY
     D(2,1) = NUXY
     D(2,2) = 1.0
C
C    NUMERICAL INTEGRATION PARAMETERS.
C    WEIGHTS
     W1 = 0.555555555555556D0
     W2 = 0.888888888888889D0
     W3 = W1
C    ABSCISSA
     A3 = 0.774596669241483D0
     A2 = 0.0D0
     A1 = -A3
C
     WT1 = 0.236926885056189
     WT2 = 0.478628670499366
     WT3 =0.568888888888889
     WT4 = WT2
     WT5 = WT1
C
     B1 = -0.906179845938664
     B2 = -0.538469310105683
     B3 = 0.0D0
     B4 = -B2
     B5 = -B1
C    APPLY ACTUAL INTEGRATION LIMITS:
C    SUBSCRIPT I REFERS TO H, J TO THETA
     T = THICK/2
     WI(1) = T*W1
     WI(2) = T*W2
     WI(3) = WI(1)
     AI(1) = T*A1
     AI(2) = T*A2
     AI(3) = T*A3
C
     WJ(1) = P4*WT1
     WJ(2) = P4*WT2
     WJ(3) = P4*WT3
     WJ(4) = WJ(2)
     WJ(5) = WJ(1)
     AJ(1) = P4*(1.0D0+B1)
     AJ(2) = P4*(1.0D0+B2)
     AJ(3) = P4*(1.0D0+B3)
     AJ(4) = P4*(1.0D0+B4)
     AJ(5) = P4*(1.0D0+B5)
C
C
```

```
C   ZERO STIFFNESS AND P (ATDA) MATRICES
    DO 30 I=1,6
    DO 30 J=1,6
    STIFF(I,J) = 0.0D0
    P(I,J)  = 0.0D0
30  CONTINUE
C
C   START OF NUMERICAL INTEGRATION (DOUBLE) LOOP
    DO 100 I=1,3
    DO 100 J=1,5
C
C   EVALUATE B MATRIX
C
C   MATRIX [A]: Strain-displacement "shape".
    HR = AJ(I)/(R**2)
    THETA = AJ(J)
    ST = SIN(THETA)
    CT = COS(THETA)
    B = THETA - P4
    B2 = B*B
    B3 = B2*B
    B4 = B3*B
    B5 = B4*B
    A(1,1) = 0.0D0
    A(1,2) = HR
    A(1,3) = 2.0D0*HR*B
    A(1,4) = HR*(6.0D0+3.0D0*B2)
    A(1,5) = HR*(4.0D0*B3+24.0D0*B)
    A(1,6) = HR*(5.0D0*B4+60.0D0*B2)
    A(2,1) = CT/TORR
    A(2,2) = (B*CT-ST)/TORR
    A(2,3) = (B2*CT-2.0D0*B*ST)/TORR
    A(2,4) = (B3*CT-3.0D0*B2*ST)/TORR
    A(2,5) = (B4*CT-4.0D0*B3*ST)/TORR
    A(2,6) = (B5*CT-5.0D0*B4*ST)/TORR
C
C   MATRIX [G]: Strain displacement "magnitude", with actual
C   boundary conditions applied.
    G(1,1) = -0.24543692610D0
    G(1,2) = -0.03855314219OD0*R
    G(1,3) = 0.24543692610D0
    G(2,1) = 7.0D0/16.0D0
    G(2,2) = 0.049087385200D0*R
    G(2,3) = G(2,1)
    G(3,1) = 0.47746482930D0
    G(3,2) = 1.0D0/8.0D0*R
    G(3,3) = -G(3,1)
    G(4,1) = -1.0132118360D0
    G(4,2) = -0.15915494310D0*R
    G(4,3) = G(4,1)
    G(5,1) = -0.12900613770D0
    G(5,2) = -0.10132118360D0*R
    G(5,3) = -G(5,1)
    G(6,1) = 0.49276714820D0
    G(6,2) = 0.12900613770D0*R
    G(6,3) = G(6,1)
C
C   EVALUATE [BMAT] = [A] [G]
    DO 40 IT=1,2
    DO 40 JT=1,3
    BMAT(IT,JT) = 0.0D0
    DO 40 KT=1,6
40  BMAT(IT,JT) = BMAT(IT,JT) + A(IT,KT)*G(KT,JT)
C
C   EVALUATE [BT] = TRANSPOSE OF [B]
    DO 47 IT=1,2
    DO 47 JT=1,3
47  BT(JT,JT) = BMAT(IT,JT)
C
C   EVALUATE [P] = [B]T [D] [B] MATRIX
C
    DO 50 IT=1,2
    DO 50 JT=1,3
    DB(IT,JT) = 0.0
    DO 50 KT=1,2
50  DB(IT,JT) = DB(IT,JT)+D(IT,KT)*BMAT(KT,JT)
C
    DO 60 IT=1,3
    DO 60 JT=1,3
    P(IT,JT) = 0.0
    DO 60 KT=1,2
60  P(IT,JT) = P(IT,JT)+BT(IT,KT)*DB(KT,JT)
C
C   EVALUATE STIFFNESS MATRIX.
C   [STIFF] = [STIFF] PREVIOUS + CONST*WEIGHTS * [P]
    DO 70 K=1,3
    DO 70 L=1,3
70  STIFF(K,L) = STIFF(K,L)+CONST*WI(I)*WJ(J)*P(K,L)
C
100 CONTINUE
C   END OF NUMERICAL INTEGRATION LOOP
C
C   APPLY STATIC CONDENSATION: Evaluate the axial stiffness.
C
    CALL REDUCE (STIFF,STBEL)
C
C   EVALUATE FLEXIBILITY FACTOR
C
C   DISPLACEMENT/ UNIT FORCE OF EQUIVALENT STRAIGHT PIPE.
    DELTA = R/(2*3.1415937*TORR*THICK*EX)
    DELTA = DELTA * (R/(THICK/2))**2
    DELTA = DELTA * (1.0 - NUXY**2)
C   FLEXIBILITY FACTOR = FLEX
    FLEX = 1/(STBEL(1,1)*DELTA)
C   WRITE RESULTS TO SCREEN
    PRINT *,'LAMBDA =',TORR*THICK/R**2
    PRINT *,'FLEX =',FLEX
C
    END
C
    SUBROUTINE REDUCE (SE,RED)
C   *** SEE APPENDIX 3.2.1 ***
```

A3.5 PROGRAM BEL3STR.FOR

```
    PROGRAM STRPOLY
C   Program to evaluate the normalised stress distribution in
C   semi-toroidal bellows.  Based on polynomial interpolation
C   of tangential displacement.
C   Bellows stiffness evaluated by Gaussian quadrature.
C   3 points thro' thickness, 5 along arc.
C
C   Written by Donald Mackenzie, March 1989.
C
    IMPLICIT DOUBLE PRECISION (A-H,O-Z)
    DOUBLE PRECISION WI(3),AI(3),WJ(5),AJ(5),A(2,6),
   1 P(6,6),STIFF(6,6),NUXY,C(6,12),G(6,3),DETI,ST3(3,3),
```

```
     2 SG(6,3),GT(3,6),STBEL(1,1),U(3),BMAT(2,3),EP(2),SIG(2)
C
C    NOTE. In this program:
C        THETA = MERIDIONAL (AXIAL) DIRECTION
C        PHI = CIRCUMFERENTIAL (AXISYMM.) DIRECTION.

C    Input routine: Interactive.
C
C    MATERIAL PROPERTIES: Young's modulus, Poisson's ratio.
C    BELLOWS GEOMETRY: R = convolution radius, TORR = mean
C    radius from
C    symmetric axis, THICK = wall thickness.
C
     PRINT *,'INPUT PIPE RADIUS'
     READ (6,*) TORR
     PRINT *,'PIPE RAD. =',TORR
     PRINT *,'INPUT CONVOLUTION RADIUS'
     READ (6,*) R
     PRINT *,'CONV. RAD =', R
     PRINT *,'INPUT THICKNESS'
     READ (6,*) THICK
     PRINT *,' THICK. =',THICK
     PRINT *,'INPUT ELASTIC MOD.'
     READ (6,*) EX
     PRINT *,' E  =',EX
     PRINT *,'INPUT POISSON RATIO'
     READ (6,*) NUXY
     PRINT *,'NU =',NUXY
C
     V = NUXY
C    WEIGHTS
     W1 = 0.555555555555556D0
     W2 = 0.888888888888889D0
     W3 = W1
C    ABSCISSA
     A3 = 0.774596669241483D0
     A2 = 0.0D0
     A1 = -A3
C
     WT1 = 0.236926885056189
     WT2 = 0.478628670499366
     WT3 = 0.568888888888889
     WT4 = WT2
     WT5 = WT1
C
     B1 = -0.906179845938664
     B2 = -0.538469310105683
     B3 = 0.0D0
     B4 = -B2
     B5 = -B1
C    ACTUAL INTEGRATION LIMITS:
C    THRO' THICKNESS H = -THICK/2 TO THICK/2.
C    ALONG MERIDIAN  THETA = 0 TO PI/4
C    SUBSCRIPT I REFERS TO H, J TO THETA
     T = THICK/2
     WI(1) = T*W1
     WI(2) = T*W2
     WI(3) = WI(1)
     AI(1) = T*A1
     AI(2) = T*A2
     AI(3) = T*A3
C
     P4 = 3.1415927D0/4.0D0
     WJ(1) = P4*WT1
     WJ(2) = P4*WT2
     WJ(3) = P4*WT3
     WJ(4) = WJ(2)
     WJ(5) = WJ(1)
     AJ(1) = P4*(1.0D0 + B1)
     AJ(2) = P4*(1.0D0 + B2)
     AJ(3) = P4*(1.0D0 + B3)
     AJ(4) = P4*(1.0D0 + B4)
     AJ(5) = P4*(1.0D0 + B5)
C
C    STIFFNESS MATRIX CONSTANT
     CONST = 2.0D0*EX*P4*4.0D0*R*TORR/(1-NUXY**2)
C
C    ZERO STIFFNESS AND P (ATDA) MATRICES
     DO 30 I = 1,6
     DO 30 J = 1,6
     STIFF(I,J) = 0.0D0
     P(I,J) = 0.0D0
  30 CONTINUE
C    SET UP INTEGRATION DOUBLE LOOP
     DO 100 I = 1,3
     DO 100 J = 1,5
C    EVALUATE ELEMENTS OF [A]
     HR = AI(I)/(R**2)
     THETA = AJ(J)
     ST = SIN(THETA)
     CT = COS(THETA)
     B = THETA - P4
     B2 = B*B
     B3 = B2*B
     B4 = B3*B
     B5 = B4*B
     A(1,1) = 0.0D0
     A(1,2) = HR
     A(1,3) = 2.0D0*HR*B
     A(1,4) = HR*(6.0D0 + 3.0D0*B2)
     A(1,5) = HR*(4.0D0*B3 + 24.0D0*B)
     A(1,6) = HR*(5.0D0*B4 + 60.0D0*B2)
     A(2,1) = CT/TORR
     A(2,2) = (B*CT-ST)/TORR
     A(2,3) = (B2*CT-2.0D0*B*ST)/TORR
     A(2,4) = (B3*CT-3.0D0*B2*ST)/TORR
     A(2,5) = (B4*CT-4.0D0*B3*ST)/TORR
     A(2,6) = (B5*CT-5.0D0*B4*ST)/TORR
C
C    EVALUATE UPPER TRIANGHLE OF INTEGRAL MATRIX
C    WITHOUT MULTIPLYING BY WEIGHTS
     P(1,1) = A(2,1)*A(2,1)
     P(1,2) = A(2,1)*(A(2,2) + V*A(1,2))
     P(1,3) = A(2,1)*(A(2,3) + V*A(1,3))
     P(1,4) = A(2,1)*(A(2,4) + V*A(1,4))
     P(1,5) = A(2,1)*(A(2,5) + V*A(1,5))
     P(1,6) = A(2,1)*(A(2,6) + V*A(1,6))
     P(2,2) = A(1,2)*(A(1,2) + V*A(2,2)) + A(2,2)*(A(2,2) + V*A(1,2))
     P(2,3) = A(1,2)*(A(1,3) + V*A(2,3)) + A(2,2)*(A(2,3) + V*A(1,3))
     P(2,4) = A(1,2)*(A(1,4) + V*A(2,4)) + A(2,2)*(A(2,4) + V*A(1,4))
     P(2,5) = A(1,2)*(A(1,5) + V*A(2,5)) + A(2,2)*(A(2,5) + V*A(1,5))
     P(2,6) = A(1,2)*(A(1,6) + V*A(2,6)) + A(2,2)*(A(2,6) + V*A(1,6))
     P(3,3) = A(1,3)*(A(1,3) + V*A(2,3)) + A(2,3)*(A(2,3) + V*A(1,3))
     P(3,4) = A(1,3)*(A(1,4) + V*A(2,4)) + A(2,3)*(A(2,4) + V*A(1,4))
     P(3,5) = A(1,3)*(A(1,5) + V*A(2,5)) + A(2,3)*(A(2,5) + V*A(1,5))
     P(3,6) = A(1,3)*(A(1,6) + V*A(2,6)) + A(2,3)*(A(2,6) + V*A(1,6))
     P(4,4) = A(1,4)*(A(1,4) + V*A(2,4)) + A(2,4)*(A(2,4) + V*A(1,4))
     P(4,5) = A(1,4)*(A(1,5) + V*A(2,5)) + A(2,4)*(A(2,5) + V*A(1,5))
```

```
      P(4,6) = A(1,4)*(A(1,6)+V*A(2,6))+A(2,4)*(A(2,6)+V*A(1,6))
      P(5,5) = A(1,5)*(A(1,5)+V*A(2,5))+A(2,5)*(A(2,5)+V*A(1,5))
      P(5,6) = A(1,5)*(A(1,6)+V*A(2,6))+A(2,5)*(A(2,6)+V*A(1,6))
      P(6,6) = A(1,6)*(A(1,6)+V*A(2,6))+A(2,6)*(A(2,6)+V*A(1,6))
C
C     EVALUATE UPPER TRIANGLE STIFFNESS MATRIX.
C     SET [STIFF] = [STIFF] PREVIOUS + CONST*WEIGHTS * [P]
      DO 50 K=1,6
      DO 50 L=K,6
      STIFF(K,L) = STIFF(K,L)+CONST*WI(I)*WJ(J)*P(K,L)
 50   CONTINUE
100   CONTINUE
C
C     FILL IN LOWER TRIANGLE BY SYMMETRY
      DO 300 I=1,6
      J = 6-I
      K = 7-I
      DO 300 L=1,J
      STIFF(K,L) = STIFF(L,K)
300   CONTINUE
C
C     G MATRIX
      G(1,1) = -0.24543692610D0
      G(1,2) = -0.038553142190D0*R
      G(1,3) = 0.24543692610D0
      G(2,1) = 7.0D0/16.0D0
      G(2,2) = 0.049087385 20D0*R
      G(2,3) = G(2,1)
      G(3,1) = 0.47746482930D0
      G(3,2) = 1.0D0/8.0D0*R
      G(3,3) = -G(3,1)
      G(4,1) = -1.0132118360D0
      G(4,2) = -0.15915494310D0*R
      G(4,3) = G(4,1)
      G(5,1) = -0.12900613770D0
      G(5,2) = -0.10132118360D0*R
      G(5,3) = -G(5,1)
      G(6,1) = 0.49276714820D0
      G(6,2) = 0.12900613770D0*R
      G(6,3) = G(6,1)
C
C     [SG] = [STIFF] [G]
      DO 450 I=1,6
      DO 450 J=1,3
      SG(I,J) = 0.0D0
      DO 450 K=1,6
450   SG(I,J) = SG(I,J) + STIFF(I,K)*G(K,J)
C
C     EVAL [GT]
      DO 470 I=1,3
      DO 470 J=1,6
470   GT(I,J) = G(J,I)
C
C     [ST3] = [GT] [SG]
      DO 500 I=1,3
      DO 500 J=1,3
      ST3(I,J) = 0.0D0
      DO 500 K=1,6
500   ST3(I,J) = ST3(I,J) + GT(I,K)*SG(K,J)
C
C     EVALUATE THE DISPLACEMENTS
C     SOLVE THE EQUATION {F} = [K] {U} FOR U
C     STORE FORCES IN U ARRAY
C     LET F=1000 FOR NOW
      U(1) = 1000.0
      U(2) = 0.0
      U(3) = 0.0
C
      CALL SOLVE (ST3,U)
      PRINT *,'-------------------------'
      PRINT *,'LAMBDA = ',TORR*THICK/R**2
C     PRINT *,'      W1          ROT1          W2'
      WRITE (6,'(9E14.3)') U(1),U(2),U(3)
C
C     EVALUATE THE STRESSES
C     FORM THE B MATRIX AT POSITION ANGLE PTHETA
C
      PRINT *,' POS     SIG TH      SIG PH'
C
      H = 1.5*THICK
      DO 1000 IPOSN = 1,2
      H=H-THICK
      PRINT *,' SURFACE  =',H
      PTHETA = -5.0
      DO 900 ITHETA=1,19
      PTHETA = PTHETA + 5.0
      PTHETA = PTHETA*3.1415927/180.
      PST = SIN(PTHETA)
      PCT = COS(PTHETA)
      HR = H/(R**2)
      B = PTHETA - P4
      B2 = B*B
      B3 = B2*B
      B4 = B3*B
      B5 = B4*B
      A(1,1) = 0.0D0
      A(1,2) = HR
      A(1,3) = 2.0D0*HR*B
      A(1,4) = HR*(6.0D0+3.0D0*B2)
      A(1,5) = HR*(4.0D0*B3+24.0D0*B)
      A(1,6) = HR*(5.0D0*B4+60.0D0*B2)
      A(2,1) = PCT/TORR
      A(2,2) = (B*PCT-PST)/TORR
      A(2,3) = (B2*PCT-2.0D0*B*PST)/TORR
      A(2,4) = (B3*PCT-3.0D0*B2*PST)/TORR
      A(2,5) = (B4*PCT-4.0D0*B3*PST)/TORR
      A(2,6) = (B5*PCT-5.0D0*B4*PST)/TORR
C
C     [B] = [A] [G]
      DO 640 IT=1,2
      DO 640 JT=1,3
      BMAT(IT,JT) = 0.0D0
      DO 640 KT=1,6
640   BMAT(IT,JT) = BMAT(IT,JT) + A(IT,KT)*G(KT,JT)
C
C     EVALUATE STRAINS {EP} = [B][U]
      EP(1) = BMAT(1,1)*U(1)+BMAT(1,2)*U(2)+BMAT(1,3)*U(3)
      EP(2) = BMAT(2,1)*U(1)+BMAT(2,2)*U(2)+BMAT(2,3)*U(3)
C
C     EVALUATE STRESSES {SIG} = [D] {EP}
      SCON = EX/(1.0-NUXY**2)
      SIG(1) = SCON*(EP(1)+NUXY*EP(2))
      SIG(2) = SCON*(EP(1)*NUXY+EP(2))
C
C     NORMALISE STRESSES BY DIVIDING BY SIGBO
      SIGBO = 3.0*R*1000.0/(3.1415927*TORR*THICK**2)
      SIG(1) = SIG(1)/SIGBO
      SIG(2) = SIG(2)/SIGBO
```

```
      PTHETA = PTHETA * 180/3.1415729
      POS = 90 - PTHETA
      WRITE (6,'(9F10.3)') POS,SIG(1),SIG(2)
  900 CONTINUE
 1000 CONTINUE
      END
C
      SUBROUTINE SOLVE (AMAT,CVEC)
C     *** SEE APPENDIX 3.3.1***
```

A3.6 PROGRAM BEL4FLEX.FOR

```
      PROGRAM BEL4FLEX
C
C     This program evaluates the flexibility factor of a
C     semi-toroidal bellows using a finite element approach.
C     Tangential displacement interpolated by quintic polynomial.
C     Assumed constant membrane extension in axial direction.
C     The bellows stiffness matrix is evaluated using gaussian
C     quadrature, 3 points through thickness and 5 along the arc.
C
C     Written by Donald Mackenzie, March 1989.
C
C     NOTE. In this program:
C           THETA = MERIDIONAL (AXIAL) DIRECTION
C           PHI = CIRCUMFERENTIAL (AXISYMM.) DIRECTION.

      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DOUBLE PRECISION WI(3),AI(3),WJ(5),AJ(5),A(2,6),
     1 P(4,4),STIFF(4,4),NUXY,G(6,4),DET1,
     2 STBEL(1,1),AG(2,4),BMAT(2,4),BT(4,2),D(2,2),DB(2,4)
C
C     Input routine: Interactive.
C
C     MATERIAL PROPERTIES: Young's modulus, Poisson's ratio.
C     BELLOWS GEOMETRY: R = convolution radius, TORR = mean
C     radius from symmetric axis, THICK = wall thickness.
C     Here set values of geometry parameters are given. Alternatively
C     an input procedure or loop could be used.
C
      PRINT *,'INPUT PIPE RADIUS'
      READ (6,*) TORR
      PRINT *,'PIPE RAD. =',TORR
      PRINT *,'INPUT CONVOLUTION RADIUS'
      READ (6,*) R
      PRINT *,'CONV. RAD =', R
      PRINT *,'INPUT THICKNESS'
      READ (6,*) THICK
      PRINT *,' THICK  =',THICK
      PRINT *,'INPUT ELASTIC MOD.'
      READ (6,*) EX
      PRINT *,' E  =',EX
      PRINT *,'INPUT POISSON RATIO'
      READ (6,*) NUXY
      PRINT *,'NU  =',NUXY
C
C     CONSTITUTIVE MATRIX
      D(1,1) = 1.0
      D(1,2) = NUXY
      D(2,1) = NUXY
      D(2,2) = 1.0
C     WEIGHTS
      W1 = 0.55555555555555556D0
      W2 = 0.88888888888888889D0
      W3 = W1
C     ABSCISSA
      A3 = 0.77459666924148340D0
      A2 = 0.0D0
      A1 = -A3
C
      WT1 = 0.23692688505618909
      WT2 = 0.47862867049936647
      WT3 = 0.56888888888888889
      WT4 = WT2
      WT5 = WT1
C
      B1 = -0.90617984593866399
      B2 = -0.53846931010568309
      B3 = 0.0D0
      B4 = -B2
      B5 = -B1
C     ACTUAL INTEGRATION LIMITS:
C     THRO' THICKNESS H = -THICK/2 TO THICK/2.
C     ALONG MERIDIAN  THETA = 0 TO PI/4
C     SUBSCRIPT I REFERS TO H, J TO THETA
      T = THICK/2
      WI(1) = T*W1
      WI(2) = T*W2
      WI(3) = WI(1)
      AI(1) = T*A1
      AI(2) = T*A2
      AI(3) = T*A3
C
      P4 = 3.14159265358979324D0/4.0D0
      WJ(1) = P4*WT1
      WJ(2) = P4*WT2
      WJ(3) = P4*WT3
      WJ(4) = WJ(2)
      WJ(5) = WJ(1)
      AJ(1) = P4*(1.0D0+B1)
      AJ(2) = P4*(1.0D0+B2)
      AJ(3) = P4*(1.0D0+B3)
      AJ(4) = P4*(1.0D0+B4)
      AJ(5) = P4*(1.0D0+B5)
C
C     STIFFNESS MATRIX CONSTANT
      CONST = 2.0D0*EX*P4*4.0D0*R*TORR/(1-NUXY**2)
C
C     ZERO STIFFNESS AND P (ATDA) MATRICES
      DO 30 I=1,4
      DO 30 J=1,4
      STIFF(I,J)=0.0D0
      P(I,J) = 0.0D0
   30 CONTINUE
C
C     SET UP INTEGRATION DOUBLE LOOP
      DO 100 I=1,3
      DO 100 J=1,5
C
C     EVALUATE B MATRIX
C
C     ELEMENTS OF [A]
      H = AI(I)
      HR = AI(I)/(R**2)
      THETA = AJ(J)
      ST = SIN(THETA)
      CT = COS(THETA)
```

```
      B = THETA - P4
      B2 = B*B
      B3 = B2*B
      B4 = B3*B
      B5 = B4*B
      A(1,1) = 0.0D0
      A(1,2) = HR
      A(1,3) = 2.0D0*HR*B
      A(1,4) = HR*(6.0D0 + 3.0D0*B2)
      A(1,5) = HR*(4.0D0*B3 + 24.0D0*B)
      A(1,6) = HR*(5.0D0*B4 + 60.0D0*B2)
      A(2,1) = CT/TORR
      A(2,2) = (B*CT-ST)/TORR
      A(2,3) = (B2*CT-2.0D0*B*ST)/TORR
      A(2,4) = (B3*CT-3.0D0*B2*ST)/TORR
      A(2,5) = (B4*CT-4.0D0*B3*ST)/TORR
      A(2,6) = (B5*CT-5.0D0*B4*ST)/TORR
C
C     G MATRIX
      G(1,1) = -0.24543692610D0
      G(1,2) = -0.038553142190D0*R
      G(1,3) = 0.24543692610D0
      G(2,1) = 7.0D0/16.0D0
      G(2,2) = 0.049087385 20D0*R
      G(2,3) = G(2,1)
      G(3,1) = 0.47746482930D0
      G(3,2) = 1.0D0/8.0D0*R
      G(3,3) = -G(3,1)
      G(4,1) = -1.0132118360D0
      G(4,2) = -0.15915494310D0*R
      G(4,3) = G(4,1)
      G(5,1) = -0.12900613770D0
      G(5,2) = -0.10132118360D0*R
      G(5,3) = -G(5,1)
      G(6,1) = 0.49276714820D0
      G(6,2) = 0.12900613770D0*R
      G(6,3) = G(6,1)
      DO 35 IT=1,6
      G(IT,4) = -(G(IT,1) + G(IT,3))
   35 CONTINUE
C
C     AG MATRIX
C     [AG] = [A] [G]
      DO 40 IT=1,2
      DO 40 JT=1,4
      AG(IT,JT) = 0.0D0
      DO 40 KT=1,6
   40 AG(IT,JT) = AG(IT,JT) + A(IT,KT)*G(KT,JT)
C
C     PUT [AG] INTO [B]
      DO 45 IT=1,2
      DO 45 JT=1,4
   45 BMAT(IT,JT) = AG(IT,JT)
C
C     ADD TERM TO LAST COL OF [B]
      BMAT(1,4) = BMAT(1,4) + (R-H)/R*R
      BMAT(2,4) = BMAT(2,4) + ST/TORR
C     EVALUATE TRANSPOSE OF B
      DO 47 IT=1,2
      DO 47 JT=1,4
   47 BT(JT,IT) = BMAT(IT,JT)
C
C     EVALUATE THE [P] = [B]T [D] [B] MATRIX
C     [DB] = [D] [B]
      DO 50 IT=1,2
      DO 50 JT=1,4
      DB(IT,JT) = 0.0
      DO 50 KT=1,2
   50 DB(IT,JT) = DB(IT,JT) + D(IT,KT)*BMAT(KT,JT)
C
C     [P] = [B]T [DB]
      DO 60 IT=1,4
      DO 60 JT=1,4
      P(IT,JT) = 0.0
      DO 60 KT=1,2
   60 P(IT,JT) = P(IT,JT) + BT(IT,KT)*DB(KT,JT)
C
C     EVALUATE STIFFNESS MATRIX.
C     SET [STIFF] = [STIFF] PREVIOUS + CONST*WEIGHTS * [P]
      DO 70 K=1,4
      DO 70 L=1,4
      STIFF(K,L) = STIFF(K,L) + CONST*WI(I)*WJ(J)*P(K,L)
   70 CONTINUE
C
  100 CONTINUE
C     WRITE (6,'(4E12.4)') ((STIFF(IT,JT),JT=1,4),IT=1,4)
C
C     STATIC COND TO GET BELLOWS STIFFNESS
      CALL REDUCE (STIFF,STBEL)
C
      PRINT *,'LAMBDA =',TORR*THICK/R**2
C     STIFFNESS OF ST PIPE
      DELTA = R/(2*3.1415937*TORR*THICK*EX)
      DELTA = DELTA * (R/(THICK/2))**2
      DELTA = DELTA * 0.910D0
C     FLEXIBILITY FACTOR
      FLEX = 1/(STBEL(1,1)*DELTA)
      PRINT *,'FLEX =',FLEX
      PRINT *,'----------'
 1000 CONTINUE
      END
C
      SUBROUTINE REDUCE (SE,RED)
C     *** SEE APPENDIX 3.2.1 ***
```

A3.7 PROGRAM BEL4STR.FOR

```
      PROGRAM BEL4STR
C     Program to evaluate the normalised stress distribution in
C     semi-toroidal bellows. Based on polynomial interpolation of
C     tangential displacement, with assumed constant axial membrane
C     extension.
C     Bellows stiffness evaluated by Gaussian quadrature. 3 points
C     thro' thickness, 5 along arc.
C
C     Written by Donald Mackenzie, March 1989.
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DOUBLE PRECISION WI(3),AI(3),WJ(5),AJ(5),A(2,6),
     1 P(4,4),STIFF(4,4),NUXY,G(6,4),DETI,STBEL(1,1),AG(2,4),
     2 BMAT(2,4),BT(4,2),D(2,2),DB(2,4),U(4),SIG(2),EP(2)
C
C     NOTE. In this program:
C          THETA = MERIDIONAL (AXIAL) DIRECTION
C          PHI = CIRCUMFERENTIAL (AXISYMM.) DIRECTION.
```

```
C   Input routine: Interactive.
C
C   MATERIAL PROPERTIES: Young's modulus, Poisson's ratio.
C   BELLOWS GEOMETRY: R = convolution radius, TORR = mean
C   radius from symmetric axis, THICK = wall thickness.

