Université d'Ottawa · University of Ottawa

ELASTO-PLASTIC FINITE ELEMENT FOR PIPELINES

By

Farhood Nowzartash

A thesis

Presented to the University of Ottawa in partial fulfillment of the requirements for

Master of Applied Science in Civil Engineering

Department of Civil Engineering

University of Ottawa

Ottawa, Canada

K1N 6N5

September 2002

The M.A.Sc. in Civil Engineering is a joint program

with Carleton University administered by the

Ottawa-Carleton Institute for Civil Engineering

0-612-76618-7

This thesis is dedicated to my wife, Haleh.

Acknowledgement

Special appreciation goes to Dr. Magdi Mohareb for his advice, directions and encouragement throughout this research project.

The author would like to express his sincere gratitude to Bahai National Spiritual Assembly of Canada for their financial and moral support. He would like also to thank Ottawa Bahai Local Spiritual Assembly and BIHE project committee for their kind hospitality during his stay in Canada to do his Master's degree at the University of Ottawa.

Deepest thanks go to my wife Haleh for her precious moral support, patience and tireless encouragement during my study.

The patience and sacrifice of my mother, father and sisters throughout my studies can never be forgotten. Thank you all.

Abstract

An efficient pipe element for the modeling of the inelastic behaviour of three-dimensional pipes is presented. The formulation is based on a two-node element with twelve degrees of freedom. The element consists of an elastic portion and two potentially plastic generalized 3D hinges located at both nodes. The formulation is based on a lumped plasticity approach. The behaviour of plastic hinges is characterized using recently developed interaction relations for pipe sections. The interaction relations are exact and include the effects of axial force, bi-axial bending moments, bi-axial shear, torsion and internal and/or external pressure. The element models shear deformation effects both in the elastic and plastic ranges. Therefore, it is suitable for predicting the behaviour of pipe segments subject to high shear forces.

The analysis is implemented using the displacement control scheme in order to capture the peak point of the load deformation response without numerical difficulty. The normality condition concept in conjunction with the yield hyper-surface at stress resultant level is used to approximately simulate the material nonlinearity effects. The developed technique models material nonlinearity in a very efficient way when compared to full 3D or shell finite element analysis. The model is thus particularly suitable for long pipeline systems. Solutions for simple problems are provided and compared to several other well-established elements in the ABAQUS™ library in order to assess the validity of the results and demonstrate their scope of applicability. A discussion of additional features that can be added to the analysis is also presented.

Table of Contents

Acknowledgement II

Abstract III

Table of Contents IV

List of Figures VII

Notations IX

Chapter 1

Introduction 1

1.1 General 1

1.2 Review on Interaction Relations 2

1.3 Review on Lumped Plasticity Models 4

Chapter 2

Plastic Deformations and Yield Hyper-Surface of Thin Pipes 5

2.1 General 5

2.2 Stress vs. Strain Relationship for Steel 6

2.3 Ingredients of Plasticity 7

2.3.1 Stress Yield Surface 8

2.3.2 Strain Decomposition Assumption 8

2.3.3 Plasticity Flow Rule Associated with Stress Resultants 9

2.4 Interaction Relations 10

2.4.1 Assumptions 11

2.4.2 Interaction Equations 11

2.4.3 Limits of Applicability 14

2.5 Extending Plasticity Flow Rules to Stress Resultants 15

2.5.1 The Normal to the Yield Surface 16

Chapter 3

Derivation of Stiffness Matrices 18

3.1 General 18

3.2 Types of Analysis 18

3.3 Ingredients of Non-linear Analysis 21

3.4 Elasto-Plastic Stiffness Element 22

3.5 Elastic Stiffness Matrix 24

3.6 Plastic Reduction Matrix 28

3.6.1 Loading and Unloading Behaviour 28

3.6.2 Plastic Reduction Matrices 29

3.6.2.1 Both Nodes are Plastified 29

3.6.2.2 Only the First Node is Plastified 33

3.6.2.3 Only the Second Node is Plastified 35

3.6.2.4 Both Nodes are Elastic 36

3.6.3 Tangent Stiffness Matrix 36

3.7 Coordinate Transformation 36

Chapter 4

Solution of Equilibrium Equations 40

4.1 General 40

4.2 Equilibrium Path and Response Diagrams 40

4.3 Iterative Procedures 41

4.3.1 Terminology 41

4.3.2 Control Schemes 42

4.4 Load Control Method Scheme 43

4.4.1 Procedure 43

4.4.2 Drifting From the Yield Surface 47

4.4.2.1 Correction of Drift Crossing the Yield Surface 47

4.4.2.2 Correction of Drift Along the Tangent Plane 49

4.4.3 Convergence Criteria 51

4.5 Displacement Control Scheme 52

Chapter 5

Verification Problems 59

5.1 General 59

5.2 Finite Elements in ABAQUS 59
5.2.1 B33 Beam Element 60
5.2.2 PIPE Cross-Section 60
5.2.3 PIPE31 Beam Element 61
5.2.4 FRAME3D Element 62
5.2.5 ELBOW31 Element 63
5.3 Verification Problems 63
5.3.1 Problem 1: Bending Moments 64
5.3.2 Problem 2: Axial Force and Bending Moments 70
5.3.3 Problem 3: Torsion and Bending Moments 73
5.3.4 Problem 4: Shear and Bending Moments 75
5.3.5 Problem 5: Internal Pressure and Bending Moments 78
5.3.6 Problem 6: A 3D Pipeline Problem 80

Chapter 6

Conclusions and Recommendations 82

References 85

Appendix A1

Derivatives of the Yield Surface Equations 90

Appendix A2

Displacement Control vs. Load Control 95

Appendix B 98

Program Listings and Programmer's Guide 98
B.1 General 98
B.2 Main Form 99
B.3 Sample Input File 109
B.4 Modules Code List 110
B.5 Classes Code List 122
B.6 Input Files 136
B.7 Output File for Problem 1 139

List of Figures

Fig. 2.1 A Piece of Pipe under general loading 5
Fig. 2.2 Stress vs. Strain Curve for Structural Steel 6
Fig. 2.3 Stress vs. Strain Curve for Pipe Steel 7
Fig. 2.4 Idealized Elastic-perfectly Plastic Material 7
Fig. 2.5 Strain Decomposition Assumption 9
Fig. 2.6 Normality criterion and Plasticity Flow Rule in 2D 9
Fig. 2.7 In-plane Axes 10
Fig. 2.8 Interaction Surface between Uniaxial Moments, Axial Force and Pressure 13
Fig. 2.9 Interaction Surface Uniaxial Shear force, Uniaxial Moments and Torsion 14
Fig. 3.1 Types of Analysis 20
Fig. 3.2 In-plane Deformation of a Pipe Section 21
Fig. 3.3 Actual Shape of a Plastic Hinge (Longitudinal section) 21
Fig. 3.4 Elastic-Plastic Elements 22
Fig. 3.5 Coordinate Systems and Element Nodal Quantities 23
Fig. 3.6 Elements Connection to a Node 23
Fig. 3.7 Active DOF of a 2D Cantilever Beam 24
Fig. 3.8 An Arbitrary Stress Status of an Element Node 28
Fig. 3.9 An Arbitrary Stress Status of a Plastified Element Node 31
Fig. 3.10 Global and Local Coordinate Systems 37
Fig. 4.1 Response Diagram and Equilibrium Path 41
Fig. 4.2 Categorized breakdown in nonlinear solution methods 42
Fig. 4.3 Incremental Control Techniques and their Limitations 43
Fig. 4.4 Material Non-Linear Analysis by Load Control Method 44
Fig. 4.5 Plastic Hinge Formation at a Fraction of the Load Increment 48
Fig. 4.6 Secant Root Finding Method 49
Fig. 4.7 Force-Point Drift from Yield Surface and Radial Correction 50
Fig. 4.8 Radial correction in a single step and multiple steps 50
Fig. 4.9 Displacement Control Scheme 56

Fig. 5.1 Default Integration Points in ABAQUS for a 3D Pipe Cross-Section 61
Fig. 5.2 Sketch of Problem 1 64
Fig. 5.3 Problem 1: No plastification 65
Fig. 5.4 Problem 1: Node 3 Plastified 65
Fig. 5.5 Problem 1: Nodes 2 and 3 plastified 66
Fig. 5.6 Comparison of Results with ABAQUS B33 for Problem 1 66
Fig. 5.7 Comparison of Results with ABAQUS PIPE31 for Problem 1 67
Fig. 5.8 Comparison of Results with ABAQUS FRAME3D for Problem 1 68
Fig. 5.9 Comparison of Results with ABAQUS ELBOW31 for Problem 1 69
Fig. 5.10 Sketch of Problem 2 70
Fig. 5.11 Comparison of Results with ABAQUS for Problem 2 71
Fig. 5.12 ABAQUS FRAME3D and Exact Yield Surfaces for a DN90 STD Section 72
Fig. 5.13 Sketch of Problem 3 73
Fig. 5.14 Problem 3: No Plastification 73
Fig. 5.15 Comparison of Results with ABAQUS for Problem 3 74
Fig. 5.16 Sketch of Problem 4 75
Fig. 5.17 Problem 4: No Plastification 75
Fig. 5.18 Problem 4: Node 3 is Plastified 76
Fig. 5.19 Problem 4: Node 2 and 3 are Plastified 77
Fig. 5.20 Comparison of Results with ABAQUS Elements for Problem 4 78
Fig. 5.21 Sketch of Problem 5 78
Fig. 5.22 Comparison of Results with ABAQUS Elements for Problem 5 79
Fig. 5.23 Pipes under Various Pressure Ratios for a Pipe Section DN90 STD 80
Fig. 5.24 Sketch of Problem 6 81
Fig. 5.25 Comparison of results with ABAQUS FRAME3D for problem 6 81

Notations

The following symbols are used in this thesis

1, 2, 3	local coordinate system axes, 1 is the longitudinal axis, 2 and 3 are two perpendicular axes in the section plane
α,β	representative stress resultant components
$\{\Delta\}$	displacements vector for the whole structure
$\{\Delta\}_{12\times1}$	element nodal displacement vector
$\Delta_{1_i},\Delta_{2_i},\Delta_{3_i}$	displacements along axes 1,2 or 3 of node i
$\Delta_{1_j},\Delta_{2_j},\Delta_{3_j}$	displacements along axes 1,2 or 3 of node j
$\{\Delta_i\}_{3\times1},\{\Delta_j\}_{3\times1}$	nodal displacements at node i and j, respectively
$\{\Delta_G\}_{12\times1}$	element nodal displacement vector in global coordinate system
$\{\Delta_n^i\}$	nodal displacement at *n*th increment and *i*th iteration
Δ_{ref}	the largest member of displacement vector
δ	prescribed incremental displacement
ε	strain
ε^e	elastic strain
ε_{ij}	strain tensor (i, j = 1,2,3)
ε^p	plastic strain
ρ	fraction of load increment that crosses the yield surface
ρ_l	lower bound estimate of ρ in secant root-finding method
ρ_r	better estimate of ρ in secant root-finding method
ρ_u	upper bound estimation of ρ in secant root-finding method
σ	stress
σ_{ij}	stress tensor (i,j = 1,2,3)

σ_u	ultimate stress of steel material
σ_y	yield stress of steel material
τ	circumferential shear stress
τ_r	ratio of circumferential shear stress to circumferential stress resistance in the absence of other stresses
Φ	yield (hyper) surface
η_n	load vector magnitude at *n*th increment
ν	Poisson's ratio
κ	bending to shear rigidity ratio (Eq. 3.3)
π	Pi number (equal to 3.1415926)
$\theta_{1i}, \theta_{2i}, \theta_{3i}$	rotation about axes 1,2 or 3 of node i
$\theta_{1j}, \theta_{2j}, \theta_{3j}$	rotation about axes 1,2 or 3 of node j
$\{\theta_i\}_{3\times1}, \{\theta_j\}_{3\times1}$	nodal rotations at nodes i and j, respectively
ξ	user specified allowable tolerance for convergence (Sec. 4.4.3)
$\psi(\sigma_{ij})$	yield function (function of stress tensor components σ_{ij})
A	pipe cross-section area
A_p	axial force resistance of the pipe in the absence of other stresses
A_r	reduced area for shear ($2(1+\nu)/(4+3\nu)A$ for thin walled pipes)
D	pipe outside diameter
$d\varepsilon^e$	incremental elastic strain
$d\varepsilon^p$	incremental plastic strain
$d\lambda$	scalar determining the plastic deformation magnitude
$d\lambda_i$, $d\lambda_j$	scalars determining the plastic deformation magnitudes for nodes i and j, respectively
$\{d\Delta_n^i\}$	incremental nodal displacement at *n*th increment and *i*th iteration
$\{d\Delta_i\}_{6\times1}, \{d\Delta_j\}_{6\times1}$	incremental displacement vector of node i and j, respectively
$\{d\Delta_i^e\}_{6\times1}, \{d\Delta_j^e\}_{6\times1}$	incremental elastic displacement vector of node i and j, respectively

$\{d\Delta_i^p\}_{6\times1}, \{d\Delta_j^p\}_{6\times1}$ incremental plastic displacement vector of node i and j, respectively

$\{d\overline{\Delta}_{n1}^i\}$ nodal displacements due to the applied load at *n*th increment and *i*th iteration

$\{d\overline{\overline{\Delta}}_{n1}^i\}$ nodal displacements due to the unbalanced load at *n*th increment and *i*th iteration

$d\eta_n^i$ incremental load factor at *n*th increment and *i*th iteration

$\{dF_i\}_{6\times1}, \{dF_j\}_{6\times1}$ incremental force vector for nodes i and j, respectively

$\{dP_n\}$ incremental applied load at *n*th increment

du_k^p associated generalized incremental plastic strain

E modulus of elasticity

e convergence error

$\{F\}$ nodal force vector for the whole structure

$\{F\}_{12\times1}$ element nodal force vector

F_1 axial force

F_{1i}, F_{2i}, F_{3i} nodal forces along axes 1,2 or 3 of node i, respectively

F_{1j}, F_{2j}, F_{3j} nodal forces along axes 1,2 or 3 of node j, respectively

F_2 shear force along local axis 2

F_3 shear force along local axis 3

$\{F_i\}_{3\times1}, \{F_j\}_{3\times1}$ nodal forces at node i and j, respectively

$\{F_G\}_{12\times1}$ element nodal force vector in global coordinate system

$\{F_n^i\}$ nodal force vector at *n*th increment and *i*th iteration

F_{r1} ratio of F_1 to A_p

F_{r2} ratio of F_2 to V_p

F_{r3} ratio of F_3 to V_p

G shear modulus

$[G]_{12\times2}$ matrix of yield surface derivatives for element

$\{G_i\}_{6\times1}, \{G_j\}_{6\times1}$ yield surface derivatives vectors for nodes i and j, respectively

I	moment of inertia of pipe section
I_2	moment of inertia about local axis 2
I_3	moment of inertia about local axis 3
i_1, i_2, i_3	unit vectors based on local coordinate system
i_X, i_Y, i_Z	unit vectors based on global coordinate system
J	St-Venant torsional rigidity constant
$[K]$	stiffness matrix for the whole structure
$[K]_{12\times12}$	element (tangent) stiffness matrix
$[K_n^i]$	structure stiffness matrix at *n*th increment and *i*th iteration
$[K^e]_{12\times12}$	element elastic stiffness matrix
$[K_{ii}^e]_{6\times6}$	upper left portion of elastic stiffness matrix
$[K_{ij}^e]_{6\times6}$	upper right portion of elastic stiffness matrix
$[K_{jj}^e]_{6\times6}$	lower right portion of elastic stiffness matrix
$[K_G]_{12\times12}$	element (tangent) stiffness matrix in the global coordinate system
$[K^m]_{12\times12}$	element plastic reduction matrix
L	pipe element length
M_1	twisting moment
$M_{1_i}, M_{2_i}, M_{3_i}$	nodal moments about axes 1,2 or 3 of node i, respectively
$M_{1_j}, M_{2_j}, M_{3_j}$	nodal moments about axes 1,2 or 3 of node j, respectively
M_2	bending moments about axis 2
M_3	bending moments about axis 3
$\{M_i\}_{3\times1}, \{M_j\}_{3\times1}$	nodal moments at node i and j, respectively
M_p	bending moment resistance of pipe in the absence of other stresses
M_{r1}	ratio of M_1 to T_p
M_{r2}	ratio of M_2 to M_p
M_{r3}	ratio of M_3 to M_p
N	total number of unknowns in displacement (or load) vector

$\{P\}$	total applied load
$\{P_n\}$	applied load at *n*th increment
p	internal and external pressure difference (+ for internal pressure)
p_p	pressure resistance in absence of other stresses
p_r	ratio of p to p_p
$\{R_n^i\}$	unbalanced load at *n*th increment and *i*th iteration
R_{ref}	the largest member of unbalanced load vector
r	pipe radius at the mid-surface of the section
r_{1X}, r_{1Y}, r_{1Z}	direction cosines of element local 1-axis with respect to the global axes
r_{2X}, r_{2Y}, r_{2Z}	direction cosines of element local 2-axis with respect to the global axes
r_{3X}, r_{3Y}, r_{3Z}	direction cosines of element local 3-axis with respect to the global axes
$[T]_{12\times12}$	element coordinate transformation matrix
t	pipe thickness
T_p	twisting moments resistance of pipe in the absence of other stresses
V_p	shear resistance of pipe in the absence of other stresses
X, Y, Z	global coordinate system axes
X_k	nodal forces (k =1 to 6 refers to F_1, F_2, F_3, M_1, M_2, M_3, respectively)
x_k	stress resultants (k =1 to 7 corresponds to F_1, F_2, F_3, M_1, M_2, M_3 and p, respectively)

Chapter 1

Introduction

1.1 General

In the most general case, elevated pipes are subject to combinations of seven stress resultants. These are biaxial shears, biaxial moments, twisting moments, axial force and internal and/or external pressure. These seven internal forces are induced by vertical, transverse, longitudinal and pressure loads. Sources of vertical loads include the pipe self-weight, the weight of the fluids they convey and the weight of permanent equipment such as valves. Sources of lateral loads include wind and earthquake. Thermal effects, friction and Poisson's ratio effect are potential sources of axial loads. Furthermore, pipes are subject to internal pressure caused by the action of fluid they convey and possibly external pressure in the case of submerged pipes. In order to prevent excessive axial forces due to thermal effects, it is common to introduce pipe loops at regular intervals into the pipeline. Although these loops provide flexible axial links between pipe segments and reduce the axial forces significantly, they introduce twisting moments in the loop arms. This observation suggests the necessity of including the twisting moment in the analysis. A stocky pipe section will fully plastify under combinations of all seven stress-resultants. In principle, a relationship between all seven variables would define a yield surface in a seven dimensional hyperspace.