    PRINT *,'INPUT PIPE RADIUS'
    READ (6,*) TORR
    PRINT *,'PIPE RAD. = ',TORR
    PRINT *,'INPUT CONVOLUTION RADIUS'
    READ (6,*) R
    PRINT *,'CONV. RAD = ', R
    PRINT *,'INPUT THICKNESS'
    READ (6,*) THICK
    PRINT *,' THICK. = ',THICK
    PRINT *,'INPUT ELASTIC MOD.'
    READ (6,*) EX
    PRINT *,' E  = ',EX
    PRINT *,'INPUT POISSON RATIO'
    READ (6,*) NUXY
    PRINT *,'NU = ',NUXY
C
    D(1,1) = 1.0
    D(1,2) = NUXY
    D(2,1) = NUXY
    D(2,2) = 1.0
C   WEIGHTS
    W1 = 0.555555555555556D0
    W2 = 0.888888888888889D0
    W3 = W1
C   ABSCISSA
    A3 = 0.774596669241483D0
    A2 = 0.0D0
    A1 = -A3
C
    WT1 = 0.236926885056189
    WT2 = 0.478628670499366
    WT3 =0.568888888888889
    WT4 = WT2
    WT5 = WT1
C
    B1 = -0.906179845938664
    B2 = -0.538469310105683
    B3 = 0.0D0
    B4 = -B2
    B5 = -B1
C   ACTUAL INTEGRATION LIMITS:
C   THRO' THICKNESS H = -THICK/2 TO THICK/2.
C   ALONG MERIDIAN  THETA = 0 TO PI/4
C   SUBSCRIPT I REFERS TO H, J TO THETA
    T = THICK/2
    WI(1) = T*W1
    WI(2) = T*W2
    WI(3) = WI(1)
    AI(1) = T*A1
    AI(2) = T*A2
    AI(3) = T*A3
C
    P4 = 3.1415927D0/4.0D0
    WJ(1) = P4*WT1
    WJ(2) = P4*WT2
    WJ(3) = P4*WT3
    WJ(4) = WJ(2)
    WJ(5) = WJ(1)
    AJ(1) = P4*(1.0D0+B1)
    AJ(2) = P4*(1.0D0+B2)
    AJ(3) = P4*(1.0D0+B3)
    AJ(4) = P4*(1.0D0+B4)
    AJ(5) = P4*(1.0D0+B5)
C
C   STIFFNESS MATRIX CONSTANT
C    DO 1000 NTIME=1,20
    CONST = 2.0D0*EX*P4*4.0D0*R*TORR/(1-NUXY**2)
C
C   ZERO STIFFNESS AND P (ATDA) MATRICES
    DO 30 I=1,4
    DO 30 J=1,4
    STIFF(I,J) = 0.0D0
    P(I,J) = 0.0D0
30  CONTINUE
C
C   SET UP INTEGRATION DOUBLE LOOP
    DO 100 I=1,3
    DO 100 J=1,5
C
C   EVALUATE B MATRIX
C
C   ELEMENTS OF [A]
    H = AI(I)
    HR = AI(I)/(R**2)
    THETA = AJ(J)
    ST = SIN(THETA)
    CT = COS(THETA)
    B = THETA - P4
    B2 = B*B
    B3 = B2*B
    B4 = B3*B
    B5 = B4*B
    A(1,1) = 0.0D0
    A(1,2) = HR
    A(1,3) = 2.0D0*HR*B
    A(1,4) = HR*(6.0D0+3.0D0*B2)
    A(1,5) = HR*(4.0D0*B3+24.0D0*B)
    A(1,6) = HR*(5.0D0*B4+60.0D0*B2)
    A(2,1) = CT/TORR
    A(2,2) = (B*CT-ST)/TORR
    A(2,3) = (B2*CT-2.0D0*B*ST)/TORR
    A(2,4) = (B3*CT-3.0D0*B2*ST)/TORR
    A(2,5) = (B4*CT-4.0D0*B3*ST)/TORR
    A(2,6) = (B5*CT-5.0D0*B4*ST)/TORR
C
C   G MATRIX
    G(1,1) = -0.24543692610D0
    G(1,2) = -0.0385531421900D0*R
    G(1,3) = 0.24543692610D0
    G(2,1) = 7.0D0/16.0D0
    G(2,2) = 0.04908738520D0*R
    G(2,3) = G(2,1)
    G(3,1) = 0.47746482930D0
    G(3,2) = 1.0D0/8.0D0*R
    G(3,3) = -G(3,1)
    G(4,1) = -1.0132118360D0
    G(4,2) = -0.15915494310D0*R
    G(4,3) = G(4,1)
    G(5,1) = -0.12900613770D0
    G(5,2) = -0.10132118360D0*R
    G(5,3) = -G(5,1)
    G(6,1) = 0.49276714820D0
```

```
      G(6,2) = 0.12900613770D0*R
      G(6,3) = G(6,1)
      DO 35 IT=1,6
      G(IT,4) = -(G(IT,1)+G(IT,3))
   35 CONTINUE
C
C     AG MATRIX
C     [AG] = [A] [G]
      DO 40 IT=1,2
      DO 40 JT=1,4
      AG(IT,JT) = 0.0D0
      DO 40 KT=1,6
   40 AG(IT,JT) = AG(IT,JT) + A(IT,KT)*G(KT,JT)
C
C     PUT [AG] INTO [B]
      DO 45 IT=1,2
      DO 45 JT=1,4
   45 BMAT(IT,JT) = AG(IT,JT)
C
C     ADD TERM TO LAST COL OF [B]
      BMAT(1,4) = BMAT(1,4) + (R-H)/R*R
      BMAT(2,4) = BMAT(2,4) + ST/TORR
C     EVALUATE TRANSPOSE OF B
      DO 47 IT=1,2
      DO 47 JT=1,4
   47 BT(JT,IT)=BMAT(IT,JT)
C
C     EVALUATE THE [P] = [B]T [D] [B] MATRIX
C     [DB] = [D] [B]
      DO 50 IT=1,2
      DO 50 JT=1,4
      DB(IT,JT)=0.0
      DO 50 KT=1,2
   50 DB(IT,JT) = DB(IT,JT)+D(IT,KT)*BMAT(KT,JT)
C
C     [P] = [B]T [DB]
      DO 60 IT=1,4
      DO 60 JT=1,4
      P(IT,JT)=0.0
      DO 60 KT=1,2
   60 P(IT,JT) = P(IT,JT)+BT(IT,KT)*DB(KT,JT)
C
C     EVALUATE STIFFNESS MATRIX.
C     SET [STIFF] = [STIFF] PREVIOUS + CONST*WEIGHTS * [P]
      DO 70 K=1,4
      DO 70 L=1,4
      STIFF(K,L) = STIFF(K,L)+CONST*WI(I)*WJ(J)*P(K,L)
   70 CONTINUE
C
  100 CONTINUE
C
      U(1) = 1000.0
      U(2) = 0.0
      U(3) = 0.0
      U(4) = 0.0
      CALL SOLVE (STIFF,U)
      PRINT *,'LAMBDA =',TORR*THICK/(R**2)
C
C     EVALUATE THE STRESSES
C     FORM THE B MATRIX AT POSITION ANGLE PTHETA
      PTHETA = -5.0
C
      PRINT *,'TOP SURFACE'
      PRINT *,'  POS     SIG TH      SIG PH'
C
      H = 1.5*THICK
      DO 1000 IPOS=1,2
      H = H-THICK
      PRINT *,'H =',H
      DO 900 ITHETA=1,19
      PTHETA = PTHETA + 5.0
      PTHETA = PTHETA*3.1415927/180.
      PST = SIN(PTHETA)
      PCT = COS(PTHETA)
      H = T
      HR = H/(R**2)
      B = PTHETA - P4
      B2 = B*B
      B3 = B2*B
      B4 = B3*B
      B5 = B4*B
C
      A(1,1) = 0.0D0
      A(1,2) = HR
      A(1,3) = 2.0D0*HR*B
      A(1,4) = HR*(6.0D0+3.0D0*B2)
      A(1,5) = HR*(4.0D0*B3+24.0D0*B)
      A(1,6) = HR*(5.0D0*B4+60.0D0*B2)
      A(2,1) = PCT/TORR
      A(2,2) = (B*PCT-PST)/TORR
      A(2,3) = (B2*PCT-2.0D0*B*PST)/TORR
      A(2,4) = (B3*PCT-3.0D0*B2*PST)/TORR
      A(2,5) = (B4*PCT-4.0D0*B3*PST)/TORR
      A(2,6) = (B5*PCT-5.0D0*B4*PST)/TORR
C
C     AG MATRIX
C     [AG] = [A] [G]
      DO 640 IT=1,2
      DO 640 JT=1,4
      AG(IT,JT) = 0.0D0
      DO 640 KT=1,6
  640 AG(IT,JT) = AG(IT,JT) + A(IT,KT)*G(KT,JT)
C
C     PUT [AG] INTO [B]
      DO 645 IT=1,2
      DO 645 JT=1,4
  645 BMAT(IT,JT) = AG(IT,JT)
C
C     ADD TERM TO LAST COL OF [B]
      BMAT(1,4) = BMAT(1,4) + (R-H)/R*R
      BMAT(2,4) = BMAT(2,4) + PST/TORR
C
C     EVALUATE STRAINS {EP} = [B][U]
      EP(1) = BMAT(1,1)*U(1)+BMAT(1,2)*U(2)+BMAT(1,3)*U(3)
      EP(2) = BMAT(2,1)*U(1)+BMAT(2,2)*U(2)+BMAT(2,3)*U(3)
C
C     EVALUATE STRESSES {SIG} = [D] {EP}
      SCON = EX/(1.0-NUXY**2)
      SIG(1) = SCON*(EP(1)+NUXY*EP(2))
      SIG(2) = SCON*(EP(1)*NUXY+EP(2))
C
C     NORMALISE STRESSES BY DIVIDING BY SIGBO
      SIGBO = 3.0*R*1000.0/(3.1415927*TORR*THICK**2)
      SIG(1) = SIG(1)/SIGBO
      SIG(2) = SIG(2)/SIGBO
      PTHETA = PTHETA * 180/3.1415729
      POS = 90 - PTHETA
      WRITE (6,'(9F10.3)') POS,SIG(1),SIG(2)
```

```
900  CONTINUE
1000 CONTINUE
     END
C
     SUBROUTINE SOLVE (AMAT,CVEC)
C    *** SEE APPENDIX 3.3.1 ***
```

APPENDIX 4.

ANSYS USER ELEMENT CODE FOR ELEMENTS BEL1, PB1, PB2 AND PB3.

A4.1 ANSYS USER ELEMENT BEL1 SOURCE CODE.

```
      PROGRAM ANSYS
C
C     ANSYS VERSION 4.3A
C
C     ********** ANSYS USER ELEMENT CODE **************
C     ******* LINEAR ELASTIC BELLOWS ELEMENT ***********
C                     DONALD MACKENZIE, JAN 1989.
C
C     SOURCE CODE FOR A LINEAR ELASTIC 3-D BELLOWS ELEMENT
C     SUBJECT TO AXIAL FORCE LOADING.
C
C     BASED ON 6 FOURRIER TERM COEFFICIENT SOLUTION.
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      EXTERNAL MAIN,STOPER
      EXTERNAL ELSHFN
      CALL NNDIM
      CALL MAIN
      CALL STOPER
      END
```

A4.1.1 SUBROUTINE USEREL

```
      SUBROUTINE USEREL (ITYP,IPARM,KYSUB,KEY3D,KDOF,KUNSYM,KTRANS)
C     ******** DEFINE PARAMETERS FOR ANSYS USER ELEMENT ***********
      INTEGER IPARM(20,12),KYSUB(9),ITYP,JTYPE,KEY3D,KDOF,KUNSYM,KTRANS
C
C     DETERMINE TYPE OF ELEMENT AND THEN BYPASS IF NOT USER ELEMENT
      JTYPE = IPARM(ITYP,3)
      IF (JTYPE .NE. 100) GO TO 100
C
C        ********** SET 3-D KEY **********
      KEY3D = 1
C
C        ********** DEFINE DOF SET AT EACH NODE **********
      KDOF = 14
C
C        ********** SET UNSYMMETRIC MATRIX KEY **********
      KUNSYM = 0
C
C     DEFINE PATTERN FOR ELEMENT TO GLOBAL TRANSFORMATION
      KTRANS = 3
C
C        ********** DEFINE NUMBER OF NODES **********
      IPARM(ITYP,8) = 2
C
C        ***** DEFINE NUMBER OF TEMPERATURES (DELTEM,TEMPER)
      IPARM(ITYP,11) = 0
C
C        ***** DEFINE NUMBER OF PRESSURES (PRESS) *******
      IPARM(ITYP,6) = 0
C
C        ***** SET ZEROED VARIABLES (NOITUEP)
      IPARM(ITYP,12) = 0
C
C        ***** DEFINE NUMBER OF REAL CONSTANTS FOR ELEMENT (RVR)
      IPARM(ITYP,10) = 4
C
C        ***** DEFINE NUMBER OF VARIABLES TO BE SAVED (SVR) ****
      IPARM(ITYP,7) = 15
C
C        ***** DEFINE NUMBER OF ROWS IN ELEMENT MATRICES (KTIK)
      IPARM(ITYP,9) = 6
C
C        ********** SET KEY TO IDENTIFY NON-LINEAR ELEMENT ***
      IPARM(ITYP,4) = 0
C
C        ********** SET KEY FOR THERMAL ELEMENT (KAN,-1) *****
      IPARM(ITYP,1) = 0
  100 RETURN
      END
```

A4.1.2 SUBROUTINE USERPT

```
      SUBROUTINE USERPT (INODE,ITYPE,KSHAPE,NNODE)
C     ********** USER SUBROUTINE FOR ANSYS PLOT SHAPE **********
C        DEFINE ELEMENT SHAPE AND NUMBER OF NODES, FOR PLOTTING
      INTEGER INODE(20),JTYPE,KSHAPE,NNODE
C     *********BYPASS IF NOT USER ELEMENT (JTYPE = 100) *********
      IF (JTYPE .NE. 100) GO TO 100
C     ****** SELECT SHAPE TO BE PLOTTED BY SETTING KSHAPE *******
      KSHAPE = 2
C        ********** SET NUMBER OF ACTUAL NODES **********
      NNODE = 2
  100 RETURN
      END
C
```

A4.1.3 SUBROUTINE ST100

```
      SUBROUTINE ST100 (IELNUM,ITYP,KELIN,KELOUT,NR,KTIK,ZS,ZASS,DAMP,
     1   GSTIF,ZSC)
C     ******* STIFFNESS PASS FOR 3-D BELLOWS ELEMENT ********
C
      EXTERNAL TRACK,GETELD,PUTELD,PROPEV,NONTBL,VZERO,MHTCH,USEERR
      INTEGER I,J,K,I3,J3,NFKEY
C
      INTEGER IELNUM,ITYP,KELIN(6),KELOUT(6),NR,KTIK,
     1 KEYERR,IOUT,NSTEPS,KFSTLD,ITTER,ITIME,NCUMIT,KRSTRT,ISPARE,
     2 K13,NPRPVL,MATST,K5,K16,IPROP,KCPDS,
     3 K20,KAY,MODE,ISYM,KAHD,IDEBUG,IXXX,
     4 ITYPE,MAT,IELEM,NROW,JTYPE,IPLOT,IPRINT,KTEMTP,KCONCV,KBICNV,
     5 KEYPLS,KEYCRP,KEYSWL,KYSUB(9),K21,NODES(20),  EPAR(50)
      REAL  ERRVAR(5)
C
      DOUBLE PRECISION
     1 DPZERO,DPHALF,DPONE,DPTWO,DPTEN,DTORAD,RADTOD,
     2 TREF,TUNIF,TOFSET, DELTIM,TIME,TIMOLD,TIME2,TIME3,DELT2,
     3 ACEL,OMEGA,CGOMEG,CGLOC,DXXX,
     4 ELMASS,XCENTR,YCENTR,ZCENTR,TFCP,SUBEX,  ERPAR(20),
     5 XYZEQ(20,3),X(20),Y(20),Z(20),  ELVOL
C
      COMMON /STCOM/ DPZERO,DPHALF,DPONE,DPTWO,DPTEN,DTORAD,RADTOD,
     1 TREF,TUNIF,TOFSET, DELTIM,TIME,TIMOLD,TIME2,TIME3,DELT2,
     2 ACEL(3),OMEGA(6),CGOMEG(6),CGLOC(3),      DXXX(16),
     3 KEYERR,IOUT,NSTEPS,KFSTLD,ITTER,ITIME,NCUMIT,KRSTRT,ISPARE,
     4 K13,NPRPVL,MATST,K5,K16,IPROP(20),KCPDS,
     5 K20,KAY(10),MODE,ISYM,KAHD,IDEBUG(10),     IXXX(41)
C
      EQUIVALENCE  (ITYPE,EPAR(1)), (MAT,EPAR(2)), (IELEM,EPAR(5)),
     1 (NROW,EPAR(7)), (JTYPE,EPAR(11)), (IPLOT,EPAR(12)),
     2 (IPRINT,EPAR(13)), (KTEMTP,EPAR(14)), (KCONCV,EPAR(16)),
     4 (KBICNV,EPAR(17)), (KEYPLS,EPAR(18)), (KEYCRP,EPAR(19)),
     5 (KEYSWL,EPAR(20)), (KYSUB(1),EPAR(21)), (K21,EPAR(30)),
     6 (NODES(1),EPAR(31))
C
      EQUIVALENCE  (ELMASS,ERPAR(1)), (XCENTR,ERPAR(2)),
     1 (YCENTR,ERPAR(3)), (ZCENTR,ERPAR(4)), (TFCP,ERPAR(5)),
     2 (SUBEX,ERPAR(6))
      EQUIVALENCE (X(1),XYZEQ(1,1)),(Y(1),XYZEQ(1,2)),(Z(1),XYZEQ(1,3))
C
      DOUBLE PRECISION
     1 ZS(KTIK,KTIK),ZASS(KTIK,KTIK),ZSC(KTIK),
     2 ALEN2,ALENG,DX,DY,DZ,EX,DENS,NUXY
C
```

```
      DOUBLE PRECISION
     1 RVR(4),SVR(15),
     2 TR(3,3),CON,ALENN1,SALP1,CALP1,SALP2,CALP2,U(240)
C
      DOUBLE PRECISION RAD,TORAD,THK,NCOR,LAMBDA,BELEN,DPPI,LS4,
     1 AMAT(6,6),BVEC(6),CVEC(6),FLEX,SECMOM,VAL,D1,TORAD3
C
      EQUIVALENCE (RVR(1),RAD), (RVR(2),TORAD), (RVR(3),THK),
     1 (RVR(4),NCOR)
C
      EQUIVALENCE (SVR(1),EX), (SVR(2),NUXY), (SVR(4),ALEN2),
     1 (SVR(5),ALENG), (SVR(6),DX), (SVR(7),DY), (SVR(8),DZ),
     2 (SVR(9),CVEC(1)), (SVR(15),STCON)
C
      DPPI = 3.14159265358979 3D0
C
      CALL TRACK(5,'ST100 ')
C
      CALL GETELD (IELNUM,ITYP,EPAR(1),ERPAR(1),CON,CON,
     1 CON,CON,RVR(1),SVR(1),XYZEQ(1,1),U(1))
C
C     ***** INITIALIZE VARIABLES FIRST TIME THRU IF NEEDED ****
      IF (KFSTLD .EQ. 0) GO TO 100
      FORCE = DPZERO
      EPEL = DPZERO
  100 CONTINUE
C
      CALL PROPEV (IELEM,MAT,ITYPE, 1,0,EX  ,1)
      CALL PROPEV (IELEM,MAT,ITYPE,10,0,DENS,1)
      CALL PROPEV (IELEM,MAT,ITYPE, 3,0,NUXY,1)
C
  120 CONTINUE
C
C         ********** VERIFY GEOMETRY **********
      DX = X(2) - X(1)
      DY = Y(2) - Y(1)
      DZ = Z(2) - Z(1)
      CON = DX**2 + DY**2
      ALEN2 = CON + DZ**2
      IF (ALEN2 .GT. DPZERO) GO TO 150
      WRITE (IOUT,2000) IELEM
 2000 FORMAT(/' *** ERROR ***'/ '  ZERO LENGTH ELEMENT ',I5)
      KEYERR = 1
C
C THIS SUBROUTINE CALL IS USED TO PASS KEYERR TO COM2 FOR NORMAL ABORTS
C
      NFKEY = 1
      CALL USEERR (NFKEY)
      GO TO 990
  150 ALENG = DSQRT(ALEN2)
      ALENN1 = DSQRT(CON)
C
C     ********** CHECK BELLOWS LENGTH AGAINST TORAD AND NCOR
C     BELLOWS LENGTH SHOULD BE ALENG = 4*TORAD*NCOR
C     ACCEPT 1% ERROR
C
      BELEN = 4*TORAD*NCOR
C
C ***** CHECK LENGTH,TORAD AND NCOR ARE VALID ****
      IF (BELEN .LT. 0.99*ALENG) GO TO 160
      IF (BELEN .GT. 1.01*ALENG) GO TO 160
C
      GO TO 180
C
  160 WRITE(IOT,2001) IELEM
 2001 FORMAT(/' *** ERROR ***'/ ' INVALID LENGTH,TORAD OR NCOR ',I5)
      KEYERR = 1
      NFKEY = 1
      CALL USEERR (NFKEY)
      GO TO 990
C
  180 CONTINUE
C
      SECMOM = (4*DPPI*RAD*(THK/2)**3)/3
C
C         ********** CALCULATE MASS AND CENTROID ********
      XCENTR = (X(1) + X(2))*DPHALF
      YCENTR = (Y(1) + Y(2))*DPHALF
      ZCENTR = (Z(1) + Z(2))*DPHALF
      ELMASS = (4*(DPPI**2)*TORAD*RAD*THK*NCOR*DENS)
C         ********** RETURN IF ERROR(S) OR CHECK RUN **********
      IF ((NSTEPS .EQ. 0) .OR. (KEYERR.EQ.1)) GO TO 990
C
C         ********** FORM TR MATRIX **********
C
C           THE TR MATRIX IS THE LOCAL TO GLOBAL CONVERSION MATRIX
      IF (ALENN1 .GT. .0001*ALENG) GO TO 200
      SALP1 = DPZERO
      CALP1 = DPONE
      GO TO 250
  200 SALP1 = DY/ALENN1
      CALP1 = DX/ALENN1
  250 SALP2 = DZ/ALENG
      CALP2 = ALENN1/ALENG
      TR(1,1) = CALP1*CALP2
      TR(2,1) = -SALP1
      TR(3,1) = -CALP1*SALP2
      TR(1,2) = SALP1*CALP2
      TR(2,2) = CALP1
      TR(3,2) = - SALP1*SALP2
      TR(1,3) = SALP2
      TR(2,3) = DPZERO
      TR(3,3) = CALP2
C
C       ******* STIFFNESS MATRIX ********
      IF (KELIN(1) .NE. 1) GO TO 400
C
      LAMBDA = THK*RAD/TORAD**2
      LS4 = (LAMBDA**2)*4
C
      AMAT(1,1) = 0.1309D0 + (1.1781D0/LS4)
      AMAT(1,2) = 0.11111D0 + (1.65969D0/LS4)
      AMAT(1,3) = -0.044444D0 + (0.25418D0/LS4)
      AMAT(1,4) = 0.0285714D0 + (0.001814D0/LS4)
      AMAT(1,5) = -0.021164D0-(0.000641D0/LS4)
      AMAT(1,6) = -0.016835D0 + (0.000283D0/LS4)
      AMAT(2,2) = 0.5236D0 + (2.87168D0/LS4)
      AMAT(2,3) = 0.213333D0/LS4
      AMAT(2,4) = -0.12249D0/LS4
      AMAT(2,5) = 0.014109D0/LS4
      AMAT(2,6) = -0.012489D0/LS4
      AMAT(3,3) = 2.0944D0 + (0.173873D0/LS4)
      AMAT(3,4) = 0.0192593D0/LS4
      AMAT(3,5) = 0.017316D0/LS4
      AMAT(3,6) = 0.0D0
      AMAT(4,4) = 4.71239D0 + (0.070646D0/LS4)
      AMAT(4,5) = 0.028132D0/LS4
      AMAT(4,6) = 0.011005D0/LS4
      AMAT(5,5) = 8.37758 + (0.38751/LS4)
      AMAT(5,6) = 0.016433D0/LS4
      AMAT(6,6) = 13.089969D0 + (0.024577D0/LS4)
C
      BVEC(1) = 3.14159D0/4
      BVEC(2) = 2.0D0/3.0D0
      BVEC(3) = -4.0D0/15.0D0
      BVEC(4) = 6.0D0/35.0D0
      BVEC(5) = -8.0D0/63.0D0
      BVEC(6) = 10.0D0/99.0D0
C
C   SOLVE SIM EQUNS USING SUBROUTINE CVECSOL
C   INITIALLY EQUATE CVEC TO BVEC FOR SUB CALL
      DO 190 I = 1,6
  190 CVEC(I) = BVEC(I)
C
```

```
      CALL CVECSOL(AMAT(1,1),CVEC(1))
C
C     EVALUATE VAL:  VAL = {B}' {C}
      VAL = (BVEC(1)*CVEC(1)+BVEC(2)*CVEC(2)+BVEC(3)*CVEC(3)
     1 +BVEC(4)*CVEC(4)+BVEC(5)*CVEC(5)+BVEC(6)*CVEC(6))
C
C     SET UP STIFFNESS MATRIX AT END I IN ELEMENT COORDINATES
C
C     EVALUATE THE MATERIAL CONST D1
      TORAD3 = TORAD**3
      D1 = (1-NUXY**2)*TORAD3/(6*EX*SECMOM)
      STCON = 1/(NCOR*4*D1*VAL)
      CALL VZERO (ZS(1,1),36)
C
      ZS(1,1) = STCON
C
C     CONVERT 3 BY 3 MATRIX FROM ELEMENT TO GLOBAL COORDINATES.
C
      CALL MHTCH (TR(1,1),ZS(1,1), 3,KTIK, 3)
C
C     FILL OUT THE COMPLETE 6 X 6 MATRIX FROM THE COMPUTED 3X3 MATRIX
      DO 300 I = 1, 3
        I3 = I + 3
        DO 300 J = 1, 3
          J3 = J + 3
          ZS(I3,J) = -ZS(I,J)
          ZS(I,J3) = -ZS(I,J)
          ZS(I3,J3) = ZS(I,J)
  300   CONTINUE
C     SET KEY THAT MATRIX WAS INDEED COMPUTED.
      KELOUT(1) = 1
C
C       ********* MASS MATRIX *****
C     SIMPLE LUMPED MASS MATRIX
C
  400 IF (KELIN(2) .NE. 1) GO TO 800
      IF (DENS .EQ. DPZERO) GO TO 800
      CALL VZERO (ZASS(1,1),36)
      ZASS(1,1) = ELMASS/2
C     FILL OUT 12X12 MASS MATRIX
      ZASS(2,2) = ZASS(1,1)
      ZASS(3,3) = ZASS(1,1)
      ZASS(4,4) = ZASS(1,1)
      ZASS(5,5) = ZASS(1,1)
      ZASS(6,6) = ZASS(1,1)
C
      CALL MHTCH(ZASS(1,1),ZS(1,1),6,KTIK,6)
      KELOUT(2) = 1
  800 CONTINUE
C
  990 CALL PUTELD (IELNUM,EPAR(1),ERPAR(1),CON,SVR(1))
C     PUTELD RESTORES DATA BACK TO FILE2
      CALL TRACK( 15,'ST100 ')
      RETURN
      END
```

SUBROUTINE CVECSOL

```
      SUBROUTINE CVECSOL(AMAT,CVEC)
C     PROG TO SOLVE SYMM MATRICES BY THE GAUSS ELIM. METHOD.
C     REF.  BREBBIA/FERRANTE PROG13
C
C     SOLVES EQUN.  [A] {C} = {B} FOR {C}
C
      INTEGER I,NEQ,J,K,I_K1,NEQ1
      DOUBLE PRECISION AMAT(6,6),BVEC(6),CVEC(6),CVAL
C
C     INITIALLY EQUATE BVEC TO CVEC
C     NO OF ELEMENTS IN ARRAY = NEQ
      NEQ=6
C
      DO 1 I=1,NEQ
    1 BVEC(I)=CVEC(I)
C
      NEQ1=NEQ-1
      DO 100 K=1,NEQ1
      CVAL=AMAT(K,K)
      K1=K+1
      DO 11 J=K1,NEQ
   11 AMAT(J,K)=AMAT(K,J)/CVAL
      IF (ABS(CVAL)-1E-6)4,4,7
    4 WRITE (6,5) K
    5 FORMAT (' **** SINGULARITY IN ROW',I5)
      GO TO 300
C
C     DIV ROW BY DIAG COEFF
C
    7 DO 8 J=K1,NEQ
    8 AMAT(K,J)=AMAT(K,J)/CVAL
      CVEC(K)=CVEC(K)/CVAL
C
C     ELIMINATE ROW UNKNOWN X(K) FROM ROW I
C
      DO 10 I=K1,NEQ
      CVAL=AMAT(I,K)
      DO 9 J=I,NEQ
    9 AMAT(I,J)=AMAT(I,J)-CVAL*AMAT(K,J)
   10 CVEC(I)=CVEC(I)-CVAL*CVEC(K)
  100 CONTINUE
C
C     COMPUTE LAST UK
C
      IF(ABS(AMAT(NEQ,NEQ))-1E-6)1,1,101
  101 CVEC(NEQ)=CVEC(NEQ)/AMAT(NEQ,NEQ)
C
C     APPLY BACKSUB TO COMPUTE REMAINING UKS
C
      DO 200 L=1,NEQ1
      K=NEQ-L
      K1=K+1
      DO 200 J=K1,NEQ
  200 CVEC(K)=CVEC(K)-AMAT(K,J)*CVEC(J)
  300 CONTINUE
      END
```

A4.1.4 SUBROUTINE SR100

```
      SUBROUTINE SR100 (IELNUM,ITYP,KELOUT,ELVOL,KTIK,ZS,ZASS,ZSC)
C ********* STRESS PASS FOR 3-D BELLOWS ELEMENT **********
C
      EXTERNAL TRACK,GETELD,PUTELD,SRPLT,MAXV
      INTEGER IPLTAY(6),I,J,IP3,JP3,NEQ
C
      INTEGER  IELNUM,ITYP,KELIN(6),KELOUT(6),NR,KTIK,
     1 KEYERR,IOUT,NSTEPS,KFSTLD,ITTER,ITIME,NCUMIT,KRSTRT,ISPARE,
     2 K13,NPRPVL,MATST,K5,K16,IPROP,KCPDS,
     3 K20,KAY,MODE,ISYM,KAHD,IDEBUG,IXXX,
     4 ITYPE,MAT,IELEM,NROW,JTYPE,IPLOT,IPRINT,KTEMTP,KCONCV,KBICNV,
     5 KEYPLS,KEYCRP,KEYSWL,KYSUB(9),K21,NODES(20),  EPAR(50)
C
      REAL  ERRVAR(5)
C
      DOUBLE PRECISION
     1 DPZERO,DPHALF,DPONE,DPTWO,DPTEN,DTORAD,RADTOD,
     2 TREF,TUNIF,TOFSET, DELTIM,TIME,TIMOLD,TIME2,TIME3,DELT2,
     3 ACEL,OMEGA,CGOMEG,CGLOC,DXXX,
     4 ELMASS,XCENTR,YCENTR,ZCENTR,TFCP,SUBEX,  ERPAR(20),
     5 XYZEQ(20,3),X(20),Y(20),Z(20),  ELVOL
C
      COMMON /STCOM/ DPZERO,DPHALF,DPONE,DPTWO,DPTEN,DTORAD,RADTOD,
     1 TREF,TUNIF,TOFSET, DELTIM,TIME,TIMOLD,TIME2,TIME3,DELT2,
     2 ACEL(3),OMEGA(6),CGOMEG(6),CGLOC(3),       DXXX(16),
```

```
     3 KEYERR,IOUT,NSTEPS,KFSTLD,ITTER,ITIME,NCUMIT,KRSTRT,ISPARE,
     4 K13,NPRPVL,MATST,K5,K16,IPR0P(20),KCPDS,
     5 K20,KAY(10),MODE,ISYM,KAHD,IDEBUG(10),     DXX(41)
C
      EQUIVALENCE  (ITYPE,EPAR(1)), (MAT,EPAR(2)), (IELEM,EPAR(5)),
     1 (NROW,EPAR(7)), (JTYPE,EPAR(11)), (IPLOT,EPAR(12)),
     2 (IPRINT,EPAR(13)), (KTEMTP,EPAR(14)), (KCONCV,EPAR(16)),
     4 (KBICNV,EPAR(17)), (KEYPLS,EPAR(18)), (KEYCRP,EPAR(19)),
     5 (KEYSWL,EPAR(20)), (KYSUB(1),EPAR(21)), (K21,EPAR(30)),
     6 (NODES(1),EPAR(31))
C
      EQUIVALENCE  (ELMASS,ERPAR(1)), (XCENTR,ERPAR(2)),
     1 (YCENTR,ERPAR(3)), (ZCENTR,ERPAR(4)), (TFCP,ERPAR(5)),
     2 (SUBEX,ERPAR(6))
C
      EQUIVALENCE (X(1),XYZEQ(1,1)),(Y(1),XYZEQ(1,2)),(Z(1),XYZEQ(1,3))
C
      DOUBLE PRECISION
     1 ZS(KTIK,KTIK),ZASS(KTIK,KTIK),
     2 ALEN2,ALENG,DX,DY,DZ,EX,NUXY,DENS
C
      DOUBLE PRECISION
     1 RVR(4),SVR(15),
     2 U(24),POSTD(19),CON
C
      DOUBLE PRECISION RAD,TORAD,THK,NCOR,LAMBDA,DPPI,LS4,
     1 CVEC(6),CONST1,CONST2,CONST3,SIGTH,SIGPH0,SIGPH9,TR(6,6),
     2 FELEM(6),DUMMY(6),ETA(11),ANGLE,
     3 BMAT(2,6),UC(6),EP(2),SIG(2),D(2,2)
C
      EQUIVALENCE (RVR(1),TORR), (RVR(2),R), (RVR(3),THICK),
     1 (RVR(4),NCOR)
C
      EQUIVALENCE (SVR(1),EX), (SVR(2),NUXY), (SVR(4),ALEN2),
     1 (SVR(5),ALENG), (SVR(6),DX), (SVR(7),DY), (SVR(8),DZ),
     2 (SVR(9),CVEC(1)),(SVR(15),STCON)
C
      DPPI = 3.14159265358979 3D0
C
      CALL TRACK (5,'SR100 ')
      CALL VZERO(TR(1,1),36)
C
      CALL GETELD (IELNUM,ITYP,EPAR(1),ERPAR(1),CON,CON,
     1 CON,CON,RVR(1),SVR(1),XYZEQ(1,1),U(1))
C
      OPEN (UNIT = 40,FILE = 'BELRES',STATUS = 'NEW')
C
C     EVALUATE BELLOWS FORCE FROM DISPLACEMENT RESULTS
      DELX = U(4)-U(1)
      DELY = U(5)-U(2)
      DELZ = U(6)-U(3)
      DELTA = SQRT(DELX**2+DELY**2+DELZ**2)
C     F = K D
      FORCE = STCON*DELTA
      WRITE (40,*) 'FORCE'
      WRITE (40,*) FORCE
C
      PRINT *,'  POSN.   SIG AXIAL   SIG HOOP'
C
C     FOR INSIDE SURF, ZETA = T/2
C     FOR OUTSIDE, ZETA = -T/2
      ZETA = -THICK/2.
      POS = -5.0
      DO 250 ILOC = 1,19
      POS = POS + 5.0
      POS = POS*3.14159 27/180.
      PC1 = COS(POS)
      PC2 = COS(POS*2.)
      PC3 = COS(POS*3.)
      PC4 = COS(POS*4.)
      PC5 = COS(POS*5.)
      PC6 = COS(POS*6.)
      PC7 = COS(POS*7.)
      PC8 = COS(POS*8.)
      PC9 = COS(POS*9.)
      PC10 = COS(POS*10.)
      PC11 = COS(POS*11.)
C
      BMAT(1,1) = -ZETA*PC1/R
      BMAT(1,2) = -ZETA*2.*PC2/R
      BMAT(1,3) = -ZETA*4.*PC4/R
      BMAT(1,4) = -ZETA*6.*PC6/R
      BMAT(1,5) = -ZETA*8.*PC8/R
      BMAT(1,6) = -ZETA*10.*PC10/R
      BMAT(2,1) = -PC1*PC1*R*.5/TORR
      BMAT(2,2) = -(PC3/3.+1.)*R*.5/TORR
      BMAT(2,3) = -(PC5/5.+PC2/3.)*R*.5/TORR
      BMAT(2,4) = -(PC7/7.+PC4/5.)*R*.5/TORR
      BMAT(2,5) = -(PC9/9.+PC6/7.)*R*.5/TORR
      BMAT(2,6) = -(PC11/11.+PC8/9.)*R*.5/TORR
C
      DCON = EX/(1.-NUXY**2)
      D(1,1) = DCON
      D(1,2) = NUXY*DCON
      D(2,1) = NUXY*DCON
      D(2,2) = DCON
C
      DO 240 I = 1,6
      UC(I) = CVEC(I)*FORCE*(1.-NUXY**2)*R**2/
     & (EX*3.14159*TORR*THICK**3)
  240 CONTINUE
C     {EP} = [B] {C}
      EP(1) = BMAT(1,1)*UC(1) + BMAT(1,2)*UC(2) + BMAT(1,3)*UC(3)
     1 + BMAT(1,4)*UC(4) + BMAT(1,5)*UC(5) + BMAT(1,6)*UC(6)
      EP(2) = BMAT(2,1)*UC(1) + BMAT(2,2)*UC(2) + BMAT(2,3)*UC(3)
     1 + BMAT(2,4)*UC(4) + BMAT(2,5)*UC(5) + BMAT(2,6)*UC(6)
C
C     EVALUATE STRESSES {SIG} = [D] {EP}
      SIG(1) = (D(1,1)*EP(1) + D(1,2)*EP(2))
      SIG(2) = (D(2,1)*EP(1) + D(2,2)*EP(2))
C
      POS = POS * 180/3.1415729
      WRITE (40,'(9F12.3)') POS,SIG(1),SIG(2)
  250 CONTINUE
C
  300 CONTINUE
C
C     ****** WRITE POSTDATA FILE *******
  350 IF (IPLOT .NE. 1) GO TO 900
C           ***** NUMBER OF FORCES (LEVEL 1) *****
      IPLTAY(2) = 3
C           ***** NUMBER OF STRESSES (LEVEL 2) *****
      IPLTAY(3) = 1
C           ***** NUMBER OF TOTAL SAVED (LEVELS 1, 2, AND 3) *****
      IPLTAY(4) = 5
C           ***** SAVE GEOMETRY FOR CONTOURS (0,NO  1,YES) *****
      IPLTAY(6) = 0
C     ****** PUT POSTDATA INFORMATION INTO POSTD ******
      POSTD(1) = SIGTH
      POSTD(2) = SIGPH0
      POSTD(3) = SIGPH9
      IF (K21 .LE. 4) GO TO 400
      IPLTAY(4) = 7
      POSTD(6) = EPEL
      POSTD(7) = EPTH
  400 CONTINUE
C
C     ***** PUT PLTARY INFORMATION ONTO FILE 12 *******
      CALL SRPLT (IELEM,ITYP,NROW,MAT,100,2,U(1),NODES(1),XYZEQ(1,1),
     1 IPLTAY(1),POSTD(1))
  900 CALL PUTELD (IELNUM,EPAR(1),ERPAR(1),CON,SVR(1))
C     PUTELD RESTORES DATA BACK TO FILE3
```