Steel pipes have traditionally been designed using elastic analysis. The need for reliable and efficient non-linear analysis methods in the design of pipelines was recognized in the latest Canadian standards (CAN/CSA – Z66.2 1996). In addition to the conventional elastic analysis methodology and related design philosophy, appendix C of the standard introduces the use of limit state design and allows the designer to take advantage of the additional capacity by deforming pipes in the plastic range. The most rigorous type of analysis would be based on full elasto-plastic 3D finite elements or shell elements with geometric nonlinearity and typically involves thousands of degrees of freedom to model short pipe segments. Despite of the accuracy of this approach, in most practical cases involving longer pipelines, it is impractical to conduct such an analysis. The objective behind the developments in this research is to partially fill the gap between simple but unrealistic elastic analysis and expensive but highly complex full three-dimensional finite element analysis. This is done by adopting an efficient elasto-plastic pipe finite element that captures the essential aspects of stocky pipe behaviour while preserving the simplicity of the model.

An early work by Porter and Powell (1971) provides a good footprint to follow. The methodology adopted in this research is similar to what they introduced with two major differences: a) including shear effects in the formulation, both in the elastic and plastic ranges; and b) adopting an exact yield surface for pipelines rather than a faceted approximation to the yield surface.

A two-node twelve-degree-of-freedom element is developed to model pipe element behaviour. The formation of a zero length generalized plastic zones at the ends of the elements approximately simulates the material non-linearity effect. The normality criterion governs the plastic hinge rotations. A displacement control iterative scheme is implemented. Thus, the model can capture the limit point without difficulty and the element provides an excellent base for adding effects of geometric nonlinear in future studies.

1.2 Review on Interaction Relations

An early work by Gerald and Becker (1957) led to a number of interaction relationships for pipes under several combinations of internal forces. They have limited their experimental and analytical work to pipes that undergo local buckling prior attaining their plastic resistance. Morris and Fenves (1969) introduced lower bound plastic interaction expressions for a number of sections including thin walled circular cross-section under combination of axial force, biaxial moment and torsion. An experimental study on pipe sections subject to combined axial force and biaxial bending moments by

Pillai and Ellis (1971) led to proposing a simplified interaction relation. Later, lower and upper bound interaction relations for tubular sections under axial force and biaxial moments were formulated by Chen and Atsuta (1972). They showed the exactness of these relations through the agreement between the two bounds. Their solution agreed with that suggested by Morris and Fenves (1969) when the torque term is omitted. Bouwkamp and Stephen (1973) conducted a series of tests on 48in. pipes subject to combinations of internal pressure, axial force and bending moments. The flexural behaviour of the pipe sections beyond the ultimate capacity was experimentally investigated by Sherman (1976). Post-buckling behaviour of tubular sections under pure torsion was studied on fibreglass pipes by Yamaki (1976). A series of tests were conducted on large fabricated tubes subjected to combination of axial load and bending (Prion and Birkemoe 1988). Obaia et. al. (1991) experimentally investigated the inelastic shear behaviour of large diameter pipes. They suggested a bending-shear capacity interaction relation based on regression analysis of their test results. Interaction equations including internal pressure, axial force and bending moment were developed by Mohareb and Murray (1999). They showed the agreement of these equations with experimental results for pipes of diameter to thickness ratio in the range of 50 and 64. The interaction relations developed were adopted in limit state design of submarine pipelines by DNV (1996). A lower bound general plastic interaction equation for pipe sections under axial load, biaxial shear, torsion, biaxial moment and internal or external pressure was developed (Mohareb 2002) by assuming statically admissible stress fields. Mohareb (2001), using Lagrange multipliers showed that the interaction relations obtained maximize the lower bound solution. Another treatment developed an upper bound solution by using kinematically admissible strain fields to be published by Mohareb (2003). The two solutions coincide demonstrating the exactness of the solution. A series of tests was conducted on steel pipes under combination of torsion, bending moment and shear forces (Ozkan 2002 and Ozkan and Mohareb 2002) demonstrated the validity of the interaction equations.

1.3 Review on Lumped Plasticity Models

Glenn et. al. (1970) presented a general procedure for tracing the load-displacement behaviour of frameworks loaded beyond the elastic range up to the ultimate load. They extended the concept of classical yield surface to stress resultants to model the plastic behaviour under combination of flexural and torsional moments and axial force as lumped plastic hinges. Porter and Powell (1971) derived a two-node elastic-plastic pipe member with two concentrated potentially plastic hinges at the end nodes. The members are assumed to yield as defined by a four-dimensional yield surface including axial force, biaxial bending moments and torsion. Internal or external pressure was assumed constant throughout the pipe. They approximated the exact yield surface by a faceted yield surface. They adopted the normality criterion and associative plasticity flow rule to predict plastic behaviour of joints. Chen and Atsuta (1976) have shown that the location of a plastic hinge at the exact maximum moment point in a frame member is not required for an accurate estimation of the member inelastic behaviour. They showed that as long as the exact location of the plastic hinge in a member is not more than one sixth of the element length away from the assumed position, the error introduced is not more than five percent. Their observation leads to a general acceptance of formulation based on two lumped plastic hinges at the ends. Orbison et al. (1982) investigated the inelastic behaviour of American wide-flange sections under the combination of axial force and biaxial bending moments by using an approximate single-equation yield surface and a lumped plasticity concept. Powell and Chen (1986) introduced a generalized plastic 3D hinges for members subject to axial load, torsion and bending moments using a flexibility approach. Later Attalla et al. (1994) incorporated into the conventional lumped plasticity approach, a zone of gradually developing plasticity. McGuire et al. (2000) provide a comprehensive summary of lumped plastic hinge formulations.

Chapter 2

Plastic Deformations and Yield Hyper-Surface of Thin Pipes

2.1 General

In general, a pipe section is under the combination of seven possible stress resultants. These are bi-axial shear, axial force, bi-axial bending moments, axial force, torsion and the difference between internal and external pressure (Fig. 2.1). An interaction equation provides a relationship between these components. By applying concepts of associative plasticity and the normality criterion the behaviour of 3D plastic hinges can be characterized.

In this chapter, the general interaction equation for thin pipes is described and the assumptions behind the formulas are discussed. Toward this goal, an overview of plasticity rules is presented, and these concepts are extended to stress resultants.

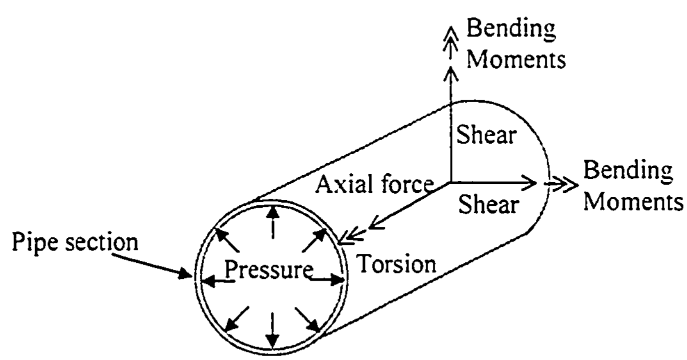

Fig. 2.1 A Piece of Pipe under General Loading

2.2 Stress vs. Strain Relationship for Steel

A typical stress vs. strain (σ-ε) relationship for structural steel from a uniaxial tension test is shown in Fig. 2.2, in which σ_y and σ_u are the yield stress and the ultimate stress, respectively. As it can be observed, structural steel shows a linear behaviour in the elastic region. In this region, the stress, σ is proportional to the strain, ε; i.e., $\sigma = E\varepsilon$ where E is the modulus of elasticity. An abrupt stiffness loss after the yield point is noticeable. In the plastic region, steel undergoes additional strain under constant stress. Structural steel usually experiences an increase in stress past the yield plateau. This phenomenon is called strain hardening. Strain hardening leads to the ultimate stress σ_u. After the ultimate stress, the engineering stress ($P/A_{un\text{-}deformed}$) gradually decreases as necking of the specimen takes place up to the rupture point. It is of interest to note that steel undergoes a permanent plastic deformation after the yield point. This is schematically presented by the unloading path in Fig. 2.2.

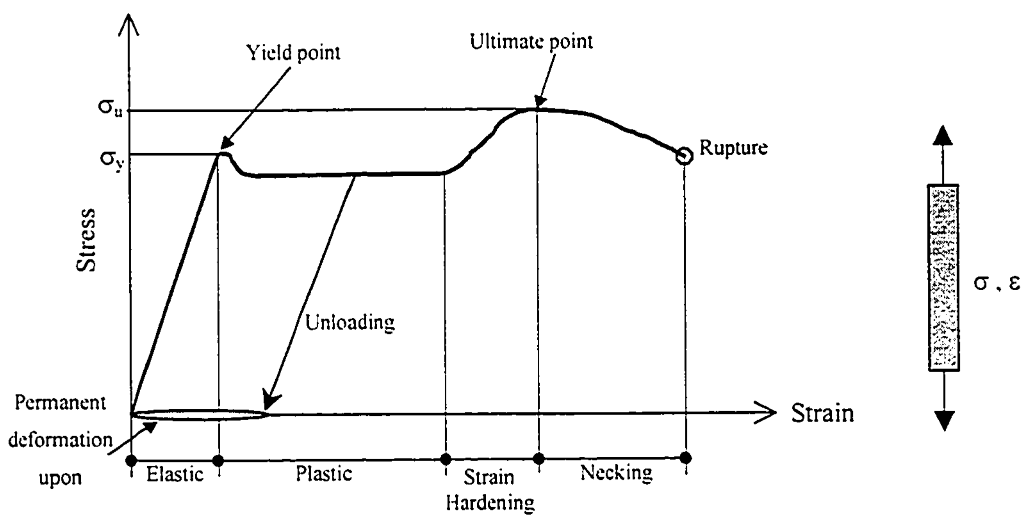

Fig. 2.2 Stress vs. Strain Curve for Structural Steel

In contrast, for pipe steel, there is no distinct yield plateau as the material yields gradually (Fig. 2.3). The yield stress is usually taken as the stress corresponding to half a percent strain.

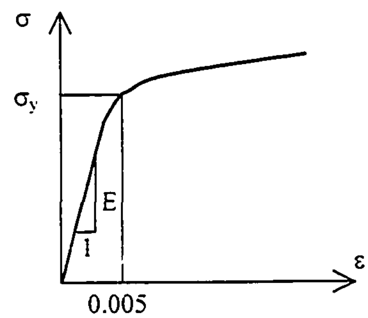

Fig. 2.3 Stress vs. Strain Curve for Pipe Steel

In this research, pipe steel is assumed elastic-perfectly plastic (Fig. 2.4). This means that steel is elastic before the yield point (ε^e, σ_y) and perfectly plastic after this point. In other words, strain hardening is neglected. The total strain has a plastic component ε^p and an elastic component ε^e. The elastic component of the strain is always reversible both before and after the yield point.

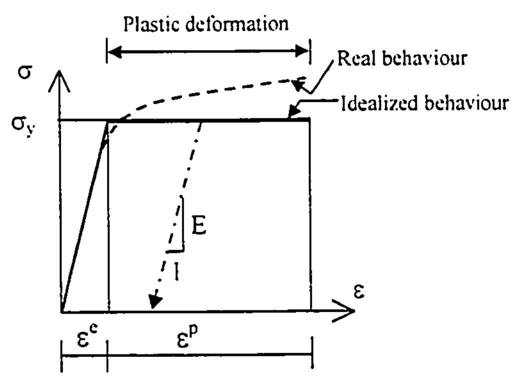

Fig. 2.4 Idealized Elastic-perfectly Plastic Material

2.3 *Ingredients of Plasticity*

Various plasticity concepts should be introduced in describing plastic behaviour. Firstly, it is necessary to determine under which conditions a given stress point is plastified. This condition is described by the yield surface introduced in section 2.3.1. Secondly, it is needed to quantify the strains of point that is known to undergo plastification. The plasticity flow rule, which is derived from normality condition, is one of the most common constitutive laws and is described in section 2.3.3.

2.3.1 Stress Yield Surface

It is assumed that there is a potential scalar function $\Psi(\sigma_{ij})$ of the components of the stress tensor σ_{ij} such that Ψ is negative when the material at that point is in the elastic range of deformation, zero when the material is yielded and a positive value is unattainable under the limitation of the formulation.

The yield condition, $\Psi(\sigma_{ij})=0$, defines a surface called the yield surface, which is independent from the particular choice of coordinate system for isotropic material such as steel. The yield condition commonly used to describe steel behaviour is the maximum distortional energy density yield criterion (Boresi and Sidebottom 1985), which is known as von Mises criterion. This criterion is

$$\frac{1}{2}\left[(\sigma_{11}-\sigma_{22})^2+(\sigma_{22}-\sigma_{33})^2+(\sigma_{11}-\sigma_{33})^2\right]+3(\sigma_{12}^2+\sigma_{23}^2+\sigma_{13}^2)-\sigma_y^2=0 \tag{2.1}$$

By applying this criterion, an interaction equation at the stress resultant level for pipes was derived (Mohareb 2002). The interaction relationship will be discussed in section 2.4.

2.3.2 Strain Decomposition Assumption

In general, an increment of strain can be assumed to consist of the sum of a recoverable incremental elastic strain, $d\varepsilon^e$ and irrecoverable incremental plastic strain, $d\varepsilon^p$ (Fig. 2.5).

$$d\varepsilon_{ij}=d\varepsilon_{ij}^e+d\varepsilon_{ij}^p \tag{2.2}$$

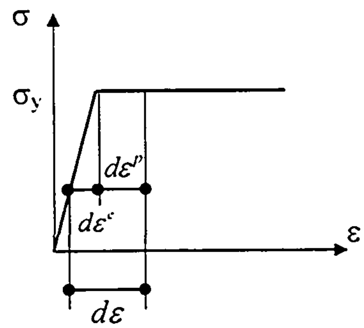

Fig. 2.5 Strain Decomposition Assumption

The strain decomposition assumption commonly used at the strain level is assumed valid for the generalized deformations.

2.3.3 Plasticity Flow Rule Associated with Stress Resultants

The flow rule relates plastic strain increments to current total stresses. The most commonly accepted plasticity flow rule is derived from the normality criterion to a plastic potential scalar function, Ψ^1. For associative plasticity, the scalar potential Ψ^1 is equal to $\Psi(\sigma_{ij})$. The Normality criterion states that, if plastic flow occurs at a point, the components of the incremental plastic strain $d\varepsilon_{ij}^p$ must be normal to the yield surface at that point. This criterion for a planar case is schematically presented in Fig. 2.6, in which α and β are two stress components (e.g., σ_{11} and σ_{22} for a biaxial state of stress).

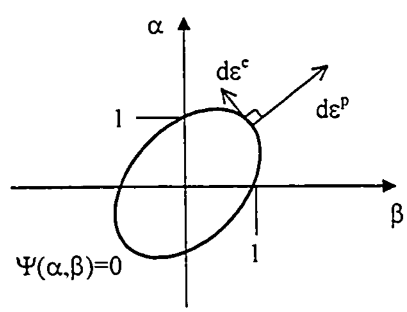

Fig. 2.6 Normality criterion and Plasticity Flow Rule in 2D

Mathematically, the associative plasticity constitutive law can be expressed as

$$d\varepsilon_{ij}^{p} = d\lambda \frac{\partial \Psi(\sigma_{ij})}{\partial \sigma_{ij}} \tag{2.3}$$

In which $d\lambda$ is a scalar representing a measure of the plastic deformation magnitude. It is indefinite if plastic flow is not constrained and a function of the resistance offered by the surrounding body when constrained.

2.4 Interaction Relations

A single-equation yield surface relating the stress resultants i.e., a function Φ in the form of $\Phi(F_1, F_2, F_3, M_1, M_2, M_3, p) = 0$ is sought. The condition $\Phi < 0$ is met for any physically possible combination of internal forces; $\Phi = 0$ corresponds to a fully plastic state of the cross section and the condition $\Phi > 0$ is unattainable under the assumptions of the formulation. This concept is analogous to the stress yield surface presented in section 2.3.1. Force F_1 is the axial force, F_2 is the shear along axis 2 (Fig. 2.7), F_3 is the shear along axis 3, M_1 is the torsion, M_2 is the bending moments about axis 2, M_3 is the bending moments about axis 3-3 and p is the internal pressure (positive) or external pressure on the pipe.

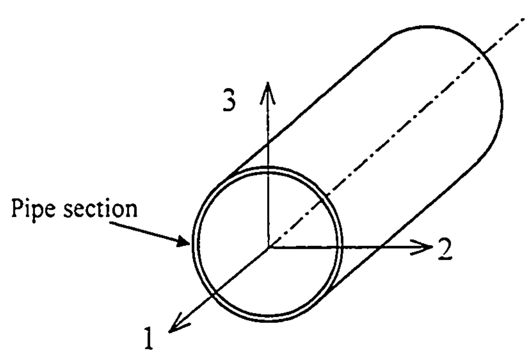

Fig. 2.7 In-plane Axes

2.4.1 Assumptions

In deriving the yield function Φ adopted in this research, the following assumptions are made (Mohareb 2002):

1) Ovalization of cross-section under bending stresses is neglected. This assumption is justifiable for pipes with low diameter to thickness ratios (e.g., $D/t < 25$).
2) A bilinear elastic-perfectly plastic approximation is adopted (Fig. 2.4). The first line passes through origin and has the same slope as the initial slope of a tension coupon tests. The second line with zero slope passes through the yield point. The additional capacity due to strain hardening is neglected. This leads to a lower bound prediction of the section plastic resistance.
3) Pipe steel is assumed to yield according to the maximum distortional energy density yield criterion (Eq. 2.1).
4) The formulation is exclusive to thin walls pipe sections (e.g., $D/t > 10$) (i.e., The variation of stress and strain along thickness is neglected.)
5) Pipe section attains its fully plastic capacity prior undergoing local buckling. This assumption is justifiable for class 1 and 2 of the steel code. It has been experimentally verified for pipes under axial force, bending moments and pressure (Mohareb and Murray 1999) with diameter to thickness ratio, D/t of 51 and 64 and pipes under bending moments, twist and shear (Ozkan and Mohareb 2002) for pipes with $D/t < 41$.

2.4.2 Interaction Equations

It has been shown (Mohareb 2002) that a single interaction equation does not exist. Instead, the exact relationships have an intrinsic form

$$\Phi_1(F_2, F_3, M_1, \tau) = \left(\frac{F_{r2}}{\tau_r}\right)^2 + \left(\frac{F_{r3}}{\tau_r}\right)^2 - \cos^2\left(\frac{\pi}{2}\frac{M_{r1}}{\tau_r}\right) = 0 \tag{2.4}$$

$$\Phi_2(F_1, M_2, M_3, p, \tau) = M_{r2}^2 + M_{r3}^2 - \left(1 - \frac{3}{4}p_r^2 - \tau_r^2\right)\cos^2\left[\frac{\frac{\pi}{2}\left(F_{r1} - \frac{1}{2}p_r\right)}{\sqrt{1 - \frac{3}{4}p_r^2 - \tau_r^2}}\right] = 0 \qquad (2.5)$$

in which,

$F_{r1} = F_1 / A_p$ is ratio of axial force to the axial resistance, (+) for tension.

$F_{r2} = F_2 / V_p$ is ratio of shear along axis 2 to the shear plastic resistance.

$F_{r3} = F_3 / V_p$ is ratio of shear along axis 3 to the shear resistance.

$M_{r1} = M_1 / T_p$ is ratio of torsion to the torsion resistance.

$M_{r2} = M_2 / M_p$ is ratio of moment about axis 2 to the moment resistance.

$M_{r3} = M_3 / M_p$ is ratio of moment about axis 3 to the moment resistance.

$p_r = p / p_p$ is ratio of pressure difference to the resistance, (+) for internal pressure.

$\tau_r = \tau / (\sigma_y / \sqrt{3})$ is ratio of circumferential shear to the circumferential resistance.

τ is the circumferential shear.

$M_p = 4r^2 t \sigma_y$ is moment resistance in absence of other stresses.

$V_p = 4rt(\sigma_y / \sqrt{3})$ is shear resistance in absence of other stresses.

$T_p = 2\pi r^2 t(\sigma_y / \sqrt{3})$ is torsion resistance in absence of other stresses.

$A_p = 2\pi r t \sigma_y$ is axial resistance in absence of other stresses.

$p_p = t\sigma_y / (r - t/2)$ is pressure resistance in absence of other stresses.

t is pipe section thickness, and

r is pipe section radius at mid-surface of the pipe.

It is observed that the derived equations are independent of the angle between resultant moment, $\sqrt{M_2^2 + M_3^2}$ and resultant shear, $\sqrt{F_2^2 + F_3^2}$. For a given general internal force combination of the yield hyper surface, the circumferential stress ratio τ_r must be identical when calculated either from Eq. 2.4 or Eq. 2.5.

The interaction relationship is seven dimensional and cannot be visualized in its general form. Special cases including only three variables are possible to illustrate. The interactions between the uniaxial moments, the axial force and the differential pressure (internal minus external pressure) is shown in Fig. 2.8. This three-dimensional surface is obtained by setting $F_{r2} = F_{r3} = M_{r1} = M_{r3} = 0$ in Eqs. 2.4 and 2.5. Another particular case is presented in Fig. 2.9. This interaction surface relates uniaxial moment, uniaxial shear and torsion and it is obtained by setting $F_{r1} = F_{r3} = M_{r3} = p_r = 0$ in Eqs. 2.4 and 2.5.

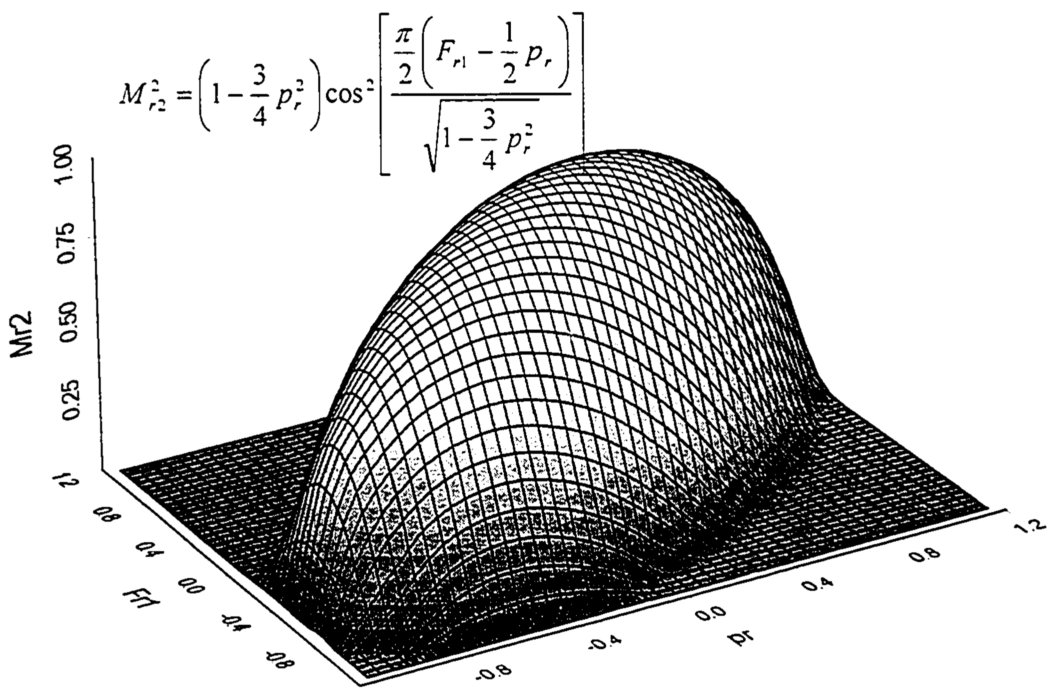

Fig.2.8 Interaction Surface between Uniaxial Moments, Axial Force and Pressure

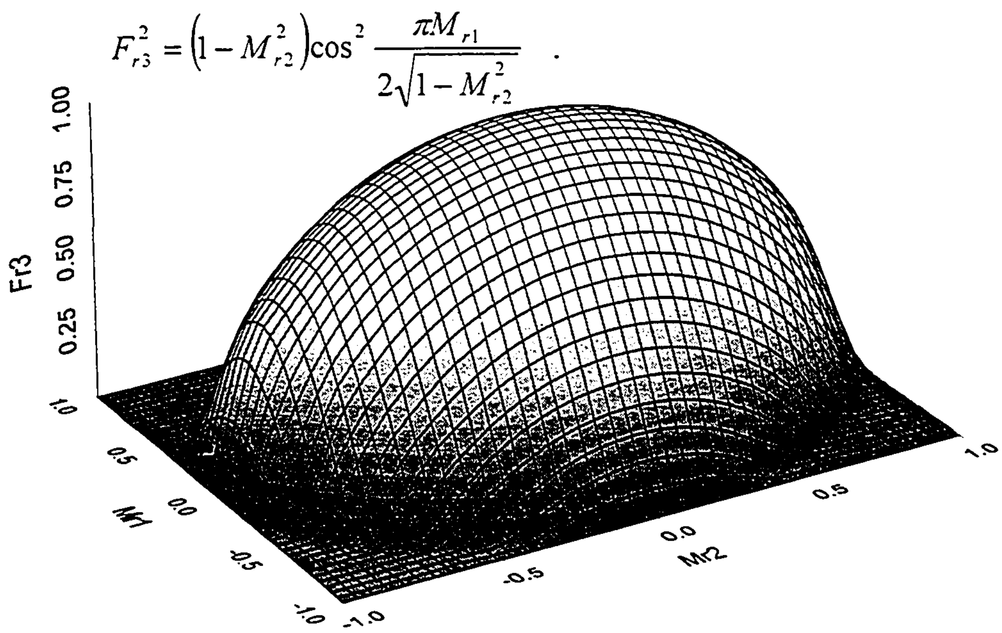

Fig. 2.9 Interaction Surface Uniaxial Shear force, Uniaxial Moments and Torsion

2.4.3 Limits of Applicability

It is important to ensure that the interaction relations are not applied outside their limits of validity. An upper bound for the domain of shearing force and twisting moment can be found by setting $\tau_r = 1$ in Eq. 2.4 giving

$$F_{r2}^2 + F_{r3}^2 \leq \cos^2(\pi M_{r1}/2) \tag{2.6}$$

The square of cosine function must be a positive value less than unity. Therefore, Eq. 2.6 leads to

$$F_{r2}^2 + F_{r3}^2 \leq \tau_r^2 \tag{2.7}$$

The bounds for pressure are obtained by setting $M_{r2}^2 + M_{r3}^2 = 0$ in Eq. 2.5 leading to

$$-2\sqrt{\frac{1-\tau_r^2}{3}} \le p_r \le 2\sqrt{\frac{1-\tau_r^2}{3}} \tag{2.8}$$

In addition, the following bounds for the axial force ratios have been obtained based on geometric constraints (Mohareb 2002)

$$\frac{1}{2}p_r - \sqrt{1-\frac{3}{4}p_r^2 - \tau_r^2} \le F_{r1} \le \frac{1}{2}p_r + \sqrt{1-\frac{3}{4}p_r^2 - \tau_r^2} \tag{2.9}$$

It should be noted that only when bounds given in Eqs. 2.6 through 2.9 are met for a given force combination, can Eqs. 2.4 and 2.5 be used to calculate the stress resultants of the section.

2.5 Extending Plasticity Flow Rules to Stress Resultants

It is required to relate the incremental generalized displacements at a plastic hinge to the stress resultants. A proper relationship may be derived by extending the plasticity flow rule to stress resultants. Using this concept Chen and Atsuta (1972) assumed that changes in the plastic generalized hinge displacements are proportional to the derivatives of the plastic potential yield surface with respect to the corresponded stress resultants. Orbison et al. (1982) proposed a single continuous yield function as a plastic potential function and utilized similar concept for relating 3D hinge forces and deformations. In both solutions, the developments were based on approximate yield surfaces as a plastic potential function. In general, the associated generalized incremental plastic strain du_k^p, corresponding to the stress resultant x_k, k = 1 to 7, was related to the generalized stress resultants through the normality condition

$$du_k^p = d\lambda \frac{\partial \Phi}{\partial x_k} \tag{2.10}$$

In Eq. 2.10. subscript k refers to stress resultant and associated generalized plastic strain caused by axial force, shear force along axis 2, shear force along axis 3, torsion, bending moments about axis 2, bending moments about axis 3, and differential pressure, respectively. In this equation, the term $\partial\Phi/\partial x_k$ is the normal to the yield surface at the point under consideration.

The above formulation is valid when a number of conditions are met. Firstly, a general yield function relating the stress resultants has to exist. Eqs. 2.4 and 2.5 satisfy this condition. Secondly, The hyper-surface has to meet Drucker's convexity condition, i.e., $\sum x_k \cdot \partial\Phi/\partial x_k > 0$. This condition is also satisfied because the derived yield surface was shown to be exact (Mohareb 2002). Thirdly, it is desirable that a single definition of the yield surface exists. Although the latter condition is not necessary, it reduces the computational efforts when compared to solutions based on piecewise potential surfaces (Orbison et.al 1982). A witness for difficulties caused by having a multiple definition for yield surface is the work by Porter and Powell (1971). This third condition is also met in the solution presented here.

2.5.1 The Normal to the Yield Surface

The yield surface was expressed as an intrinsic two-equation form. This causes some complexity in finding an expression for the normal to the yield hyper-surface. This problem can be resolved by the following method. Equations 2.4 and 2.5 are respectively, written in the following symbolic form:

$$\Phi_1(F_2, F_3, M_1, \tau) = 0 \tag{2.11}$$

$$\Phi_2(F_1, M_2, M_3, p, \tau) = 0 \tag{2.12}$$

According to the normality rule, the incremental plastic deformations have to be normal to the yield surface and orthogonal to the stress increment. Thus, the dot product of incremental plastic strains du_k^p and the generalized stress resultant increment dx_k must be zero. By taking the total differential of Eqs 2.11 and 2.12, one obtains

$$d\Phi_1 = \frac{\partial\Phi_1}{\partial F_2}dF_2 + \frac{\partial\Phi_1}{\partial F_3}dF_3 + \frac{\partial\Phi_1}{\partial M_1}dM_1 + \frac{\partial\Phi_1}{\partial\tau}d\tau = 0 \tag{2.13}$$

$$d\Phi_2 = \frac{\partial\Phi_2}{\partial F_1}dF_1 + \frac{\partial\Phi_2}{\partial M_2}dM_2 + \frac{\partial\Phi_2}{\partial M_3}dM_3 + \frac{\partial\Phi_2}{\partial p}dp + \frac{\partial\Phi_2}{\partial\tau}d\tau = 0 \tag{2.14}$$

The right hand side of Eqs. 2.13 and 2.14 vanish since the stress resultant point has to remain on the yield surface.

From Eq. 2.14, by solving for $d\tau$ and substituting into Eq. 2.13 results into an equation of the form of $\{G\}^T \cdot \{dx\} = 0$, in which

$$\{G\}^T = \left\langle -\frac{(\partial\Phi_1/\partial\tau)(\partial\Phi_2/\partial F_1)}{\partial\Phi_2/\partial\tau}, \frac{\partial\Phi_1}{\partial F_2}, \frac{\partial\Phi_1}{\partial F_3}, \frac{\partial\Phi_1}{\partial M_1}, -\frac{(\partial\Phi_1/\partial\tau)(\partial\Phi_2/\partial M_2)}{\partial\Phi_2/\partial\tau}, -\frac{(\partial\Phi_1/\partial\tau)(\partial\Phi_2/\partial M_3)}{\partial\Phi_2/\partial\tau} \right\rangle \tag{2.15}$$

Because of problems with orthogonalilty between the axial force and the internal pressure, which causes that pressure cannot be treated as a separate DOF and the complexity of analysis, it was decided to hold the pressure constant along the pipeline. However, the yield surface equation is still a function of pressure. This assumption will affect derivations in chapter 3.

Vector $\{G\}$ is orthogonal to any incremental vector $\{dx\}$ located on the yield hyper-surface at the point of interest $\{x\}$ and must be normal to the hyper-surface at that point. Therefore, Eq. 2.15 provides the expression for the normal to the yield hyper-surface. Appendix A1 provides the expression for the normal to the yield surface, an example of the computation of the derivatives and the normal to the yield surface.

Chapter 3

Derivation of Stiffness Matrices

3.1 General

The developments prescribed in this chapter are based on the stiffness method, which relates the nodal displacements of the structure to the externally applied loads. For a linear analysis, the derivations are well established. Emphasis thus will be on incorporating the plastic effects.

A two-node twelve-degree-of-freedom element is formulated to model member behaviour. Plasticity effects are incorporated into the formulation by the inclusion of potentially plastic zones at the ends of members. Plastic reduction matrices that characterize the behaviour of an element with one or two plastic hinges are derived.

The difference between various types of non-linear analysis and their limitations are discussed. Subsequently, the ingredients of the formulation including element descriptions, elastic and plastic stiffness matrices are presented.

3.2 Types of Analysis

Four different types of analysis are possible. These are: first-order elastic, second-order elastic, first-order inelastic and second-order inelastic. Each of the methods includes some assumptions that introduce approximations and limitations in the results.

A first-order elastic analysis is based on the two simplifying assumptions; a) material behaviour is linear elastic; b) deformations are small enough so that their effect on equilibrium equations is negligible. In a more realistic and elaborate representation of the structure, either of these rather restrictive assumptions can be relaxed, resulting in a non-linear formulation.

In a second-order elastic analysis, the structural material is assumed linearly elastic while the deformations are large and their effect is incorporated into the equilibrium equations.