```
CALL TRACK( 15,'SR100 ')
RETURN
END
```

A4.2 ANSYS USER ELEMENT PB1

```
      PROGRAM ANSYS
C
C               ELEMENT PB2
C               ***********
C     ANSYS USER ELEMENT CODE FOR PIPE BEND ELEMENT ELEMENT
C
C     BASED ON VLASOV THIN WALL CIRC CURVED BEAM SOLUTION
C     WITH SUPERPOSED VON KARMAN OVALISATION: 3 I/P ANS 3 O/P MODES.
C
C     CLOSED FORM INTEGRATION FOR STIFFNESS MATRIX.
C     MATRIX STATICALLY CONDENSED TO GIVE FINAL 12x12 STIFFNESS MATRIX.
C
C                  DONALD MACKENZIE  JAN 1989
C
C     ANSYS VERSION 4.3A
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      EXTERNAL MAIN,STOPER
      EXTERNAL ELSHFN
      CALL NNDIM
      CALL MAIN
      CALL STOPER
      END
```

A4.2.1 SUBROUTINE USEREL

```
      SUBROUTINE USEREL (ITYP,IPARM,KYSUB,KEY3D,KDOF,KUNSYM,KTRANS)
C
      INTEGER IPARM(20,12),KYSUB(9),ITYP,JTYPE,KEY3D,KDOF,KUNSYM,KTRANS
C     **** DETERMINE TYPE OF ELEMENT AND THEN BYPASS IF NOT USER ELEMENT****
      JTYPE = IPARM(ITYP,3)
      IF (JTYPE .NE. 100) GO TO 100
C
C        ********** SET 3-D KEY **********
      KEY3D = 1
C        ********** DEFINE DOF SET AT EACH NODE **********
      KDOF = 0
C        ********** SET UNSYMMETRIC MATRIX KEY **********
      KUNSYM = 0
C
C     ***** DEFINE PATTERN FOR ELEMENT TO GLOBAL TRANSFORMATION ***
      KTRANS = 2
C        ********** DEFINE NUMBER OF NODES **********
      IPARM(ITYP,8) = 2
C        ***** DEFINE NUMBER OF TEMPERATURES (DELTEM,TEMPER) ******
      IPARM(ITYP,11) = 2
C
C        ***** DEFINE NUMBER OF PRESSURES (PRESS) *******
C           IF THERMAL ANALYSIS, TWO TIMES NUMBER OF CONVECTION SURFACES
      IPARM(ITYP,6) = 1
C        ***** SET ZEROED VARIABLES (NOITUEP)
      IPARM(ITYP,12) = 0
C        ***** DEFINE NUMBER OF REAL CONSTANTS FOR ELEMENT (RVR) ********
      IPARM(ITYP,10) = 5
C
C        ***** DEFINE NUMBER OF VARIABLES TO BE SAVED (SVR) *****
      IPARM(ITYP,7) = 254
C        ***** DEFINE NUMBER OF ROWS IN ELEMENT MATRICES (KTIK) *****
      IPARM(ITYP,9) = 12
C        ********** SET KEY TO IDENTIFY NON-LINEAR ELEMENT **********
      IPARM(ITYP,4) = 0
C        ********** SET KEY FOR THERMAL ELEMENT (KAN,-1) **********
      IPARM(ITYP,1) = 0
  100 RETURN
      END
```

A4.2.2 SUBROUTINE USERPT

```
      SUBROUTINE USERPT (INODE,JTYPE,KSHAPE,NNODE)
C     ********** USER SUBROUTINE FOR ANSYS PLOT SHAPE **********
C     DEFINE ELEMENT SHAPE AND NUMBER OF NODES, FOR PLOTTING
      INTEGER INODE(20),JTYPE,KSHAPE,NNODE
C     *********BYPASS IF NOT USER ELEMENT (JTYPE = 100) *********
      IF (JTYPE .NE. 100) GO TO 100
C     ****** SELECT SHAPE TO BE PLOTTED BY SETTING KSHAPE *******
      KSHAPE = 2
C        ********** SET NUMBER OF ACTUAL NODES **********
      NNODE = 2
  100 RETURN
      END
```

A4.2.3 SUBROUTINE ST100

```
      SUBROUTINE ST100 (IELNUM,ITYP,KELIN,KELOUT,NR,KTIK,ZS,
     1 ZASS,DAMP,GSTIF,ZSC)
C
C     ******* STIFFNESS PASS FOR PIPE BEND ELEMENT ELEMENT *********
C           CLOSED FORM STIFFNESS MATRIX
C
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
C
C     -------- DECLARE EXTERNAL SUBROUTINES AND FUNCTIONS ----------
      EXTERNAL TRACK,GETELD,PUTELD,PROPEV,NONTBL,VZERO,MHTCH,USEERR,
     1 MATXV,MAXV,TRSUB,MAXB,MATXB,INVM,REDUCE
C
C     ------------ COMMON BLOCK DECLARATIONS ----------------
C     STANDARD ST100 INTEGER VARIABLES ASSOCIATED WITH COMMON:
      INTEGER  IELNUM,ITYP,KELIN(6),KELOUT(6),NR,KTIK,
     1 KEYERR,IOUT,NSTEPS,KFSTLD,ITTER,ITIME,NCUMIT,KRSTRT,ISPARE,
     2 K13,NPRPVL,MATST,K5,K16,IPROP,KCPDS,
     3 K20,KAY,MODE,ISYM,KAHD,IDEBUG,DXXX,
     4 ITYPE,MAT,IELEM,NROW,JTYPE,IPLOT,IPRINT,KTEMTP,KCONCV,KBICNV,
     5 KEYPLS,KEYCRP,KEYSWL,KYSUB(9),K21,NODES(20),  EPAR(50)
      REAL  ERRVAR(5)
C
C     STANDARD ST100 DOUBLE PRECISION VARIABLES ASSOCIATED WITH COMMON:
      DOUBLE PRECISION
     1 DPZERO,DPHALF,DPONE,DPTWO,DPTEN,DTORAD,RADTOD,
     2 TREF,TUNIF,TOFSET, DELTIM,TIME,TIMOLD,TIME2,TIME3,DELT2,
     3 ACEL,OMEGA,CGOMEG,CGLOC,DXXX,
     4 ELMASS,XCENTR,YCENTR,ZCENTR,TFCP,SUBEX,  ERPAR(20),
     5 XYZEQ(20,3),X(20),Y(20),Z(20),  ELVOL
C
C     ST100 COMMON BLOCK: STCOM
      COMMON /STCOM/ DPZERO,DPHALF,DPONE,DPTWO,DPTEN,DTORAD,RADTOD,
     1 TREF,TUNIF,TOFSET, DELTIM,TIME,TIMOLD,TIME2,TIME3,DELT2,
     2 ACEL(3),OMEGA(6),CGOMEG(6),CGLOC(3),       DXXX(16),
     3 KEYERR,IOUT,NSTEPS,KFSTLD,ITTER,ITIME,NCUMIT,KRSTRT,ISPARE,
     4 K13,NPRPVL,MATST,K5,K16,IPROP(20),KCPDS,
     5 K20,KAY(10),MODE,ISYM,KAHD,IDEBUG(10),     DXXX(41)
C
C     EQUIVALENCING OF STCOM VARIABLES
      EQUIVALENCE  (ITYPE,EPAR(1)), (MAT,EPAR(2)), (IELEM,EPAR(5)),
     1 (NROW,EPAR(7)), (JTYPE,EPAR(11)), (IPLOT,EPAR(12)),
     2 (IPRINT,EPAR(13)), (KTEMTP,EPAR(14)), (KCONCV,EPAR(16)),
     4 (KBICNV,EPAR(17)), (KEYPLS,EPAR(18)), (KEYCRP,EPAR(19)),
     5 (KEYSWL,EPAR(20)), (KYSUB(1),EPAR(21)), (K21,EPAR(30)),
     6 (NODES(1),EPAR(31))
      EQUIVALENCE  (ELMASS,ERPAR(1)), (XCENTR,ERPAR(2)),
     1 (YCENTR,ERPAR(3)), (ZCENTR,ERPAR(4)), (TFCP,ERPAR(5)),
     2 (SUBEX,ERPAR(6))
      EQUIVALENCE (X(1),XYZEQ(1,1)),(Y(1),XYZEQ(1,2)),(Z(1),XYZEQ(1,3))
C
C     ------------ USER DEFINED VARIABLES ----------------
C     INTEGER, REAL AND DOUBLE PRECISION VARIABLES DEFINED HERE BY USER.
```

```
C
C  DECLARE INTEGER VARIABLES:
      INTEGER I,J,K,I3,J3,I6,J6,LP(4),NSTR,NUM,KDEMO,NFKEY,
     1 KI,KO,LI,LO
C
C  DECLARE DOUBLE PRECISION VARIABLES:
      DOUBLE PRECISION ZS(KTIK,KTIK),ZASS(KTIK,KTIK),
     & ZSC(KTIK),TR(12,12),ZSTEMP(12,12),
     1 U(240),PROP(4),EX,ALPX,NUXY,DENS,ALEN2,ALENG,DX,DY,DZ,
     2 AFLU,TEMPER(2),CON,AVETEM,TRPROP(7),MPROP(1),LDPROP(1),DPPI,
     3 BRAD,ALPHA,PRAD,THICK,PHI,RVR(5),SVR(254)
      DOUBLE PRECISION ZSI(6,6),ZSO(6,6),BI(6,12),B1(6,6),B2(6,6),
     2 CA,SA,CM1,C2A,S2A,C2M1,TRIG1,TRIG2,
     3 AREA,POLMOM,SECMOM,EI,GJ,RATIO,D,B,DM1,DM2,
     4 DTERM1,DTERM2,DTERM3,NRINV,V,BTERM1,BTERM2,BB,BM1,BTERM3,
     5 BTERM4,H,CONST,THDIS(12)
      DOUBLE PRECISION CONOV,PR4,BRT2,ZSOV(3,3),ZSCUP(6,3),
     1 CUPMAT(6,3),ZSFULL(9,9),VALI(11),VALO(11),TEMP(12)
C
C --- USER EQUIVALENCING OF REAL AND SAVED VARIABLES (RVR, SVR) ----
C
C  EQUIVALENCE REAL VARIABLES RVR():
C  BRAD = BEND RAD,  ALPHA = BEND ANGLE,  PRAD = PIPE RAD,  THICK = WALL THK.
C  PHI = ORIENTATION ANGLE
      EQUIVALENCE (RVR(1),BRAD), (RVR(2),ALPHA), (RVR(3),PRAD),
     1 (RVR(4),THICK), (RVR(5),PHI)
C
C  EQUIVALENCE SAVED VARIABLES SVR():
      EQUIVALENCE (SVR(1),B1(1,1)),(SVR(37),B2(1,1)),
     1 (SVR(73),VALI(1)), (SVR(84),PROP(1)), (SVR(88),TR(1,1)),
     2 (SVR(232),VALO(1)), (SVR(243),THDIS(1))
C
C ---------- EQUIVALENCING OF MATERIAL PROPERTIES ----------
      EQUIVALENCE  (PROP(1),EX), (PROP(2),ALPX), (PROP(3),NUXY),
     1 (PROP(4),DENS)
C
      CALL TRACK(5,'ST100')
C
C  --- READING IN ELEMENT INFORMATION: SUBROUTINE GETELD ----
C
      CALL GETELD (IELNUM,ITYP,EPAR(1),ERPAR(1),CON,TEMPER(1),
     1 CON,CON,RVR(1),SVR(1),XYZBQ(1,1),U(1))
C
C   CONVERT BEND AND ORIENTATION ANGLES TO RADIANS.
      DATA DPPI / 3.14159265358979 3D0 /
      PHI = PHI*DPPI/180.0D0
      ALPHA = ALPHA*DPPI/180.0D0
C
C -- READING IN ELEMENT MATERIAL PROPERTIES: SUBROUTINE PROPEV --
C
C   SET UP INTEGER ARRAY FOR ACCESSING MATERIAL PROPERTIES
      DATA LP / 1, 2, 3,10/
C
      AVETEM = DPHALF*(TEMPER(1) + TEMPER(2))
C
      CALL PROPEV (IELEM,MAT,JTYPE,LP(1),AVETEM,PROP(1),4)
C
C ---------- GEOMETRY VALIDITY CHECK ----------
C
      DX = X(2) - X(1)
      DY = Y(2) - Y(1)
      DZ = Z(2) - Z(1)
      CON = DX**2 + DY**2
      ALEN2 = CON + DZ**2
      IF (ALEN2 .GT. 0.0) GO TO 150
      WRITE (IOUT,2000) IELEM                          WRITE
 2000 FORMAT ('ZERO LENGTH ELEMENT' ,I5)
      KEYERR = 1
      NFKEY = 1
      CALL USEERR (NFKEY)                              USEERR
      GO TO 990
C
  150 CONTINUE
C
C ---------- CALCULATE MASS AND CENTROID ----------
C
C   STILL TO BE DONE: EXISTING IS FOR STRAIGHT BEAM
      XCENTR = (X(1) + X(2))*DPHALF
      YCENTR = (Y(1) + Y(2))*DPHALF
      ZCENTR = (Z(1) + Z(2))*DPHALF
      AREA = 2*DPPI*PRAD*THICK
      AFLU = DPPI*(PRAD-(THICK/2))**2
      ELMASS =  (DENS*AREA+DENSFL*AFLU)*BRAD*ALPHA
C
C ---------- END OF CHECK RUN OR ERROR DETECTED ----------
C
      IF ((NSTEPS .EQ. 0) .OR. (KEYERR.EQ.1)) GO TO 990
C
C -------- EVALUATE THE ELEMENT TRANSFORMATION MATRIX ----------
C TRANSFORMATION MATRIX TR IS EVALUATED IN THE USER DEFINED
C SUBROUTINE TRSUB.  INFORMATION REQUIRED TO CALCULATE MATRIX VALUES
C IS PASSED IN BY ARRAY TRPROP.
C
      TRPROP(1) = DX
      TRPROP(2) = DY
      TRPROP(3) = DZ
      TRPROP(4) = PHI
      TRPROP(5) = ALPHA
      TRPROP(6) = CON
      TRPROP(7) = ALEN2
C
C ZERO THE TR MATRIX.
      CALL VZERO (TR(1,1),144)
C
      CALL TRSUB (TR,TRPROP)
C
C ---------- EVALUATE THE ELEMENT STIFFNESS MATRIX ----------
C   ELEMENT CB2.
C   CURVED CYLINDRICAL THIN WALLED BEAM ELEMENT STIFFNESS MATRIX
C   CLOSED FORM STIFFNESS MATRIX
C
C CHECK MATRIX IS REQUIRED.
      IF (KELIN(1) .NE. 1) GO TO 400
      CALL VZERO (ZS(1,1),144)
C
C   EVALUATE CONSTANTS AND TRIG FUNCTIONS.
      CA = COS(ALPHA)
      SA = SIN(ALPHA)
      CM1 = CA-1.0
      C2A = COS(2.0*ALPHA)
      S2A = SIN(2.0*ALPHA)
      C2M1 = C2A-1.0
      TRIG1 = 2.0*ALPHA-S2A
      TRIG2 = 2.0*ALPHA+S2A
C
      AREA = 2.0*DPPI*PRAD*THICK
      POLMOM = AREA*PRAD**2
      SECMOM = POLMOM/2.0
      EI = EX*SECMOM
      GJ = EX*POLMOM/(2.0+2.0*NUXY)
C
      RATIO = (PRAD/BRAD)**2
      D = RATIO/2.0
      B = 2.0*EI/(EI+GJ)
C
      DM1 = 1.0-D
      DM2 = 1.0-2.0*D
      DTERM1 = 2.0*DM1/BRAD
      DTERM2 = ALPHA*SA+DM2*CA
      DTERM3 = -ALPHA*CA+DM2*SA
C
      NRINV = -1.0/BRAD
      V = 1/(2.0*(1.0+NUXY))
C
```

```
      BB = B**2
      BM1 = 1.0-B
      BTERM1 = (2.0-B)**2
      BTERM3 = (-ALPHA*SA+CA*BM1)/BRAD
      BTERM4 = (ALPHA*CA+SA*BM1)/BRAD
C
      H = 1.0+RATIO/2.0
      CONST = EI/BRAD**3
      CONST = CONST/(1.0-NUXY**2)
      CONOV = CONST*ALPHA/RATIO
C
C *** IN PLANE STIFFNESS MATRIX ***
C
C     EVALUATE [BI] = [C1]
      CALL VZERO (BI(1,1),72)
C
      BI(1,3) = DM2
      BI(1,4) = -1.0
      BI(1,6) = 1.0
      BI(2,1) = 1.0
      BI(2,2) = 1.0
      BI(3,3) = -DTERM1
      BI(3,6) = NRINV
      BI(4,1) = ALPHA
      BI(4,2) = SA
      BI(4,3) = DTERM2
      BI(4,4) = -CA
      BI(4,5) = DTERM3
      BI(4,6) = 1.0
      BI(5,1) = 1.0
      BI(5,2) = CA
      BI(5,3) = ALPHA*CA
      BI(5,4) = SA
      BI(5,5) = ALPHA*SA
      BI(6,1) = ALPHA*NRINV
      BI(6,3) = -DTERM1*CA
      BI(6,5) = -DTERM1*SA
      BI(6,6) = NRINV
C
C     INVERT BI
      CALL INVM(BI,DET1,6,6,12)
C
C     EQUATE FIRST 6X6 OF BI WITH B1
      DO 322 I = 1,6
      DO 322 J = 1,6
  322 B1(I,J) = BI(I,J)
C     OPEN FILE DON.DAT FOR STORING STRESS RESULTS
      OPEN (UNIT = 40,FILE = 'PB2RES',STATUS = 'NEW')
C
C     EVALUATE ZSI = IN-PLANE INTEGRAL MATRIX
      CALL VZERO(ZSI(1,1),36)
C
      ZSI(1,1) = CONST*ALPHA
      ZSI(1,3) = CONST*2.0*CM1
      ZSI(1,5) = CONST*2.0*SA
      ZSI(3,1) = ZSI(1,3)
      ZSI(3,3) = CONST*H*TRIG1
      ZSI(3,5) = CONST*H*C2M1
      ZSI(5,1) = ZSI(1,5)
      ZSI(5,3) = ZSI(3,5)
      ZSI(5,5) = CONST*H*TRIG2
C
C     EVALUATE I/P STIFFNESS [ZSI] = [B1]'[ZSI] [B1]
      CALL MHTCH(B1(1,1),ZSI(1,1),6,6,6)
C
C     EVALUATE OVALISATION MATRIX ZSOV
      CALL VZERO(ZSOV(1,1),9)
C
      PR4 = PRAD**4
      BRT2 = (BRAD*THICK)**2
      ZSOV(1,1) = CONOV*(3.0*BRT2/(4.0*PR4) + 5.0/8.0)
      ZSOV(1,2) = CONOV*5.0/32.0
      ZSOV(2,1) = ZSOV(1,2)
      ZSOV(2,2) = CONOV*(75.0*BRT2/(4.0*PR4) + 17.0/32.0)
      ZSOV(2,3) = CONOV*7.0/32.0
      ZSOV(3,2) = ZSOV(2,3)
      ZSOV(3,3) = CONOV*(1225.0*BRT2/(12.0*PR4) + 37.0/72.0)
C
C     EVALUATE THE IN PLANE COUPLING MATRIX ZSCUP
      CALL VZERO(ZSCUP(1,1),18)
      CALL VZERO(CUPMAT(1,1),18)
      CONIPC = CONST * 3.0*BRAD/(2.0*PRAD)
      CUPMAT(1,1) = CONIPC*(-ALPHA/2.0)
      CUPMAT(3,1) = CONIPC*(1.0-CA)
      CUPMAT(5,1) = CONIPC*(-SA)
C
      CALL MATXB (B1(1,1),CUPMAT(1,1),ZSCUP,6,6,6,6,3,6)
C
C     FILL IN 9X9 I/P MATRIX ZSFULL
      DO 325 I = 1,6
      DO 325 J = 1,6
  325 ZSFULL(I,J) = ZSI(I,J)
      DO 326 I = 1,6
      DO 326 J = 1,3
      JP6 = J+6
  326 ZSFULL(I,JP6) = ZSCUP(I,J)
      DO 327 I = 1,3
      IP6 = I+6
      DO 327 J = 1,6
  327 ZSFULL(IP6,J) = ZSCUP(J,I)
      DO 328 I = 1,3
      IP6 = I+6
      DO 328 J = 1,3
      JP6 = J+6
  328 ZSFULL(IP6,JP6) = ZSOV(I,J)
C
      CALL VZERO(ZSI(1,1),36)
      CALL REDUCE (ZSFULL,ZSI)
C
C     EQUATE NON-ZERO TERMS FROM LAST 3 ROWS OF ZSFULL AFTER
C     REDUCTION TO THE ARRAY VALI
      VALI(1) = ZSFULL(7,1)
      VALI(2) = ZSFULL(7,2)
      VALI(3) = ZSFULL(7,3)
      VALI(4) = ZSFULL(7,4)
      VALI(5) = ZSFULL(7,5)
      VALI(6) = ZSFULL(7,6)
      VALI(7) = ZSFULL(7,7)
      VALI(8) = ZSFULL(7,8)
      VALI(9) = ZSFULL(8,8)
      VALI(10) = ZSFULL(8,9)
      VALI(11) = ZSFULL(9,9)
C
C
C *** OUT OF PLANE STIFFNESS ***
C
C     EVALUATE BI = C2
C
      CALL VZERO(BI(1,1),72)
C
      BI(1,1) = 1.0
      BI(1,4) = B
      BI(1,5) = -1.0
      BI(2,1) = 1/BRAD
      BI(2,2) = BM1/BRAD
      BI(3,3) = 1/BRAD
      BI(3,6) = NRINV
      BI(4,1) = CA
      BI(4,2) = ALPHA*CA-SA*B
      BI(4,3) = SA
      BI(4,4) = ALPHA*SA+CA*B
      BI(4,5) = -1.0
      BI(4,6) = -ALPHA
      BI(5,1) = CA/BRAD
```

```
      BI(5,2) = ALPHA*CA/BRAD
      BI(5,3) = SA/BRAD
      BI(5,4) = ALPHA*SA/BRAD
      BI(6,1) = SA*NRINV
      BI(6,2) = BTERM3
      BI(6,3) = CA/BRAD
      BI(6,4) = BTERM4
      BI(6,6) = NRINV
C
C     INVERT BI
C
      CALL INVM(BI,DET1,6,6,12)
C     EQUATE FIRST 6X6 OF BI TO B2
      DO 332 I = 1,6
      DO 332 J = 1,6
  332 B2(I,J) = BI(I,J)
C
C     EVALUATE ZSO = O/P INTEGRAL MATRIX
      CALL VZERO(ZSO(1,1),36)
C
      ZSO(2,2) = CONST*(BTERM1*0.25*TRIG1 + BB*V*0.5*TRIG2)
      ZSO(2,4) = CONST*(BB*V*0.5-BTERM1*0.25)*(1-C2A)
      ZSO(2,6) = CONST*2.0*B*V*SA
      ZSO(4,2) = ZSO(2,4)
      ZSO(4,4) = CONST*(BTERM1*0.25*TRIG2 + BB*V*0.5*TRIG1)
      ZSO(4,6) = CONST*2*B*V*(1-CA)
      ZSO(6,2) = ZSO(2,6)
      ZSO(6,4) = ZSO(4,6)
      ZSO(6,6) = CONST*2.0*ALPHA*V
C
C     EVALUATE O/P STIFFNESS [ZSO] = [BI]'[ZSO] [BI]
      CALL MHTCH(B2(1,1),ZSO(1,1),6,6,6)
C
C     EVALUATE THE OUT OF PLANE COUPLING MATRIX ZSCUP
      CALL VZERO(ZSCUP(1,1),18)
      CALL VZERO(CUPMAT(1,1),18)
      CONOPC = CONST * 3.0*BRAD/(4.0*PRAD)*(2.0-B)
      CUPMAT(2,1) = CONOPC*(CA-1.0)
      CUPMAT(4,1) = CONOPC*(SA)
C
      CALL MATXB (B2(1,1),CUPMAT(1,1),ZSCUP,6,6,6,6,3,6)
C
C     FILL IN 9X9 I/P MATRIX ZSFULL
      DO 335 I = 1,6
      DO 335 J = 1,6
  335 ZSFULL(I,J) = ZSO(I,J)
      DO 336 I = 1,3
      DO 336 J = 1,3
      JP6 = J + 6
  336 ZSFULL(I,JP6) = ZSCUP(I,J)
      DO 337 I = 1,3
      IP6 = I + 6
      DO 337 J = 1,6
  337 ZSFULL(IP6,J) = ZSCUP(J,I)
      DO 338 I = 1,3
      IP6 = I + 6
      DO 338 J = 1,3
      JP6 = J + 6
  338 ZSFULL(IP6,JP6) = ZSOV(I,J)
C
      CALL VZERO(ZSO(1,1),36)
      CALL REDUCE (ZSFULL,ZSO)
C
C
C     EQUATE NON-ZERO TERMS FROM LAST 3 ROWS OF 9X9 ZSFULL
C     AFTER REDUCTION TO THE ARRAY VALO
      VALO(1) = ZSFULL(7,1)
      VALO(2) = ZSFULL(7,2)
      VALO(3) = ZSFULL(7,3)
      VALO(4) = ZSFULL(7,4)
      VALO(5) = ZSFULL(7,5)
      VALO(6) = ZSFULL(7,6)
      VALO(7) = ZSFULL(7,7)
      VALO(8) = ZSFULL(7,8)
      VALO(9) = ZSFULL(8,8)
      VALO(10) = ZSFULL(8,9)
      VALO(11) = ZSFULL(9,9)
C
C     ASSEMBLE FULL 12x12 REDUCED STIFFNESS MATRIX
C     I/P TERMS IN I-J ODD LOCATIONS, O/P TERMS IN EVEN
C
C     ZERO ZS
      CALL VZERO (ZS(1,1),144)
      DO 360 I = 1,6
       KO = 2*I-1
       KE = 2*I
        DO 360 J = 1,6
         LO = 2*J-1
         LE = 2*J
      ZS(KO,LO) = ZSI(I,J)
      ZS(KE,LE) = ZSO(I,J)
  360 CONTINUE
C
C ---------- STIFFNESS MATRIX TRANSFORMATION ----------
C ELEMENT MATRICES ARE TRANSFORMED TO THE GLOBAL CO-ORD.
C SYSTEM BY THE ANSYS SUBROUTINE MHTCH.
      CALL MHTCH (TR(1,1),ZS(1,1), KTIK,KTIK, KTIK)
C
C     SET KEY THAT MATRIX WAS COMPUTED.
      KELOUT(1) = 1
C
  400 CONTINUE
C
C ------ EVALUATE ELEMENT LOAD VECTOR -------
C
C CHECK MATRIX IS REQUIRED.
  800 IF (KELIN(5) .NE. 1) GO TO 990
C
      CALL VZERO (TEMP(1),12)
      CALL VZERO (ZSC(1),12)
C
      ZSC(3) = BRAD*ALPX*(AVETEM-TREF)
      ZSC(9) = ZSC(3)
C
      CALL VZERO (TEMP(1),12)
C
      CALL MATXV(TR(1,1),ZSC(1),TEMP,12,12)
C
      DO 803 I = 1,12
      THDIS(I) = TEMP(I)
  803 TEMP(I) = -TEMP(I)
C
      CALL VZERO (ZSC(1),12)
C
      CALL MAXV(ZS(1,1),TEMP(1),ZSC,12,12)
C
C     SET KEY THAT MATRIX WAS COMPUTED.
      KELOUT(5) = 1
C
C ---------- OUTPUT ELEMENT DATA TO FILE 12 ----------
C ELEMENT DATA IS OUTPUT TO FILE 12 BY THE SUBROUTINE PUTELD.
C STANDARD:
  990 CALL PUTELD (IELNUM,EPAR(1),ERPAR(1),CON,SVR(1))
C
      CALL TRACK( 15,'ST100 ')
      RETURN
      END
C
```

SUBROUTINE REDUCE

```
      SUBROUTINE REDUCE (SE,RED)
C
C     STATIC CONDENSATION OF STIFFNESS MATRICES
C     REDUCES I/P AND O/P 9x9 MATRICES TO 6x6.
C     BASED ON GAUSSIAN ELIMINATION: REF. COOK.
C
C     NO REDUCTION OF FORCE VECTOR REQUIRED AS NODELESS
C     DOF HAVE ZERO CORRESPONDING GENERALISED FORCES.
C
C     FULL, SYMMETRIC STIFFNESS MATRIX SE.
C     CONDENSATION OPERATIONS ON LOWER TRIANGLE OF SE.
C     SIZE OF FULL MATRIX = NSIZE
C     NUMBER OF DOF TO BE REDUCED = NUM.  DOF TO BE REDUCED
C     STORED IN LAST NUM DOF.
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DOUBLE PRECISION SE(9,9),RED(6,6)
C
      NSIZE = 9
      NUM = 3
C     CONDENSATION OF LOWER TRIANGLE OF SE.
      DO 30 K=1,NUM
      LL = NSIZE - K
      KK = LL + 1
      DO 20 L=1,LL
      IF (SE(KK,L) .EQ. 0.0) GO TO 20
      DUM = SE(KK,L)/SE(KK,KK)
      DO 10 M=1,L
   10 SE(L,M) = SE(L,M) - SE(KK,M)*DUM
   20 CONTINUE
   30 CONTINUE
C     FILL IN THE UPPER TRIANGLE BY SYMMETRY.
      DO 40 K=1,LL
      DO 40 L=1,K
   40 SE(L,K) = SE(K,L)
C
C     EQUATE FIRST 6 ROWS AND COLUMNS OF SE TO RED
C     RED IS THE REDUCED IN-PLANE MATRIX
      DO 50 I=1,6
      DO 50 J=1,6
   50 RED(I,J) = SE(I,J)
      RETURN
      END
C
```

SUBROUTINE TRSUB