Materials such as pipe steel will typically exhibit initially linear elastic response, followed by a plastic response. Therefore, it is possible to incorporate inelastic effects into the analysis to properly model the material behaviour beyond the elastic limit, while assuming small deformations (i.e., deformation effect on the equilibrium equation is negligible.). This type of analysis is called a first-order inelastic analysis and models material plasticity.

A more general type of analysis is the second-order inelastic analysis, in which both assumptions made in first-order elastic analysis are relaxed. In this type of analysis, not only the changes in material properties are taken into consideration but also the effects of large deformations are considered.

Analysis outcomes based on various analysis representations for a simple cantilever beam-column is schematically shown in Fig. 3.1. In the figure, a representative load magnitude P is plotted against the lateral displacement Δ at the cantilever tip.

In first-order elastic analysis (curve 1), the applied load P is linearly proportional to the response Δ. This type of analysis is the simplest to conduct and is performed without iteration. In general, it can accurately predict the initial part of the structural response in which the structure behaves more or less in a linear manner. First-order elastic analysis is in general useful under service load levels.

The second curve depicts a second-order elastic analysis. Due to the presence of the axial force P, the member exhibits a softer response than the first curve. The softening behaviour of the member is accounted for in the analysis through a geometric non-linear matrix.

In a first-order inelastic analysis (curve 3), the effect of material non-linearity is accounted for through the introduction of a plastic reduction matrix. Unlike second-order elastic analysis, a first-order inelastic analysis predicts a failure load for the structure when a mechanism takes place. This type of analysis is helpful to find the ultimate capacity of a structure based on cross-sectional strength. The developments presented in this thesis lie in this category, although extensions to include geometric non-linearity are possible.

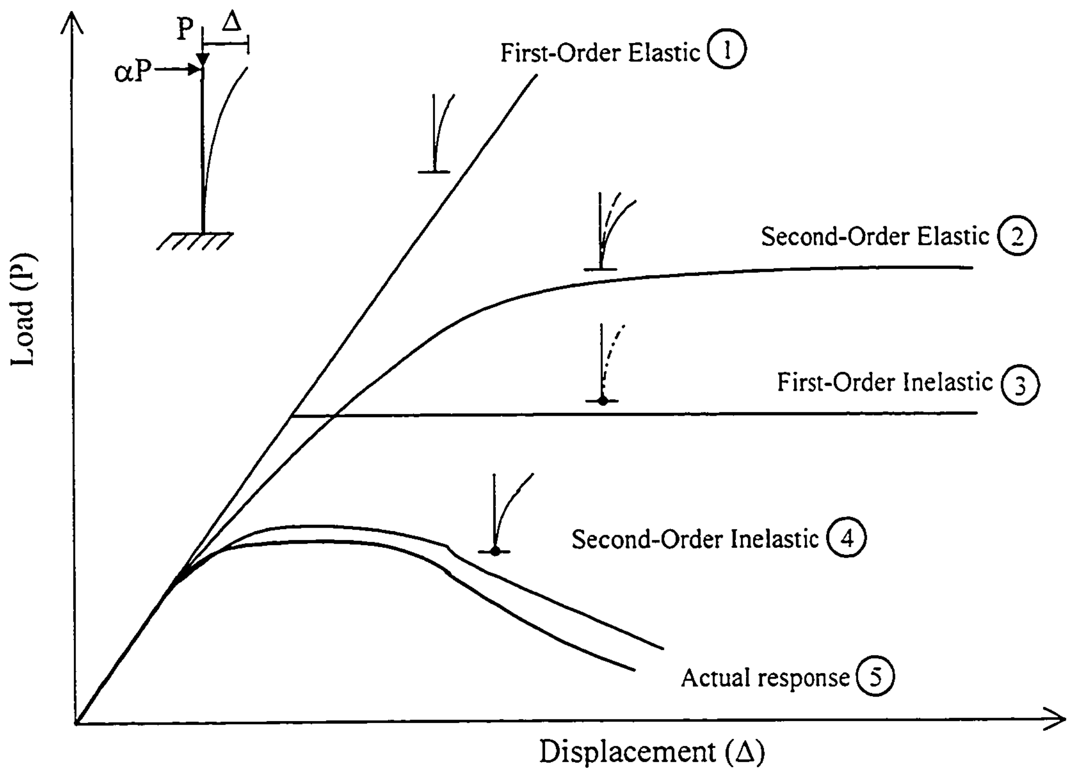

Fig. 3.1 Types of Analysis

Second-order inelastic analysis (curve 4) is more realistic than the previous analysis types and is able to capture non-linear effects due to both material and geometric effects. The method cannot represent the actual response under very large displacement, which would need a three-dimensional geometrically non-linear finite element analysis based on shell elements. There are two major sources for this limitation: a) cross-sectional distortion (Fig. 3.2) and b) the spread of plasticity (Fig. 3.3).

As shown in Fig. 3.1, second-order inelastic analysis based on the assumption that pipe cross-section remains perfectly circular after deformation, includes some approximation and will not capture the exact structural behaviour in general. In the analysis, it is assumed that each pipe member is uni-dimensional. Therefore, possible cross-sectional distortions (Fig. 3.2) cannot be captured. Cross sectional distortion induces changes to the internal strain energy, which are negligible in the pre-peak range of deformation for thicker pipes. In the post-peak range of deformation, cross-section distortions can be significant and are captured only by a 3D analysis (i.e. shell elements, 3D brick element and etc.).

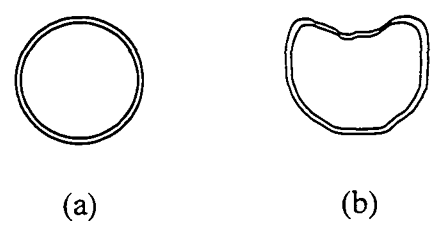

Fig. 3.2 In-plane Deformation of a Pipe Section
(a) Initial Shape; (b) Deformed Shape

In the method prescribed here, the plasticity effects are assumed to be concentrated in lumped plastic hinges at the two nodes of the elements. In contrast, a real plastic hinge is not concentrated at a point (Fig. 3.3) and has a finite length, which is neglected in the derivation, under the lumped plasticity (Sec. 3.4). Bruneau et. al (1998) showed that the difference between the assumption based on lumped plastic model and that based on spread of plasticity is negligible in most practical cases.

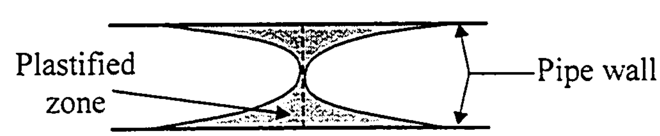

Fig. 3.3 Actual Shape of a Plastic Hinge (Longitudinal section)

3.3 *Ingredients of Non-linear Analysis*

Several ingredients are involved in a non-linear analysis. These will be presented in this chapter and chapter 4. These are

1) A suitable element definition (Section 3.4)
2) Nodal displacements and loads (Section 3.4)

3) Element elastic stiffness matrix $[K^e]$ (Section 3.5)
4) Element plastic reduction matrix $[K^m]$ (Section 3.6)
5) Coordinates transformation (Section 3.7)
6) Assembly of the structure stiffness matrix, load vector of the elements to build the stiffness matrix and load vector of the whole structure.
7) Strategies to trace equilibrium path (Chapter 4)

3.4 Elasto-Plastic Stiffness Element

The element derived consists of an elastic member with two potentially plastic hinges of zero length at each end (Fig. 3.4). This approximates real plastic hinges of finite length. Two systems of coordinates are introduced (Fig. 3.5a, b), local 1-2-3 and global X-Y-Z coordinate systems. The nodal displacements in the local coordinate system (Fig. 3.5c), corresponding to internal forces (Fig. 3.5d) and sign convention are shown. The first and second nodes of an element are denoted as i and j, respectively.

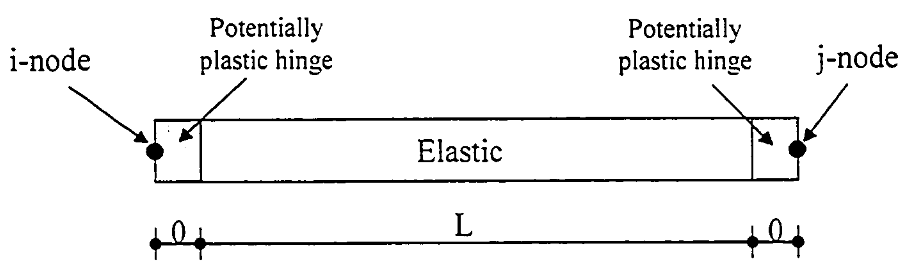

Fig. 3.4 Elastic-Plastic Elements

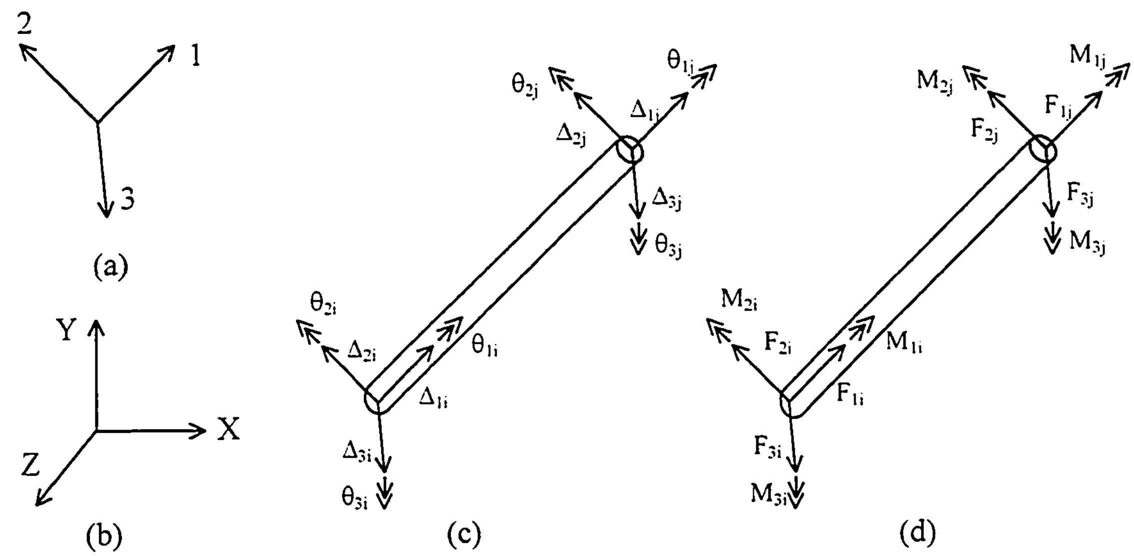

Fig. 3.5 Coordinate Systems and Element Nodal Quantities

(a) Local coordinates (b) Global coordinates (c) Displacements and rotations

(d) Forces and moments

An assembly of three pipe elements denoted as l, m and n connected to a common external node O at their nodes l_j, m_i and n_i, respectively is shown in Fig. 3.6. When all three nodes l_j, m_i and n_i are elastic, all three rotations are equal to those of the external node O. When one or two of these nodes undergo plastic deformations, the rotations will assume different values from those of node O. These values will be given through the plastic reduction matrix (Sec. 3.6). It is physically impossible that all three nodes are plastified, in which case the rotation equilibrium condition of joint O would be violated. Thus, the rotations of the remaining node will here coincide with those of node O for a stable structure.

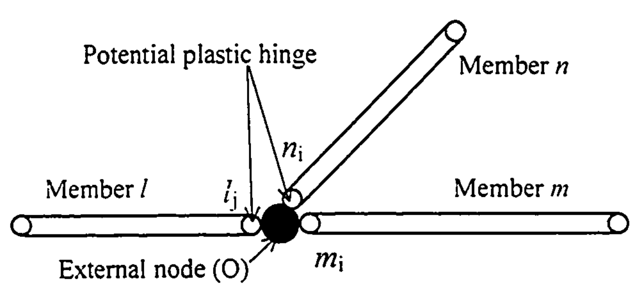

Fig. 3.6 Elements Connection to a Node

3.5 Elastic Stiffness Matrix

For short spans (e.g. *L/D*<10), shear deformation effects of beam tend to be significant. In order to include these effects into the formulation, one can consider a cantilever with an end moment *M* and transverse force *F* acting on its tip (Fig. 3.7). It is required to calculate the corresponding tip displacement and rotation, including shear deformation effects. By using Costigliano's theorem, these relations are

$$\begin{Bmatrix} \Delta \\ \theta \end{Bmatrix} = \begin{bmatrix} \dfrac{L^3}{3EI} + \dfrac{L}{GA_r} & -\dfrac{L^2}{2EI} \\ -\dfrac{L^2}{2EI} & \dfrac{L}{EI} \end{bmatrix} \begin{Bmatrix} F \\ M \end{Bmatrix} \tag{3.1}$$

in which A_r is the reduced shear area. For a thin walled pipe section, Blevins (1984) gives

$$A_r = \frac{2(1+\nu)}{4+3\nu} A \tag{3.2}$$

Fig. 3.7 Active DOF of a 2D Cantilever Beam

The matrix in Eq. 3.1 is the flexibility matrix in which ν is the Poisson's ratio. By inverting the flexibility matrix the stiffness matrix corresponding the two degrees of freedom at the node j, $[K_j^e]_{2\times2}$ is found as

$$\left[K_j^e\right]_{2\times2} = \frac{1}{1+\kappa}\begin{bmatrix} \frac{12EI}{L^3} & \frac{6EI}{L^2} \\ \frac{6EI}{L^2} & (4+\kappa)\frac{EI}{L} \end{bmatrix} \tag{3.3}$$

in which $\kappa = (12EI)/(L^2 GA_r)$ is the ratio of bending to shear rigidity. By considering the equilibrium conditions and symmetry property of the stiffness matrix, other sub-matrices of the full element stiffness matrix are obtained as:

$$\left[K^e\right]_{4\times4} = \frac{1}{1+\kappa}\begin{bmatrix} \frac{12EI}{L^3} & -\frac{6EI}{L^2} & -\frac{12EI}{L^3} & -\frac{6EI}{L^2} \\ & (4+\kappa)\frac{EI}{L} & \frac{6EI}{L^2} & (2-\kappa)\frac{EI}{L} \\ & & \frac{12EI}{L^3} & \frac{6EI}{L^2} \\ Sym. & & & (4+\kappa)\frac{EI}{L} \end{bmatrix} \tag{3.4}$$

The DOFs of node i along the local axes 1-2-3 and its rotations about these axes are numbered 1 through 6 respectively. For node j they are numbered 7 to 12. For a pipe section, the moment of inertia about both centroidal axes is equal (e.g. $I_2 = I_3 = I$). Using Eq. 3.4, the 3D stiffness matrix, can be expressed as

$$\left[K^{e}\right]_{12\times 12}=\begin{bmatrix}\left[K^{e}_{ii}\right]_{6\times 6} & \left[K^{e}_{ij}\right]_{6\times 6}\\ \left[K^{e}_{ij}\right]^{T}_{6\times 6} & \left[K^{e}_{jj}\right]_{6\times 6}\end{bmatrix}_{12\times 12}=\frac{1}{1+\kappa}\times \tag{3.5}$$

$$\left[\begin{array}{cccccc:cccccc}
\frac{EA}{L} & 0 & 0 & 0 & 0 & 0 & -\frac{EA}{L} & 0 & 0 & 0 & 0 & 0\\
 & \frac{12EI}{L^3} & 0 & 0 & 0 & \frac{6EI}{L^2} & 0 & \frac{-12EI}{L^3} & 0 & 0 & 0 & \frac{6EI}{L^2}\\
 & & \frac{12EI}{L^3} & 0 & -\frac{6EI}{L^2} & 0 & 0 & 0 & \frac{-12EI}{L^3} & 0 & \frac{-6EI}{L^2} & 0\\
 & & & \frac{GJ}{L} & 0 & 0 & 0 & 0 & 0 & \frac{-GJ}{L} & 0 & 0\\
 & & & & \frac{(4+\kappa)EI}{L} & 0 & 0 & 0 & \frac{6I}{L^2} & 0 & \frac{(2-\kappa)I}{L} & 0\\
 & & & & & \frac{(4+\kappa)EI}{L} & 0 & \frac{-6EI}{L^2} & 0 & 0 & 0 & \frac{(2-\kappa)EI}{L}\\
\hdashline
 & & & & & & \frac{EA}{L} & 0 & 0 & 0 & 0 & 0\\
 & & & & & & & \frac{12EI}{L^3} & 0 & 0 & 0 & \frac{-6EI}{L^2}\\
 & & & & & & & & \frac{12EI}{L^3} & 0 & \frac{6EI}{L^2} & 0\\
 & & & & & & & & & \frac{GJ}{L} & 0 & 0\\
 & Sym. & & & & & & & & & \frac{(4+\kappa)EI}{L} & 0\\
 & & & & & & & & & & & \frac{(4+\kappa)EI}{L}
\end{array}\right]$$

in which $J = 2I$ is the torsional rigidity constant, $G = E/2(1+\nu)$ is shear modulus and ν is Poisson's ratio.