```
      SUBROUTINE TRSUB  (TR,TRPROP)
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
C
      EXTERNAL VZERO,MAXB,MATXB
C
      DOUBLE PRECISION T(3,3),TI(3,3),TJ(3,3),TR(12,12),
     1 THETA(3,3),TRPROP(7),DX,DY,DZ,PHI,ALPHA,CON,ALEN2,
     2 SALP1,SALP2,SALP3,CALP1,CALP2,CALP3
C
      DX = TRPROP(1)
      DY = TRPROP(2)
      DZ = TRPROP(3)
      PHI = TRPROP(4)
      ALPHA = TRPROP(5)
      CON = TRPROP(6)
      ALEN2 = TRPROP(7)
C
      CALL VZERO (T(1,1),9)
      CALL VZERO (THETA(1,1),9)
      CALL VZERO (TI(1,1),9)
      CALL VZERO (TJ(1,1),9)
C
      ALENG = SQRT(ALEN2)
      ALENN1 = SQRT(CON)
C
      IF (ALENN1 .GT. .0001*ALENG) GO TO 200
      SALP1 = 0.0
      CALP1 = 1.0
      GO TO 250
  200 SALP1 = DY/ALENN1
      CALP1 = DX/ALENN1
  250 SALP2 = DZ/ALENG
      CALP2 = ALENN1/ALENG
      SALP3 = SIN(PHI)
      CALP3 = COS(PHI)
C     DEFINE EQUIVALENT STRAIGHT BEAM 3X3 TR MATRIX
      T(1,1) = CALP1*CALP2
      T(2,1) = -CALP1*SALP2*SALP3-SALP1*CALP3
      T(3,1) = SALP1*SALP3-CALP1*SALP2*CALP3
      T(1,2) = SALP1*CALP2
      T(2,2) = CALP1*CALP3-SALP1*SALP2*SALP3
      T(3,2) = -CALP1*SALP3-SALP1*SALP2*CALP3
      T(1,3) = SALP2
      T(2,3) = CALP2*SALP3
      T(3,3) = CALP2*CALP3
C     DEFINE CURVE-STRAIGHT BEAM NODE ROTATION MATRIX THETA
      THETA(1,1) = COS(ALPHA/2)
      THETA(3,1) = SIN(ALPHA/2)
      THETA(2,2) = 1.0
      THETA(1,3) = -(THETA(3,1))
      THETA(3,3) = THETA(1,1)
C     MULTIPLY T MATRIX BY THETA AND THETA TRANSPOSE RESP TO GET 3X3
C     NODAL TR MATRICES TI AND TJ.
C     SUBROUTINES ATIMB AND ATIMB USED FOR MULT.
C
C     MAXB AND MATXB ARE ANSYS IN-HOUSE MATRIX ROUTINES.
      CALL MAXB(THETA(1,1),T(1,1),TI(1,1),3,3,3,3,3,3)
      CALL MATXB(THETA(1,1),T(1,1),TJ(1,1),3,3,3,3,3,3)
C
C FILL OUT 12X12 TR MATRIX FROM TI AND TJ
      DO 260 I=1,3
         I3=I+3
           DO 260 J=1,3
             J3=J+3
         TR(I,J) = TI(I,J)
         TR(I3,J3) = TI(I,J)
  260 CONTINUE
      DO 270 I=1,3
        I6 = I+6
        I9 = I+9
          DO 270 J=1,3
            J6=J+6
            J9=J+9
        TR(I6,J6) = TJ(I,J)
        TR(I9,J9) = TJ(I,J)
  270 CONTINUE
      END
```

SUBROUTINE INVM

```
      SUBROUTINE INVM(A,D,N,NX,MX)
C
C     THIS PROGRAM COMPUTE THE INVERSE OF A MATRIX
C     USING THE GAUSS ELIMINATION METHOD
C
C     A : RECTANGULAR ARRAY OF SIZE  N X 2N
C     D : DETERMINANT
C     N : ORDER OF A
C     NX: ROW
C     MX: COLUMN
C
```

```
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION A(NX,MX)
      NM1 = N-1
      NP1 = N+1
      NX2 = N*2
C
C     PUT A UNIT MATRIX IN THE ARGUMENTED PART OF A
C
      DO 2 I = 1,N
       IPN = I+N
       DO 1 J = 1,N
        JPN = J+N
    1   A(I,JPN) = 0.0
    2  A(I,IPN) = 1
C
C     APPLY THE ELIMINATION PROCESS
C
      DO 10 K = 1,NM1
      KP1 = K+1
       C = A(K,K)
       IF (ABS(C)-1.0E-10) 3,3,8
    3  DO 6 J = KP1,N
        IF (ABS(A(J,K))-1.0E-10) 6,6,4
    4   DO 5 L = K,NX2
         C = A(K,L)
         A(K,L) = A(J,L)
    5    A(J,L) = C
        C = A(K,K)
        GOTO 8
    6   CONTINUE
    7  WRITE(6,6999) K
       D = 0.0
       GOTO 15
    8  DO 9 J = KP1,NX2
    9   A(K,J) = A(K,J)/C
       DO 10 I = KP1,N
        C = A(I,K)
        DO 10 J = KP1,NX2
   10     A(I,J) = A(I,J)-C*A(K,J)
      IF (ABS(A(N,N))-1.0E-10) 7,7,11
   11 DO 12 J = NP1,NX2
   12  A(N,J) = A(N,J)/A(N,N)
C
C     APPLY THE BACKSUBSTITUTION PROCESS
C
      DO 13 L = 1,NM1
       K = N-L
       KP1 = K+1
       DO 13 I = NP1,NX2
        DO 13 J = KP1,N
   13     A(K,J) = A(K,J)-A(K,J)*A(J,I)
C
C     PUT THE INVERSE IN THE FIRST  N X N  POSITIONS
C
      DO 14 I = 1,N
       DO 14 J = 1,N
        JPN = J+N
   14   A(I,J) = A(I,JPN)
   15 RETURN
 6999 FORMAT(24H **** SINGULARITY IN ROW,I2)
      END
```

A4.2.4 SUBROUTINE SR100

```
      SUBROUTINE SR100 (IELNUM,ITYP,KELOUT,ELVOL,KTIK,ZS,ZASS,ZSC)
C
C ********* STRESS PASS FOR 3-D ELBOW ELEMENT ********
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      EXTERNAL TRACK,GETELD,PUTELD,SRPLT,MAXV,VZERO
      INTEGER IPLTAY(6),IEP
C
C ***** START STCOM STORAGE ********
      INTEGER IELNUM,ITYP,KELIN(6),KELOUT(6),NR,KTIK,
     1 KEYERR,IOUT,NSTEPS,KFSTLD,ITTER,ITIME,NCUMIT,KRSTRT,ISPARE,
     2 K13,NPRPVL,MATST,K5,K16,IPROP,KCPDS,
     3 K20,KAY,MODE,ISYM,KAHD,IDEBUG,DXXX,
     4 ITYPE,MAT,IELEM,NROW,JTYPE,IPLOT,IPRINT,KTEMTP,KCONCV,KBICNV,
     5 KEYPLS,KEYCRP,KEYSWL,KYSUB(9),K21,NODES(20),  EPAR(50)
      REAL  ERRVAR(5)
C
      DOUBLE PRECISION
     1 DPZERO,DPHALF,DPONE,DPTWO,DPTEN,DTORAD,RADTOD,
     2 TREF,TUNIF,TOFSET, DELTIM,TIME,TIMOLD,TIME2,TIME3,DELT2,
     3 ACEL,OMEGA,CGOMEG,CGLOC,DXXX,
     4 ELMASS,XCENTR,YCENTR,ZCENTR,TFCP,SUBEX,  ERPAR(20),
     5 XYZEQ(20,3),X(20),Y(20),Z(20),  ELVOL
C
      COMMON /STCOM/ DPZERO,DPHALF,DPONE,DPTWO,DPTEN,DTORAD,RADTOD,
     1 TREF,TUNIF,TOFSET, DELTIM,TIME,TIMOLD,TIME2,TIME3,DELT2,
     2 ACEL(3),OMEGA(6),CGOMEG(6),CGLOC(3),      DXXX(16),
     3 KEYERR,IOUT,NSTEPS,KFSTLD,ITTER,ITIME,NCUMIT,KRSTRT,ISPARE,
     4 K13,NPRPVL,MATST,K5,K16,IPROP(20),KCPDS,
     5 K20,KAY(10),MODE,ISYM,KAHD,IDEBUG(10),     DXXX(41)
C
      EQUIVALENCE  (ITYPE,EPAR(1)), (MAT,EPAR(2)), (IELEM,EPAR(5)),
     1 (NROW,EPAR(7)), (JTYPE,EPAR(11)), (IPLOT,EPAR(12)),
     2 (IPRINT,EPAR(13)), (KTEMTP,EPAR(14)), (KCONCV,EPAR(16)),
     4 (KBICNV,EPAR(17)), (KEYPLS,EPAR(18)), (KEYCRP,EPAR(19)),
     5 (KEYSWL,EPAR(20)), (KYSUB(1),EPAR(21)), (K21,EPAR(30)),
     6 (NODES(1),EPAR(31))
C
      EQUIVALENCE  (ELMASS,ERPAR(1)), (XCENTR,ERPAR(2)),
     1 (YCENTR,ERPAR(3)), (ZCENTR,ERPAR(4)), (TFCP,ERPAR(5)),
     2 (SUBEX,ERPAR(6))
      EQUIVALENCE (X(1),XYZEQ(1,1)),(Y(1),XYZEQ(1,2)),(Z(1),XYZEQ(1,3))
C
      DOUBLE PRECISION                                *UPD*
     1 ZS(KTIK,KTIK),ZASS(KTIK,KTIK),ZSC(KTIK),
     2 BRAD,ALPHA,PRAD,THICK,PHI,EX,ALPX,NUXY,DENS,DPPI,PROP(4),
     3 RVR(5),SVR(254),PRESS(1),B1(6,6),B2(6,6),
     4 TR(12,12),U(24),POSTD(20),CON,A1(6),BI(2,9),DMATI(2,2),
     5 SMI(3,9),DOFI(9),VALI(11),EPI(2),SIGI(2),ULOC(12),
     6 DOPO(9),SMO(3,9),DMATO(3,3),A2(3,6),BO(3,9),BOB(3,6),
     7 VALO(11),EPO(3),SIGO(3),SIG(3),SFACT,THDIS(12)
C
      EQUIVALENCE (RVR(1),BRAD), (RVR(2),ALPHA), (RVR(3),PRAD),
     1 (RVR(4),THICK), (RVR(5),PHI)
C
      EQUIVALENCE (SVR(1),B1(1,1)),(SVR(37),B2(1,1)),
     1 (SVR(73),VALI(1)), (SVR(84),PROP(1)), (SVR(88),TR(1,1)),
     2 (SVR(232),VALO(1)), (SVR(243),THDIS(1))
C
C  EQUIVALENCE MATERIAL PROPERTIES PROP():
      EQUIVALENCE  (PROP(1),EX), (PROP(2),ALPX), (PROP(3),NUXY),
     1 (PROP(4),DENS)
C
      DATA DPPI / 3.14159265358979 3D0 /
C
      CALL TRACK (5,'SR100 ')
      CALL GETELD (IELNUM,ITYP,EPAR(1),ERPAR(1),CON,CON,
     1 PRESS(1),CON,RVR(1),SVR(1),XYZEQ(1,1),U(1))
C
C
C ~~~~~~~~~~~~~~~~ STRESS PASS ~~~~~~~~~~~~~~~~~~
C STRESSES ARE EVALUATED AROUND CIRC AT BOTH NODES
C
C   EVALUATE I/P CONSTITUTIVE MATRIX [DMATI]
      CALL VZERO (DMATI(1,1),4)
      DCON = EX/(1.0-NUXY**2)
      DMATI(1,1) = DCON
      DMATI(1,2) = DCON*NUXY
      DMATI(2,1) = DMATI(1,2)
```

```
      DMAT1(2,2) = DCON
C
      CALL VZERO (DMATO(1,1),9)
      DMATO(1,1) = DCON
      DMATO(1,2) = DCON*NUXY
      DMATO(2,1) = DMATO(1,2)
      DMATO(2,2) = DCON
      DMATO(3,3) = DCON*(1-NUXY)/2
C
C     EVALUATE LOCAL  DISPLACEMENT VECTOR ULOC (IN LOCAL CSYS)
C
C     SUBTRACT THERMAL DISPLACEMENTS AS OBTAINED FROM DIRECT
C     STIFFNESS PROCEDURE IN ST100
C
      DO 10 I = 1,12
      U(I) = U(I)+THDIS(I)
   10 CONTINUE
C
      CALL MAXV (TR(1,1),U(1),ULOC,12,12)
C
C     ******* RECOVER NODELESS DOF. ********
C     IN-PLANE DOF
C     SMI CONTAINS LAST 3 ROWS OF I/P ZSFULL AFTER CONDENSATION
      CALL VZERO (SMI(1,1),27)
      SMI(1,1) = VALI(1)
      SMI(1,2) = VALI(2)
      SMI(1,3) = VALI(3)
      SMI(1,4) = VALI(4)
      SMI(1,5) = VALI(5)
      SMI(1,6) = VALI(6)
      SMI(1,7) = VALI(7)
      SMI(1,8) = VALI(8)
      SMI(2,8) = VALI(9)
      SMI(2,9) = VALI(10)
      SMI(3,9) = VALI(11)
      SMI(2,7) = SMI(1,8)
      SMI(3,8) = SMI(2,9)
C
C     DOFI = I/P DOF IN LOCAL CSYS.
C     THE KNOWN (BEAM) DOF STORED IN FIRST 6 ROWS.
C     NODELESS DOF RECOVERED AND STORED IN LAST 3.
      CALL VZERO (DOFI(1),9)
      DOFI(1) = ULOC(1)
      DOFI(2) = ULOC(3)
      DOFI(3) = ULOC(5)
      DOFI(4) = ULOC(7)
      DOFI(5) = ULOC(9)
      DOFI(6) = ULOC(11)
C
      DO 30 J = 1,3
      JJ = 6+J
      DUM = 0.0
      K = JJ - 1
      DO 20 L = 1,K
   20 DUM = DUM + SMI(J,L)*DOFI(L)
   30 DOFI(JJ) = (0.0 - DUM)/SMI(J,JJ)
C
C     OUT OF PLANE DEGREES OF FREEDOM
C
      CALL VZERO (SMO(1,1),27)
C     SMO CONTAINS LAST 3 ROWS OF O/P ZSFULL AFTER CONDENSATION
      SMO(1,1) = VALO(1)
      SMO(1,2) = VALO(2)
      SMO(1,3) = VALO(3)
      SMO(1,4) = VALO(4)
      SMO(1,5) = VALO(5)
      SMO(1,6) = VALO(6)
      SMO(1,7) = VALO(7)
      SMO(1,8) = VALO(8)
      SMO(2,8) = VALO(9)
      SMO(2,9) = VALO(10)
      SMO(3,9) = VALO(11)
      SMO(2,7) = SMO(1,8)
      SMO(3,8) = SMO(2,9)
C
C     DOFO = O/P DOF IN LOCAL CSYS.
C     THE KNOWN (BEAM) DOF STORED IN FIRST 6 ROWS.
C     NODELESS DOF RECOVERED AND STORED IN LAST 3.
      DOFO(1) = ULOC(2)
      DOFO(2) = ULOC(4)
      DOFO(3) = ULOC(6)
      DOFO(4) = ULOC(8)
      DOFO(5) = ULOC(10)
      DOFO(6) = ULOC(12)
C
      DO 50 J = 1,3
      JJ = 6+J
      DUM = 0.0
      K = JJ - 1
      DO 40 L = 1,K
   40 DUM = DUM + SMO(J,L)*DOFO(L)
   50 DOFO(JJ) = (0.0 - DUM)/SMO(J,JJ)
C
C     <<<<<<< SET UP LOOP FOR STRESS AND STRAIN EVALUATION >>>>>>>
C
      RATIO = (PRAD/BRAD)**2
      D = RATIO/2.0
      B = (2*(1+NUXY))/(2+NUXY)
      OPCON = PRAD/(BRAD**2)
C
C     PHIB IS THE MERIDIONAL ANGLE FOR STRESS EVALUATION
      PHIB = -ALPHA
      DO 2000 JJNODE = 1,2
      PHIB = PHIB+ALPHA
      WRITE (40,*) 'ELEMENT'
      WRITE (40,*) IELNUM
      WRITE (40,*) 'NODE'
      WRITE (40,*) JJNODE
C
C     SURF = -1 INNER,  0 MIDDLE,   +1 OUTER
      SURF = 1.0
      SFACT = 1.0
C
      WRITE (40,*) ' THETA    SIG(PHI)     SIG(THETA) TAU   SIGE'
C     LOOP IEP EVALUATES STRAIN/STRESS AT LOCATIONS AROUND CIRC
C     AT CURRENT NODAL LOCATION.
C
      THETA = -10.0
C
      DO 100 IEP = 1,36
C
      THETA = THETA + 10.0
      THETA = THETA * DPPI/180.0
      PHIB = PHIB * DPPI/180.0
C
C     ********* IN PLANE STRESS AND STRAIN ***********
C
C     EVALUATE IN PLANE B MATRIX
C     I/P A MATRIX, A1. (MULT BY THE 1/R USUALLY OUTSIDE THE BRACKETS)
      CALL VZERO (A1(1),6)
      CT = COS(THETA)
      ST = SIN(THETA)
      CP = COS(PHIB)
      SP = SIN(PHIB)
      CONST = PRAD*CT/BRAD
      A1(1) = -CONST/BRAD
      A1(3) = (D+CONST)*2.0*SP/BRAD
      A1(5) = (D+CONST)*-2.0*CP/BRAD
      CALL VZERO (BI(1,1),18)
C
C     EVALUATE I/P [BI] MATRIX
C     BENDING CONTRIBUTIONS-FIRST 6 COLUMNS. : [BI] = 1/R [A1] [B1]
C     ONLY ROW 1 IS POPULATED:
      DO 70 I = 1,6
```

```
      BI(1,J) = A1(1)*B1(1,J)+A1(3)*B1(3,J)+A1(5)*B1(5,J)
   70 CONTINUE
C
C     PUT I/P OV. MATRIX IN LAST THREE COLUMNS OF BI
      BI(1,7) = (CT*COS(2.0*THETA) + 0.5*ST*SIN(2.0*THETA))/BRAD
      BI(1,8) = (CT*COS(4.0*THETA) + 0.25*ST*SIN(4.0*THETA))/BRAD
      BI(1,9) = (CT*COS(6.0*THETA)
     1 + (1.0/6.0)*ST*SIN(6.0*THETA))/BRAD
C     CHOOSE INNER, MIDDLE OR OUTER SURFACE FOR STRESS EVALUATION
      BI(2,7) = SURF*-3.0*THICK/2.0*COS(2.0*THETA)/PRAD**2
      BI(2,8) = SURF*-15.0*THICK/2.0*COS(4.0*THETA)/PRAD**2
      BI(2,9) = SURF*-35.0*THICK/2.0*COS(6.0*THETA)/PRAD**2
C
C     EVALUATE STRAINS {EPI} = [BI] {DOFI} = {EPI(PHI) EPI(THETA)}
      CALL MAXV (BI(1,1),DOFI(1),EPI,2,9)
C
C     EVALUATE STRESSES {SIGI} = [D] {EPI} = {SIGI(PHI) SIGI(THETA)}
      CALL MAXV (DMATI(1,1),EPI(1),SIGI,2,2)
C       EVALUATE STRESS FACTOR = SIGI/(MR/I)
      SIGI(1) = SIGI(1)*SFACT
      SIGI(2) = SIGI(2)*SFACT
C
C     ******* OUT OF PLANE STRESS AND STRAIN ********
C
C     EVALUATE O/P B MATRIX BO
C     O/P A MATRIX A2. MULT BY PRAD/(BRAD**2) USUALLY OUTSIDE BRACKETS
      CALL VZERO (A2(1,1),18)
      A2(1,2) = -(2.0-B)*SP*ST*OPCON
      A2(1,4) = (2.0-B)*CP*ST*OPCON
      A2(3,2) = B*CP*OPCON
      A2(3,4) = B*SP*OPCON
      A2(3,6) = OPCON
C
      CALL VZERO (BOB(1,1),18)
      CALL MAXB (A2(1,1),B2(1,1),BOB,3,6,3,3,6,6)
C     PUT BOB IN FIRST 6 COLUMNS OF BO
      CALL VZERO (BO(1,1),27)
      DO 80 I=1,3
      DO 80 J=1,6
      BO(I,J) = BOB(I,J)
   80 CONTINUE
C     EVALUATE OVALISATION B MATRIX IN LAST 3 COLS. OF BO
      BO(1,7) = (CT*SIN(2.0*THETA) - 0.5*ST*COS(2.0*THETA))/BRAD
      BO(1,8) = (CT*SIN(4.0*THETA) - 0.25*ST*COS(4.0*THETA))/BRAD
      BO(1,9) = (CT*SIN(6.0*THETA)
     1 - (1.0/6.0)*ST*COS(6.0*THETA))/BRAD
C
      BO(2,7) = SURF*-3.0*THICK/2*SIN(2.0*THETA)/PRAD**2
      BO(2,8) = SURF*-15.0*THICK/2*SIN(4.0*THETA)/PRAD**2
      BO(2,9) = SURF*-35.0*THICK/2*SIN(6.0*THETA)/PRAD**2
C     EVALUATE STRAINS {EPO} = [BO] {DOFO} = {EPO(PHI) EPO(THETA) GAMMA}
      CALL MAXV (BO(1,1),DOFO(1),EPO,3,9)
C
C     EVALUATE STRESSES {SIGO} = [D] {EPO} = {SIGI(PHI) SIGI(THETA) TAU}
      CALL MAXV (DMATO(1,1),EPO(1),SIGO,3,3)
C       EVALUATE STRESS FACTOR = SIGO/(MR/I)
      SIGO(1) = SIGO(1)*SFACT
      SIGO(2) = SIGO(2)*SFACT
      SIGO(3) = SIGO(3)*SFACT
C
      THETA = THETA*180.0/DPPI
C       WRITE (40,'(6F10.2)') THETA,SIGO(1),SIGO(2),SIGO(3)
C
C     EVALUATE TOTAL (SUMMED) STRESSES
      SIG(1) = SIGI(1) + SIGO(1)
      SIG(2) = SIGI(2) + SIGO(2)
      SIG(3) = SIGO(3)
C
C     VON MISE STRESS
      SIGE = ((SIG(1)-SIG(2))**2 + SIG(1)**2 +SIG(2)**2 ) / 2.0
      SIGE = DSQRT(SIGE + 3*SIG(3)**2)
      WRITE (40,'(5F16.3)') THETA,SIG(1),SIG(2),SIG(3),SIGE

  100 CONTINUE
C
      PHIB = PHIB*180.0/DPPI
 2000 CONTINUE
C
      IF (IPLOT .NE. 1) GO TO 900
C           ***** NUMBER OF FORCES (LEVEL 1) *****
      IPLTAY(2) = 12
C           ***** NUMBER OF STRESSES (LEVEL 2) *****
      IPLTAY(3) = 4
C           ***** NUMBER OF TOTAL SAVED (LEVELS 1, 2, AND 3)
      IPLTAY(4) = 20
C           ***** SAVE GEOMETRY FOR CONTOURS (0,NO  1,YES)
      IPLTAY(6) = 0
C
C     ****** PUT POSTDATA INFORMATION INTO POSTD ******
C
C     ***** PUT PLTARY INFORMATION ONTO FILE 12 *******
      CALL SRPLT (IELEM,ITYP,NROW,MAT,100,2,U(1),NODES(1),XYZEQ(1,1),
     1 IPLTAY(1),POSTD(1))
  900 CALL PUTELD (IELNUM,EPAR(1),ERPAR(1),CON,SVR(1))
C     PUTELD RESTORES DATA BACK TO FILE3
      CALL TRACK( 15,'SR100 ')
      RETURN
      END
```

A4.3 ELEMENT PB2 ANSYS USER ELEMENT SOURCE CODE

```
C            ELEMENT PB2
C            ***********
C  ANSYS USER ELEMENT CODE FOR PIPE BEND ELEMENT ELEMENT
C
C  BASED ON VLASOV THIN WALL CIRC CURVED BEAM SOLUTION
C  WITH SUPERPOSED LINEAR OVALISATION: 3 I/P AND 3 O/P MODES.
C
C  CLOSED FORM INTEGRATION FOR BEAM STIFFNESS MATRIX.
C  NUMERICAL INTEGRATION OF OVALISATION AND COUPLING MATRICES.
C  MATRIX STATICALLY CONDENSED TO GIVE FINAL 12x12 STIFFNESS MATRIX.
C
C              DONALD MACKENZIE  OCT 1989
C
C  ANSYS VERSION 4.3A
   IMPLICIT DOUBLE PRECISION (A-H,O-Z)
   EXTERNAL MAIN,STOPER
   EXTERNAL ELSHFN
   CALL NNDIM
   CALL MAIN
   CALL STOPER
   END
```

A4.3.1 SUBROUTINE USEREL

```
   SUBROUTINE USEREL (ITYP,IPARM,KYSUB,KEY3D,KDOF,KUNSYM,KTRANS)
C
   INTEGER IPARM(20,12),KYSUB(9),ITYP,JTYPE,KEY3D,KDOF,KUNSYM,KTRANS
C  **** DETERMINE TYPE OF ELEMENT AND THEN BYPASS IF NOT USER ELEMENT
   JTYPE = IPARM(ITYP,3)
   IF (JTYPE .NE. 100) GO TO 100
C
C     ********** SET 3-D KEY **********
   KEY3D = 1
C     ********** DEFINE DOF SET AT EACH NODE *********
   KDOF = 0
C     ********** SET UNSYMMETRIC MATRIX KEY **********
   KUNSYM = 0
C
C    ***** DEFINE PATTERN FOR ELEMENT TO GLOBAL TRANSFORMATION ***
   KTRANS = 2
C     ********** DEFINE NUMBER OF NODES **********
   IPARM(ITYP,8) = 2
C     ***** DEFINE NUMBER OF TEMPERATURES (DELTEM,TEMPER) ******
   IPARM(ITYP,11) = 2
C
C     ***** DEFINE NUMBER OF PRESSURES (PRESS) *******
C    IF THERMAL ANALYSIS, TWO TIMES NUMBER OF CONVECTION SURFACES
   IPARM(ITYP,6) = 1
C     ***** SET ZEROED VARIABLES (NOITUEP)
   IPARM(ITYP,12) = 0
C     ***** DEFINE NUMBER OF REAL CONSTANTS FOR ELEMENT (RVR) *
   IPARM(ITYP,10) = 5
C
C     ***** DEFINE NUMBER OF VARIABLES TO BE SAVED (SVR) *****
   IPARM(ITYP,7) = 364
C     ***** DEFINE NUMBER OF ROWS IN ELEMENT MATRICES (KTIK) **
   IPARM(ITYP,9) = 12
C     ********** SET KEY TO IDENTIFY NON-LINEAR ELEMENT *****
   IPARM(ITYP,4) = 0
C     ********** SET KEY FOR THERMAL ELEMENT (KAN,-1) *******
   IPARM(ITYP,1) = 0
 100 RETURN
   END
```

A4.3.2 SUBROUTINE USERPT

```
   SUBROUTINE USERPT (INODE,JTYPE,KSHAPE,NNODE)
C ********** USER SUBROUTINE FOR ANSYS PLOT SHAPE **********
C  DEFINE ELEMENT SHAPE AND NUMBER OF NODES, FOR PLOTTING
   INTEGER INODE(20),JTYPE,KSHAPE,NNODE
C  *********BYPASS IF NOT USER ELEMENT (JTYPE = 100) *********
   IF (JTYPE .NE. 100) GO TO 100
C  ****** SELECT SHAPE TO BE PLOTTED BY SETTING KSHAPE *******
   KSHAPE = 2
C     ********** SET NUMBER OF ACTUAL NODES **********
   NNODE = 2
 100 RETURN
   END
```

A4.3.3 SUBROUTINE ST100

```
   SUBROUTINE ST100 (IELNUM,ITYP,KELIN,KELOUT,NR,KTIK,ZS,
   1 ZASS,DAMP,GSTIF,ZSC)
C
C ******* STIFFNESS PASS FOR PIPE BEND ELEMENT ELEMENT *********
C          CLOSED FORM BEAM STIFFNESS MATRIX
C  NUMERICAL INTEGRATION OF OVALISATION AND COUPLING MATRICES.
C
   IMPLICIT DOUBLE PRECISION (A-H,O-Z)
C
C  -------- DECLARE EXTERNAL SUBROUTINES AND FUNCTIONS ---------
   EXTERNAL TRACK,GETELD,PUTELD,PROPEV,NONTBL,VZERO,MHTCH,USEERR,
   1 MAXV,TRSUB,MAXB,MATXB,INVM,REDUCE,INTEG
C
C  ---------- COMMON BLOCK DECLARATIONS ----------
C  STANDARD ST100 INTEGER VARIABLES ASSOCIATED WITH COMMON:
   INTEGER  IELNUM,ITYP,KELIN(6),KELOUT(6),NR,KTIK,
   1 KEYERR,IOUT,NSTEPS,KFSTLD,ITTER,ITIME,NCUMIT,KRSTRT,ISPARE,
   2 K13,NPRPVL,MATST,KS,K16,IPROP,KCPDS,
   3 K20,KAY,MODE,ISYM,KAHD,IDEBUG,IXXX,
   4 ITYPE,MAT,IELEM,NROW,JTYPE,IPLOT,IPRINT,KTEMTP,KCONCV,KBICNV,
   5 KEYPLS,KEYCRP,KEYSWL,KYSUB(9),K21,NODES(20),  EPAR(50)
   REAL  ERRVAR(5)
C
C  STANDARD ST100 DOUBLE PRECISION VARIABLES ASSOCIATED WITH COMMON:
   DOUBLE PRECISION
   1 DPZERO,DPHALF,DPONE,DPTWO,DPTEN,DTORAD,RADTOD,
   2 TREF,TUNIF,TOFSET, DELTIM,TIME,TIMOLD,TIME2,TIME3,DELT2,
   3 ACEL,OMEGA,CGOMEG,CGLOC,DXXX,
   4 ELMASS,XCENTR,YCENTR,ZCENTR,TFCP,SUBEX,  ERPAR(20),
   5 XYZEQ(20,3),X(20),Y(20),Z(20),  ELVOL
C
C  ST100 COMMON BLOCK: STCOM
   COMMON /STCOM/ DPZERO,DPHALF,DPONE,DPTWO,DPTEN,DTORAD,RADTOD,
   1 TREF,TUNIF,TOFSET, DELTIM,TIME,TIMOLD,TIME2,TIME3,DELT2,
   2 ACEL(3),OMEGA(6),CGOMEG(6),CGLOC(3),      DXXX(16),
   3 KEYERR,IOUT,NSTEPS,KFSTLD,ITTER,ITIME,NCUMIT,KRSTRT,ISPARE,
   4 K13,NPRPVL,MATST,KS,K16,IPROP(20),KCPDS,
   5 K20,KAY(10),MODE,ISYM,KAHD,IDEBUG(10),     IXXX(41)
C
C  EQUIVALENCING OF STCOM VARIABLES
   EQUIVALENCE  (ITYPE,EPAR(1)), (MAT,EPAR(2)), (IELEM,EPAR(5)),
   1 (NROW,EPAR(7)), (JTYPE,EPAR(11)), (IPLOT,EPAR(12)),
   2 (IPRINT,EPAR(13)), (KTEMTP,EPAR(14)), (KCONCV,EPAR(16)),
   4 (KBICNV,EPAR(17)), (KEYPLS,EPAR(18)), (KEYCRP,EPAR(19)),
   5 (KEYSWL,EPAR(20)), (KYSUB(1),EPAR(21)), (K21,EPAR(30)),
   6 (NODES(1),EPAR(31))
   EQUIVALENCE  (ELMASS,ERPAR(1)), (XCENTR,ERPAR(2)),
   1 (YCENTR,ERPAR(3)), (ZCENTR,ERPAR(4)), (TFCP,ERPAR(5)),
   2 (SUBEX,ERPAR(6))
   EQUIVALENCE (X(1),XYZEQ(1,1)),(Y(1),XYZEQ(1,2)),(Z(1),XYZEQ(1,3))
C
C  ---------- USER DEFINED VARIABLES ----------
C  INTEGER, REAL AND DOUBLE PRECISION VARIABLES DEFINED HERE BY USER.
```