For long members, κ tends to zero and the stiffness matrix becomes similar to that of other standard references (i.e. Kardestuncer 1974, Kassimali 1999, McGuire et. al. 2000).

$$
\left[K^e\right]_{12\times 12}\Big|_{\kappa=0} = \begin{bmatrix} \left[K^e_{II}\right]_{6\times 6} & \left[K^e_{IJ}\right]_{6\times 6} \\ \left[K^e_{IJ}\right]^T_{6\times 6} & \left[K^e_{JJ}\right]_{6\times 6} \end{bmatrix}_{12\times 12} = \tag{3.6}
$$

$$
\left[\begin{array}{cccccc:cccccc}
\frac{EA}{L} & 0 & 0 & 0 & 0 & 0 & \frac{-EA}{L} & 0 & 0 & 0 & 0 & 0 \\
 & \frac{12EI}{L^3} & 0 & 0 & 0 & \frac{6EI}{L^2} & 0 & \frac{-12EI}{L^3} & 0 & 0 & 0 & \frac{6EI}{L^2} \\
 & & \frac{12EI}{L^3} & 0 & -\frac{6EI}{L^2} & 0 & 0 & 0 & \frac{-12EI}{L^3} & 0 & \frac{-6EI}{L^2} & 0 \\
 & & & \frac{GJ}{L} & 0 & 0 & 0 & 0 & 0 & \frac{-GJ}{L} & 0 & 0 \\
 & & & & \frac{4EI}{L} & 0 & 0 & 0 & \frac{6I}{L^2} & 0 & \frac{2I}{L} & 0 \\
 & & & & & \frac{4EI}{L} & 0 & \frac{-6EI}{L^2} & 0 & 0 & 0 & \frac{2EI}{L} \\
\hdashline
 & & & & & & \frac{EA}{L} & 0 & 0 & 0 & 0 & 0 \\
 & & & & & & & \frac{12EI}{L^3} & 0 & 0 & 0 & \frac{-6EI}{L^2} \\
 & & & & & & & & \frac{12EI}{L^3} & 0 & \frac{6EI}{L^2} & 0 \\
 & & & & & & & & & \frac{GJ}{L} & 0 & 0 \\
 & Sym. & & & & & & & & & \frac{4EI}{L} & 0 \\
 & & & & & & & & & & & \frac{4EI}{L}
\end{array}\right]
$$

For elastic deformations

$$
\left[K^e\right]_{12\times 12}\{\Delta\}_{12\times 1} = \{F\}_{12\times 1} \tag{3.7}
$$

in which the nodal vector $\{F\}$ and nodal displacement vector $\{\Delta\}$ are

$$
\{F\}_{12\times 1} = \begin{Bmatrix} \{F_I\}_{3\times 1} \\ \{M_I\}_{3\times 1} \\ \{F_J\}_{3\times 1} \\ \{M_J\}_{3\times 1} \end{Bmatrix} \quad ; \{\Delta\}_{12\times 1} = \begin{Bmatrix} \{\Delta_I\}_{3\times 1} \\ \{\theta_I\}_{3\times 1} \\ \{\Delta_J\}_{3\times 1} \\ \{\theta_J\}_{3\times 1} \end{Bmatrix} \tag{3.8}
$$

3.6 Plastic Reduction Matrix

3.6.1 Loading and Unloading Behaviour

In order to simplify the problem, the discussion is restricted to a member subject to two types of stress resultants α and β (e.g., bending moments and axial force). The yield surface relating the internal forces $\Phi(\alpha, \beta) = 0$ is shown in Fig. 3.8. Each stress resultant state is defined by a point in the α-β plane. Although the discussion is related to a 2D yield surface, the concepts illustrated remain valid for multi-dimensional yield surfaces.

It is assumed that at a particular stage in the analysis, node i of a member has reached the fully plastic state while node j is still elastic (Fig. 3.8). Under the next increment of loading, the point representing the internal forces at node j may move in any direction in the α-β plane, while the force point of node i has two possibilities depending on the behaviour of the rest of the structure. a) It may unload elastically along path io (line a) or b) It can move along the tangent to the yield surface (line b).

Mathematically, elastic unloading is detected when the scalar $d\lambda$ in Eq. 2.3 gives a negative value (Sections 3.6.2.1, 3.6.2.2 and 3.6.2.3 give the expressions for $d\lambda$ at both nodes.) Otherwise, the point will move along the tangent to the yield surface plane.

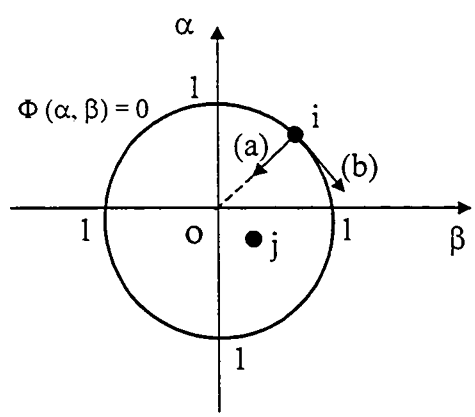

Fig. 3.8 An Arbitrary Stress Status of an Element Node

3.6.2 Plastic Reduction Matrices

Four different situations may arise in an element

1- Both nodes are plastified

2- Only the first node is plastified

3- Only the second node is plastified

4- Both nodes are elastic

An element plastic reduction matrix for each of the above cases has to be formulated separately. This is presented in sections 3.6.2.1 through 3.6.2.4.

3.6.2.1 Both Nodes are Plastified

According to the strain decomposition assumption, the joint generalized displacement increment can be decomposed into elastic and plastic contributions. In vector form, this is expressed as

$$\begin{Bmatrix} \{d\Delta_i\}_{6\times1} \\ \{d\Delta_j\}_{6\times1} \end{Bmatrix}_{12\times1} = \begin{Bmatrix} \{d\Delta_i^e\}_{6\times1} \\ \{d\Delta_j^e\}_{6\times1} \end{Bmatrix}_{12\times1} + \begin{Bmatrix} \{d\Delta_i^p\}_{6\times1} \\ \{d\Delta_j^p\}_{6\times1} \end{Bmatrix}_{12\times1} \tag{3.9}$$

in which

$\{d\Delta_i\}_{6\times1}$ is the incremental displacements vector of node i

$\{d\Delta_j\}_{6\times1}$ is the incremental displacements vector of node j

$\{d\Delta_i^e\}_{6\times1}$ is the incremental elastic displacements vector of node i

$\{d\Delta_j^e\}_{6\times1}$ is the incremental elastic displacements vector of node j

$\{d\Delta_i^p\}_{6\times1}$ is the incremental plastic displacements vector of node i

$\{d\Delta_j^p\}_{6\times1}$ is the incremental plastic displacements vector of node j

The elastic force-displacement relationship is valid for incremental end forces and incremental displacements, i.e.,

$$\begin{Bmatrix} \{dF_i\}_{6\times1} \\ \{dF_j\}_{6\times1} \end{Bmatrix}_{12\times1} = \left[K^e\right]_{12\times12} \begin{Bmatrix} \{d\Delta_i^e\}_{6\times1} \\ \{d\Delta_j^e\}_{6\times1} \end{Bmatrix}_{12\times1} \tag{3.10}$$

The normality rule postulates that the incremental plastic deformation is normal to a plastic potential function, Ψ. By taking the yield function, Φ as the plastic potential function (i.e. $\Psi = \Phi$) and knowing that the normality rule is applicable to both nodes separately, Eq. 2.3 gives

$$\begin{Bmatrix} \{d\Delta_i^p\}_{6\times1} \\ \{d\Delta_j^p\}_{6\times1} \end{Bmatrix}_{12\times1} = \begin{bmatrix} \left\{ \left. \dfrac{\partial\Phi}{\partial X_k} \right|_i \right\}_{6\times1} & \{0\}_{6\times1} \\ \{0\}_{6\times1} & \left\{ \left. \dfrac{\partial\Phi}{\partial X_k} \right|_j \right\}_{6\times1} \end{bmatrix}_{12\times2} \times \begin{Bmatrix} d\lambda_i \\ d\lambda_j \end{Bmatrix}_{2\times1} \tag{3.11}$$

in which

$d\lambda_i$ is a scalar indicating the magnitude of the plastic deformation for node i

$d\lambda_j$ is a scalar indicating the magnitude of the plastic deformation for node j

$\left\{\partial\Phi/\partial X_k|_i\right\}_{6\times1}$ are derivatives of yield surface with respect to nodal force X_k for node i

$\left\{\partial\Phi/\partial X_k|_j\right\}_{6\times1}$ are derivatives of yield surface with respect to nodal force X_k for node j

$k = 1\ldots6$ denotes the component of nodal forces

By defining matrix $[G]$ such that

$$[G] = \begin{bmatrix} \{G_i\}_{6\times1} & \{0\}_{6\times1} \\ \{0\}_{6\times1} & \{G_j\}_{6\times1} \end{bmatrix} = \begin{bmatrix} \left\{ \left. \dfrac{\partial\Phi}{\partial X_k} \right|_i \right\}_{6\times1} & \{0\}_{6\times1} \\ \{0\}_{6\times1} & \left\{ \left. \dfrac{\partial\Phi}{\partial X_k} \right|_j \right\}_{6\times1} \end{bmatrix}_{12\times2} \tag{3.12}$$

Eq. 3.11 can be written in the more compact form as

$$\begin{Bmatrix} \{d\Delta_I^p\}_{6\times1} \\ \{d\Delta_J^p\}_{6\times1} \end{Bmatrix}_{12\times1} = [G]_{12\times2} \begin{Bmatrix} d\lambda_I \\ d\lambda_J \end{Bmatrix}_{2\times1} \tag{3.13}$$

This matrix is the core of development of the plastic reduction matrix. At plastified nodes, the plastic reduction matrix reduces the stiffness of the element and ensures that the elastic contribution to the total deformation is tangent to the yield surface at the force point.

According to the normality flow rule, the plastic deformation and the incremental force vector are orthogonal, i.e.,

$$\left\langle \left\langle d\Delta_I^p \right\rangle_{1\times6} \quad \left\langle d\Delta_J^p \right\rangle_{1\times6} \right\rangle_{1\times12} \cdot \begin{Bmatrix} \{dF_I\}_{6\times1} \\ \{dF_J\}_{6\times1} \end{Bmatrix}_{12\times1} = 0 \tag{3.14}$$

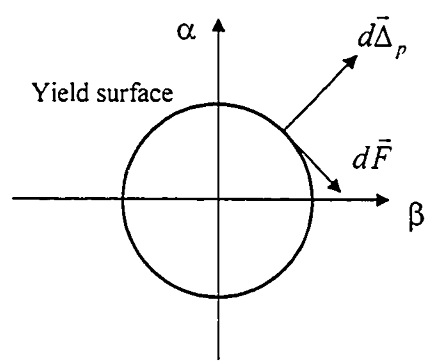

Fig. 3.9 An Arbitrary Stress Status of a Plastified Element Node

Substituting Eq. 3.13 into Eq. 3.14, results into

$$\left\langle d\lambda_I \quad d\lambda_J \right\rangle_{1\times2} [G]_{2\times12}^T \begin{Bmatrix} \{dF_I\}_{6\times1} \\ \{dF_J\}_{6\times1} \end{Bmatrix}_{12\times1} = 0 \tag{3.15}$$

For a given displacement increment, λ_I and λ_J are positive scalars, therefore

$$[G]^T_{2\times12}\begin{Bmatrix}\{dF_i\}_{6\times1}\\\{dF_j\}_{6\times1}\end{Bmatrix}_{12\times1}=\begin{Bmatrix}0\\0\end{Bmatrix}_{2\times1} \tag{3.16}$$

Pre-multiplying Eq. 3.9 by $[K^e]_{12\times12}$ results into

$$[K^e]_{12\times12}\begin{Bmatrix}\{d\Delta_i\}_{6\times1}\\\{d\Delta_j\}_{6\times1}\end{Bmatrix}_{12\times1}=[K^e]_{12\times12}\begin{Bmatrix}\{d\Delta_i^e\}_{6\times1}\\\{d\Delta_j^e\}_{6\times1}\end{Bmatrix}_{12\times1}+[K^e]_{12\times12}\begin{Bmatrix}\{d\Delta_i^p\}_{6\times1}\\\{d\Delta_j^p\}_{6\times1}\end{Bmatrix}_{12\times1} \tag{3.17}$$

Substituting Eqs. 3.13 and 3.10 into Eq. 3.17 gives

$$[K^e]_{12\times12}\begin{Bmatrix}\{d\Delta_i\}_{6\times1}\\\{d\Delta_j\}_{6\times1}\end{Bmatrix}_{12\times1}=\begin{Bmatrix}\{dF_i\}_{6\times1}\\\{dF_j\}_{6\times1}\end{Bmatrix}_{12\times1}+[K^e]_{12\times12}[G]_{12\times2}\begin{Bmatrix}d\lambda_i\\d\lambda_j\end{Bmatrix}_{2\times1} \tag{3.18}$$

Pre-multiplying both sides of the Eq. 3.18 by $[G]^T_{2\times12}$ and using the identity in Eq.3.16 result into

$$[G]^T_{2\times12}[K^e]_{12\times12}\begin{Bmatrix}\{d\Delta_i\}_{6\times1}\\\{d\Delta_j\}_{6\times1}\end{Bmatrix}_{12\times1}=[G]^T_{2\times12}[K^e]_{12\times12}[G]_{12\times2}\begin{Bmatrix}d\lambda_i\\d\lambda_j\end{Bmatrix}_{2\times1} \tag{3.19}$$

When both ends are plastified, the inverse of matrix $[G]^T[K][G]$ exists and the following derivation holds true. Other cases when one or two nodes remains elastic will result in a singularity of the matrix $[G]^T[K][G]$ and will be treated separately under sections 3.6.2.2 through 3.6.2.3. By pre-multiplying both sides of Eq. 3.19 by $\left[[G]^T_{2\times12}[K^e]_{12\times12}[G]_{12\times2}\right]^{-1}$, the vector of scalars $\langle d\lambda_i \quad d\lambda_j\rangle^T$ is obtained as

$$\begin{Bmatrix}d\lambda_i\\d\lambda_j\end{Bmatrix}_{2\times1}=\left[[G]^T_{2\times12}[K^e]_{12\times12}[G]_{12\times2}\right]^{-1}_{2\times2}[G]^T_{2\times12}[K^e]_{12\times12}\begin{Bmatrix}\{d\Delta_i\}_{6\times1}\\\{d\Delta_j\}_{6\times1}\end{Bmatrix}_{12\times1} \tag{3.20}$$

If, after a plastic hinge has been formed at a particular node, the force redistribution that takes place under loading of the structure is such as to reduce the resultant action on that section, it may unload elastically. The appearance of a negative element in the $\{d\lambda\}$ vector at the end of a load increment is a signal of unloading, and is corrected in the subsequent iteration accordingly.

The next step is deriving the plastic reduction matrix. Substituting Eq. 3.9 into Eq. 3.10 and using identity in Eq. 3.13 results into

$$\begin{Bmatrix} \{dF_i\}_{6\times1} \\ \{dF_j\}_{6\times1} \end{Bmatrix}_{12\times1} = [K^e]_{12\times12} \left\{ \begin{Bmatrix} \{d\Delta_i\}_{6\times1} \\ \{d\Delta_j\}_{6\times1} \end{Bmatrix}_{12\times1} - [G]_{12\times2} \begin{Bmatrix} d\lambda_i \\ d\lambda_j \end{Bmatrix}_{2\times1} \right\}_{12\times1} \tag{3.21}$$

Substituting Eq. 3.20 into Eq. 3.21 yields

$$\begin{Bmatrix} \{dF_i\}_{6\times1} \\ \{dF_j\}_{6\times1} \end{Bmatrix}_{12\times1} = \left[[K^e]_{12\times12} + [K^m]_{12\times12} \right] \begin{Bmatrix} \{d\Delta_i\}_{6\times1} \\ \{d\Delta_j\}_{6\times1} \end{Bmatrix}_{12\times1} \tag{3.22}$$

in which $[K^m]$, the plastic reduction matrix is given by

$$[K^m]_{12\times12} = -[K^e]_{12\times12} \, [G]_{12\times2} \left[[G]^T_{2\times12} \, [K^e]_{12\times12} \, [G]_{12\times2} \right]^{-1}_{2\times2} [G]^T_{2\times12} \, [K^e]_{12\times12} \tag{3.23}$$

3.6.2.2 Only the First Node is Plastified

Vector $\{G_i\}$ is nonzero only when the first node is plastified. Similarly $\{G_j\}$ is nonzero only when the second node is plastified. When any (or both) nodes are elastic, the above formulation would result in a singular matrix $\left[[G]^T_{2\times12}[K^e]_{12\times12} \, [G]_{12\times2}\right]$. Therefore, Eq. 3.23 is no longer applicable and the formulation is modified as follows.