```
C
C  DECLARE INTEGER VARIABLES:
      INTEGER I,J,K,I3,J3,I6,J6,LP(4),NSTR,NUM,KDEMO,NFKEY,
     1 KI,KO,LI,LO
C
C  DECLARE DOUBLE PRECISION VARIABLES:
      DOUBLE PRECISION ZS(KTIK,KTIK),ZASS(KTIK,KTIK),
     & ZSC(KTIK),TR(12,12),
     1 U(240),PROP(4),EX,ALPX,NUXY,DENS,ALEN2,ALENG,DX,DY,DZ,
     2 AFLU,TEMPER(2),CON,AVETEM,TRPROP(7),MPROP(1),LDPROP(1),DPPI,
     3 BRAD,ALPHA,PRAD,THICK,PHI,RVR(5),SVR(364)
      DOUBLE PRECISION ZSI(6,6),ZSO(6,6),BI(6,12),B1(6,6),B2(6,6),
     2 CA,SA,CM1,C2A,S2A,C2M1,TRIG1,TRIG2,
     3 AREA,POLMOM,SECMOM,EI,GJ,RATIO,D,B,DM1,DM2,
     4 DTERM1,DTERM2,DTERM3,NRINV,V,BTERM1,BTERM2,BB,BM1,BTERM3,
     5 BTERM4,H,CONST
      DOUBLE PRECISION CONOV,PR4,BRT2,ZSOVI(6,6),CPINTI(6,6),
     1 ZSCUP(6,6),ZSFULL(12,12),ZSOVO(6,6),
     2 CPINTO(6,6),SUBDATA(6),SMI(6,12),SMO(6,12)
C
C --- USER EQUIVALENCING OF REAL AND SAVED VARIABLES (RVR, SVR) ---
C
C  EQUIVALENCE REAL VARIABLES RVR():
C  BRAD = BEND RAD,  ALPHA = BEND ANGLE,  PRAD = PIPE RAD,  THICK = WALL THK.
C  PHI = ORIENTATION ANGLE
      EQUIVALENCE (RVR(1),BRAD), (RVR(2),ALPHA), (RVR(3),PRAD),
     1 (RVR(4),THICK), (RVR(5),PHI)
C
C  EQUIVALENCE SAVED VARIABLES SVR():
      EQUIVALENCE (SVR(1),B1(1,1)),(SVR(37),B2(1,1)),
     1 (SVR(73),SMI(1,1)), (SVR(145),PROP(1)), (SVR(149),TR(1,1)),
     2 (SVR(293),SMO(1))
C
C  ---------- EQUIVALENCING OF MATERIAL PROPERTIES ---------
      EQUIVALENCE  (PROP(1),EX), (PROP(2),ALPX), (PROP(3),NUXY),
     1 (PROP(4),DENS)
C
      CALL TRACK(5,'ST100')
C
C  ---- READING IN ELEMENT INFORMATION: SUBROUTINE GETELD ----
C
      CALL GETELD (IELNUM,ITYP,EPAR(1),ERPAR(1),CON,TEMPER(1),
     1 CON,CON,RVR(1),SVR(1),XYZBQ(1,1),U(1))
C
C  CONVERT BEND AND ORIENTATION ANGLES TO RADIANS.
      DATA DPPI / 3.14159265358979D0 /
      PHI = PHI*DPPI/180.0D0
      ALPHA = ALPHA*DPPI/180.0D0
C
C -- READING IN ELEMENT MATERIAL PROPERTIES: SUBROUTINE PROPEV --
C
C  SET UP INTEGER ARRAY FOR ACCESSING MATERIAL PROPERTIES
      DATA LP / 1, 2, 3,10/
C
      AVETEM = DPHALF*(TEMPER(1) + TEMPER(2))
C
      CALL PROPEV (IELEM,MAT,ITYPE,LP(1),AVETEM,PROP(1),4)
C
C  ---------- GEOMETRY VALIDITY CHECK ----------
C
      DX = X(2) - X(1)
      DY = Y(2) - Y(1)
      DZ = Z(2) - Z(1)
      CON = DX**2 + DY**2
      ALEN2 = CON + DZ**2
      IF (ALEN2 .GT. 0.0) GO TO 150
      WRITE (IOUT,2000) IELEM
 2000 FORMAT ('ZERO LENGTH ELEMENT' ,I5)
      KEYERR = 1
      NFKEY = 1
      CALL USEERR (NFKEY)
      GO TO 990
C
 150  CONTINUE
C
C  ---------- CALCULATE MASS AND CENTROID ----------
C
C   STILL TO BE DONE: EXISTING IS FOR STRAIGHT BEAM
      XCENTR = (X(1) + X(2))*DPHALF
      YCENTR = (Y(1) + Y(2))*DPHALF
      ZCENTR = (Z(1) + Z(2))*DPHALF
      AREA = 2*DPPI*PRAD*THICK
      AFLU = DPPI*(PRAD-(THICK/2))**2
      ELMASS =  (DENS*AREA + DENSFL*AFLU)*BRAD*ALPHA
C
C  ---------- END OF CHECK RUN OR ERROR DETECTED ----------
C
      IF ((NSTEPS .EQ. 0) .OR. (KEYERR.EQ.1)) GO TO 990
C
C  ---------- EVALUATE THE ELEMENT TRANSFORMATION MATRIX ----------
C TRANSFORMATION MATRIX TR IS EVALUATED IN THE USER DEFINED
C SUBROUTINE TRSUB.  INFORMATION REQUIRED TO CALCULATE MATRIX VALUES
C IS PASSED IN BY ARRAY TRPROP.
C
      TRPROP(1) = DX
      TRPROP(2) = DY
      TRPROP(3) = DZ
      TRPROP(4) = PHI
      TRPROP(5) = ALPHA
      TRPROP(6) = CON
      TRPROP(7) = ALEN2
C
C ZERO THE TR MATRIX.
      CALL VZERO (TR(1,1),144)
C
      CALL TRSUB (TR,TRPROP)
C
C  ---------- EVALUATE THE ELEMENT STIFFNESS MATRIX ----------
C   ELEMENT CB2.
C   CURVED CYLINDRICAL THIN WALLED BEAM ELEMENT STIFFNESS MATRIX
C   CLOSED FORM.
C   OVALISATION AND COUPLING MATRICES NUMERICALLY INTEGRATED
C   IN SUBROUTINE INTEGRATE.
C
C   OPEN FILE MAT.DAT FOR MATRIX PRINTOUT
      OPEN (UNIT=40,FILE='PB4RES',STATUS='NEW')
C CHECK MATRIX IS REQUIRED.
      IF (KELIN(1) .NE. 1) GO TO 400
      CALL VZERO (ZS(1,1),144)
C
C   SET UP ARRAY SUBDATA WITH REQD. SUBROUTINE INFO
      SUBDATA(1) = BRAD
      SUBDATA(2) = ALPHA
      SUBDATA(3) = PRAD
      SUBDATA(4) = THICK
      SUBDATA(5) = EX
      SUBDATA(6) = NUXY
C
C   'INTEG' EVALUATES I/P AND OUT O/P OVALISATION STIFFNESS
C   MATRICES, AND I/P AND O/P COUPLING INTEGRAL MATRICES.
      CALL VZERO(ZSOVI(1,1),36)
      CALL VZERO(ZSOVO(1,1),36)
      CALL VZERO(CPINTI(1,1),36)
      CALL VZERO(CPINTO(1,1),36)
C
      CALL INTEG (ZSOVI,ZSOVO,CPINTI,CPINTO,SUBDATA)
C
C  EVALUATE CONSTANTS AND TRIG FUNCTIONS.
      CA = COS(ALPHA)
      SA = SIN(ALPHA)
      CM1 = CA-1.0
      C2A = COS(2.0*ALPHA)
      S2A = SIN(2.0*ALPHA)
      C2M1 = C2A-1.0
```

```
      TRIG1 = 2.0*ALPHA-S2A
      TRIG2 = 2.0*ALPHA+S2A
C
      AREA = 2.0*DPPI*PRAD*THICK
      POLMOM = AREA*PRAD**2
      SECMOM = POLMOM/2.0
      EI = EX*SECMOM
      GJ = EX*POLMOM/(2.0+2.0*NUXY)
C
      RATIO = (PRAD/BRAD)**2
      D = RATIO/2.0
      B = 2.0*EI/(EI+GJ)
C
      DM1 = 1.0-D
      DM2 = 1.0-2.0*D
      DTERM1 = 2.0*DM1/BRAD
      DTERM2 = ALPHA*SA+DM2*CA
      DTERM3 = -ALPHA*CA+DM2*SA
C
      NRINV = -1.0/BRAD
      V = 1/(2.0*(1.0+NUXY))
C
      BB = B**2
      BM1 = 1.0-B
      BTERM1 = (2.0-B)**2
      BTERM3 = (-ALPHA*SA+CA*BM1)/BRAD
      BTERM4 = (ALPHA*CA+SA*BM1)/BRAD
C
      H = 1.0+RATIO/2.0
      CONST = EI/BRAD**3
      CONST = CONST/(1.0-NUXY**2)
      CONOV = CONST*ALPHA/RATIO
C
C *** IN PLANE STIFFNESS MATRIX ***
C
C     EVALUATE [BI] = [C1]
      CALL VZERO (BI(1,1),72)
C
      BI(1,3) = DM2
      BI(1,4) = -1.0
      BI(1,6) = 1.0
      BI(2,1) = 1.0
      BI(2,2) = 1.0
      BI(3,3) = -DTERM1
      BI(3,6) = NRINV
      BI(4,1) = ALPHA
      BI(4,2) = SA
      BI(4,3) = DTERM2
      BI(4,4) = -CA
      BI(4,5) = DTERM3
      BI(4,6) = 1.0
      BI(5,1) = 1.0
      BI(5,2) = CA
      BI(5,3) = ALPHA*CA
      BI(5,4) = SA
      BI(5,5) = ALPHA*SA
      BI(6,1) = ALPHA*NRINV
      BI(6,3) = -DTERM1*CA
      BI(6,5) = -DTERM1*SA
      BI(6,6) = NRINV
C
C     INVERT BI
      CALL INVM(BI,DET1,6,6,12)
C
C     EQUATE FIRST 6X6 OF BI WITH B1
      DO 322 I=1,6
      DO 322 J=1,6
  322 B1(I,J) = BI(I,J)
C
C     EVALUATE ZSI = IN-PLANE INTEGRAL MATRIX
      CALL VZERO(ZSI(1,1),36)
C
      ZSI(1,1) = CONST*ALPHA
      ZSI(1,3) = CONST*2.0*CM1
      ZSI(1,5) = CONST*2.0*SA
      ZSI(3,1) = ZSI(1,3)
      ZSI(3,3) = CONST*H*TRIG1
      ZSI(3,5) = CONST*H*C2M1
      ZSI(5,1) = ZSI(1,5)
      ZSI(5,3) = ZSI(3,5)
      ZSI(5,5) = CONST*H*TRIG2
C
C     EVALUATE I/P STIFFNESS [ZSI]=[B1]'[ZSI][B1]
      CALL MHTCH(B1(1,1),ZSI(1,1),6,6,6)
C
C     EVALUATE I/P COUPLING STIFFNESS
      CALL VZERO(ZSCUP(1,1),36)
      CALL MATXB (B1(1,1),CPINT1(1,1),ZSCUP,6,6,6,6,6,6)
C
C     FILL IN 12X12 I/P MATRIX ZSFULL
      CALL VZERO(ZSFULL(1,1),144)
      DO 325 I=1,6
      DO 325 J=1,6
  325 ZSFULL(I,J) = ZSI(I,J)
C
      DO 326 I=1,6
      DO 326 J=1,6
      JP6 = J+6
  326 ZSFULL(I,JP6) = ZSCUP(I,J)
C
      DO 327 I=1,6
      IP6 = I+6
      DO 327 J=1,6
  327 ZSFULL(IP6,J) = ZSCUP(J,I)
C
      DO 328 I=1,6
      IP6 = I+6
      DO 328 J=1,6
      JP6 = J+6
  328 ZSFULL(IP6,JP6) = ZSOVI(I,J)
C
C
      CALL VZERO(ZSI(1,1),36)
      CALL REDUCE (ZSFULL,ZSI)
C
C     EQUATE LAST 6 ROWS OF ZSFULL AFTER
C     REDUCTION TO THE ARRAY SMI
      CALL VZERO (SMI(1,1),72)
      DO 330 I=1,6
        IP6 = I+6
      DO 330 J=1,12
  330 SMI(I,J) = ZSFULL(IP6,J)
C
C *** OUT OF PLANE STIFFNESS ***
C
C     EVALUATE BI = C2
C
      CALL VZERO(BI(1,1),72)
C
      BI(1,1) = 1.0
      BI(1,4) = B
      BI(1,5) = -1.0
      BI(2,1) = 1/BRAD
      BI(3,2) = BM1/BRAD
      BI(3,3) = 1/BRAD
      BI(3,6) = NRINV
      BI(4,1) = CA
      BI(4,2) = ALPHA*CA-SA*B
      BI(4,3) = SA
      BI(4,4) = ALPHA*SA+CA*B
      BI(4,5) = -1.0
      BI(4,6) = -ALPHA
      BI(5,1) = CA/BRAD
      BI(5,2) = ALPHA*CA/BRAD
```

```
      BI(5,3) = SA/BRAD
      BI(5,4) = ALPHA*SA/BRAD
      BI(6,1) = SA*NRINV
      BI(6,2) = BTERM3
      BI(6,3) = CA/BRAD
      BI(6,4) = BTERM4
      BI(6,6) = NRINV
C
C     INVERT BI
C
      CALL INVM(BI,DET1,6,6,12)
C     EQUATE FIRST 6X6 OF BI TO B2
      DO 332 I=1,6
      DO 332 J=1,6
  332 B2(I,J) = BI(I,J)
C
C     EVALUATE ZSO = O/P INTEGRAL MATRIX
      CALL VZERO(ZSO(1,1),36)
C
      ZSO(2,2) = CONST*(BTERM1*0.25*TRIG1+BB*V*0.5*TRIG2)
      ZSO(2,4) = CONST*(BB*V*0.5-BTERM1*0.25)*(1-C2A)
      ZSO(2,6) = CONST*2.0*B*V*SA
      ZSO(4,2) = ZSO(2,4)
      ZSO(4,4) = CONST*(BTERM1*0.25*TRIG2+BB*V*0.5*TRIG1)
      ZSO(4,6) = CONST*2*B*V*(1-CA)
      ZSO(6,2) = ZSO(2,6)
      ZSO(6,4) = ZSO(4,6)
      ZSO(6,6) = CONST*2.0*ALPHA*V
C
C     EVALUATE O/P STIFFNESS [ZSO] = [BI]'[ZSO] [BI]
      CALL MHTCH(B2(1,1),ZSO(1,1),6,6,6)
C
C     EVALUATE THE OUT OF PLANE COUPLING MATRIX ZSCUP
      CALL VZERO(ZSCUP(1,1),36)
      CALL MATXB (B2(1,1),CPINTO(1,1),ZSCUP,6,6,6,6,6,6)
C
C     FILL IN 9X9 I/P MATRIX ZSFULL
      CALL VZERO(ZSFULL(1,1),144)
      DO 335 I=1,6
      DO 335 J=1,6
  335 ZSFULL(I,J) = ZSO(I,J)
C
      DO 336 I=1,6
      DO 336 J=1,6
      JP6 = J+6
  336 ZSFULL(I,JP6) = ZSCUP(I,J)
C
      DO 337 I=1,6
      IP6 = I+6
      DO 337 J=1,6
  337 ZSFULL(IP6,J) = ZSCUP(J,I)
C
      DO 338 I=1,6
      IP6 = I+6
      DO 338 J=1,6
      JP6 = J+6
  338 ZSFULL(IP6,JP6) = ZSOVO(I,J)
C
      CALL VZERO(ZSO(1,1),36)
      CALL REDUCE (ZSFULL,ZSO)
C
C     EQUATE LAST 6 ROWS OF ZSFULL
C     AFTER REDUCTION TO THE ARRAY SMO
C
      CALL VZERO (SMO(1,1),72)
      DO 350 I=1,6
        IP6 = I+6
      DO 350 J=1,12
  350 SMO(I,J) = ZSFULL(IP6,J)
C     ASSEMBLE FULL 12x12 REDUCED STIFFNESS MATRIX
C     I/P TERMS IN I-J ODD LOCATIONS, O/P TERMS IN EVEN
C
C     ZERO ZS
      CALL VZERO (ZS(1,1),144)
      DO 360 I=1,6
      KO = 2*I-1
      KE = 2*I
        DO 360 J=1,6
        LO = 2*J-1
        LE = 2*J
      ZS(KO,LO) = ZSI(I,J)
      ZS(KE,LE) = ZSO(I,J)
  360 CONTINUE
C
C ---------- STIFFNESS MATRIX TRANSFORMATION ----------
C ELEMENT MATRICES ARE TRANSFORMED TO THE GLOBAL CO-ORD.
C SYSTEM BY THE ANSYS SUBROUTINE MHTCH.
      CALL MHTCH (TR(1,1),ZS(1,1), KTIK,KTIK, KTIK)
C
C     SET KEY THAT MATRIX WAS COMPUTED.
      KELOUT(1) = 1
C
  400 CONTINUE
  800 CONTINUE
C
C ---------- OUTPUT ELEMENT DATA TO FILE 12 ----------
C ELEMENT DATA IS OUTPUT TO FILE 12 BY THE SUBROUTINE PUTELD.
C
  990 CALL PUTELD (IELNUM,EPAR(1),ERPAR(1),CON,SVR(1))
C
      CALL TRACK( 15,'ST100 ')
      RETURN
      END
```

SUBROUTINE INTEG

```
      SUBROUTINE INTEG (STOI,STOO,STCUPI,STCUPO,SUBDATA)
C
C     NUMERICAL INTEGRATION ROUTINE FOR IN PLANE AND
C     OUT OF PLANE OVALISATION AND COUPLING STIFFNESS MATRICES.
C
C     INTEGRATE THRO' THICKNESS: 3 POINT RULE
C     INTEGRATE AROUND CIRC.: 24 POINT RULE
C     INTEGRATE ALONG AXIS: 5 POINT
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      EXTERNAL MAXB,MATXB,VZERO
      DOUBLE PRECISION
     & AI(3),WI(3),AJ(24),WJ(24),BI(2,3),D(2,2),DBI(2,6),
     1 BTDBI(6,6),BNT(6,2),NUXY,STOI(6,6),
     2 BO(2,3),DBO(2,6),BTDBO(6,6),STOO(6,6),
     3 AK(5),WK(5),AMATI(6,2),
     4 ATDBI(6,6),STCUPI(6,6),BCUP(3,3),DO(3,3),AMATO(6,3),
     5 DBCUPN(3,6),ATDBO(6,6),STCUPO(6,6),SUBDATA(6),
     6 N(3,6),BN(2,6),N1,N2,BCUPN(3,6)
C
C     READ IN ELEMENT PROPERTIES
      BRAD = SUBDATA(1)
      ALPHA = SUBDATA(2)
      PRAD = SUBDATA(3)
      THICK = SUBDATA(4)
      EX = SUBDATA(5)
      NUXY = SUBDATA(6)
C     WEIGHTS AND ABSCISSAS
C
C     3 POINT RULE: INTEGRATION LIMITS +/- 1
      AI(1) = -0.774596669241483
      AI(2) = 0.0
      AI(3) = 0.774596669241483
C
      WI(1) = 0.555555555555553
      WI(2) = 0.888888888888889
      WI(3) = 0.555555555555553
```

```
C
C   24 POINT RULE: INTEGRATION LIMITS 0 TO 2 PI
C   ABSCISSA
    DATA (AJ(I),I=1,24)/
   1 0.0151197943026776 , 0.079392582916506 , 0.193916213993074,
   2 0.356836145971435 , 0.565480438598976 , 0.816423930689125,
   3 1.105546397970765 , 1.428100566150951 , 1.778790157089124,
   4 2.151856885770058 , 2.541175023904087 , 2.940351989944773,
   5 3.342833318055227 , 3.742010284095913 , 4.131328422229942,
   6 4.504395150910876 , 4.855084741849049 , 5.177638910029235,
   7 5.466761377310875 , 5.717704869401024 , 5.926349162028565,
   8 6.089269094006926 , 6.203792725083494 , 6.268065513697324/
C   WEIGHTS
    DATA (WJ(I),I=1,24)/
   1 0.038771116880962 , 0.089634000925077 , 0.139101676526740,
   2 0.186291998762931 , 0.230424767197769 , 0.270774378315854,
   3 0.306678040343665 , 0.337546129710709 , 0.362871758252947,
   4 0.382239063958693 , 0.395330028457241 , 0.401929694667374,
   5 0.401929694667374 , 0.395330028457241 , 0.382239063958693,
   6 0.362871758252947 , 0.337546129710709 , 0.306678040343665,
   7 0.270774378315854 , 0.230424767197769 , 0.186291998762931,
   8 0.139101676526740 , 0.089634000925077 , 0.038771116880962/
C
C   5 POINT RULE: INTEGRATION LIMITS + /- 1
C   ABSCISSA
    AK(1) = -0.906179845938664
    AK(2) = -0.538469310105683
    AK(3) = 0.000000000000000
    AK(4) = 0.538469310105683
    AK(5) = 0.906179845938664
C   WEIGHTS
    WK(1) = 0.236926885056181
    WK(2) = 0.478628670499366
    WK(3) = 0.568888888888889
    WK(4) = 0.478628670499366
    WK(5) = 0.236926885056181
C
C   CONVERT ALPHA TO RADIANS
C
C   CONVERT THRO' THICKNESS POINTS TO ACTUAL INTEGRATION LIMITS
    T = THICK/2.0
    DO 10 I=1,3
    AI(I) = AI(I)*T
 10 WI(I) = WI(I)*T
C
C   CONVERT AXIAL POINTS TO ACTUAL INTEGRATION LIMITS
    ALP = ALPHA/2.0
    DO 20 I=1,5
    AK(I) = ALP*(1+AK(I))
 20 WK(I) = ALP*WK(I)
C
    DCON = EX/(1.0-NUXY**2)
C   CONSTITUTIVE MATRIX - LEAVE OUT DCON
    D(1,1) = 1.0
    D(1,2) = NUXY
    D(2,1) = NUXY
    D(2,2) = 1.0
C   CONSTANTS FOR OV AND CUP STIFFNESS
    CONOV = DCON*BRAD*PRAD
    CCUPI = DCON*PRAD
    CCUPO = DCON*PRAD**2/BRAD
C   SET UP INTEGRATION DOUBLE LOOP THRO' THICK & ROUND CIRC
C
    CALL VZERO (STOI(1,1),36)
    CALL VZERO (STOO(1,1),36)
    CALL VZERO (STCUPI(1,1),36)
    CALL VZERO (STCUPO(1,1),36)
C
    DO 300 I=1,3
    DO 300 J=1,24
    DO 300 K=1,5
C
C   AI(I) = THRO' THICK LOCATION H
C   AJ(J) = ROUND CIRC POSN. THETA
C   AK(K) = AXIAL POSN PHIB
    H = AI(I)
    THETA = AJ(J)
    PHIB = AK(K)
C
C   LINEAR SHAPE FUNCTION MATRIX
    N1 = 1.0 - PHIB/ALPHA
    N2 = PHIB/ALPHA
    CALL VZERO (N(1,1),18)
    N(1,1) = N1
    N(1,4) = N2
    N(2,2) = N1
    N(2,5) = N2
    N(3,3) = N1
    N(3,6) = N2
C
    CT = COS(THETA)
    CT2 = COS(2.0*THETA)
    CT4 = COS(4.0*THETA)
    CT6 = COS(6.0*THETA)
    ST = SIN(THETA)
    ST2 = SIN(2.0*THETA)
    ST4 = SIN(4.0*THETA)
    ST6 = SIN(6.0*THETA)
C
C   **** IN PLANE OVALISATION ****
C   EVALUATE [N]T[B]T[D][B][N], MULT BY (WI(I)*WJ(J)) AND ADD TO LAST
C   VALUE:  [STIFF] = [STIFF] + [BTDB] WI WJ
C
    CALL VZERO (BI(1,1),6)
C   [BI]
    BI(1,1) = (CT*CT2+0.5*ST*ST2)/BRAD
    BI(1,2) = (CT*CT4+0.25*ST*ST4)/BRAD
    BI(1,3) = (CT*CT6+(1.0/6.0)*ST*ST6)/BRAD
C
    R2 = PRAD**2
    BI(2,1) = -3.0*H*CT2/R2
    BI(2,2) = -15.0*H*CT4/R2
    BI(2,3) = -35.0*H*CT6/R2
C
C   EVALUATE [B][N]
    CALL VZERO (BN(1,1),12)
    CALL MAXB (BI(1,1),N(1,1),BN,2,3,2,2,6,3)
C
C   EVALUATE [BNT] = TRANSPOSE OF [BN]
    CALL VZERO (BNT(1,1),12)
    DO 30 IT=1,2
    DO 30 JT=1,6
 30 BNT(JT,IT)=BN(IT,JT)
C
C   EVALUATE [DBI]=[D][BN]
    CALL VZERO (DBI(1,1),12)
    CALL MAXB (D(1,1),BN(1,1),DBI,2,2,2,2,6,2)
C
C   EVALUATE [BTDBI] = [BN]T [DBI]
    CALL VZERO (BTDBI(1,1),36)
    CALL MAXB (BNT(1,1),DBI(1,1),BTDBI,6,2,6,6,6,2)
C
C   EVALUATE STIFFNESS MATRIX.
C   [STOI] = [STOI] PREVIOUS + CONOV*WEIGHTS * [P]
    DO 70 KT=1,6
    DO 70 L=1,6
 70 STOI(KT,L) = STOI(KT,L) +
   & CONOV*WI(I)*WJ(J)*WK(K)*BTDBI(KT,L)
C
C   **** OUT OF PLANE OVALISATION ****
C   [BO]
    CALL VZERO (BO(1,1),6)
    BO(1,1) = (CT*ST2-0.5*ST*CT2)/BRAD
    BO(1,2) = (CT*ST4-0.25*ST*CT4)/BRAD
```

```
      BO(1,3) = (CT*ST6-(1.0/6.0)*ST*CT6)/BRAD
C
      R2 = PRAD**2
      BO(2,1) = -3.0*H*ST2/R2
      BO(2,2) = -15.0*H*ST4/R2
      BO(2,3) = -35.0*H*ST6/R2
C
C     EVALUATE [B][N]
      CALL VZERO (BN(1,1),12)
      CALL MAXB (BO(1,1),N(1,1),BN,2,3,2,2,6,3)
C
C     EVALUATE [BNT] = TRANSPOSE OF [BN]
      CALL VZERO (BNT(1,1),12)
      DO 130 IT=1,2
      DO 130 JT=1,6
  130 BNT(JT,IT)=BN(IT,JT)
C
C
C     EVALUATE [DBO]=[D][BN]
      CALL VZERO (DBO(1,1),12)
      CALL MAXB (D(1,1),BN(1,1),DBO,2,2,2,2,6,2)
C
C     EVALUATE [BTDBO] = [BN]T [DBO]
      CALL VZERO (BTDBO(1,1),36)
      CALL MAXB (BNT(1,1),DBO(1,1),BTDBO,6,2,6,6,6,2)
C
C     EVALUATE STIFFNESS MATRIX.
C     [STOO] = [STOO] PREVIOUS + CONOV*WEIGHTS * [BTDBO]
      DO 170 KT=1,6
      DO 170 L=1,6
  170 STOO(KT,L) = STOO(KT,L)+
     & CONOV*WI(I)*WJ(J)*WK(K)*BTDBO(KT,L)
C
C
C **** IN PLANE COUPLING INTEGRAL MATRIX ****
C     EVALUATE [B1]T [AMATI]T [D] [BI] [N], MUILT BY (WI(I)*WJ(J)*WK(K))
C     AND ADD TO LAST VALUE: [STIFF] = [STIFF] + [BTDB] WI WJ WK
C
C     EVALUATE [AMATI]T [DBI]
      SP = SIN(PHIB)
      CP = COS(PHIB)
C
C     TRANSPOSE OF IN PLANE AMAT: AMATI
C
      SMALLD = (PRAD/BRAD)**2/2.0
      RCTOR = PRAD*CT/BRAD
      CALL VZERO (AMATI(1,1),12)
      AMATI(1,1) = -RCTOR
      AMATI(3,1) = 2.0*(SMALLD+RCTOR)*SP
      AMATI(5,1) = -2.0*(SMALLD+RCTOR)*CP
C
C     EVALUATE [AI]T [DBI]
      CALL VZERO(ATDBI(1,1),36)
      CALL MAXB (AMATI(1,1),DBI(1,1),ATDBI,6,2,6,6,6,2)
C
C     EVALUATE INTEGRAL MATRIX.
C     [STCUPI] = [STCUPI] PREVIOUS + CCUPI*WEIGHTS * [P]
      DO 200 KT=1,6
      DO 200 LT=1,6
  200 STCUPI(KT,LT) = STCUPI(KT,LT)+
     1 CCUPI*WI(I)*WJ(J)*WK(K)*ATDBI(KT,LT)
C
C     **** OUT OF PLANE COUPLING INTEGRAL MATRIX ****
C     THE O/P MATRICES ARE LARGER THAN I/P DUE TO TORSION
C     O/P CONSTITUTIVE MATRIX
      CALL VZERO (DO(1,1),9)
      DO(1,1) = 1.0
      DO(1,2) = NUXY
      DO(2,1) = NUXY
      DO(2,2) = 1.0
      DO(3,3) = (1-NUXY)/2.0
C
C     --- O/P COUPLING [B] IS 3X3.
C     O/P [B] FOR COUPLIN = [BCUP]
      CALL VZERO (BCUP(1,1),9)
      DO 210 IT = 1,2
      DO 210 JT = 1,3
  210 BCUP(IT,JT) = BO(IT,JT)
C     [AMATO] T
      CALL VZERO (AMATO(1,1),18)
      SMALLB = (2*(1+NUXY))/(2+NUXY)
C
      AMATO(2,1) = -(2.0-SMALLB)*SP*ST
      AMATO(4,1) = (2.0-SMALLB)*CP*ST
      AMATO(2,3) = SMALLB*CP
      AMATO(4,3) = SMALLB*SP
      AMATO(6,3) = 1.0
C
C     EVALUATE [BCUP] [N]
      CALL VZERO (BCUPN(1,1),18)
      CALL MAXB (BCUP(1,1),N(1,1),BCUPN,3,3,3,3,6,3)
C     [DO] [BCUP]
      CALL VZERO (DBCUPN(1,1),18)
      CALL MAXB (DO(1,1),BCUPN(1,1),DBCUPN,3,3,3,3,6,3)
C
C     EVALUATE [AO]T [DBCUPN]
      CALL VZERO (ATDBO(1,1),36)
      CALL MAXB (AMATO(1,1),DBCUPN(1,1),ATDBO,6,3,6,6,6,3)
C
C     EVALUATE INTEGRAL MATRIX.
C     [STCUPO] = [STCUPO] PREVIOUS + CCUPO*WEIGHTS * [P]
      DO 240 KT=1,6
      DO 240 LT=1,6
  240 STCUPO(KT,LT) = STCUPO(KT,LT)+
     1 CCUPO*WI(I)*WJ(J)*WK(K)*ATDBO(KT,LT)
C
  300 CONTINUE
C     END OF NUMERICAL INTEGRATION LOOP
      RETURN
      END
C
```

SUBROUTINE REDUCE

```
      SUBROUTINE REDUCE (SE,RED)
C
C     STATIC CONDENSATION OF STIFFNESS MATRICES
C     REDUCES I/P AND O/P 12x12 MATRICES TO 6x6.
C     BASED ON GAUSSIAN ELIMINATION: REF. COOK.
C
C     NO REDUCTION OF FORCE VECTOR REQUIRED AS NODELESS
C     DOF HAVE ZERO CORRESPONDING GENERALISED FORCES.
C
C     FULL, SYMMETRIC STIFFNESS MATRIX SE.
C     CONDENSATION OPERATIONS ON LOWER TRIANGLE OF SE.
C     SIZE OF FULL MATRIX = NSIZE
C     NUMBER OF DOF TO BE REDUCED = NUM.  DOF TO BE REDUCED
C     STORED IN LAST NUM DOF.
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DOUBLE PRECISION SE(12,12),RED(6,6)
C
      CALL VZERO (RED(1,1),36)
      NSIZE = 12
      NUM = 6
C     CONDENSATION OF LOWER TRIANGLE OF SE.
      DO 30 K=1,NUM
      LL = NSIZE - K
      KK = LL + 1
      DO 20 L=1,LL
      IF (SE(KK,L) .EQ. 0.0) GO TO 20
      DUM = SE(KK,L)/SE(KK,KK)
      DO 10 M=1,L
```

```
10  SE(L,M) = SE(L,M) - SE(KK,M)*DUM
20  CONTINUE
30  CONTINUE
C   FILL IN THE UPPER TRIANGLE BY SYMMETRY.
    DO 40 K=1,LL
    DO 40 L=1,K
40  SE(L,K) = SE(K,L)
C
C   EQUATE FIRST 6 ROWS AND COLUMNS OF SE TO RED
C   RED IS THE REDUCED IN-PLANE MATRIX
    DO 50 I=1,6
    DO 50 J=1,6
50  RED(I,J) = SE(I,J)
    RETURN
    END
C
```

SUBROUTINE TRSUB

```
    SUBROUTINE TRSUB  (TR,TRPROP)
C
    IMPLICIT DOUBLE PRECISION (A-H,O-Z)
C
    EXTERNAL VZERO,MAXB,MATXB
C
    DOUBLE PRECISION T(3,3),TI(3,3),TJ(3,3),TR(12,12),
   1 THETA(3,3),TRPROP(7),DX,DY,DZ,PHI,ALPHA,CON,ALEN2,
   2 SALP1,SALP2,SALP3,CALP1,CALP2,CALP3
C
    DX=TRPROP(1)
    DY=TRPROP(2)
    DZ=TRPROP(3)
    PHI=TRPROP(4)
    ALPHA=TRPROP(5)
    CON=TRPROP(6)
    ALEN2=TRPROP(7)
C
    CALL VZERO (T(1,1),9)
    CALL VZERO (THETA(1,1),9)
    CALL VZERO (TI(1,1),9)
    CALL VZERO (TJ(1,1),9)
C
    ALENG = SQRT(ALEN2)
    ALENN1 = SQRT(CON)
C
    IF (ALENN1 .GT. .0001*ALENG) GO TO 200
    SALP1 = 0.0
    CALP1 = 1.0
    GO TO 250
200 SALP1 = DY/ALENN1
    CALP1 = DX/ALENN1
250 SALP2 = DZ/ALENG
    CALP2 = ALENN1/ALENG
    SALP3 = SIN(PHI)
    CALP3 = COS(PHI)
C   DEFINE EQUIVALENT STRAIGHT BEAM 3X3 TR MATRIX
    T(1,1) = CALP1*CALP2
    T(2,1) = -CALP1*SALP2*SALP3-SALP1*CALP3
    T(3,1) = SALP1*SALP3-CALP1*SALP2*CALP3
    T(1,2) = SALP1*CALP2
    T(2,2) = CALP1*CALP3-SALP1*SALP2*SALP3
    T(3,2) = -CALP1*SALP3-SALP1*SALP2*CALP3
    T(1,3) = SALP2
    T(2,3) = CALP2*SALP3
    T(3,3) = CALP2*CALP3
C   DEFINE CURVE-STRAIGHT BEAM NODE ROTATION MATRIX THETA
    THETA(1,1) = COS(ALPHA/2)
    THETA(3,1) = SIN(ALPHA/2)
    THETA(2,2) = 1.0
    THETA(1,3) = -(THETA(3,1))
    THETA(3,3) = THETA(1,1)
C   MULTIPLY T MATRIX BY THETA AND THETA TRANSPOSE RESP TO GET 3X3
C   NODAL TR MATRICES TI AND TJ.
C   SUBROUTINES ATIMB AND ATIMB USED FOR MULT.
C
C   MAXB AND MATXB ARE ANSYS IN-HOUSE MATRIX ROUTINES.
    CALL MAXB(THETA(1,1),T(1,1),TI(1,1),3,3,3,3,3,3)
    CALL MATXB(THETA(1,1),T(1,1),TJ(1,1),3,3,3,3,3,3)
C
C FILL OUT 12X12 TR MATRIX FROM TI AND TJ
    DO 260  I=1,3
       I3=I+3
         DO 260 J=1,3
           J3=J+3
       TR(I,J) = TI(I,J)
       TR(I3,J3) = TI(I,J)
260 CONTINUE
    DO 270 I=1,3
      I6 = I+6
      I9 = I+9
        DO 270 J=1,3
          J6=J+6
          J9=J+9
      TR(I6,J6) = TJ(I,J)
      TR(I9,J9) = TJ(I,J)
270 CONTINUE
    END
```

SUBROUTINE INVM