By writing matrix $\left[[G]^T_{2\times12}[K^e]_{12\times12} \, [G]_{12\times2}\right]$ in a sub-matrix form, matrix multiplication gives

$$[G]^T[K][G]=\begin{bmatrix}\langle G_i\rangle^T_{1\times6} & \langle 0\rangle_{1\times6}\\ \langle 0\rangle_{1\times6} & \langle G_j\rangle^T_{1\times6}\end{bmatrix}_{2\times12}\begin{bmatrix}[K^e_{ii}]_{6\times6} & [K^e_{ij}]_{6\times6}\\ [K^e_{ij}]^T_{6\times6} & [K^e_{jj}]_{6\times6}\end{bmatrix}_{12\times12}\begin{bmatrix}\{G_i\}_{6\times1} & \{0\}_{6\times1}\\ \{0\}_{6\times1} & \{G_j\}_{6\times1}\end{bmatrix}_{12\times2}$$

$$=\begin{bmatrix}\langle G_i\rangle^T_{1\times6}[K^e_{ii}]_{6\times6}\{G_i\}_{6\times1} & \langle G_i\rangle^T_{1\times6}[K^e_{ij}]_{6\times6}\{G_j\}_{6\times1}\\ \langle G_j\rangle^T_{1\times6}[K^e_{ij}]_{6\times6}\{G_i\}_{6\times1} & \langle G_j\rangle^T_{1\times6}[K^e_{jj}]_{6\times6}\{G_j\}_{6\times1}\end{bmatrix}_{2\times2} \tag{3.24}$$

Also

$$[G]^T[K]\{d\Delta\}=\begin{bmatrix}\langle G_i\rangle^T_{1\times6} & \langle 0\rangle_{1\times6}\\ \langle 0\rangle_{1\times6} & \langle G_j\rangle^T_{1\times6}\end{bmatrix}_{2\times12}\begin{bmatrix}[K^e_{ii}]_{6\times6} & [K^e_{ij}]_{6\times6}\\ [K^e_{ij}]^T_{6\times6} & [K^e_{jj}]_{6\times6}\end{bmatrix}_{12\times12}\begin{Bmatrix}\{d\Delta_i\}_{6\times1}\\ \{d\Delta_j\}_{6\times1}\end{Bmatrix}_{12\times1}$$

$$=\begin{Bmatrix}\langle G_i\rangle^T_{1\times6}[K^e_{ii}]_{6\times6}\{d\Delta_i\}_{6\times1}+\langle G_i\rangle^T_{1\times6}[K^e_{ij}]_{6\times6}\{d\Delta_j\}_{6\times1}\\ \langle G_j\rangle^T_{1\times6}[K^e_{ij}]_{6\times6}\{d\Delta_i\}_{6\times1}+\langle G_j\rangle^T_{1\times6}[K^e_{jj}]_{6\times6}\{d\Delta_j\}_{6\times1}\end{Bmatrix}_{2\times1} \tag{3.25}$$

When only the first node is plastified and the second node is still elastic $\{G_j\}$ vanishes. This vanishes the corresponding terms in Eqs. 3.24 and 3.25. Substituting Eqs. 3.24 and 3.25 into Eq. 3.19 results into

$$\begin{Bmatrix}\langle G_i\rangle^T_{1\times6}\left([K^e_{ii}]_{6\times6}\{d\Delta_i\}_{6\times1}+[K^e_{ij}]_{6\times6}\{d\Delta_j\}_{6\times1}\right)\\ 0\end{Bmatrix}_{2\times1}=\begin{bmatrix}\langle G_i\rangle^T_{1\times6}[K^e_{ii}]_{6\times6}\{G_i\}_{6\times1} & 0\\ 0 & 0\end{bmatrix}_{2\times2}\begin{Bmatrix}d\lambda_i\\ d\lambda_j\end{Bmatrix}_{2\times1} \tag{3.26}$$

Therefore,

$$\begin{Bmatrix}d\lambda_i\\ d\lambda_j\end{Bmatrix}_{2\times1}=\begin{Bmatrix}\dfrac{\langle G_i\rangle^T_{1\times6}[K^e_{ii}]_{6\times6}\{d\Delta_i\}_{6\times1}+\langle G_i\rangle^T_{1\times6}[K^e_{ij}]_{6\times6}\{d\Delta_j\}}{\langle G_i\rangle^T_{1\times6}[K^e_{ii}]_{6\times6}\{G_i\}_{6\times1}}\\ 0\end{Bmatrix}_{2\times1} \tag{3.27}$$

Also substituting Eqs. 3.24 and 3.25 into Eq. 3.23 gives

$$\left[K^{m}\right]_{12\times12}=-\frac{1}{\left\langle G_{I}\right\rangle_{1\times6}^{T}\left[K_{II}^{e}\right]_{6\times6}\left\{G_{I}\right\}_{6\times1}}\left[K^{e}\right]_{12\times12}\left[G\right]_{12\times2}\left[G\right]_{2\times12}^{T}\left[K^{e}\right]_{12\times12} \tag{3.28}$$

Equation 3.28 is the plastic reduction matrix when the first node is plastified while the second node remains elastic.

3.6.2.3 Only the Second Node is Plastified

In the same manner, appropriate equations can be derived for a case that the first end is elastic and the second end is plastic. When only the second node plastifies, the first node remains elastic $\{G_I\}$ vanishes. This vanishes the corresponding term in Eqs. 3.24 and 3.25. Substituting Eqs. 3.24 and 3.25 into Eq. 3.19 results into

$$\begin{Bmatrix} 0 \\ \left\langle G_{J}\right\rangle_{1\times6}^{T}\left(\left[K_{IJ}^{e}\right]_{6\times6}\left\{d\Delta_{I}\right\}_{6\times1}+\left[K_{JJ}^{e}\right]_{6\times6}\left\{d\Delta_{J}\right\}_{6\times1}\right)\end{Bmatrix}_{2\times1}=\begin{bmatrix}0 & 0\\ 0 & \left\langle G_{J}\right\rangle_{1\times6}^{T}\left[K_{JJ}^{e}\right]_{6\times6}\left\{G_{J}\right\}_{6\times1}\end{bmatrix}_{2\times2}\begin{Bmatrix} d\lambda_{I}\\ d\lambda_{J}\end{Bmatrix}_{2\times1} \tag{3.29}$$

Therefore,

$$\begin{Bmatrix} d\lambda_{I}\\ d\lambda_{J}\end{Bmatrix}_{2\times1}=\begin{Bmatrix} 0 \\ \dfrac{\left\langle G_{J}\right\rangle^{T}\left[K_{IJ}^{e}\right]_{6\times6}\left\{d\Delta_{I}\right\}_{6\times1}+\left\langle G_{J}\right\rangle_{1\times6}^{T}\left[K_{JJ}^{e}\right]_{6\times6}\left\{d\Delta_{J}\right\}_{6\times1}}{\left\langle G_{J}\right\rangle_{1\times6}^{T}\left[K_{JJ}^{e}\right]_{6\times6}\left\{G_{J}\right\}_{6\times1}}\end{Bmatrix}_{2\times1} \tag{3.30}$$

Substituting Eqs. 3.24 and 3.25 into Eq. 3.23 gives

$$\left[K^{m}\right]_{12\times12}=-\frac{1}{\left\langle G_{J}\right\rangle_{1\times6}^{T}\left[K_{JJ}^{e}\right]_{6\times6}\left\{G_{J}\right\}_{6\times1}}\left[K^{e}\right]_{12\times12}\left[G\right]_{12\times2}\left[G\right]_{2\times12}^{T}\left[K^{e}\right]_{12\times12} \tag{3.31}$$

Equation 3.31 gives the plastic reduction matrix when the second node is plastified while the first node remains elastic.

3.6.2.4 Both Nodes are Elastic

In this case, material non-linear behaviour does not take place and both the plastic reduction matrix and the plastic deformation magnitude vector are zeros. Equations 3.20 and 3.23 reduce to

$$\begin{Bmatrix} d\lambda_i \\ d\lambda_j \end{Bmatrix} = \begin{Bmatrix} 0 \\ 0 \end{Bmatrix} \tag{3.32}$$

and

$$\left[K^{m}\right]_{12\times12} = \left[0\right]_{12\times12} \tag{3.33}$$

3.6.3 Tangent Stiffness Matrix

The tangent stiffness matrix is the summation of the elastic stiffness matrix (Eq. 3.5) and the plastic reduction matrix (Eq. 3.23, 3.28 or 3.31). i.e.,

$$\left[K\right]_{12\times12} = \left[K^{e}\right]_{12\times12} + \left[K^{m}\right]_{12\times12} \tag{3.34}$$

3.7 Coordinate Transformation

To be able to model a pipe system with general geometry it is needed to transform load and displacement vectors from global to local coordinate systems and vice versa. To achieve this goal, it is assumed that coordinates i_X, i_Y and i_Z (Fig. 3.10a) of an arbitrary vector $\vec{v}$ in the global coordinate system X-Y-Z are known (Fig. 3.10b). It is required to find its coordinates i_1, i_2 and i_3 in local coordinate system 1-2-3.

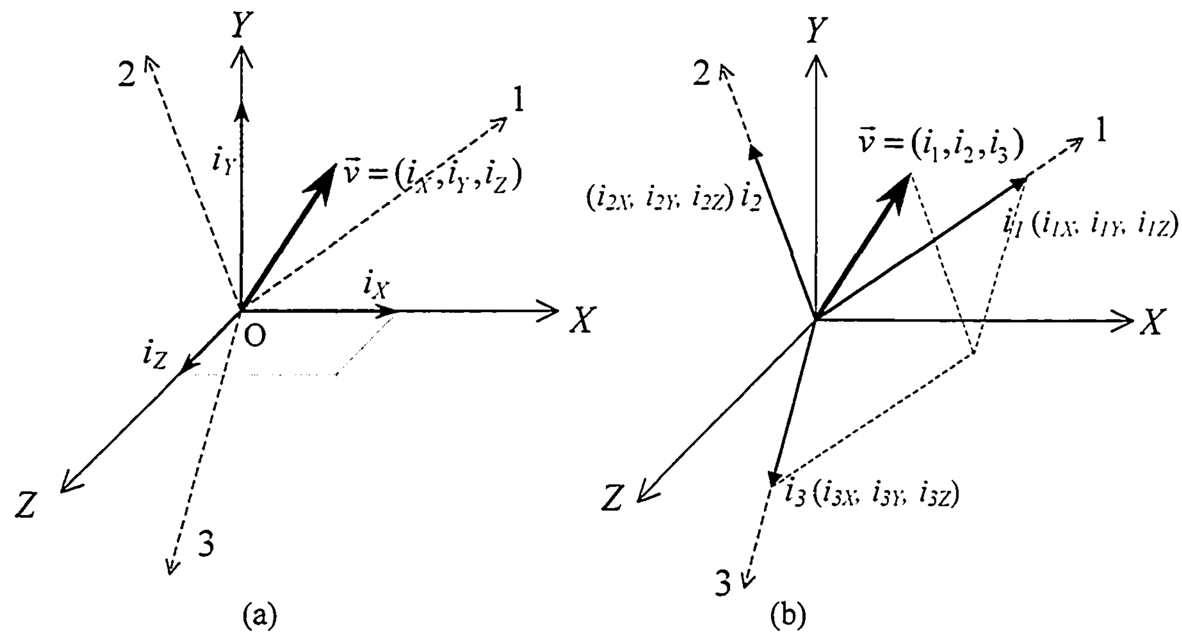

Fig. 3.10 Global and Local Coordinate Systems

From geometry, the following relationship exists between the two sets of coordinates

$$\begin{Bmatrix} i_1 \\ i_2 \\ i_3 \end{Bmatrix}_{3\times1} = [r]_{3\times3} \begin{Bmatrix} i_X \\ i_Y \\ i_Z \end{Bmatrix}_{3\times1} \tag{3.35}$$

in which the elements of transformation matrix $[r]_{3\times3}$ as defined as in Fig. 3.10b are:

$$[r]_{3\times3} = \begin{bmatrix} r_{1X} & r_{1Y} & r_{1Z} \\ r_{2X} & r_{2Y} & r_{2Z} \\ r_{3X} & r_{3Y} & r_{3Z} \end{bmatrix} = \begin{bmatrix} r_{1X} & r_{1Y} & r_{1Z} \\ -\dfrac{r_{1X}r_{1Y}}{\sqrt{r_{1X}^2+r_{1Z}^2}} & \sqrt{r_{1X}^2+r_{1Z}^2} & -\dfrac{r_{1Y}r_{1Z}}{\sqrt{r_{1X}^2+r_{1Z}^2}} \\ -\dfrac{r_{1Z}}{\sqrt{r_{1X}^2+r_{1Z}^2}} & 0 & \dfrac{r_{1X}}{\sqrt{r_{1X}^2+r_{1Z}^2}} \end{bmatrix} \tag{3.36}$$

in which r_{1X}, r_{1Y} and r_{1Z} are direction cosines of axis 1 with respect to the global coordinate axes, r_{2X}, r_{2Y} and r_{2Z} are direction cosines of axis 2 and r_{3X}, r_{3Y} and r_{3Z} are direction cosines of axis 3 with respect to the global coordinate axes.

Axis 3 is defined as the normal to the 1OY plane and is given by the cross product $i_3 = i_1 \times (0,1,0)$; Axis 2 is defined from the orthogonal relation $i_2 = i_3 \times i_1$. When vector $\bar{v}$ is along the axis Y, i_1 coincides with (0,1,0) and the above axis definition must be modified. In this special case, the unit vector $i_1 = i_Y$ while the unit vector $i_3 = i_Z$ gives the direction of the axis 3. The unit vector i_2 is established using the cross product $i_3 \times i_1 = -i_X$. Thus, the transformation matrix is given as

$$[r]^{1}_{3\times3} = \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3.37}$$

The transformation law in Eq. 3.35 is valid for any vector and $[r]_{3\times3}$ is thus applied to the forces, moments, displacements and rotations at each node. (i.e. $\{F_i\}_{3\times1}$, $\{F_j\}_{3\times1}$, $\{M_i\}_{3\times1}$, $\{M_j\}_{3\times1}$, $\{\Delta_i\}_{3\times1}$, $\{\Delta_j\}_{3\times1}$, $\{\theta_i\}_{3\times1}$ and $\{\theta_j\}_{3\times1}$)

For all twelve degrees of freedom of an element, the transformation matrix relating the nodal quantities in the local coordinate system to those in the global coordinate system is

$$[T] = \begin{bmatrix} [r]_{3\times3} & 0 & 0 & 0 \\ 0 & [r]_{3\times3} & 0 & 0 \\ 0 & 0 & [r]_{3\times3} & 0 \\ 0 & 0 & 0 & [r]_{3\times3} \end{bmatrix}_{12\times12} \tag{3.38}$$

Matrix 3.38 is orthogonal, i.e. $[T]^T = [T]^{-1}$

Matrix $[T]$ can be used to relate the nodal load vector and displacement vector from local coordinate system into global coordinate system as

$$[T]_{12\times12}\{F_G\}_{12\times1} = \{F\}_{12\times1} \tag{3.39}$$

$$[T]_{12\times12}\{\Delta_G\}_{12\times1} = \{\Delta\}_{12\times1} \tag{3.40}$$

in which $\{F_G\}_{12\times1}$ is the nodal force vector in global coordinate system

$\{F\}_{12\times1}$ is the nodal force vector in local coordinate system

$\{\Delta_G\}_{12\times1}$ is the nodal displacement vector in global coordinate system

$\{\Delta\}_{12\times1}$ is the nodal displacement vector in local coordinate system

Starting with the equilibrium equation $[F]=[K]\{\Delta\}$ in the local coordinate system and using identities Eq. 3.39 and 3.40 , gives

$$[T]_{12\times12}\{F_G\}_{12\times1} = [K]_{12\times12}[T]_{12\times12}\{\Delta_G\}_{12\times1} \tag{3.41}$$

Pre-multiplying both sides of Eq. 3.41 by $[T]^T_{12\times12}$ yields

$$\{F_G\}_{12\times1} = [T]^T_{12\times12}[K]_{12\times12}[T]_{12\times12}\{\Delta_G\}_{12\times1} \tag{3.42}$$

Equation 3.42 signifies that element stiffness matrix in the global coordinates $[K_G]$, is related to the element stiffness matrix in the local coordinates $[K]$ by

$$[K_G]_{12\times12} = [T]^T_{12\times12}[K]_{12\times12}[T]_{12\times12} \tag{3.43}$$

Chapter 4

Solution of Equilibrium Equations

4.1 General

Different iterative procedures are available for the solution of non-linear finite element equations. In these methods, external loads are typically applied in several increments. The equilibrium condition for each increment is established using a number of iterations within a specific tolerance. Although iterative methods are computationally expensive, they can follow the response path closely, when compared to other non-iterative incremental methods.

In this chapter, the terminology used in iterative methods is introduced. The load control algorithm is described in detail, as it is the basis of other more elaborate iterative schemes. The discussion is then extended to the displacement control method, which is used in the program developed as part of this thesis.

4.2 Equilibrium Path and Response Diagrams

The nonlinear response diagrams of a structure can be schematically represented in a two-dimensional plot as a relationship between a representative load value and a representative displacement value as shown in Fig. 4.1. The qualifier "representative" value implies a choice among many possible candidates. The smooth curve shown is referred to as the equilibrium path. Each point on the path represents a possible equilibrium configuration or state of the structure and the objective here is to trace this path. Either a load or a displacement can be selected as a control parameter. It is also possible to define a control parameter as a combination of the representative load and the representative displacement.

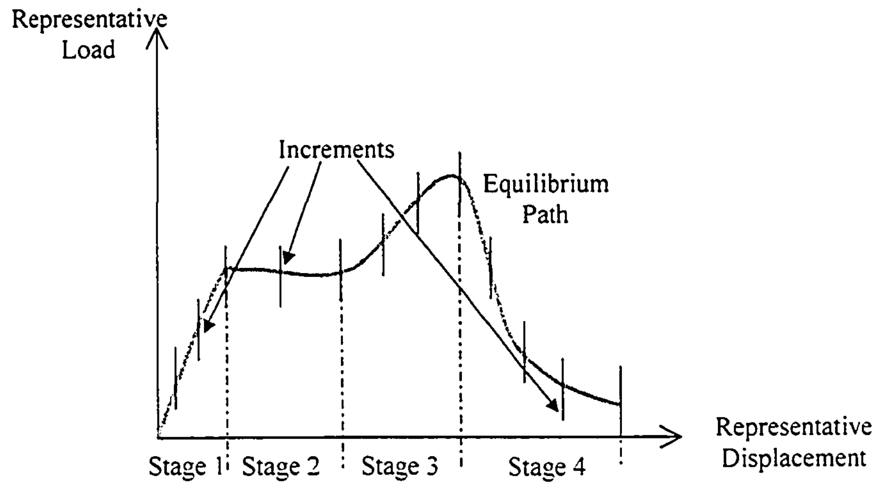

Fig. 4.1 Response Diagram and Equilibrium Path

4.3 Iterative Procedures

4.3.1 Terminology

In all types of non-linear analysis, it is required to follow the equilibrium response of the structure by adopting an iterative solution procedure. The basic idea is to follow the structural response path by gradually incrementing the control parameter and the structural state. A stepwise method helps attaining convergence, avoids extraneous roots, and gives an insight into the structural behaviour, when compared to a method based on total displacement. The following terminology is devised after Felippa (2000).

The process includes a categorized breakdown into stages (Fig. 4.1), incremental steps and iterative steps. Processing a complex loading history generally involves performing a series of analysis stages. Control parameters are varied dependently in each stage and are characterized by a single stage control parameter. Stages are linked together in a sense that the ending point of one is the starting point for another. For a non-complex loading history, it is possible to use only one stage. In order to advance the solution, each stage is broken into increments (Fig. 4.2). Over each increment, the state and control parameters undergo a finite change. If needed, some iteration is performed over each increment to eliminate the residual error between the equilibrium path and the approximate linearized solution. The converged solution after each increment is normally of interest to the analyst, as it represents an equilibrium state.

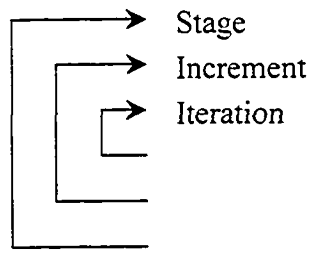

Fig. 4.2 Categorized breakdown in nonlinear solution methods

A successful iterative procedure scheme must include the following

1) A controlling parameter capable of describing the progress of the computation along the equilibrium path.
2) An iterative method that solves a set of simultaneous nonlinear equilibrium equations along the loading path.
3) Efficient convergence criteria

4.3.2 Control Schemes

Different control schemes have been devised to perform non-linear analysis. These are force control, displacement control and arc-length control. The force control method is the easiest to implement but cannot generally capture the limit point of the equilibrium path (Fig. 4.3a). In this scheme, the loads are applied as prescribed increments to the structure and the system of equations are solved iteratively for the corresponding incremental displacements. The load control scheme fails in the neighbourhood of the limit point. In the displacement control scheme, a selected displacement is prescribed and the corresponding load increment is obtained and kept constant throughout subsequent iterations (Fig. 4.3b). This scheme can capture the limit point and the post-limit behaviour of the structure but fails in snapback regions, which are characteristic of shell buckling. The arc-length scheme is based on a parameter including both displacements and loads (Fig. 4.3). It is prescribed by means of a constraint equation, which is added to the set of equilibrium equations. This method is generally the most elaborate to implement. It is useful to capture snapback behaviour of structures (Fig. 4.3c). Snapback behaviour is encountered in buckling analysis of rings and shells. In this research, the analysis is based on beam idealization of pipe sections rather than a shell analysis. Therefore, local buckling phenomena are not modelled and snapback behaviour is not captured. As a result, the displacement control method is judged most suitable, since it can predict both the limit point and post-peak behaviour of the system.

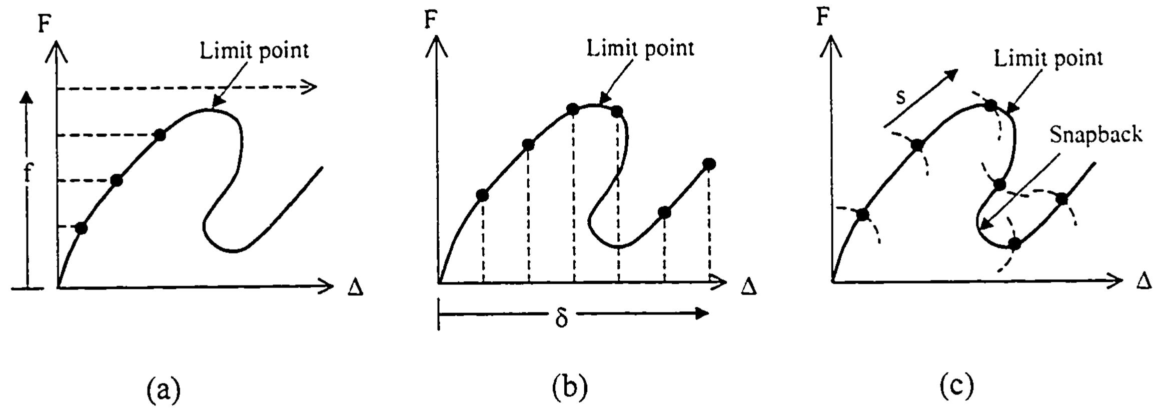

Fig. 4.3 Incremental Control Techniques and their Limitations

(a) Load control; (b) Displacement control; (c) Arc-length control

4.4 Load Control Method Scheme

4.4.1 Procedure

Load control is the simplest method to implement. Displacement control can be considered as a variation of the load control method. Therefore, by explaining the load control scheme, the idea behind displacement control scheme is simplified.

A schematic representation of load control method is shown in Fig. 4.4 for a structure with a representative load P. In this figure, point 1 on the equilibrium path is assumed to be known and the goal is to find point 2 corresponding to a prescribed load increment dP_n.

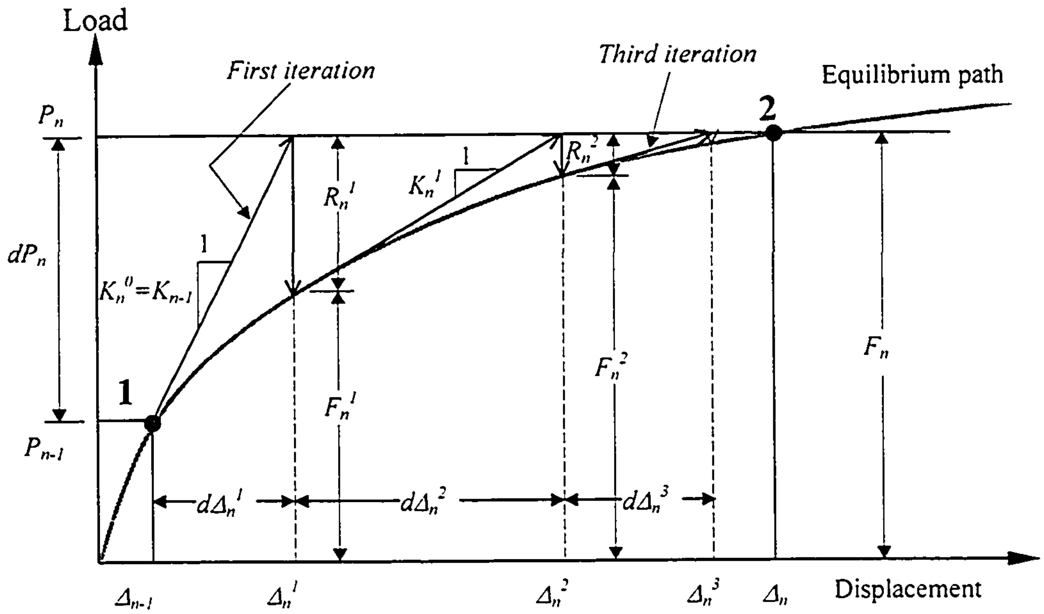

Fig. 4.4 Material Non-Linear Analysis by Load Control Method

Under the load control scheme, it is assumed that equilibrium has been established in a previous step *n-1*, i.e. Δ_{n-1} corresponding to a load level P_{n-1} that is known (point 1). A prescribed load increment $\{dP_n\}$ is then applied to the system. The value of applied load $\{P_{n-1}+dP_n\}$ remains constant during the following iterations and an iterative procedure is adopted to eliminate the unbalanced forces $\{R_n^i\}$ that exist at the end of iterations. The presence of unbalanced forces indicates that equilibrium between the externally applied forces and internally calculated forces is violated. The unbalanced forces arise because of the linearization process in which the updated stiffness matrix is formulated based on the previous configuration (first iteration at point 1 in this case) and the internal forces in the structure.

The procedure for implementing the load control method is based on the Newton-Raphson iterative technique. Unless otherwise stated, subscripts refer to the load increment number and superscripts refer to the iteration number within the load increment. All vectors and matrices are those for the whole structure and are referred to the global coordinate system (Section 3.7).

The procedure is as follows:

1) Calculate load increment $\{dP_n\}$

$$\{dP_n\} = \eta_n \{P\} \tag{4.1}$$

in which $\{P\}$ is the total applied loads vector and η_n is a load factor

2) Set iteration number i as zero.

3) Set the unbalance load $\{R_n^0\}$ as the load increment $\{dP_n\}$.

4) Calculate the total external force vector $\{P_n\}$ from

$$\{P_n\} = \{P_{n-1}\} + \{dP_n\} \tag{4.2}$$

5) Set the initial stiffness matrix of the current increment $[K_n^0]$ as the stiffness matrix of the final iteration of the previous increment $[K_{n-1}]$. For the first increment, elastic behaviour is assumed.

6) Increment the iteration number i.

7) After imposing boundary conditions, solve the following system of equations for the structure incremental displacement vector $\{d\Delta_n^i\}$.

$$[K_n^{i-1}]\{d\Delta_n^i\} = \{R_n^{i-1}\} \tag{4.3}$$

8) Calculate the structure total nodal displacement vector at the end of the ith iteration $\{\Delta_n^i\}$ using

$$\{\Delta_n^i\} = \{\Delta_{n-1}\} + \sum_{j=1}^{i} \{d\Delta_n^j\} \tag{4.4}$$

9) For each element, individually perform the following

   i. Extract the element nodal displacement vector $\{\Delta_G\}_{12\times1}$ from $\{\Delta'_n\}$.

   ii. Transform $\{\Delta_G\}_{12\times1}$ to local coordinate system $\{\Delta\}_{12\times1}$ (Sec. 3.7).

   iii. Evaluate the element nodal forces $\{F\}_{12\times1}$ based on new $\{\Delta\}_{12\times1}$ and stiffness matrix from the previous iteration, using

$$\{F\}_{12\times1} = [K]_{12\times12}\{\Delta\}_{12\times1} \tag{4.5}$$

   iv. Determine whether each of the element nodes is plastified. In doing so, the circumferential shear τ for each of the two nodes is calculated from Eq. 2.4, and then substituted into Eq. 2.5. A value of Eq. 2.5 greater than zero is the indication of the formation of a plastic hinge at that point. If this violation of the yield condition occurs, the nodal forces should be scaled back and returned to the yield surface. The algorithm is discussed in Sec. 4.4.2. This scaling back results into a contribution to the residual forces $\{R'_n\}$ (step12).

   v. Update the element stiffness matrix $[K]_{12\times12}$. Four possibilities may arise

      a. The nodal forces lie within the yield surface. In this case, the element stiffness matrix is assumed elastic and is evaluated according to Eq. 3.5.

      b. Both nodes are plastified. The corresponding matrix is provided in Eq.3.23.

      c. Only the first node is plastified. The element stiffness matrix is calculated according to Eq. 3.28.

      d. Only the second node is plastified. The element stiffness matrix is calculated according to Eq. 3.31.

   vi. Transform the element stiffness matrix and load vector into the global coordinate system (Sec. 3.7).

10) Form the structure internal force vector at the end of the ith iteration $\{F'_n\}$ by assembling the element force vectors (evaluated in step vi).

11) Assemble the elements stiffness matrices (evaluated in step vi) to form the structure stiffness matrix $[K_n^i]$ based on the newest configuration.

12) Calculate the residual force $\{R_n^i\}$ at the end of the current iteration from

$$\{R_n^i\} = \{P_n\} - \{F_n^i\} \tag{4.6}$$

13) Check whether the convergence criterion (Sec.4.4.3) is met. If not repeat steps 6 to 12 otherwise go to the next step.

14) Prescribe another load increment η_{n+1} and repeat steps 1 to 13 until all load increments are applied or no solution could be found within the specified tolerance. The second case is generally the indication of exceeding the limit point.

4.4.2 Drifting From the Yield Surface

An element node that is known to plastify has to remain on the yield surface when no unloading takes place. During the iteration process, this requirement can be violated in two cases. The first case is when nodal forces cross the yield surface. This signifies the formation of a plastic hinge and techniques for correcting this discrepancy are presented in Sec. 4.4.2.1. The second case takes place when the plastic reduction matrix constrains the nodal forces at a plastic hinge to move along the tangent plane to the yield surface. In principle, these nodal forces drift away from the yield surface and the yield criterion is violated. Techniques for correcting this drift are presented in Sec. 4.4.2.2.

4.4.2.1 Correction of Drift Crossing the Yield Surface

During the iteration process, nodal forces of one or more elements may reach the yield surface at a fraction of the applied load increment. This signifies the formation of a plastic hinge during that load increment. The plastic hinge formation at a fraction of applied load increment is shown in Fig. 4.5 in a simplified representative (α-β) stress resultant space. Points 1 and 2 depict the forces of a given node before and after the application of load increment assuming elastic loading, respectively. The goal here is to find the fraction ρ of the load increment that meets the yield condition. In Fig. 4.5 this point is denoted A.

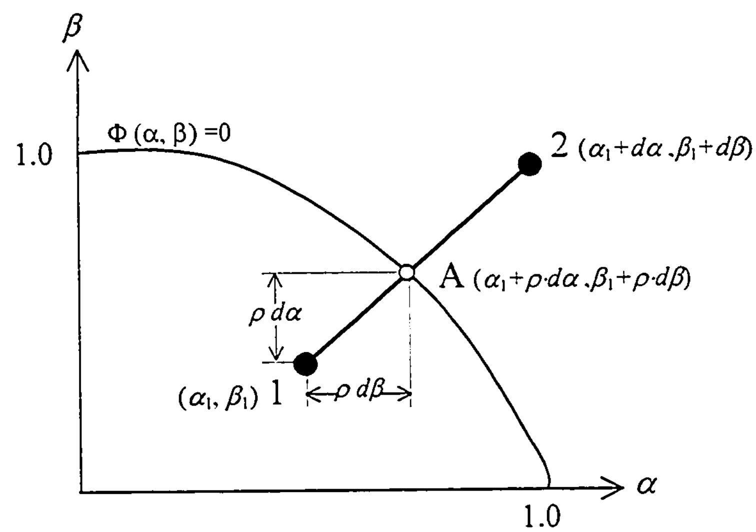

Fig. 4.5 Plastic Hinge Formation at a Fraction of the Load Increment

By knowing that $\Phi(\alpha_1, \beta_1) < 0$ and $\Phi(\alpha_1 + d\alpha, \beta_1 + d\beta) > 0$, it is required to find ratio ρ, $0 \le \rho \le 1.0$ such that $\Phi(\alpha_1 + \rho d\alpha, \beta_1 + \rho d\beta) = 0$. The secant root finding method (Fig. 4.6) is utilized in the program. Using linear interpolation, a better estimation ρ_r of sought root ρ is

$$\rho_r = \rho_u - \frac{[\Phi(\alpha, \beta)|_{\rho_u} - 1](\rho_l - \rho_u)}{\Phi(\alpha, \beta)|_{\rho_l} - \Phi(\alpha, \beta)|_{\rho_u}} \tag{4.7}$$

In Eq. 4.7, ρ_l is the lower limit of the root domain and ρ_u is the upper limit of the root domain. The initial guess values are taken as $\rho_l = 0.0$ and $\rho_u = 1.0$ corresponding to points 1 and 2, respectively. The ratio ρ_r is an improved guess of the sought value ρ and can be used to narrow down the domain of the solution. Iterations are performed until the solution is found within a specified tolerance. By locating point A on the yield surface, the tangent stiffness matrix can be calculated as described in Sec. 3.6.3.

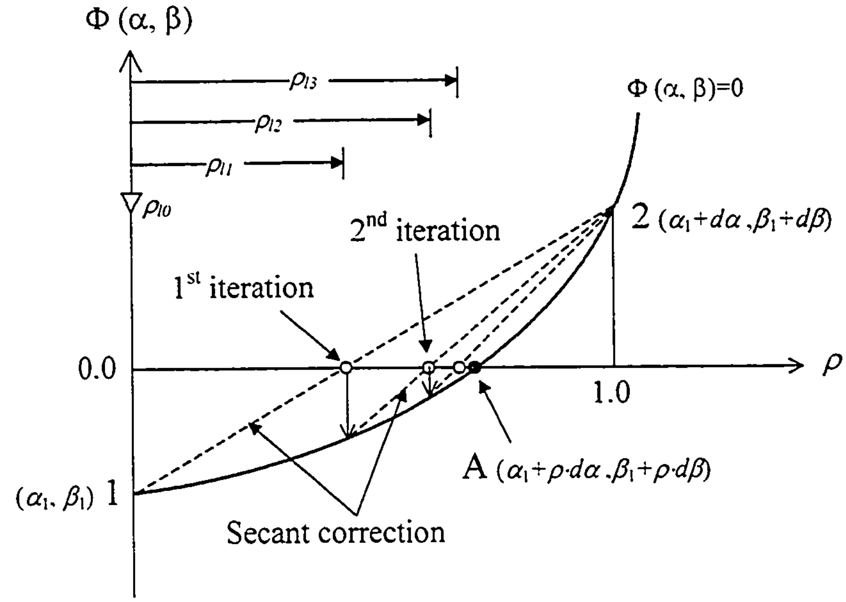

Fig. 4.6 Secant Root Finding Method

4.4.2.2 . Correction of Drift Along the Tangent Plane

At a plastic hinge, the plastic reduction matrix ensures that the element incremental nodal forces are constrained to move along the plane tangent to the yield surface (Point A in Fig. 4.7). As a result, these forces will drift from the yield surface. Therefore, the yield criterion will be violated. The element stiffness is formulated based on the gradient vector $\{G_A\}$ at point A. The applied load increment results in a state of internal forces at B, which violates the yield condition. If the stiffness matrix for the next increment is formulated at B (instead of D) with the corresponding gradient vector $\{G_B\}$, the force-point moves further to point C. This drift from the yield surface is inconsistence with the assumption of an elastic-perfectly plastic material.