```
    SUBROUTINE INVM(A,D,N,NX,MX)
C
C   THIS PROGRAM COMPUTE THE INVERSE OF A MATRIX
C   USING THE GAUSS ELIMINATION METHOD
C
C   A : RECTANGULAR ARRAY OF SIZE  N X 2N
C   D : DETERMINANT
C   N : ORDER OF A
C   NX: ROW
C   MX: COLUMN
C
    IMPLICIT DOUBLE PRECISION (A-H,O-Z)
    DIMENSION A(NX,MX)
    NM1=N-1
    NP1=N+1
    NX2=N*2
C
C   PUT A UNIT MATRIX IN THE ARGUMENTED PART OF A
C
    DO 2 I=1,N
     IPN=I+N
     DO 1 J=1,N
      JPN=J+N
  1   A(I,JPN)=0.0
  2 A(I,IPN)=1
C
C   APPLY THE ELIMINATION PROCESS
C
    DO 10 K=1,NM1
    KP1=K+1
     C=A(K,K)
     IF (ABS(C)-1.0E-10) 3,3,8
  3  DO 6 J=KP1,N
      IF (ABS(A(J,K))-1.0E-10) 6,6,4
  4    DO 5 L=K,NX2
        C=A(K,L)
        A(K,L)=A(J,L)
  5     A(J,L)=C
       C=A(K,K)
       GOTO 8
  6    CONTINUE
  7  WRITE(6,6999) K
```

```
      D = 0.0
      GOTO 15
    8 DO 9 J = KP1,NX2
    9   A(K,J) = A(K,J)/C
      DO 10 I = KP1,N
        C = A(I,K)
        DO 10 J = KP1,NX2
   10     A(I,J) = A(I,J)-C*A(K,J)
      IF (ABS(A(N,N))-1.0E-10) 7,7,11
   11 DO 12 J = NP1,NX2
   12   A(N,J) = A(N,J)/A(N,N)
C
C     APPLY THE BACKSUBSTITUTION PROCESS
C
      DO 13 L = 1,NM1
        K = N-L
        KP1 = K+1
        DO 13 I = NP1,NX2
          DO 13 J = KP1,N
   13       A(K,I) = A(K,I)-A(K,J)*A(J,I)
C
C     PUT THE INVERSE IN THE FIRST  N X N  POSITIONS
C
      DO 14 I = 1,N
        DO 14 J = 1,N
          JPN = J+N
   14     A(I,J) = A(I,JPN)
   15 RETURN
 6999 FORMAT(24H **** SINGULARITY IN ROW,I2)
      END
```

A4.3.4 SUBROUTINE SR100

```
      SUBROUTINE SR100 (IELNUM,ITYP,KELOUT,ELVOL,KTIK,ZS,ZASS,ZSC)
C
C ********* STRESS PASS FOR 3-D CYL BEAM  ELEMENT ********
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      EXTERNAL TRACK,GETELD,PUTELD,SRPLT,MAXV,VZERO
      INTEGER IPLTAY(6),IEP
C
C ***** START STCOM STORAGE ********
      INTEGER  IELNUM,ITYP,KELIN(6),KELOUT(6),NR,KTIK,
     1 KEYERR,IOUT,NSTEPS,KFSTLD,ITTER,ITIME,NCUMIT,KRSTRT,ISPARE,
     2 K13,NPRPVL,MATST,K5,K16,IPROP,KCPDS,
     3 K20,KAY,MODE,ISYM,KAHD,IDEBUG,DXXX,
     4 ITYPE,MAT,IELEM,NROW,JTYPE,IPLOT,IPRINT,KTEMTP,KCONCV,KBICNV,
     5 KEYPLS,KEYCRP,KEYSWI,KYSUB(9),K21,NODES(20),  EPAR(50)
      REAL  ERRVAR(5)
C
      DOUBLE PRECISION
     1 DPZERO,DPHALF,DPONE,DPTWO,DPTEN,DTORAD,RADTOD,
     2 TREF,TUNIF,TOFSET, DELTIM,TIME,TIMOLD,TIME2,TIME3,DELT2,
     3 ACEL,OMEGA,CGOMEG,CGLOC,DXXX,
     4 ELMASS,XCENTR,YCENTR,ZCENTR,TFCP,SUBEX,  ERPAR(20),
     5 XYZEQ(20,3),X(20),Y(20),Z(20),  ELVOL
C
      COMMON /STCOM/ DPZERO,DPHALF,DPONE,DPTWO,DPTEN,DTORAD,RADTOD,
     1 TREF,TUNIF,TOFSET, DELTIM,TIME,TIMOLD,TIME2,TIME3,DELT2,
     2 ACEL(3),OMEGA(6),CGOMEG(6),CGLOC(3),      DXXX(16),
     3 KEYERR,IOUT,NSTEPS,KFSTLD,ITTER,ITIME,NCUMIT,KRSTRT,ISPARE,
     4 K13,NPRPVL,MATST,K5,K16,IPROP(20),KCPDS,
     5 K20,KAY(10),MODE,ISYM,KAHD,IDEBUG(10),     DXXX(41)
C
      EQUIVALENCE  (ITYPE,EPAR(1)), (MAT,EPAR(2)), (IELEM,EPAR(5)),
     1 (NROW,EPAR(7)), (JTYPE,EPAR(11)), (IPLOT,EPAR(12)),
     2 (IPRINT,EPAR(13)), (KTEMTP,EPAR(14)), (KCONCV,EPAR(16)),
     4 (KBICNV,EPAR(17)), (KEYPLS,EPAR(18)), (KEYCRP,EPAR(19)),
     5 (KEYSWI,EPAR(20)), (KYSUB(1),EPAR(21)), (K21,EPAR(30)),
     6 (NODES(1),EPAR(31))
C
      EQUIVALENCE  (ELMASS,ERPAR(1)), (XCENTR,ERPAR(2)),
     1 (YCENTR,ERPAR(3)), (ZCENTR,ERPAR(4)), (TFCP,ERPAR(5)),
     2 (SUBEX,ERPAR(6))
      EQUIVALENCE (X(1),XYZEQ(1,1)),(Y(1),XYZEQ(1,2)),(Z(1),XYZEQ(1,3))
C
      DOUBLE PRECISION
     1 ZS(KTIK,KTIK),ZASS(KTIK,KTIK),ZSC(KTIK),
     2 BRAD,ALPHA,PRAD,THICK,PHI,EX,ALPX,NUXY,DENS,DPPI,PROP(4),
     3 RVR(5),SVR(364),PRESS(1),B1(6,6),B2(6,6),
     4 TR(12,12),U(24),POSTD(20),CON,A1(6),B1(2,12),DMATI(2,2),
     5 SMI(6,12),DOFI(12),EPI(2),SIGI(2),ULOC(12),
     6 DOPO(12),SMO(6,12),DMATO(3,3),A2(3,6),BO(3,12),BOB(3,6),
     7 EPO(3),SIGO(3),SIG(3),BN(2,6),BOVI(2,3),BOVO(2,3),N(2,6),
     8 N1,N2
C
      EQUIVALENCE (RVR(1),BRAD), (RVR(2),ALPHA), (RVR(3),PRAD),
     1 (RVR(4),THICK), (RVR(5),PHI)
C
C     EQUIVALENCE SAVED VARIABLES SVR():
      EQUIVALENCE (SVR(1),B1(1,1)),(SVR(37),B2(1,1)),
     1 (SVR(73),SMI(1,1)), (SVR(145),PROP(1)), (SVR(149),TR(1,1)),
     2 (SVR(293),SMO(1))
C
C     EQUIVALENCE MATERIAL PROPERTIES PROP():
      EQUIVALENCE  (PROP(1),EX), (PROP(2),ALPX), (PROP(3),NUXY),
     1 (PROP(4),DENS)
C
      DATA DPPI / 3.14159265358979 3D0 /
C
      CALL TRACK (5,'SR100 ')
      CALL GETELD (IELNUM,ITYP,EPAR(1),ERPAR(1),CON,CON,
     1 PRESS(1),CON,RVR(1),SVR(1),XYZEQ(1,1),U(1))
C
C     CONVERT ALPHA TO RADIANS
      ALPHA = ALPHA*DPPI/180.0
C
C ~~~~~~~~~~~~~~~~ STRESS PASS ~~~~~~~~~~~~~~~~~~
C STRESSES ARE EVALUATED AROUND CIRC AT BOTH NODES
C
C     EVALUATE I/P CONSTITUTIVE MATRIX [DMATI]
      CALL VZERO (DMATI(1,1),4)
      DCON = EX/(1.0-NUXY**2)
      DMATI(1,1) = DCON
      DMATI(1,2) = DCON*NUXY
      DMATI(2,1) = DMATI(1,2)
      DMATI(2,2) = DCON
C
      CALL VZERO (DMATO(1,1),9)
      DMATO(1,1) = DCON
      DMATO(1,2) = DCON*NUXY
      DMATO(2,1) = DMATO(1,2)
      DMATO(2,2) = DCON
      DMATO(3,3) = DCON*(1-NUXY)/2
C
C     EVALUATE LOCAL  DISPLACEMENT VECTOR ULOC (IN LOCAL CSYS)
C     FROM GLOBAL DISP. VECTOR U.
      CALL MAXV (TR(1,1),U(1),ULOC,12,12)
C
C     ******* RECOVER NODELESS DOF. ********
C     IN-PLANE DOF
C     SMI CONTAINS LAST 6 ROWS OF I/P ZSFULL AFTER CONDENSATION
C
C     DOFI = I/P DOF IN LOCAL CSYS.
C     THE  KNOWN (BEAM) DOF STORED IN FIRST 6 ROWS.
C     NODELESS DOF RECOVERED AND STORED IN LAST 6.
      CALL VZERO (DOFI(1),9)
      DOFI(1) = ULOC(1)
      DOFI(2) = ULOC(3)
      DOFI(3) = ULOC(5)
      DOFI(4) = ULOC(7)
      DOFI(5) = ULOC(9)
      DOFI(6) = ULOC(11)
```

```
C
C   RECOVERY ALGORITHM
    DO 30 J=1,6
    JJ = 6+J
    DUM = 0.0
    K = JJ - 1
    DO 20 L=1,K
 20 DUM = DUM + SMI(J,L)*DOFI(L)
 30 DOFI(JJ) = (0.0 - DUM)/SMI(J,JJ)
C
C   OUT OF PLANE DEGREES OF FREEDOM
C
C   SMO CONTAINS LAST 6 ROWS OF O/P ZSFULL AFTER CONDENSATION
C   DOFO =  O/P DOF IN LOCAL CSYS.
C   THE KNOWN (BEAM) DOF STORED IN FIRST 6 ROWS.
C   NODELESS DOF RECOVERED AND STORED IN LAST 6.
    DOFO(1) = ULOC(2)
    DOFO(2) = ULOC(4)
    DOFO(3) = ULOC(6)
    DOFO(4) = ULOC(8)
    DOFO(5) = ULOC(10)
    DOFO(6) = ULOC(12)
C
C   RECOVERY ALGORITHM
    DO 50 J=1,6
    JJ = 6+J
    DUM = 0.0
    K = JJ - 1
    DO 40 L=1,K
 40 DUM = DUM + SMO(J,L)*DOFO(L)
 50 DOFO(JJ) = (0.0 - DUM)/SMO(J,JJ)
C
C   ALL NODAL AND NODELESS DOF NOW EVALUATED.
C
C   < < < < < < < SET UP LOOP FOR STRESS AND STRAIN EVALUATION > > > > > > >
C
C   LOCATION: NODE I
C   PHIB IS THE MERIDIONAL ANGLE FOR STRESS EVALUATION
C
    RATIO = (PRAD/BRAD)**2
    D = RATIO/2.0
    B = (2*(1+NUXY))/(2+NUXY)
    OPCON = PRAD/(BRAD**2)
C
    PRINT *,'INPUT SURFACE FOR STRESS CALCS'
    PRINT *,'-1 = INNER'
    PRINT *,'0 = MIDDLE'
    PRINT *,'1 = OUTER'
    READ (5,*) SURF
C
    PRINT *,'INPUT STRESS NORMALISATION FACTOR'
    READ (5,*) SFACT
C
C   LOOP IEP EVALUATES STRAIN/STRESS AT LOCATIONS AROUND CIRC
C   AT NODE I, IN STEPS OF 10 DEGREES.
C   HERE, FOR DEVELOPMENT TESTS, IN-PLANE STRESSES EVALUATED.
    WRITE (40,*) '    THETA    SIGPHI    SIGTH    TAU'
    THETA = -10.0
    DO 100 IEP=1,36
C
    THETA = THETA + 10.0
    THETA = THETA * DPPI/180.0
C
C   ********* IN PLANE STRESS AND STRAIN ***********
C
C   EVALUATE IN PLANE B MATRIX
C   I/P A MATRIX, A1. (MULT BY THE 1/R USUALLY OUTSIDE THE BRACKETS)
    CALL VZERO (A1(1),6)
    CT = COS(THETA)
    ST = SIN(THETA)
    CP = 1.0
    SP = 0.0
    CONST = PRAD*CT/BRAD
    A1(1) = -CONST/BRAD
    A1(3) = (D+CONST)*2.0*SP/BRAD
    A1(5) = (D+CONST)*-2.0*CP/BRAD
    CALL VZERO (BI(1,1),24)
C
C   EVALUATE I/P [BI] MATRIX
C   BENDING CONTRIBUTIONS-FIRST 6 COLUMNS. : [BI] = 1/R [A1] [B1]
C   ONLY ROW 1 IS POPULATED:
    DO 70 I=1,6
    BI(1,I) = A1(1)*B1(1,I)+A1(3)*B1(3,I)+A1(5)*B1(5,I)
 70 CONTINUE
C
C   PUT I/P OV. MATRIX IN LAST SIX COLUMNS OF BI
    BI(1,7) = (CT*COS(2.0*THETA) + 0.5*ST*SIN(2.0*THETA))/BRAD
    BI(1,8) = (CT*COS(4.0*THETA) + 0.25*ST*SIN(4.0*THETA))/BRAD
    BI(1,9) = (CT*COS(6.0*THETA)
   1 + (1.0/6.0)*ST*SIN(6.0*THETA))/BRAD
C
    BI(2,7) = SURF*-3.0*THICK/2.0*COS(2.0*THETA)/PRAD**2
    BI(2,8) = SURF*-15.0*THICK/2.0*COS(4.0*THETA)/PRAD**2
    BI(2,9) = SURF*-35.0*THICK/2.0*COS(6.0*THETA)/PRAD**2
C
C   EVALUATE STRAINS {EPI} = [BI] {DOFI} = {EPI(PHI)  EPI(THETA)}
    CALL MAXV (BI(1,1),DOFI(1),EPI,2,12)
C
C   EVALUATE STRESSES {SIGI} = [D] {EPI} = {SIGI(PHI)  SIGI(THETA)}
    CALL MAXV (DMATI(1,1),EPI(1),SIGI,2,2)
    SIGI(1) = SIGI(1)/SFACT
    SIGI(2) = SIGI(2)/SFACT
C
C   ******* OUT OF PLANE STRESS AND STRAIN ********
C
C   EVALUATE O/P B MATRIX BO
C   O/P A MATRIX A2. MULT BY PRAD/(BRAD**2) USUALLY OUTSIDE BRACKETS
    CALL VZERO (A2(1,1),18)
    A2(1,2) = -(2.0-B)*SP*ST*OPCON
    A2(1,4) = (2.0-B)*CP*ST*OPCON
    A2(3,2) = B*CP*OPCON
    A2(3,4) = B*SP*OPCON
    A2(3,6) = OPCON
C
    CALL VZERO (BOB(1,1),18)
    CALL MAXB (A2(1,1),B2(1,1),BOB,3,6,3,3,6,6)
C
C   PUT BOB IN FIRST 6 COLUMNS OF BO
    CALL VZERO (BO(1,1),36)
    DO 80 I=1,3
    DO 80 J=1,6
    BO(I,J) = BOB(I,J)
 80 CONTINUE
C   NODE I
C   EVALUATE OVALISATION B MATRIX IN LAST SIX COLS. OF BO
    BO(1,7) = (CT*SIN(2.0*THETA) - 0.5*ST*COS(2.0*THETA))/BRAD
    BO(1,8) = (CT*SIN(4.0*THETA) - 0.25*ST*COS(4.0*THETA))/BRAD
    BO(1,9) = (CT*SIN(6.0*THETA)
   1 - (1.0/6.0)*ST*COS(6.0*THETA))/BRAD
C
    BO(2,7) = SURF*-3.0*THICK/2.0*SIN(2.0*THETA)/PRAD**2
    BO(2,8) = SURF*-15.0*THICK/2.0*SIN(4.0*THETA)/PRAD**2
    BO(2,9) = SURF*-35.0*THICK/2.0*SIN(6.0*THETA)/PRAD**2
C
C   EVALUATE STRAINS {EPO} = [BO] {DOFO} = {EPO(PHI) EPO(THETA) GAMMA}
    CALL MAXV (BO(1,1),DOFO(1),EPO,3,12)
C
C   EVALUATE STRESSES {SIGO} = [D] {EPO} = {SIGI(PHI) SIGI(THETA) TAU}
    CALL MAXV (DMATO(1,1),EPO(1),SIGO,3,3)
C   REMOVE 'C' FOR O/P STRESSES
C   SIGO(1) = SIGO(1)/SFACT
C   SIGO(2) = SIGO(2)/SFACT
C   SIGO(3) = SIGO(3)/SFACT
C
```

```
      THETA = THETA*180.0/DPPI
C
C     IN-PLANE STRESSES:
      WRITE (40,'(4F16.3)') THETA,SIGI(1),SIGI(2),SIG(3)
  100 CONTINUE
C
      IF (IPLOT .NE. 1) GO TO 900
C           ***** NUMBER OF FORCES (LEVEL 1) *****
      IPLTAY(2) = 12
C           ***** NUMBER OF STRESSES (LEVEL 2) *****
      IPLTAY(3) = 4
C           ***** NUMBER OF TOTAL SAVED (LEVELS 1, 2, AND 3)
      IPLTAY(4) = 20
C           ***** SAVE GEOMETRY FOR CONTOURS (0,NO  1,YES)
      IPLTAY(6) = 0
C
C     ****** PUT POSTDATA INFORMATION INTO POSTD ******
C
C     ***** PUT PLTARY INFORMATION ONTO FILE 12 *******
      CALL SRPLT (IELEM,ITYP,NROW,MAT,100,2,U(1),NODES(1),XYZEQ(1,1),
     1 IPLTAY(1),POSTD(1))
  900 CALL PUTELD (IELNUM,EPAR(1),ERPAR(1),CON,SVR(1))
C     PUTELD RESTORES DATA BACK TO FILE3
      CALL TRACK( 15,'SR100 ')
      RETURN
      END
```

A4.4 ANSYS USER ELEMENT PB3

```
      PROGRAM ANSYS
C                ELEMENT PB3
C                **********
C     ANSYS USER ELEMENT CODE FOR PIPE BEND ELEMENT
C
C     BASED ON VLASOV THIN WALL CIRC CURVED BEAM SOLUTION
C     AND PIECEWISE QUINTIC POLYNOMIAL OVALISATION.
C
C     MATRIX STATICALLY CONDENSED TO GIVE FINAL 12x12 STIFFNESS MATRIX.
C
C                DONALD MACKENZIE  OCT 1989
C
C     ANSYS VERSION 4.3A
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      EXTERNAL MAIN,STOPER
      EXTERNAL ELSHFN
      CALL NNDIM
      CALL MAIN
      CALL STOPER
      END
```

A4.4.1 SUBROUTINE USEREL

```
      SUBROUTINE USEREL (ITYP,IPARM,KYSUB,KEY3D,KDOF,KUNSYM,KTRANS)
C
      INTEGER IPARM(20,12),KYSUB(9),ITYP,JTYPE,KEY3D,KDOF,KUNSYM,KTRANS
C     **** DETERMINE TYPE OF ELEMENT AND THEN BYPASS IF NOT USER ELEMENT*
      JTYPE = IPARM(ITYP,3)
      IF (JTYPE .NE. 100) GO TO 100
C
C        ********** SET 3-D KEY **********
      KEY3D = 1
C        ********** DEFINE DOF SET AT EACH NODE **********
      KDOF = 0
C        ********** SET UNSYMMETRIC MATRIX KEY **********
      KUNSYM = 0
C
C     ***** DEFINE PATTERN FOR ELEMENT TO GLOBAL TRANSFORMATION ***
      KTRANS = 2
C        ********** DEFINE NUMBER OF NODES **********
      IPARM(ITYP,8) = 2
C        ***** DEFINE NUMBER OF TEMPERATURES (DELTEM,TEMPER) ******
      IPARM(ITYP,11) = 2
C
C        ***** DEFINE NUMBER OF PRESSURES (PRESS) *******
C     IF THERMAL ANALYSIS, TWO TIMES NUMBER OF CONVECTION SURFACES
      IPARM(ITYP,6) = 1
C        ***** SET ZEROED VARIABLES (NOITUEP)
      IPARM(ITYP,12) = 0
C        ***** DEFINE NUMBER OF REAL CONSTANTS FOR ELEMENT (RVR) ***
      IPARM(ITYP,10) = 5
C
C        ***** DEFINE NUMBER OF VARIABLES TO BE SAVED (SVR) *****
      IPARM(ITYP,7) = 253
C        ***** DEFINE NUMBER OF ROWS IN ELEMENT MATRICES (KTIK) ****
      IPARM(ITYP,9) = 12
C        ********** SET KEY TO IDENTIFY NON-LINEAR ELEMENT *******
      IPARM(ITYP,4) = 0
C        ********** SET KEY FOR THERMAL ELEMENT (KAN,-1) *********
      IPARM(ITYP,1) = 0
  100 RETURN
      END
```

A4.4.2 SUBROUTINE USERPT

```
      SUBROUTINE USERPT (INODE,JTYPE,KSHAPE,NNODE)
C ********** USER SUBROUTINE FOR ANSYS PLOT SHAPE **********
C     DEFINE ELEMENT SHAPE AND NUMBER OF NODES, FOR PLOTTING
      INTEGER INODE(20),JTYPE,KSHAPE,NNODE
C     *********BYPASS IF NOT USER ELEMENT (JTYPE = 100) *********
      IF (JTYPE .NE. 100) GO TO 100
C     ****** SELECT SHAPE TO BE PLOTTED BY SETTING KSHAPE *******
      KSHAPE = 2
C        ********** SET NUMBER OF ACTUAL NODES **********
      NNODE = 2
  100 RETURN
      END
```

A4.4.3 SUBROUTINE ST100

```
      SUBROUTINE ST100 (IELNUM,ITYP,KELIN,KELOUT,NR,KTIK,ZS,
     1 ZASS,DAMP,GSTIF,ZSC)
C
C ******* STIFFNESS PASS FOR PIPE BEND ELEMENT PB3 *********
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
C
C     -------- DECLARE EXTERNAL SUBROUTINES AND FUNCTIONS ----------
      EXTERNAL TRACK,GETELD,PUTELD,PROPEV,NONTBL,VZERO,MHTCH,USEERR,
     1 MATXV,MAXV,TRSUB,MAXB,MATXB,INVM,REDUCE
C
C     ---------- COMMON BLOCK DECLARATIONS ----------
C     STANDARD ST100 INTEGER VARIABLES ASSOCIATED WITH COMMON:
      INTEGER  IELNUM,ITYP,KELIN(6),KELOUT(6),NR,KTIK,
     1 KEYERR,IOUT,NSTEPS,KFSTLD,ITTER,ITIME,NCUMIT,KRSTRT,ISPARE,
     2 K13,NPRPVL,MATST,K5,K16,IPROP,KCPDS,
     3 K20,KAY,MODE,ISYM,KAHD,IDEBUG,DXXX,
     4 ITYPE,MAT,IELEM,NROW,JTYPE,IPLOT,IPRINT,KTEMTP,KCONCV,KBICNV,
     5 KEYPLS,KEYCRP,KEYSWL,KYSUB(9),K21,NODES(20),  EPAR(50)
      REAL  ERRVAR(5)
C
C     STANDARD ST100 DOUBLE PRECISION VARIABLES ASSOCIATED WITH COMMON:
      DOUBLE PRECISION
     1 DPZERO,DPHALF,DPONE,DPTWO,DPTEN,DTORAD,RADTOD,
     2 TREF,TUNIF,TOFSET, DELTIM,TIME,TIMOLD,TIME2,TIME3,DELT2,
     3 ACEL,OMEGA,CGOMEG,CGLOC,DXXX,
     4 ELMASS,XCENTR,YCENTR,ZCENTR,TFCP,SUBEX,  ERPAR(20),
     5 XYZEQ(20,3),X(20),Y(20),Z(20),  ELVOL
C
C     ST100 COMMON BLOCK: STCOM
      COMMON /STCOM/ DPZERO,DPHALF,DPONE,DPTWO,DPTEN,DTORAD,RADTOD,
     1 TREF,TUNIF,TOFSET, DELTIM,TIME,TIMOLD,TIME2,TIME3,DELT2,
     2 ACEL(3),OMEGA(6),CGOMEG(6),CGLOC(3),      DXXX(16),
     3 KEYERR,IOUT,NSTEPS,KFSTLD,ITTER,ITIME,NCUMIT,KRSTRT,ISPARE,
     4 K13,NPRPVL,MATST,K5,K16,IPROP(20),KCPDS,
     5 K20,KAY(10),MODE,ISYM,KAHD,IDEBUG(10),     DXX(41)
C
C     EQUIVALENCING OF STCOM VARIABLES
      EQUIVALENCE  (ITYPE,EPAR(1)), (MAT,EPAR(2)), (IELEM,EPAR(5)),
     1 (NROW,EPAR(7)), (JTYPE,EPAR(11)), (IPLOT,EPAR(12)),
     2 (IPRINT,EPAR(13)), (KTEMTP,EPAR(14)), (KCONCV,EPAR(16)),
     4 (KBICNV,EPAR(17)), (KEYPLS,EPAR(18)), (KEYCRP,EPAR(19)),
     5 (KEYSWL,EPAR(20)), (KYSUB(1),EPAR(21)), (K21,EPAR(30)),
     6 (NODES(1),EPAR(31))
      EQUIVALENCE  (ELMASS,ERPAR(1)), (XCENTR,ERPAR(2)),
     1 (YCENTR,ERPAR(3)), (ZCENTR,ERPAR(4)), (TFCP,ERPAR(5)),
     2 (SUBEX,ERPAR(6))
      EQUIVALENCE (X(1),XYZEQ(1,1)),(Y(1),XYZEQ(1,2)),(Z(1),XYZEQ(1,3))
C
```

```
C ---------- USER DEFINED VARIABLES ----------
C INTEGER, REAL AND DOUBLE PRECISION VARIABLES DEFINED HERE BY USER.
C
C DECLARE INTEGER VARIABLES:
      INTEGER I,J,K,I3,J3,I6,J6,LP(4),NSTR,NUM,KDEMO,NFKEY,
     1 KI,KO,LI,LO
C
C DECLARE DOUBLE PRECISION VARIABLES:
      DOUBLE PRECISION ZS(KTIK,KTIK),ZASS(KTIK,KTIK),
     & ZSC(KTIK),TR(12,12),ZSTEMP(12,12),
     1 U(240),PROP(4),EX,ALPX,NUXY,DENS,ALEN2,ALENG,DX,DY,DZ,
     2 AFLU,TEMPER(2),CON,AVETEM,TRPROP(7),MPROP(1),LDPROP(1),DPPI,
     3 BRAD,ALPHA,PRAD,THICK,PHI,RVR(5),SVR(253)
      DOUBLE PRECISION ZSI(6,6),ZSO(6,6),BI(6,12),B1(6,6),B2(6,6),
     2 CA,SA,CM1,C2A,S2A,C2M1,TRIG1,TRIG2,
     3 AREA,POLMOM,SECMOM,EI,GJ,RATIO,D,B,DM1,DM2,
     4 DTERM1,DTERM2,DTERM3,NRINV,V,BTERM1,BTERM2,BB,BM1,BTERM3,
     5 BTERM4,H,CONST
      DOUBLE PRECISION CONOV,PR4,BRT2,ZSOV(2,2),ZSCUP(6,2),
     1 CUPMAT(6,2),ZSFULL(8,8),SMI(2,8),SMO(2,8)
C
C ---- USER EQUIVALENCING OF REAL AND SAVED VARIABLES (RVR, SVR) ----
C
C EQUIVALENCE REAL VARIABLES RVR():
C BRAD = BEND RAD,  ALPHA = BEND ANGLE,  PRAD = PIPE RAD,  THICK = WALL THK.
C PHI = ORIENTATION ANGLE
      EQUIVALENCE (RVR(1),BRAD), (RVR(2),ALPHA), (RVR(3),PRAD),
     1 (RVR(4),THICK), (RVR(5),PHI)
C
C EQUIVALENCE SAVED VARIABLES SVR():
      EQUIVALENCE (SVR(1),B1(1,1)),(SVR(37),B2(1,1)),
     1  (SVR(73),PROP(1)), (SVR(77),TR(1,1)),
     2 (SVR(221),SMI(1,1)), (SVR(237),SMO(1,1))
C
C ---------- EQUIVALENCING OF MATERIAL PROPERTIES ----------
      EQUIVALENCE  (PROP(1),EX), (PROP(2),ALPX), (PROP(3),NUXY),
     1 (PROP(4),DENS)
C
      CALL TRACK(5,'ST100')
C
C  ---- READING IN ELEMENT INFORMATION: SUBROUTINE GETELD ----
C
      CALL GETELD (IELNUM,ITYP,EPAR(1),ERPAR(1),CON,TEMPER(1),
     1 CON,CON,RVR(1),SVR(1),XYZBQ(1,1),U(1))
C
C   OPEN FILE PB3RES.DAT FOR OUTPUT OF ELEMENT INFORMATION
C   REQUIRED AT DEVELOPMENT STAGE.  (IN THIS FINAL DEVELOPMENT
C   VERSION ONLY STRESS DISTRIBUTIONS AT NODE I WRITTEN TO FILE).
      OPEN (UNIT=40,FILE='PB3RES',STATUS='NEW')C
C   CONVERT BEND AND ORIENTATION ANGLES TO RADIANS.
      DATA DPPI / 3.14159265358979 3D0 /
      PHI = PHI*DPPI/180.0D0
      ALPHA = ALPHA*DPPI/180.0D0
C
C -- READING IN ELEMENT MATERIAL PROPERTIES: SUBROUTINE PROPEV --
C
C   SET UP INTEGER ARRAY FOR ACCESSING MATERIAL PROPERTIES
      DATA LP / 1, 2, 3,10/
C
      AVETEM = DPHALF*(TEMPER(1) + TEMPER(2))
C
      CALL PROPEV (IELEM,MAT,JTYPE,LP(1),AVETEM,PROP(1),4)
C
C ---------- GEOMETRY VALIDITY CHECK ----------
C
      DX = X(2) - X(1)
      DY = Y(2) - Y(1)
      DZ = Z(2) - Z(1)
      CON = DX**2 + DY**2
      ALEN2 = CON + DZ**2
      IF (ALEN2 .GT. 0.0) GO TO 150
      WRITE (IOUT,2000) IELEM
 2000 FORMAT ('ZERO LENGTH ELEMENT' ,I5)
      KEYERR = 1
      NFKEY = 1
      CALL USEERR (NFKEY)
      GO TO 990
C
  150 CONTINUE
C
C ---------- CALCULATE MASS AND APPROX. CENTROID ----------
C
      XCENTR = (X(1) + X(2))*DPHALF
      YCENTR = (Y(1) + Y(2))*DPHALF
      ZCENTR = (Z(1) + Z(2))*DPHALF
      AREA = 2*DPPI*PRAD*THICK
      AFLU = DPPI*(PRAD-(THICK/2))**2
      ELMASS =  (DENS*AREA+DENSFL*AFLU)*BRAD*ALPHA
C
C ---------- END OF CHECK RUN OR ERROR DETECTED ----------
C
      IF ((NSTEPS .EQ. 0) .OR. (KEYERR.EQ.1)) GO TO 990
C
C -------- EVALUATE THE ELEMENT TRANSFORMATION MATRIX ---------
C TRANSFORMATION MATRIX TR IS EVALUATED IN THE SUBROUTINE TRSUB.
C INFORMATION REQUIRED TO CALCULATE MATRIX VALUES IS PASSED IN
C BY ARRAY TRPROP.
C
      TRPROP(1) = DX
      TRPROP(2) = DY
      TRPROP(3) = DZ
      TRPROP(4) = PHI
      TRPROP(5) = ALPHA
      TRPROP(6) = CON
      TRPROP(7) = ALEN2
C
C ZERO THE TR MATRIX.
      CALL VZERO (TR(1,1),144)
C
      CALL TRSUB (TR,TRPROP)
C
C ---------- EVALUATE THE ELEMENT STIFFNESS MATRIX ----------
C
C            ******* BEAM BENDING ********
C
C             CLOSED FORM STIFFNESS MATRIX
C
C CHECK MATRIX IS REQUIRED.
      IF (KELIN(1) .NE. 1) GO TO 400
      CALL VZERO (ZS(1,1),144)
C
C   EVALUATE CONSTANTS AND TRIG FUNCTIONS.
      CA = COS(ALPHA)
      SA = SIN(ALPHA)
      CM1 = CA-1.0
      C2A = COS(2.0*ALPHA)
      S2A = SIN(2.0*ALPHA)
      C2M1 = C2A-1.0
      TRIG1 = 2.0*ALPHA-S2A
      TRIG2 = 2.0*ALPHA+S2A
C
      AREA = 2.0*DPPI*PRAD*THICK
      POLMOM = AREA*PRAD**2
      SECMOM = POLMOM/2.0
      EI = EX*SECMOM
      GJ = EX*POLMOM/(2.0+2.0*NUXY)
C
      RATIO = (PRAD/BRAD)**2
      D = RATIO/2.0
      B = 2.0*EI/(EI+GJ)
C
      DM1 = 1.0-D
```

```
      DM2 = 1.0-2.0*D
      DTERM1 = 2.0*DM1/BRAD
      DTERM2 = ALPHA*SA+DM2*CA
      DTERM3 = -ALPHA*CA+DM2*SA
C
      NRINV = -1.0/BRAD
      V = 1/(2.0*(1.0+NUXY))
C
      BB = B**2
      BM1 = 1.0-B
      BTERM1 = (2.0-B)**2
      BTERM3 = (-ALPHA*SA+CA*BM1)/BRAD
      BTERM4 = (ALPHA*CA+SA*BM1)/BRAD
C
      H = 1.0+RATIO/2.0
      CONST = EI/BRAD**3
      CONST = CONST/(1.0-NUXY**2)
      CONOV = CONST*ALPHA/RATIO
C
C *** IN PLANE STIFFNESS MATRIX ***
C
C     EVALUATE IN-PLANE DISPLACEMENT FIELD CONSTANTS,
C     MATRIX [B1] = INV. [C1]
      CALL VZERO (BI(1,1),72)
C
C     PUT [C1] IN [BI]
      BI(1,3) = DM2
      BI(1,4) = -1.0
      BI(1,6) = 1.0
      BI(2,1) = 1.0
      BI(2,2) = 1.0
      BI(3,3) = -DTERM1
      BI(3,6) = NRINV
      BI(4,1) = ALPHA
      BI(4,2) = SA
      BI(4,3) = DTERM2
      BI(4,4) = -CA
      BI(4,5) = DTERM3
      BI(4,6) = 1.0
      BI(5,1) = 1.0
      BI(5,2) = CA
      BI(5,3) = ALPHA*CA
      BI(5,4) = SA
      BI(5,5) = ALPHA*SA
      BI(6,1) = ALPHA*NRINV
      BI(6,3) = -DTERM1*CA
      BI(6,5) = -DTERM1*SA
      BI(6,6) = NRINV
C
C     INVERT BI
      CALL INVM(BI,DET1,6,6,12)
C
C     EQUATE FIRST 6X6 OF BI WITH B1
      DO 322 I=1,6
      DO 322 J=1,6
  322 B1(I,J) = BI(I,J)
C
C     EVALUATE ZSI = MATRIX OF IN-PLANE INTEGRALS
      CALL VZERO(ZSI(1,1),36)
C
      ZSI(1,1) = CONST*ALPHA
      ZSI(1,3) = CONST*2.0*CM1
      ZSI(1,5) = CONST*2.0*SA
      ZSI(3,1) = ZSI(1,3)
      ZSI(3,3) = CONST*H*TRIG1
      ZSI(3,5) = CONST*H*C2M1
      ZSI(5,1) = ZSI(1,5)
      ZSI(5,3) = ZSI(3,5)
      ZSI(5,5) = CONST*H*TRIG2
C
C     EVALUATE I/P STIFFNESS [ZSI] = [B1]'[ZSI] [B1]
      CALL MHTCH(B1(1,1),ZSI(1,1),6,6,6)
C
C     EVALUATE OVALISATION MATRIX ZSOV AND IN PLANE COUPLING
C     INTEGRALS [ZSCUP] IN SUBROUTINE OVAL
      CALL OVAL (PRAD,BRAD,THICK,ALPHA,EX,NUXY,ZSOV,CUPMAT)
C
C     EVALUATE COUPLING STIFFNESS
      CALL MATXB (B1(1,1),CUPMAT(1,1),ZSCUP,6,6,6,6,2,6)
C
C     ASSEMBLE 9x9 IN-PLANE STIFFNESS MATRIX
      DO 325 I=1,6
      DO 325 J=1,6
  325 ZSFULL(I,J) = ZSI(I,J)
      DO 326 I=1,6
      DO 326 J=1,2
      JP6 = J+6
  326 ZSFULL(I,JP6) = ZSCUP(I,J)
C
      DO 327 I=1,2
      IP6 = I+6
      DO 327 J=1,6
  327 ZSFULL(IP6,J) = ZSCUP(J,I)
      DO 328 I=1,2
      DO 328 J=1,2
      IP6 = I+6
      JP6 = J+6
      ZSFULL(IP6,JP6) = ZSOV(I,J)
  328 CONTINUE
C
C     STATICALLY CONDENSE I/P STIFFNESS TO 6x6 MATRIX ZSI
      CALL VZERO(ZSI(1,1),36)
      CALL REDUCE (ZSFULL,ZSI)
C
C     STORE ELEMENTS OF REDUCED ZSFULL REQUIRED FOR D.O.F
C     RECOVERY IN SR100
      DO 330 I=1,2
      IP6 = I+6
      DO 330 J=1,8
  330 SMI(I,J) = ZSFULL(IP6,J)
C
C
C        *** OUT OF PLANE STIFFNESS MATRIX ***
C
C     EVALUATE O/P DISPLACEMENT FIELD CONSTANTS,
C     MATRIX [B2] = INV. [C2]
C     PUT [C2] IN [BI]
C
      CALL VZERO(BI(1,1),72)
      BI(1,1) = 1.0
      BI(1,4) = B
      BI(1,5) = -1.0
      BI(2,1) = 1/BRAD
      BI(2,2) = BM1/BRAD
      BI(3,3) = 1/BRAD
      BI(3,6) = NRINV
      BI(4,1) = CA
      BI(4,2) = ALPHA*CA-SA*B
      BI(4,3) = SA
      BI(4,4) = ALPHA*SA+CA*B
      BI(4,5) = -1.0
      BI(4,6) = -ALPHA
      BI(5,1) = CA/BRAD
      BI(5,2) = ALPHA*CA/BRAD
      BI(5,3) = SA/BRAD
      BI(5,4) = ALPHA*SA/BRAD
      BI(6,1) = SA*NRINV
      BI(6,2) = BTERM3
      BI(6,3) = CA/BRAD
      BI(6,4) = BTERM4
      BI(6,6) = NRINV
C
C     INVERT BI
C
```

```
      CALL INVM(BI,DET1,6,6,12)
C     EQUATE FIRST 6X6 OF BI TO B2
      DO 332 I=1,6
      DO 332 J=1,6
  332 B2(I,J) = BI(I,J)
C
C     EVALUATE ZSO = MATRIX OF O/P INTEGRALS
      CALL VZERO(ZSO(1,1),36)
C
      ZSO(2,2) = CONST*(BTERM1*0.25*TRIG1+BB*V*0.5*TRIG2)
      ZSO(2,4) = CONST*(BB*V*0.5-BTERM1*0.25)*(1-C2A)
      ZSO(2,6) = CONST*2.0*B*V*SA
      ZSO(4,2) = ZSO(2,4)
      ZSO(4,4) = CONST*(BTERM1*0.25*TRIG2+BB*V*0.5*TRIG1)
      ZSO(4,6) = CONST*2*B*V*(1-CA)
      ZSO(6,2) = ZSO(2,6)
      ZSO(6,4) = ZSO(4,6)
      ZSO(6,6) = CONST*2.0*ALPHA*V
C
C     EVALUATE O/P STIFFNESS MATRIX [ZSO]=[BI]'[ZSO][BI]
      CALL MHTCH(B2(1,1),ZSO(1,1),6,6,6)
C
C     EVALUATE THE OUT OF PLANE OVALISATION STIFFNESS
C     AND COUPLING INTEGRALS [ZSCUP] IN SUBROUTINE OVALO
      CALL VZERO(ZSCUP(1,1),24)
      CALL VZERO(CUPMAT(1,1),24)
      CALL VZERO(ZSOV(1,1),4)
C
      CALL OVALO (PRAD,BRAD,THICK,ALPHA,EX,NUXY,ZSOV,CUPMAT)
C     EVALUATE COUPLING STIFFNESS MATRIX
      CALL MATXB (B2(1,1),CUPMAT(1,1),ZSCUP,6,6,6,6,2,6)
C
C     ASSEMBLE 9X9 O/P MATRIX ZSFULL
      DO 335 I=1,6
      DO 335 J=1,6
  335 ZSFULL(I,J) = ZSO(I,J)
      DO 336 I=1,6
      DO 336 J=1,2
      JP6 = J+6
  336 ZSFULL(I,JP6) = ZSCUP(I,J)
      DO 337 I=1,2
      IP6 = I+6
      DO 337 J=1,6
  337 ZSFULL(IP6,J) = ZSCUP(J,I)
      DO 338 I=1,2
      IP6 = I+6
      DO 338 J=1,2
      JP6 = J+6
  338 ZSFULL(IP6,JP6) = ZSOV(I,J)
C
C     REDUCE O/P STIFFNESS TO 6x6 MATRIX ZSO
      CALL VZERO(ZSO(1,1),36)
      CALL REDUCE (ZSFULL,ZSO)
C
C     SAVE REDUCED ZSFULL TERMS REQUIRED FOR D.O.F. RECOVERY
C     IN SR100
      DO 340 I=1,2
      IP6 = I+6
      DO 340 J=1,8
  340 SMO(I,J) = ZSFULL(IP6,J)
C
C     ASSEMBLE FULL I/P-O/P 12x12 STIFFNESS MATRIX
C     I/P TERMS IN I-J ODD LOCATIONS, O/P TERMS IN EVEN
C
C     ZERO ZS
      CALL VZERO (ZS(1,1),144)
      DO 360 I=1,6
      KO = 2*I-1
      KE = 2*I
        DO 360 J=1,6
        LO = 2*J-1
        LE = 2*J
      ZS(KO,LO) = ZSI(I,J)
      ZS(KE,LE) = ZSO(I,J)
  360 CONTINUE
C
C ---------- STIFFNESS MATRIX TRANSFORMATION ----------
C ELEMENT MATRICES ARE TRANSFORMED TO THE GLOBAL CO-ORD.
C SYSTEM BY THE ANSYS SUBROUTINE MHTCH.
      CALL MHTCH (TR(1,1),ZS(1,1), KTIK,KTIK, KTIK)
C
C     SET KEY THAT MATRIX WAS COMPUTED.
      KELOUT(1) = 1
C
  400 CONTINUE
C
C ---------- OUTPUT ELEMENT DATA TO FILE 12 ----------
C ELEMENT DATA IS OUTPUT TO FILE 12 BY THE SUBROUTINE PUTELD.
C
  990 CALL PUTELD (IELNUM,EPAR(1),ERPAR(1),CON,SVR(1))
C
      CALL TRACK( 15,'ST100 ')
      RETURN
      END
C
```

SUBROUTINE REDUCE

```
      SUBROUTINE REDUCE (SE,RED)
C
C     STATIC CONDENSATION OF STIFFNESS MATRICES
C     REDUCES I/P AND O/P 7x7 MATRICES TO 6x6.
C     BASED ON GAUSSIAN ELIMINATION: REF.
C     COOK R.D.  CONCEPTS AND APPLICATIONS OF FINITE ELEMENT ANALYSIS
C     JOHN WILEY & SONS
C
C     NO REDUCTION OF FORCE VECTOR REQUIRED AS NODELESS
C     DOF HAVE ZERO CORRESPONDING GENERALISED FORCES.
C
C     FULL, SYMMETRIC STIFFNESS MATRIX SE.
C     CONDENSATION OPERATIONS ON LOWER TRIANGLE OF SE.
C     SIZE OF FULL MATRIX = NSIZE
C     NUMBER OF DOF TO BE REDUCED = NUM.  DOF TO BE REDUCED
C     STORED IN LAST NUM DOF.
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DOUBLE PRECISION SE(8,8),RED(6,6)
C
      NSIZE = 8
      NUM = 2
C     CONDENSATION OF LOWER TRIANGLE OF SE.
      DO 30 K=1,NUM
      LL = NSIZE - K
      KK = LL + 1
      DO 20 L=1,LL
      IF (SE(KK,L) .EQ. 0.0) GO TO 20
      DUM = SE(KK,L)/SE(KK,KK)
      DO 10 M=1,L
   10 SE(L,M) = SE(L,M) - SE(KK,M)*DUM
   20 CONTINUE
   30 CONTINUE
C     FILL IN THE UPPER TRIANGLE BY SYMMETRY.
      DO 40 K=1,LL
      DO 40 L=1,K
   40 SE(L,K) = SE(K,L)
C
C     EQUATE FIRST 6 ROWS AND COLUMNS OF SE TO RED
C     RED IS THE REDUCED IN-PLANE MATRIX
      DO 50 I=1,6
      DO 50 J=1,6
   50 RED(I,J) = SE(I,J)
```

```
      RETURN
      END
C
```

SUBROUTINE TRSUB

```
      SUBROUTINE TRSUB  (TR,TRPROP)
C
C     ROUTINE TO EVALUATE 12x12 TRANSFORMATION MATRIX OF A CURVED BEAM.
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
C
      EXTERNAL VZERO,MAXB,MATXB
C
      DOUBLE PRECISION T(3,3),TI(3,3),TJ(3,3),TR(12,12),
     1 THETA(3,3),TRPROP(7),DX,DY,DZ,PHI,ALPHA,CON,ALEN2,
     2 SALP1,SALP2,SALP3,CALP1,CALP2,CALP3
C
      DX = TRPROP(1)
      DY = TRPROP(2)
      DZ = TRPROP(3)
      PHI = TRPROP(4)
      ALPHA = TRPROP(5)
      CON = TRPROP(6)
      ALEN2 = TRPROP(7)
C
      CALL VZERO (T(1,1),9)
      CALL VZERO (THETA(1,1),9)
      CALL VZERO (TI(1,1),9)
      CALL VZERO (TJ(1,1),9)
C
      ALENG = SQRT(ALEN2)
      ALENN1 = SQRT(CON)
C
      IF (ALENN1 .GT. .0001*ALENG) GO TO 200
      SALP1 = 0.0
      CALP1 = 1.0
      GO TO 250
  200 SALP1 = DY/ALENN1
      CALP1 = DX/ALENN1
  250 SALP2 = DZ/ALENG
      CALP2 = ALENN1/ALENG
      SALP3 = SIN(PHI)
      CALP3 = COS(PHI)
C     DEFINE EQUIVALENT STRAIGHT BEAM 3X3 TR MATRIX
      T(1,1) = CALP1*CALP2
      T(2,1) = -CALP1*SALP2*SALP3-SALP1*CALP3
      T(3,1) = SALP1*SALP3-CALP1*SALP2*CALP3
      T(1,2) = SALP1*CALP2
      T(2,2) = CALP1*CALP3-SALP1*SALP2*SALP3
      T(3,2) = -CALP1*SALP3-SALP1*SALP2*CALP3
      T(1,3) = SALP2
      T(2,3) = CALP2*SALP3
      T(3,3) = CALP2*CALP3
C     DEFINE CURVE-STRAIGHT BEAM NODE ROTATION MATRIX THETA
      THETA(1,1) = COS(ALPHA/2)
      THETA(3,1) = SIN(ALPHA/2)
      THETA(2,2) = 1.0
      THETA(1,3) = -(THETA(3,1))
      THETA(3,3) = THETA(1,1)
C     MULTIPLY T MATRIX BY THETA AND THETA TRANSPOSE RESP TO GET 3X3
C     NODAL TR MATRICES TI AND TJ.
C     SUBROUTINES ATIMB AND ATIMB USED FOR MULT.
C
C     MAXB AND MATXB ARE ANSYS IN-HOUSE MATRIX ROUTINES.
      CALL MAXB(THETA(1,1),T(1,1),TI(1,1),3,3,3,3,3,3)
      CALL MATXB(THETA(1,1),T(1,1),TJ(1,1),3,3,3,3,3,3)
C
C FILL OUT 12X12 TR MATRIX FROM TI AND TJ
      DO 260 I = 1,3
         I3 = I+3
         DO 260 J = 1,3
            J3 = J+3
         TR(I,J) = TI(I,J)
         TR(I3,J3) = TI(I,J)
  260 CONTINUE
      DO 270 I = 1,3
         I6 = I+6
         I9 = I+9
         DO 270 J = 1,3
            J6 = J+6
            J9 = J+9
         TR(I,J6) = TJ(I,J)
         TR(I9,J9) = TJ(I,J)
  270 CONTINUE
      END
```

SUBROUTINE INVM

```
      SUBROUTINE INVM(A,D,N,NX,MX)
C
C     THIS PROGRAM COMPUTE THE INVERSE OF A MATRIX
C     USING THE GAUSS ELIMINATION METHOD. REF:
C     BREBBIA CA & FERRANTE AJ   COMPUTATIONAL METHODS FOR
C     THE SOLUTION OF ENGINEERING PROBLEMS    PENTECH PRESS.
C
C     A : RECTANGULAR ARRAY OF SIZE  N X 2N
C     D : DETERMINANT
C     N : ORDER OF A
C     NX: ROW
C     MX: COLUMN
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION A(NX,MX)
      NM1 = N-1
      NP1 = N+1
      NX2 = N*2
C
C     PUT A UNIT MATRIX IN THE ARGUMENTED PART OF A
C
      DO 2 I = 1,N
       IPN = I+N
       DO 1 J = 1,N
        JPN = J+N
    1   A(I,JPN) = 0.0
    2 A(I,IPN) = 1
C
C     APPLY THE ELIMINATION PROCESS
C
      DO 10 K = 1,NM1
      KP1 = K+1
       C = A(K,K)
       IF (ABS(C)-1.0E-10) 3,3,8
    3  DO 6 J = KP1,N
        IF (ABS(A(J,K))-1.0E-10) 6,6,4
    4   DO 5 L = K,NX2
         C = A(K,L)
         A(K,L) = A(J,L)
    5    A(J,L) = C
        C = A(K,K)
        GOTO 8
    6   CONTINUE
    7   WRITE(6,6999) K
        D = 0.0
        GOTO 15
    8   DO 9 J = KP1,NX2
    9    A(K,J) = A(K,J)/C
       DO 10 I = KP1,N
        C = A(I,K)
        DO 10 J = KP1,NX2
   10     A(I,J) = A(I,J)-C*A(K,J)
      IF (ABS(A(N,N))-1.0E-10) 7,7,11
   11 DO 12 J = NP1,NX2
```

```
 12  A(N,J)=A(N,J)/A(N,N)
C
C   APPLY THE BACKSUBSTITUTION PROCESS
C
    DO 13 L=1,NM1
     K=N-L
     KP1=K+1
     DO 13 I=NP1,NX2
      DO 13 J=KP1,N
 13     A(K,I)=A(K,I)-A(K,J)*A(J,I)
C
C   PUT THE INVERSE IN THE FIRST  N X N  POSITIONS
C
    DO 14 I=1,N
     DO 14 J=1,N
      JPN=J+N
 14   A(I,J)=A(I,JPN)
 15 RETURN
6999 FORMAT(24H **** SINGULARITY IN ROW,I2)
    END
C
C
```

SUBROUTINE OVAL

```
    SUBROUTINE OVAL (PR,BR,THICK,ALPHA,EX,NUXY,STIFF,CUPMAT)
C
C   ROUTINE TO EVALUATE IN-PLANE OVALISATION STIFFNESS MATRIX
C   AND COUPLING INTEGRAL MATRIX.
C
    IMPLICIT DOUBLE PRECISION (A-H,O-Z)
    DOUBLE PRECISION NUXY,ACIRC(5),WCIRC(5),
   2 ATHK(3),WTHK(3),A(2,6),D(2,2),DB1(2,2),DB2(2,2),
   3 DB3(2,2),DB4(2,2),
   3 B1(2,2),B2(2,2),B3(2,2),B4(2,2),WALP(3),AALP(3),
   5 CUP1(6,2),CUP2(6,2),CUP3(6,2),CUP4(6,2),CUP(6,2),
   6 CUPMAT(6,2),HM1(6,2),HM2(6,2),HM3(6,2),HM4(6,2),
   7 BDB1(2,2),BDB2(2,2),BDB3(2,2),BDB4(2,2),BDB(2,2),
   8 STIFF(2,2)
C
    PR2 = PR**2
    CONST = EX*PR*BR/(1.0-NUXY**2)
    CONCUP = CONST/BR
    DD = (PR/BR)**2/2.
C
    CALL HMAT (HM1,HM2,HM3,HM4,PR)
C
    D(1,1) = 1.
    D(1,2) = NUXY
    D(2,1) = NUXY
    D(2,2) = 1.
C
    DO 21 I=1,6
    DO 21 J=1,2
 21  CUPMAT(I,J) = 0.
C
C    WEIGHTS
    W1 = 0.555555555555556
    W2 = 0.888888888888889
    W3 = W1
C   ABSCISSA
    AB3 = 0.774596669241483
    AB2 = 0.0
    AB1 = -AB3
C
    WT1 = 0.236926885056189
    WT2 = 0.478628670499366
    WT3 =0.568888888888889
    WT4 = WT2
    WT5 = WT1
C
    BA1 = -0.906179845938664
    BA2 = -0.538469310105683
    BA3 = 0.0D0
    BA4 = -BA2
    BA5 = -BA1
C   ACTUAL INTEGRATION LIMITS:
C   THRO' THICKNESS H = -THICK/2 TO THICK/2.
C   AROUND CIRC A = - PI/4 TO PI/4
C   ALONG AXIS, 0 TO ALPHA
    T = THICK/2.0
    WTHK(1) = T*W1
    WTHK(2) = T*W2
    WTHK(3) = WTHK(1)
    ATHK(1) = T*AB1
    ATHK(2) = T*AB2
    ATHK(3) = T*AB3
C
    WALP(1) = ALPHA/2.*W1
    WALP(2) = ALPHA/2.*W2
    WALP(3) = ALPHA/2.*W3
    AALP(1) = ALPHA/2.*(1.+AB1)
    AALP(2) = ALPHA/2.*(1.+AB2)
    AALP(3) = ALPHA/2.*(1.+AB3)
C
    P4 = 3.1415927D0/4.0D0
    WCIRC(1) = P4*WT1
    WCIRC(2) = P4*WT2
    WCIRC(3) = P4*WT3
    WCIRC(4) = WCIRC(2)
    WCIRC(5) = WCIRC(1)
    ACIRC(1) = P4*BA1
    ACIRC(2) = P4*BA2
    ACIRC(3) = P4*BA3
    ACIRC(4) = P4*BA4
    ACIRC(5) = P4*BA5
C
C   INITIALISE ARRAYS
    DO 24 I=1,2
    DO 24 J=1,2
 24  STIFF(I,J) = 0.
    DO 25 I=1,6
    DO 25 J=1,2
    CUP1(I,J) = 0.
 25  CONTINUE
    DO 27 I=1,6
    DO 27 J=1,2
 27  CUP2(I,J) = 0.
    DO 28 I=1,6
    DO 28 J=1,2
 28  CUP3(I,J) = 0.
    DO 29 I=1,6
    DO 29 J=1,2
 29  CUP4(I,J) = 0.
    DO 30 I=1,6
    DO 30 J=1,2
 30  CUP(I,J) = 0.
C
C   START NUMERICAL INTEGRATION LOOP
C
    DO 100 ITHK=1,3
    H = ATHK(ITHK)
    DO 100 ICIRC=1,5
    AL = ACIRC(ICIRC)
    DO 100 IALP=1,3
    AALPP = AALP(IALP)
C
    AL2 = AL**2
    AL3 = AL*AL2
    AL4 = AL*AL3
    AL5 = AL*AL4
C
    A(1,1) = 0.
```

```
      A(1,2) = H/PR2
      A(1,3) = 2.*H*AL/PR2
      A(1,4) = H*(3.*AL2+6.)/PR2
      A(1,5) = H*(4.*AL3+24.*AL)/PR2
      A(1,6) = H*(5.*AL4+60.*AL2)/PR2
C
      TH = AL+P4
      ST = SIN(TH)
      CT = COS(TH)
      A(2,1) = -ST/BR
      A(2,2) = -(AL*ST + CT)/BR
      A(2,3) = -(AL2*ST + 2.*AL*CT)/BR
      A(2,4) = -(AL3*ST + 3.*AL2*CT)/BR
      A(2,5) = -(AL4*ST + 4.*AL3*CT)/BR
      A(2,6) = -(AL5*ST + 5.*AL4*CT)/BR
C
      BM1 = -PR/BR*CT
      BM2 = 2.*(DD+PR/BR*CT)*SIN(AALPP)
      BM3 = -2.*(DD+PR/BR*CT)*COS(AALPP)
C
      CALL MULT (B1,A,HM1)
      CALL BTDB (BDB1,B1,D)
      CALL MULTDB (DB1,D,B1)
      CUP1(1,1) = BM1*DB1(2,1)
      CUP1(1,2) = BM1*DB1(2,2)
      CUP1(3,1) = BM2*DB1(2,1)
      CUP1(3,2) = BM2*DB1(2,2)
      CUP1(5,1) = BM3*DB1(2,1)
      CUP1(5,2) = BM3*DB1(2,2)
C
      TH = AL+3.*P4
      ST = SIN(TH)
      CT = COS(TH)
      A(2,1) = -ST/BR
      A(2,2) = -(AL*ST + CT)/BR
      A(2,3) = -(AL2*ST + 2.*AL*CT)/BR
      A(2,4) = -(AL3*ST + 3.*AL2*CT)/BR
      A(2,5) = -(AL4*ST + 4.*AL3*CT)/BR
      A(2,6) = -(AL5*ST + 5.*AL4*CT)/BR
C
      BM1 = -PR/BR*CT
      BM2 = 2.*(DD+PR/BR*CT)*SIN(AALPP)
      BM3 = -2.*(DD+PR/BR*CT)*COS(AALPP)
C
      CALL MULT (B2,A,HM2)
      CALL BTDB (BDB2,B2,D)
      CALL MULTDB (DB2,D,B2)
      CUP2(1,1) = BM1*DB2(2,1)
      CUP2(1,2) = BM1*DB2(2,2)
      CUP2(3,1) = BM2*DB2(2,1)
      CUP2(3,2) = BM2*DB2(2,2)
      CUP2(5,1) = BM3*DB2(2,1)
      CUP2(5,2) = BM3*DB2(2,2)
C
      TH = AL+5.*P4
      ST = SIN(TH)
      CT = COS(TH)
      A(2,1) = -ST/BR
      A(2,2) = -(AL*ST + CT)/BR
      A(2,3) = -(AL2*ST + 2.*AL*CT)/BR
      A(2,4) = -(AL3*ST + 3.*AL2*CT)/BR
      A(2,5) = -(AL4*ST + 4.*AL3*CT)/BR
      A(2,6) = -(AL5*ST + 5.*AL4*CT)/BR
C
      BM1 = -PR/BR*CT
      BM2 = 2.*(DD+PR/BR*CT)*SIN(AALPP)
      BM3 = -2.*(DD+PR/BR*CT)*COS(AALPP)
C
      CALL MULT (B3,A,HM3)
      CALL BTDB (BDB3,B3,D)
      CALL MULTDB (DB3,D,B3)
      CUP3(1,1) = BM1*DB3(2,1)
      CUP3(1,2) = BM1*DB3(2,2)
      CUP3(3,1) = BM2*DB3(2,1)
      CUP3(3,2) = BM2*DB3(2,2)
      CUP3(5,1) = BM3*DB3(2,1)
      CUP3(5,2) = BM3*DB3(2,2)
C
      TH = AL+7.*P4
      ST = SIN(TH)
      CT = COS(TH)
      A(2,1) = -ST/BR
      A(2,2) = -(AL*ST + CT)/BR
      A(2,3) = -(AL2*ST + 2.*AL*CT)/BR
      A(2,4) = -(AL3*ST + 3.*AL2*CT)/BR
      A(2,5) = -(AL4*ST + 4.*AL3*CT)/BR
      A(2,6) = -(AL5*ST + 5.*AL4*CT)/BR
C
      BM1 = -PR/BR*CT
      BM2 = 2.*(DD+PR/BR*CT)*SIN(AALPP)
      BM3 = -2.*(DD+PR/BR*CT)*COS(AALPP)
C
      CALL MULT (B4,A,HM4)
      CALL BTDB (BDB4,B4,D)
      CALL MULTDB (DB4,D,B4)
      CUP4(1,1) = BM1*DB4(2,1)
      CUP4(1,2) = BM1*DB4(2,2)
      CUP4(3,1) = BM2*DB4(2,1)
      CUP4(3,2) = BM2*DB4(2,2)
      CUP4(5,1) = BM3*DB4(2,1)
      CUP4(5,2) = BM3*DB4(2,2)
C
      DO 40 I = 1,2
      DO 40 J = 1,2
 40   BDB(I,J) = BDB1(I,J) + BDB2(I,J) + BDB3(I,J) + BDB4(I,J)
      DO 44 I = 1,6
      DO 44 J = 1,2
      CUP(I,J) = (CUP1(I,J) + CUP2(I,J) + CUP3(I,J) + CUP4(I,J))/BR
 44   CONTINUE
C
      DO 50 I = 1,2
      DO 50 J = 1,2
 50   STIFF(I,J) = STIFF(I,J) +
     1 BDB(I,J)*CONST*WTHK(ITHK)*WCIRC(ICIRC)*WALP(IALP)
C
      DO 60 I = 1,6
      DO 60 J = 1,2
      CUPMAT(I,J) = CUPMAT(I,J) +
     1 CUP(I,J)*CONST*WTHK(ITHK)*WCIRC(ICIRC)*WALP(IALP)
 60   CONTINUE
 100  CONTINUE
      RETURN
      END
```

SUBROUTINE OVALO

```
      SUBROUTINE OVALO (PR,BR,THICK,ALPHA,EX,NUXY,STIFF,CUPMAT)
C
C     ROUTINE TO EVALUATE O/P OVALISATION STIFFNESS MATRIX
C     AND COUPLING INTEGRAL MATRIX.
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DOUBLE PRECISION NUXY,ACIRC(5),WCIRC(5),
     2 ATHK(3),WTHK(3),A(2,6),D(2,2),DB1(2,2),DB2(2,2),
     3 DB3(2,2),DB4(2,2),
     3 B1(2,2),B2(2,2),B3(2,2),B4(2,2),WALP(3),AALP(3),
     5 CUP1(6,2),CUP2(6,2),CUP3(6,2),CUP4(6,2),CUP(6,2),
     6 CUPMAT(6,2),HM1(6,2),HM2(6,2),HM3(6,2),HM4(6,2),
     7 BDB1(2,2),BDB2(2,2),BDB3(2,2),BDB4(2,2),BDB(2,2),
     8 STIFF(2,2)
C
      PR2 = PR**2
      CONST = EX*PR*BR/(1.0-NUXY**2)
```

```
      CONCUP = CONST/BR
      DD = (PR/BR)**2/2.
      B = (2*(1+NUXY))/(2+NUXY)
C
      CALL HMAT (HM1,HM2,HM3,HM4,PR)
C
      D(1,1) = 1.
      D(1,2) = NUXY
      D(2,1) = NUXY
      D(2,2) = 1.
C
      DO 21 I=1,6
      DO 21 J=1,2
   21 CUPMAT(I,J) = 0.
C
C     WEIGHTS
      W1 = 0.5555555555555556
      W2 = 0.8888888888888889
      W3 = W1
C     ABSCISSA
      AB3 = 0.774596669241483
      AB2 = 0.0
      AB1 = -AB3
C
      WT1 = 0.236926885056189
      WT2 = 0.478628670499366
      WT3 = 0.568888888888889
      WT4 = WT2
      WT5 = WT1
C
      BA1 = -0.906179845938664
      BA2 = -0.538469310105683
      BA3 = 0.0D0
      BA4 = -BA2
      BA5 = -BA1
C     ACTUAL INTEGRATION LIMITS:
C     THRO' THICKNESS H = -THICK/2 TO THICK/2.
C     AROUND CIRC A = - PI/4 TO PI/4
C     ALONG AXIS, 0 TO ALPHA
      T = THICK/2.0
      WTHK(1) = T*W1
      WTHK(2) = T*W2
      WTHK(3) = WTHK(1)
      ATHK(1) = T*AB1
      ATHK(2) = T*AB2
      ATHK(3) = T*AB3
C
      WALP(1) = ALPHA/2.*W1
      WALP(2) = ALPHA/2.*W2
      WALP(3) = ALPHA/2.*W3
      AALP(1) = ALPHA/2.*(1.+AB1)
      AALP(2) = ALPHA/2.*(1.+AB2)
      AALP(3) = ALPHA/2.*(1.+AB3)
C
      P4 = 3.1415927D0/4.0D0
      WCIRC(1) = P4*WT1
      WCIRC(2) = P4*WT2
      WCIRC(3) = P4*WT3
      WCIRC(4) = WCIRC(2)
      WCIRC(5) = WCIRC(1)
      ACIRC(1) = P4*BA1
      ACIRC(2) = P4*BA2
      ACIRC(3) = P4*BA3
      ACIRC(4) = P4*BA4
      ACIRC(5) = P4*BA5
C
      DO 24 I=1,2
      DO 24 J=1,2
   24 STIFF(I,J) = 0.
      DO 25 I=1,6
      DO 25 J=1,2
      CUP1(I,J) = 0.
   25 CONTINUE
      DO 27 I=1,6
      DO 27 J=1,2
   27 CUP2(I,J) = 0.
      DO 28 I=1,6
      DO 28 J=1,2
   28 CUP3(I,J) = 0.
      DO 29 I=1,6
      DO 29 J=1,2
   29 CUP4(I,J) = 0.
      DO 30 I=1,6
      DO 30 J=1,2
   30 CUP(I,J) = 0.
C     START NUMERICAL INTEGRATION LOOP
      DO 100 ITHK=1,3
      H = ATHK(ITHK)
      DO 100 ICIRC=1,5
      AL = ACIRC(ICIRC)
      DO 100 IALP=1,3
      AALPP = AALP(IALP)
C
      AL2 = AL**2
      AL3 = AL*AL2
      AL4 = AL*AL3
      AL5 = AL*AL4
C
      A(1,1) = 0.
      A(1,2) = H/PR2
      A(1,3) = 2.*H*AL/PR2
      A(1,4) = H*(3.*AL2+6.)/PR2
      A(1,5) = H*(4.*AL3+24.*AL)/PR2
      A(1,6) = H*(5.*AL4+60.*AL2)/PR2
C
      TH = AL+P4+P4
      ST = SIN(TH)
      CT = COS(TH)
      A(2,1) = -ST/BR
      A(2,2) = -(AL*ST + CT)/BR
      A(2,3) = -(AL2*ST + 2.*AL*CT)/BR
      A(2,4) = -(AL3*ST + 3.*AL2*CT)/BR
      A(2,5) = -(AL4*ST + 4.*AL3*CT)/BR
      A(2,6) = -(AL5*ST + 5.*AL4*CT)/BR
C
C
      CALL MULT (B1,A,HM1)
      CALL BTDB (BDB1,B1,D)
      CALL MULTDB (DB1,D,B1)
C
      BM1 = -(2-B)*SIN(AALPP)*ST
      BM2 = (2-B)*COS(AALPP)*ST
C
      CUP1(2,1) = BM1*DB1(2,1)
      CUP1(2,2) = BM1*DB1(2,2)
      CUP1(4,1) = BM2*DB1(2,1)
      CUP1(4,2) = BM2*DB1(2,2)
C
      TH = AL+3.*P4+P4
      ST = SIN(TH)
      CT = COS(TH)
      A(2,1) = -ST/BR
      A(2,2) = -(AL*ST + CT)/BR
      A(2,3) = -(AL2*ST + 2.*AL*CT)/BR
      A(2,4) = -(AL3*ST + 3.*AL2*CT)/BR
      A(2,5) = -(AL4*ST + 4.*AL3*CT)/BR
      A(2,6) = -(AL5*ST + 5.*AL4*CT)/BR
C
C
      CALL MULT (B2,A,HM2)
      CALL BTDB (BDB2,B2,D)
      CALL MULTDB (DB2,D,B2)
      BM1 = -(2-B)*SIN(AALPP)*ST
      BM2 = (2-B)*COS(AALPP)*ST
```

```
C
      CUP2(2,1) = BM1*DB2(2,1)
      CUP2(2,2) = BM1*DB2(2,2)
      CUP2(4,1) = BM2*DB2(2,1)
      CUP2(4,2) = BM2*DB2(2,2)
C
      TH = AL+5.*P4+P4
      ST = SIN(TH)
      CT = COS(TH)
      A(2,1) = -ST/BR
      A(2,2) = -(AL*ST + CT)/BR
      A(2,3) = -(AL2*ST + 2.*AL*CT)/BR
      A(2,4) = -(AL3*ST + 3.*AL2*CT)/BR
      A(2,5) = -(AL4*ST + 4.*AL3*CT)/BR
      A(2,6) = -(AL5*ST + 5.*AL4*CT)/BR
C
C
      CALL MULT (B3,A,HM3)
      CALL BTDB (BDB3,B3,D)
      CALL MULTDB (DB3,D,B3)
      BM1 = -(2-B)*SIN(AALPP)*ST
      BM2 = (2-B)*COS(AALPP)*ST
C
      CUP3(2,1) = BM1*DB3(2,1)
      CUP3(2,2) = BM1*DB3(2,2)
      CUP3(4,1) = BM2*DB3(2,1)
      CUP3(4,2) = BM2*DB3(2,2)
C
      TH = AL+7.*P4+P4
      ST = SIN(TH)
      CT = COS(TH)
      A(2,1) = -ST/BR
      A(2,2) = -(AL*ST + CT)/BR
      A(2,3) = -(AL2*ST + 2.*AL*CT)/BR
      A(2,4) = -(AL3*ST + 3.*AL2*CT)/BR
      A(2,5) = -(AL4*ST + 4.*AL3*CT)/BR
      A(2,6) = -(AL5*ST + 5.*AL4*CT)/BR
C
      CALL MULT (B4,A,HM4)
      CALL BTDB (BDB4,B4,D)
      CALL MULTDB (DB4,D,B4)
      BM1 = -(2-B)*SIN(AALPP)*ST
      BM2 = (2-B)*COS(AALPP)*ST
C
      CUP4(2,1) = BM1*DB4(2,1)
      CUP4(2,2) = BM1*DB4(2,2)
      CUP4(4,1) = BM2*DB4(2,1)
      CUP4(4,2) = BM2*DB4(2,2)
C
      DO 40 I=1,2
      DO 40 J=1,2
   40 BDB(I,J) = BDB1(I,J) + BDB2(I,J) + BDB3(I,J) + BDB4(I,J)
      DO 44 I=1,6
      DO 44 J=1,2
      CUP(I,J) = (CUP1(I,J)+ CUP2(I,J)+ CUP3(I,J)+ CUP4(I,J))
     1 *(PR/BR**2)
   44 CONTINUE
C
      DO 50 I=1,2
      DO 50 J=1,2
   50 STIFF(I,J) = STIFF(I,J) +
     1 BDB(I,J)*CONST*WTHK(ITHK)*WCIRC(ICIRC)*WALP(IALP)
C
      DO 60 I=1,6
      DO 60 J=1,2
      CUPMAT(I,J) = CUPMAT(I,J) +
     1 CUP(I,J)*CONST*WTHK(ITHK)*WCIRC(ICIRC)*WALP(IALP)
   60 CONTINUE
  100 CONTINUE
      RETURN
      END
```

SUBROUTINE HMAT

```
      SUBROUTINE HMAT (HM1,HM2,HM3,HM4,PR)
C
C     ROUTINE TO EVALUATE OVALISATION SHAPE FUNCTION MATRICES [H]
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DOUBLE PRECISION F(6,6),G1(6),G2(6),G3(6),G4(6),
     1 GM1(6,4),GM2(6,4),GM3(6,4),GM4(6,4),
     2 HM1(6,2),HM2(6,2),HM3(6,2),HM4(6,2),PR
C
C     F MATRIX
      F(1,1) = 0.4614468578
      F(1,2) = -0.2454369261
      F(1,3) = -0.038553314219*PR
      F(1,4) = F(1,1)
      F(1,5) = -F(1,2)
      F(1,6) = F(1,3)
      F(2,1) = -1.144574688
      F(2,2) = 7.0/16.0
      F(2,3) = 0.049087385 21*PR
      F(2,4) = -F(2,1)
      F(2,5) = F(2,2)
      F(2,6) = -F(2,3)
      F(3,1) = 1.0/8.0
      F(3,2) = .4774648293
      F(3,3) = 1./8.*PR
      F(3,4) = F(3,1)
      F(3,5) = -F(3,2)
      F(3,6) = F(3,3)
      F(4,1) = 1.130906434
      F(4,2) = -1.0132118 36
      F(4,3) = -.1591549431*PR
      F(4,4) = -F(4,1)
      F(4,5) = F(4,2)
      F(4,6) = -F(4,3)
      F(5,1) = -.1013211836
      F(5,2) = -.1290061377
      F(5,3) = F(5,1)*PR
      F(5,4) = F(5,1)
      F(5,5) = -F(5,2)
      F(5,6) = F(5,3)
      F(6,1) = -.4984044817
      F(6,2) = .4927671482
      F(6,3) = .1290061377*PR
      F(6,4) = -F(6,1)
      F(6,5) = F(6,2)
      F(6,6) = -F(6,3)
C
C     G VECTORS
      DO 10 I=1,6
      G1(I) = F(I,2)-.5*F(I,4)
      G2(I) = -.5*F(I,1)
      G3(I) = .5*F(I,4)
      G4(I) = 0.5*F(I,1)+F(I,5)
   10 CONTINUE
C
C     G MATRICES
      DO 20 I=1,6
      GM1(I,1) = G1(I)
      GM1(I,2) = G4(I)
      GM1(I,3) = G3(I)
      GM1(I,4) = G2(I)
C
      GM2(I,1) = G2(I)
      GM2(I,2) = G1(I)
      GM2(I,3) = G4(I)
      GM2(I,4) = G3(I)
C
      GM3(I,1) = G3(I)
      GM3(I,2) = G2(I)
      GM3(I,3) = G1(I)
```

```
      GM3(I,4) = G4(I)
C
      GM4(1,1) = G4(I)
      GM4(1,2) = G3(I)
      GM4(1,3) = G2(I)
      GM4(1,4) = G1(I)
   20 CONTINUE
C
      DO 23 I = 1,6
      HM1(1,1) = -GM1(1,1) - GM1(1,3)
      HM1(1,2) =  GM1(1,2) + GM1(1,4)
      HM2(1,1) = -GM2(1,1) - GM2(1,3)
      HM2(1,2) =  GM2(1,2) + GM2(1,4)
      HM3(1,1) = -GM3(1,1) - GM3(1,3)
      HM3(1,2) =  GM3(1,2) + GM3(1,4)
      HM4(1,1) = -GM4(1,1) - GM4(1,3)
      HM4(1,2) =  GM4(1,2) + GM4(1,4)
   23 CONTINUE
      RETURN
      END
```

SUBROUTINE MULT

```
      SUBROUTINE MULT (C,A,B)
C
C     ROUTINE TO EVALUATE [C] = [A] [B]
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DOUBLE PRECISION A(2,6),B(6,2),C(2,2)
C
      DO 10 I = 1,2
      DO 10 J = 1,2
      C(I,J) = 0.
      DO 10 K = 1,6
   10 C(I,J) = C(I,J) + A(I,K)*B(K,J)
      RETURN
      END
```

SUBROUTINE BTDB

```
      SUBROUTINE BTDB(BDB,B,D)
C
C     ROUTINE TO EVALUATE BTDB MATRICES
C
      DOUBLE PRECISION BDB(2,2),B(2,2),D(2,2),BB(2,2)
C
      DO 10 I = 1,2
      DO 10 J = 1,2
      BB(I,J) = 0.
      DO 10 K = 1,2
   10 BB(I,J) = BB(I,J) + D(I,K)*B(K,J)
C
      DO 20 I = 1,2
      DO 20 J = 1,2
      BDB(I,J) = 0.
      DO 20 K = 1,2
   20 BDB(I,J) = BDB(I,J) + B(K,I)*BB(K,J)
      RETURN
      END
```

SUBROUTINE MULTDB

```
      SUBROUTINE MULTDB (C,A,B)
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DOUBLE PRECISION A(2,2),B(2,2),C(2,2)
C
      DO 10 I = 1,2
      DO 10 J = 1,2
      C(I,J) = 0.
      DO 10 K = 1,2
   10 C(I,J) = C(I,J) + A(I,K)*B(K,J)
      RETURN
      END
```

SUBROUTINE CMULT

```
      SUBROUTINE CMULT (C,A,B)
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DOUBLE PRECISION A(6,2),B(2,2),C(6,2)
C
      DO 10 I = 1,6
      DO 10 J = 1,2
      C(I,J) = 0.
      DO 10 K = 1,2
   10 C(I,J) = C(I,J) + A(I,K)*B(K,J)
      RETURN
      END
C
```

A4.4.4 SUBROUTINE SR100

```
      SUBROUTINE SR100 (IELNUM,ITYP,KELOUT,ELVOL,KTIK,ZS,ZASS,ZSC)
C
C ********* STRESS PASS FOR ELEMENT PB3 ********
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      EXTERNAL TRACK,GETELD,PUTELD,SRPLT,MAXV,VZERO
      INTEGER IPLTAY(6),IEP
C
C ***** START STCOM STORAGE ********
      INTEGER  IELNUM,ITYP,KELIN(6),KELOUT(6),NR,KTIK,
     1 KEYERR,IOUT,NSTEPS,KFSTLD,ITTER,ITIME,NCUMIT,KRSTRT,ISPARE,
     2 K13,NPRPVL,MATST,K5,K16,IPROP,KCPDS,
     3 K20,KAY,MODE,ISYM,KAHD,IDEBUG,IXXX,
     4 ITYPE,MAT,IELEM,NROW,JTYPE,IPLOT,IPRINT,KTEMTP,KCONCV,KBICNV,
     5 KEYPLS,KEYCRP,KEYSWL,KYSUB(9),K21,NODES(20),  EPAR(50)
      REAL  ERRVAR(5)
C
      DOUBLE PRECISION
     1 DPZERO,DPHALF,DPONE,DPTWO,DPTEN,DTORAD,RADTOD,
     2 TREF,TUNIF,TOFSET, DELTIM,TIME,TIMOLD,TIME2,TIME3,DELT2,
     3 ACEL,OMEGA,CGOMEG,CGLOC,DXXX,
     4 ELMASS,XCENTR,YCENTR,ZCENTR,TFCP,SUBEX,  ERPAR(20),
     5 XYZEQ(20,3),X(20),Y(20),Z(20),  ELVOL
C
      COMMON /STCOM/ DPZERO,DPHALF,DPONE,DPTWO,DPTEN,DTORAD,RADTOD,
     1 TREF,TUNIF,TOFSET, DELTIM,TIME,TIMOLD,TIME2,TIME3,DELT2,
     2 ACEL(3),OMEGA(6),CGOMEG(6),CGLOC(3),       DXXX(16),
     3 KEYERR,IOUT,NSTEPS,KFSTLD,ITTER,ITIME,NCUMIT,KRSTRT,ISPARE,
     4 K13,NPRPVL,MATST,K5,K16,IPROP(20),KCPDS,
     5 K20,KAY(10),MODE,ISYM,KAHD,IDEBUG(10),     IXXX(41)
C
      EQUIVALENCE  (ITYPE,EPAR(1)), (MAT,EPAR(2)), (IELEM,EPAR(5)),
     1 (NROW,EPAR(7)), (JTYPE,EPAR(11)), (IPLOT,EPAR(12)),
     2 (IPRINT,EPAR(13)), (KTEMTP,EPAR(14)), (KCONCV,EPAR(16)),
     4 (KBICNV,EPAR(17)), (KEYPLS,EPAR(18)), (KEYCRP,EPAR(19)),
     5 (KEYSWL,EPAR(20)), (KYSUB(1),EPAR(21)), (K21,EPAR(30)),
     6 (NODES(1),EPAR(31))
C
      EQUIVALENCE  (ELMASS,ERPAR(1)), (XCENTR,ERPAR(2)),
     1 (YCENTR,ERPAR(3)), (ZCENTR,ERPAR(4)), (TFCP,ERPAR(5)),
     2 (SUBEX,ERPAR(6))
      EQUIVALENCE (X(1),XYZEQ(1,1)),(Y(1),XYZEQ(1,2)),(Z(1),XYZEQ(1,3))
C
      DOUBLE PRECISION
     1 ZS(KTIK,KTIK),ZASS(KTIK,KTIK),ZSC(KTIK),
     2 BRAD,ALPHA,PRAD,THICK,PHI,EX,ALPX,NUXY,DENS,DPPI,PROP(4),
     3 RVR(5),SVR(253),PRESS(1),B1(6,6),B2(6,6),
```

```
     4 TR(12,12),U(24),POSTD(20),CON,A1(6),BI(2,8),DMATI(2,2),
     5 SMI(2,8),DOFI(8),EPI(2),SIGI(2),ULOC(12),A(2,6),BBI(2,2),
     6 DOFO(8),SMO(2,8),DMATO(3,3),A2(3,6),BO(3,8),BOB(3,6),
     7 EPO(3),SIGO(3),SIG(3),BN(2,6),BOVI(2,3),BOVO(2,3),
     8 HM1(6,2),HM2(6,2),HM3(6,2),HM4(6,4),SFACT
C
      EQUIVALENCE (RVR(1),BRAD), (RVR(2),ALPHA), (RVR(3),PRAD),
     1 (RVR(4),THICK), (RVR(5),PHI)
C
C     EQUIVALENCE SAVED VARIABLES SVR():
      EQUIVALENCE (SVR(1),B1(1,1)),(SVR(37),B2(1,1)),
     1  (SVR(73),PROP(1)), (SVR(77),TR(1,1)),
     2 (SVR(221),SMI(1,1)), (SVR(237),SMO(1,1))
C
C     EQUIVALENCE MATERIAL PROPERTIES PROP():
      EQUIVALENCE  (PROP(1),EX), (PROP(2),ALPX), (PROP(3),NUXY),
     1 (PROP(4),DENS)
C
      DATA DPPI / 3.14159265358979 3D0 /
C
      CALL TRACK (5,'SR100 ')
      CALL GETELD (IELNUM,ITYP,EPAR(1),ERPAR(1),CON,CON,
     1 PRESS(1),CON,RVR(1),SVR(1),XYZEQ(1,1),U(1))
C
C     CONVERT ALPHA TO RADIANS
      ALPHA = ALPHA*DPPI/180.0
C
C ~~~~~~~~~~~~~~~~~ STRESS PASS ~~~~~~~~~~~~~~~~~~
C
C     EVALUATE I/P CONSTITUTIVE MATRIX [DMATI]
      CALL VZERO (DMATI(1,1),4)
      DCON = EX/(1.0-NUXY**2)
      DMATI(1,1) = DCON
      DMATI(1,2) = DCON*NUXY
      DMATI(2,1) = DMATI(1,2)
      DMATI(2,2) = DCON
C
C     EVALUATE O/P CONSTITUTIVE MATRIX
      CALL VZERO (DMATO(1,1),9)
      DMATO(1,1) = DCON
      DMATO(1,2) = DCON*NUXY
      DMATO(2,1) = DMATO(1,2)
      DMATO(2,2) = DCON
      DMATO(3,3) = DCON*(1-NUXY)/2
C
C     EVALUATE LOCAL  DISPLACEMENT VECTOR ULOC (IN LOCAL CSYS)
C     FROM GLOBAL DISP. VECTOR U.
      CALL MAXV (TR(1,1),U(1),ULOC,12,12)
C
C     ******* RECOVER NODELESS DOF. ********
C     ---- IN-PLANE DOF ----
C     SMI CONTAINS LAST 2 ROWS OF I/P ZSFULL AFTER CONDENSATION
C
C     DOFI =  I/P DOF IN LOCAL CSYS.
C     THE  KNOWN (BEAM) DOF STORED IN FIRST 6 ROWS.
C     NODELESS DOF RECOVERED AND STORED IN LAST 2.
      CALL VZERO (DOFI(1),8)
      DOFI(1) = ULOC(1)
      DOFI(2) = ULOC(3)
      DOFI(3) = ULOC(5)
      DOFI(4) = ULOC(7)
      DOFI(5) = ULOC(9)
      DOFI(6) = ULOC(11)
C
C     RECOVERY ALGORITHM: REF COOK
      DO 30 J=1,2
      JJ = 6+J
      DUM = 0.0
      K = JJ - 1
      DO 20 L=1,K
   20 DUM = DUM + SMI(J,L)*DOFI(L)
   30 DOFI(JJ) = (0.0 - DUM)/SMI(J,JJ)
C
C     OUT OF PLANE DEGREES OF FREEDOM
C
C     SMO CONTAINS LAST 6 ROWS OF O/P ZSFULL AFTER CONDENSATION
C     DOFO =  O/P DOF IN LOCAL CSYS.
C     THE  KNOWN (BEAM) DOF STORED IN FIRST 6 ROWS.
C     NODELESS DOF RECOVERED AND STORED IN LAST 2.
      DOFO(1) = ULOC(2)
      DOFO(2) = ULOC(4)
      DOFO(3) = ULOC(6)
      DOFO(4) = ULOC(8)
      DOFO(5) = ULOC(10)
      DOFO(6) = ULOC(12)
C
C     RECOVERY ALGORITHM
      DO 50 J=1,2
      JJ = 6+J
      DUM = 0.0
      K = JJ - 1
      DO 40 L=1,K
   40 DUM = DUM + SMO(J,L)*DOFO(L)
   50 DOFO(JJ) = (0.0 - DUM)/SMO(J,JJ)
C
C     ALL NODAL AND NODELESS DOF NOW EVALUATED.
C
C     <<<<<<< SET UP LOOP FOR STRESS AND STRAIN EVALUATION >>>>>>>
C
C     FOR DEVELOPMENT ONLY STRESSES AT NODE I WERE EVALUATED: PHI=0
C
      RATIO = (PRAD/BRAD)**2
      PR2 = PRAD**2
      D = RATIO/2.0
      B = (2*(1+NUXY))/(2+NUXY)
      OPCON = PRAD/(BRAD**2)
      P4 = 3.1415927/4.
C
C     AN INTERACTIVE SURFACE AND STRESS NORMALISATION FACTOR
C     INPUT WAS INCLUDED.
C
C     CHOOSE H = INNER/MID/OUTER SURFACE
C           = -THICK/2.0,THICK/2
      PRINT *,'-------------------------'
      PRINT *,'INPUT SURFACE FOR STRESS CALCS'
      PRINT *,'-1 -- INNER'
      PRINT *,'0 -- MIDDLE'
      PRINT *,'1 -- OUTER'
      READ (5,*) SURF
      PRINT *,'INPUT STRESS NORMALISATION FACTOR'
      READ (5,*) SFACT
C
      WRITE (40,8) IELNUM
C     REMOVE BELOW C IF I/P OUTPUT REQD. IN FILE PB2RES.DAT
C     WRITE (40,*) '   ALPHA   SIGPTH   SIGPHI   POSN'
      POSITN = -5.
      POSN = 40.
C     STRESSES EVALUATED AT NODE I IN STEPS OF 5 DEGREES
      DO 1000 IQRT =1,4
      AL = -50.
      DO 100 IALP = 1,18
C
      AL = AL+5.
      AL = AL*DPPI/180.
      POSITN = POSITN + 5
C
C     ********* IN PLANE STRESS AND STRAIN ***********
C     EVALUATE I/P [B] MATRIX
      AL2 = AL**2
      AL3 = AL*AL2
      AL4 = AL*AL3
      AL5 = AL*AL4
C
      H = -THICK/2*SURF
```

```
      A(1,1) = 0.
      A(1,2) = H/PR2
      A(1,3) = 2.*H*AL/PR2
      A(1,4) = H*(3.*AL2+6.)/PR2
      A(1,5) = H*(4.*AL3+24.*AL)/PR2
      A(1,6) = H*(5.*AL4+60.*AL2)/PR2
C
      IF (IQRT.EQ.1) TH = AL+P4
      IF (IQRT.EQ.2) TH = AL+3.*P4
      IF (IQRT.EQ.3) TH = AL+5.*P4
      IF (IQRT.EQ.4) TH = AL+7.*P4
C
      ST = SIN(TH)
      CT = COS(TH)
      A(2,1) = -ST/BRAD
      A(2,2) = -(AL*ST + CT)/BRAD
      A(2,3) = -(AL2*ST + 2.*AL*CT)/BRAD
      A(2,4) = -(AL3*ST + 3.*AL2*CT)/BRAD
      A(2,5) = -(AL4*ST + 4.*AL3*CT)/BRAD
      A(2,6) = -(AL5*ST + 5.*AL4*CT)/BRAD
C
      CALL VZERO (HM1(1,1),12)
      CALL VZERO (HM2(1,1),12)
      CALL VZERO (HM3(1,1),12)
      CALL VZERO (HM4(1,1),12)
      CALL HMAT (HM1,HM2,HM3,HM4,PR)
C
      IF (IQRT.EQ.1) GO TO 52
      IF (IQRT.EQ.2) GO TO 54
      IF (IQRT.EQ.3) GO TO 56
      IF (IQRT.EQ.4) GO TO 58
C
 52   CALL MULT (BBI,A,HM1)
      GO TO 59
 54   CALL MULT (BBI,A,HM2)
      GO TO 59
 56   CALL MULT (BBI,A,HM3)
      GO TO 59
 58   CALL MULT (BBI,A,HM4)
      GO TO 59
C
 59   CONTINUE
      DO 60 I=1,2
      DO 60 J=1,8
 60   BI(I,J) = 0.0
C
C     LOCATION AT PHI = 0,  SIN PHI = 0, COS PHI = 1
C     BEAM BENDING STRAIN DISP IN FIRST 6 COLUMNS.
      CALL VZERO (A1(1),6)
      CONST = PRAD*CT/BRAD
      A1(1) = -CONST/BRAD
C     A1(3) = 2.*(D+CONST)*SIN(PHI)/BRAD = 0 FOR PHI = 0
      A1(5) = -2.*(D+CONST)/BRAD
C                          *COS(PHI) FOR PHI NOT 0
      CALL VZERO (BI(1,1),16)
      DO 70 I=1,6
      BI(2,I) = A1(1)*BI(1,I)+A1(3)*BI(3,I)+A1(5)*BI(5,I)
 70   CONTINUE
C
      BI(1,7) = BBI(1,1)
      BI(1,8) = BBI(1,2)
      BI(2,7) = BBI(2,1)
      BI(2,8) = BBI(2,2)
C
C     EVALUATE STRAINS {EPI} = [BI] {DOFI} = {EPI(PHI)  EPI(THETA)}
      CALL MAXV (BI(1,1),DOFI(1),EPI,2,8)
C
C     EVALUATE STRESSES {SIGI} = [D] {EPI} = {SIGI(PHI)  SIGI(THETA)}
      CALL MAXV (DMATI(1,1),EPI(1),SIGI,2,2)
C     EVALUATE STRESS FACTOR = SIGI/(MR/I)
      SIGI(1) = SIGI(1)*SFACT
      SIGI(2) = SIGI(2)*SFACT
C
      AL = AL*180.0/DPPI
C
C     REMOVE BELOW C IF I/P STRESS OUTPUT TO PB3RES.DAT REQD
C     WRITE (40,'(4F16.3)') AL,SIGI(1),SIGI(2),POSITN
 100  CONTINUE
C
C
C     ******* OUT OF PLANE STRESS AND STRAIN ********
C
C
      WRITE (40,*) ' POSN   SIGTH     SIGPHI    TAU'
      AL = -50.
      DO 200 IALP = 1,18
C
      AL = AL+5.
      AL = AL*DPPI/180.
      POSN = POSN + 5
C
      AL2 = AL**2
      AL3 = AL*AL2
      AL4 = AL*AL3
      AL5 = AL*AL4
C
      CALL VZERO (A(1,1),12)
      H = -THICK/2*SURF
      A(1,1) = 0.
      A(1,2) = H/PR2
      A(1,3) = 2.*H*AL/PR2
      A(1,4) = H*(3.*AL2+6.)/PR2
      A(1,5) = H*(4.*AL3+24.*AL)/PR2
      A(1,6) = H*(5.*AL4+60.*AL2)/PR2
C
      IF (IQRT.EQ.1) TH = AL+P4+P4
      IF (IQRT.EQ.2) TH = AL+3.*P4+P4
      IF (IQRT.EQ.3) TH = AL+5.*P4+P4
      IF (IQRT.EQ.4) TH = AL+7.*P4+P4
C
      ST = SIN(TH)
      CT = COS(TH)
      A(2,1) = -ST/BRAD
      A(2,2) = -(AL*ST + CT)/BRAD
      A(2,3) = -(AL2*ST + 2.*AL*CT)/BRAD
      A(2,4) = -(AL3*ST + 3.*AL2*CT)/BRAD
      A(2,5) = -(AL4*ST + 4.*AL3*CT)/BRAD
      A(2,6) = -(AL5*ST + 5.*AL4*CT)/BRAD
C
      CALL VZERO (HM1(1,1),12)
      CALL VZERO (HM2(1,1),12)
      CALL VZERO (HM3(1,1),12)
      CALL VZERO (HM4(1,1),12)
      CALL HMAT (HM1,HM2,HM3,HM4,PR)
C
      CALL VZERO (BBI(1,1),4)
      IF (IQRT.EQ.1) GO TO 152
      IF (IQRT.EQ.2) GO TO 154
      IF (IQRT.EQ.3) GO TO 156
      IF (IQRT.EQ.4) GO TO 158
C
 152  CALL MULT (BBI,A,HM1)
      GO TO 159
 154  CALL MULT (BBI,A,HM2)
      GO TO 159
 156  CALL MULT (BBI,A,HM3)
      GO TO 159
 158  CALL MULT (BBI,A,HM4)
      GO TO 159
C
 159  CONTINUE
C
C     PUT OV TERMS IN LAST 2 COLS OF BO
C
```

```
      CALL VZERO (BO(1,1),24)
      BO(1,7) = BBI(1,1)
      BO(1,8) = BBI(1,2)
      BO(2,7) = BBI(2,1)
      BO(2,8) = BBI(2,2)
C
C     EVALUATE O/P B MATRIX BO
C     O/P A MATRIX A2. MULT BY PRAD/(BRAD**2) USUALLY OUTSIDE BRACKETS
      CALL VZERO (A2(1,1),18)
C
C     AT NODE I:
      CP = 1.0
      SP = 0.0
C
      A2(1,2) = -(2.0-B)*SP*ST*OPCON
      A2(1,4) = (2.0-B)*CP*ST*OPCON
      A2(3,2) = B*CP*OPCON
      A2(3,4) = B*SP*OPCON
      A2(3,6) = OPCON
C
      CALL VZERO (BOB(1,1),18)
      CALL MAXB (A2(1,1),B2(1,1),BOB,3,6,3,3,6,6)
C     PUT BOB IN FIRST 6 COLUMNS OF BO
C     NOTE THAT FIRST TWO ROWS ARE SWAPPED ROUND FOR CONSISTANCY WITH
C     OVALISATION STRAIN-DISP MATRIX
      DO 160 J = 1,6
      BO(1,J) = BOB(2,J)
      BO(2,J) = BOB(1,J)
      BO(3,J) = BOB(3,J)
  160 CONTINUE
C
C     EVALUATE STRAINS {EPO} = [BI] {DOFO} = {EPO(PHI)  EPO(THETA) TAU}
      CALL MAXV (BO(1,1),DOFO(1),EPO,3,8)
C
C     EVALUATE STRESSES {SIGO} = [D] {EPI} = {SIGO(PHI) SIGO(THETA)  GAMMA}
      CALL MAXV (DMATO(1,1),EPO(1),SIGO,3,3)
C     EVALUATE STRESS FACTOR = SIGI/(MR/I)
      SIGO(1) = SIGO(1)*SFACT
      SIGO(2) = SIGO(2)*SFACT
      SIGO(3) = SIGO(3)*SFACT
C
C
      AL = AL*180.0/DPPI
      WRITE (40,'(4F16.3)') POSN,SIGO(1),SIGO(2),SIGO(3)
  200 CONTINUE
 1000 CONTINUE
C
C     DUMMY WRITE TO FILE12.  ALL STRESSES ACTUALLY WRITTEN TO
C     PB3RES.DAT
C
      IF (IPLOT .NE. 1) GO TO 900
C           ***** NUMBER OF FORCES (LEVEL 1) *****
      IPLTAY(2) = 12
C           ***** NUMBER OF STRESSES (LEVEL 2) *****
      IPLTAY(3) = 4
C           ***** NUMBER OF TOTAL SAVED (LEVELS 1, 2, AND 3)
      IPLTAY(4) = 20
C           ***** SAVE GEOMETRY FOR CONTOURS (0,NO  1,YES)
      IPLTAY(6) = 0
C
C     ****** PUT POSTDATA INFORMATION INTO POSTD ******
C
C     ***** PUT PLTARY INFORMATION ONTO FILE 12 *******
      CALL SRPLT (IELEM,ITYP,NROW,MAT,100,2,U(1),NODES(1),XYZEQ(1,1),
     1 IPLTAY(1),POSTD(1))
  900 CALL PUTELD (IELNUM,EPAR(1),ERPAR(1),CON,SVR(1))
C     PUTELD RESTORES DATA BACK TO FILE3
      CALL TRACK( 15,'SR100 ')
      RETURN
      END
```