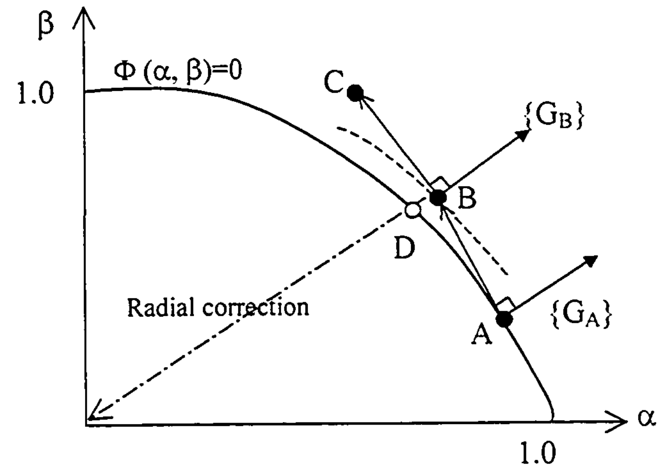

Fig. 4.7 Force-Point Drift from Yield Surface and Radial Correction

Among the proposed approaches for preventing force-point drift, the radial return correction (Fig. 4.7) has been chosen in this research. In this method, the intersection point D of the yield surface and a line passing through the drifted point B and the origin is taken to correct the drift. For a highly nonlinear system, it is necessary to do this correction in several steps as shown in Fig. 4.8. As can be seen, there is a difference, albeit small, between the result of a multiple step correction (point D), and one-step correction (point C). This error is very small for a perfectly plastic material and is judged insignificant in this research. Therefore, a radial return in a single step is adopted.

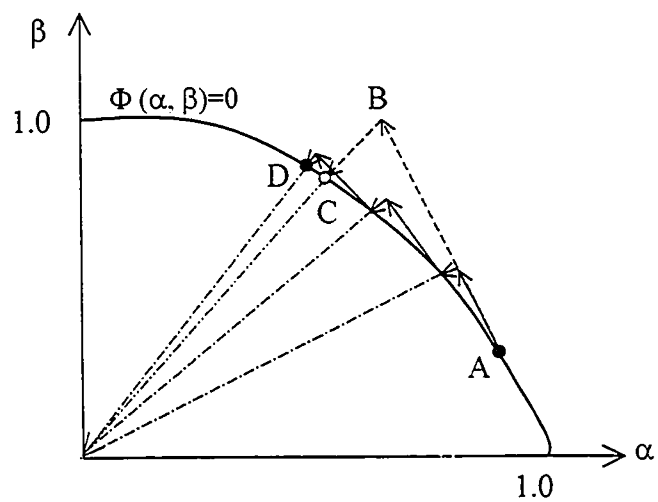

Fig. 4.8 Radial correction in a single step and multiple steps

4.4.3 Convergence Criteria

Efficient convergence criteria are another important part of an iterative scheme. These criteria govern the stop point of iterations in an increment. Convergence criteria must be defined such that they locate a point on the equilibrium path within an acceptable tolerance. Conservative criteria can cause more calculations without a significant improvement in the accuracy of the results. On the other hand, liberal criteria may cause large errors, or even divergence of the solution. It is important to strike a balance between the computation time and the attained accuracy by defining efficient convergence criteria.

One of the suggested norms for testing convergence is the "Modified absolute norm". This norm was originally introduced by Bergan and Clough (1972) based on displacements.

$$e = \frac{1}{N}\sum_{k=1}^{N}\left|\frac{d\Delta_k}{\Delta_{ref}}\right| \tag{4.8}$$

in which e is the calculated error, N is the total number of unknown displacement components, $d\Delta_k$ is the kth element of the incremental displacement vector $\{d\Delta_n^i\}$, and depending on the units of Δ_{ref} is taken as either the largest components of translation or rotation within the total displacement vector $\{\Delta_n^i\}$.

The norm used in this research is a similar one although based on forces.

$$e = \frac{1}{N}\sum_{k=1}^{N}\left|\frac{R_k}{R_{ref}}\right| \tag{4.9}$$

In Eq. 4.9 R_k is the kth element of the unbalanced force vector $\{R_n^i\}$ and R_{ref} is the largest component of the unbalanced force vector.

A convergence criterion of $e \leq \xi$ can be used where the user-defined tolerance ξ is usually of the order of 10^{-2} to 10^{-6} depending on the desired accuracy and the nature of the problem.

4.5 Displacement Control Scheme

The load control method fails to trace the response of a structure in the neighbourhood of the limit point. Increasing the number of iterations and using smaller load steps yield better estimations for this point. In practice, this approach causes singularity at the limit point and divergence may occur. Further, load control cannot capture the post-peak behaviour of the structure.

Displacement control is an effective technique that avoids singularity near the limit point. In this method, a single displacement component is selected as a controlling parameter and its value is prescribed. The corresponding load level η_n is taken as unknown. Unlike the load control technique, the load level is not constant during iterations and is adjusted in the iteration process in order to maintain constant the prescribed control displacement. Hence, Eq. 4.3 must be modified as

$$\left[K_n^{i-1}\right]\left\{d\Delta_n^i\right\}=\left\{R_n^{i-1}\right\}+d\eta_n^i\{P\} \tag{4.10}$$

in which $d\eta_n^i$ is the increment of load incremental factor at the ith iteration of nth increment.

The following procedure is adapted from that suggested by Ramm (1980). Bathe (1983) added some improvements to the technique. For the sake of simplicity, it is assumed that the stiffness expression is such that the prescribed displacement component $d\Delta_{n2}^i=\delta$ is the last member in the nodal displacement vector. Writing Eq. 4.10 in a partitioned form yields

$$\begin{bmatrix}\left[K_{n11}^{i-1}\right] & \left\{K_{n12}^{i-1}\right\}\\ \left\langle K_{n21}^{i-1}\right\rangle & K_{n22}^{i-1}\end{bmatrix}\begin{Bmatrix}\left\{d\Delta_{n1}^i\right\}\\ \delta\end{Bmatrix}=\begin{Bmatrix}\left\{R_{n1}^{i-1}\right\}\\ R_{n2}^{i-1}\end{Bmatrix}+d\eta_n^i\begin{Bmatrix}\{P_1\}\\ P_2\end{Bmatrix} \tag{4.11}$$

By interchanging the variables, one obtains

$$\begin{bmatrix} \left[K_{n11}^{i-1}\right] & -\{P_1\} \\ \left\langle K_{n21}^{i-1}\right\rangle & -P_2 \end{bmatrix} \begin{Bmatrix} \{d\Delta_{n1}^{i}\} \\ d\eta_n^{i} \end{Bmatrix} = \begin{Bmatrix} \{R_{n1}^{i-1}\} \\ R_{n2}^{i-1} \end{Bmatrix} - \delta \begin{Bmatrix} \{K_{n12}^{i-1}\} \\ K_{n22}^{i-1} \end{Bmatrix} \tag{4.12}$$

In principle, the system of Eq. 4.12 can be solved for the unknown vector $\left\langle \left\langle d\Delta_{n1}^{i}\right\rangle^T \; d\eta_n^{i}\right\rangle^T$. However, the loss of the symmetry of the matrix on the right hand side appears to be a severe weakness in the procedure. This problem is circumvented as follows. Equation 4.11 can be expressed in the alternative form

$$\left[K_{n11}^{i-1}\right] \{d\Delta_{n1}^{i}\} = d\eta_n^{i} \{P_1\} + \left(\{R_{n1}^{i-1}\} - \{K_{n12}^{i-1}\} \cdot \delta\right) \tag{4.13}$$

$$\left\langle K_{n21}^{i-1}\right\rangle \{d\Delta_{n1}^{i}\} = d\eta_n^{i} P_2 + R_{n2}^{i-1} - K_{n22}^{i-1} \cdot \delta \tag{4.14}$$

Equation 4.13 is linear in the load parameter $d\eta_n^{i}$. Therefore, the displacement increment vector $\{d\Delta_n^{i}\}$ can be decomposed into two parts, corresponding to each of the two terms of the right hand side of Eq. 4.13, as

$$\{d\Delta_{n1}^{i}\} = d\eta_n^{i} \{d\bar{\Delta}_{n1}^{i}\} + \{d\bar{\bar{\Delta}}_{n1}^{i}\} \tag{4.15}$$

in which $\{d\bar{\Delta}_{n1}^{i}\}$ is the nodal incremental displacement due to the applied load and $\{d\bar{\bar{\Delta}}_{n1}^{i}\}$ is the nodal incremental displacement due to unbalanced load.

From Eqs. 4.13 and 4.15, the two components of the displacement vectors can be obtained as

$$\left[K_{n11}^{i-1}\right] \{d\bar{\Delta}_{n1}^{i}\} = \{P_1\} \tag{4.16}$$

$$\left[K_{n11}^{i-1}\right] \{d\bar{\bar{\Delta}}_{n1}^{i}\} = \{R_{n1}^{i-1}\} - \{K_{n12}^{i-1}\} \cdot \delta \tag{4.17}$$

Substituting $\{d\Delta'_n\}$ from Eq. 4.15 into Eq. 4.14 and solving for the load parameter $d\eta_n^i$ yields

$$d\eta_n^i = \frac{K_{n22}^i\,\delta + \left\langle K_{n21}^{i-1}\right\rangle \left\{d\bar{\bar{\Delta}}_{n1}^i\right\} - R_{n2}^{i-1}}{P_2 - \left\{K_{n21}^{i-1}\right\} \left\{d\bar{\Delta}_{n1}^i\right\}} \tag{4.18}$$

Equations 4.15 to 4.18 are the core of the displacement control method iteration. The value of δ is prescribed at the first iteration of an increment. It will be kept zero during the next iterations until the next increment. Therefore, after the first iteration Eq. 4.17 is changed into

$$\left[K_{n11}^{i-1}\right] \left\{d\bar{\bar{\Delta}}_{n1}^i\right\} = \left\{R_{n1}^{i-1}\right\} \tag{4.19}$$

Moreover, Eq. 4.18 is changed into

$$d\eta_n^i = \frac{\left\langle K_{n21}^{i-1}\right\rangle \left\{d\bar{\bar{\Delta}}_{n1}^i\right\} - R_{n2}^{i-1}}{P_2 - \left\{K_{n21}^{i-1}\right\} \left\{d\bar{\Delta}_{n1}^i\right\}} \tag{4.20}$$

For the displacement control method to be successful, it is essential to choose a control DOF that increases monotonically with the applied loads. This is possible in most practical problems.

In Eq. 4.11, it was assumed that the controlling displacement parameter is the last member of the nodal displacement vector. In general, this is not always the case. Therefore, it is needed to extend the scheme into a more general case as follows.

Assume the controlling displacement parameter of a system with m DOF is the pth DOF, where $p \leq m$. The stiffness matrix of the system is an $m \times m$ matrix and can be partitioned as

$$[K]_{m\times m} = \begin{bmatrix} [a]_{(p-1)\times(p-1)} & \{b\}_{(p-1)\times 1} & [c]_{(p-1)\times(m-p)} \\ \langle b\rangle^T_{1\times(p-1)} & e & \langle d\rangle^T_{1\times(m-p)} \\ [c]^T_{(m-p)\times(p-1)} & \{d\}_{(m-p)\times 1} & [f]_{(m-p)\times(m-p)} \end{bmatrix}_{m\times m} \tag{4.21}$$

The stiffness matrix sub-matrices in Eq. 4.11 are $\left[K^{t-1}_{n11}\right]$, $\left\{K^{t-1}_{n12}\right\}$, $\left\langle K^{t-1}_{n21}\right\rangle$ and K^{t-1}_{n22}. By using Eq. 4.21, sub matrix $\left[K^{t-1}_{n11}\right]$ is defined as

$$\left[K^{t-1}_{n11}\right] = \begin{bmatrix} [a]_{(p-1)\times(p-1)} & [c]_{(p-1)\times(m-p)} \\ [c]^T_{(p-1)\times(m-p)} & [f]_{(m-p)\times(m-p)} \end{bmatrix}_{(m-1)\times(m-1)} \tag{4.22}$$

Sub matrices $\left\{K^{t-1}_{n12}\right\}$ and $\left\langle K^{t-1}_{n21}\right\rangle$ are defined as

$$\left\{K^{t-1}_{n12}\right\} = \left\langle K^{t-1}_{n21}\right\rangle^T = \begin{Bmatrix} \{b\}_{(p-1)\times 1} \\ \{d\}_{(m-p)\times 1} \end{Bmatrix}_{(m-1)\times 1} = \left\langle \langle b\rangle^T_{1\times(p-1)} \quad \langle d\rangle^T_{1\times(m-p)} \right\rangle_{1\times(m-1)} \tag{4.23}$$

Moreover, the last sub-matrix K^{t-1}_{n22}, which is actually a single value, is defined as

$$K^{t-1}_{n22} = e \tag{4.24}$$

By using the same method, the reduced vector of applied load $\{P_1\}$ is equal to applied load vector $\{P\}$ without the element corresponding to the controlling parameter, while P_2 is equal to that member of $\{P\}$ corresponding to the controlling parameter. A similar argument can be made for the unbalanced force and displacement vectors.

The above definitions are exactly equivalent to re-ordering the stiffness matrix, and corresponding load and displacement vectors so that the controlling displacement component becomes the last member in the nodal displacement vector.

The main idea of this method is schematically represented in Fig. 4.9. In this figure, point 1 on the response path is assumed to be known and the goal is to find point 2. The iteration number is shown by a circled value.

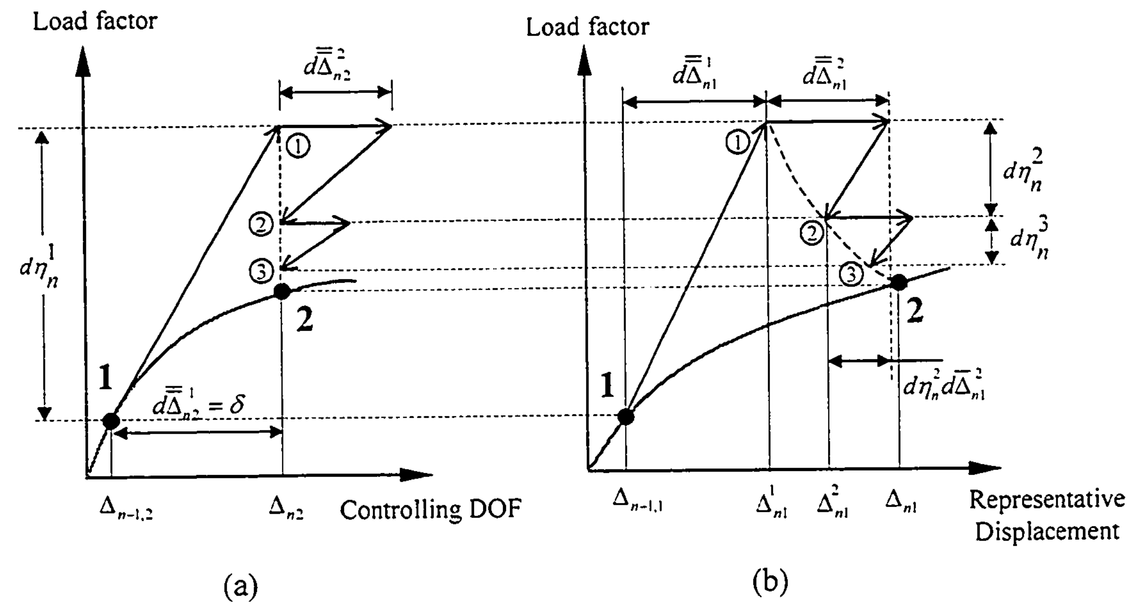

Fig. 4.9 Displacement Control Scheme

(a) Load vs. Controlling DOF; (b) Load vs. a representative DOF

As illustrated in Fig. 4.9a, during the first iteration the control displacement is incremented by a user specified amount δ. The control displacement parameter is kept zero throughout subsequent iterations while the corresponding load level $d\eta_n^i$ and other displacement components (Fig. 4.9b) are updated through an iterative procedure. The iteration will stop when the equilibrium conditions have been met (point 2). This point is the starting point of next displacement increment to the controlling DOF.

The process for implementing the displacement control method follows. A subscript refers to the increment and a superscript refers to the iteration within that increment. Vectors and matrices are those for the whole structure and are referred to the global coordinate system.

1) Form the reduced applied load vector $\{P_1\}$ from $\{P\}$ by removing P_2, the member corresponding to the control DOF.
2) Set the control incremental displacement parameter $d\Delta^i_{n2}$ to the prescribed value δ.
3) Set iteration number i to zero.
4) Set the initial stiffness matrix for the current increment $\left[K^0_n\right]$ equal to that of the final iteration of the previous increment $\left[K_{n-1}\right]$.
5) Set the initial unbalanced load $\left\{R^0_n\right\}$ to zero.
6) Impose boundary conditions.
7) Partition the stiffness matrix into $\left[K^i_{n11}\right]$, $\left\{K^i_{n12}\right\}$, $\left\langle K^i_{n21}\right\rangle$, K^i_{n22} according to Eqs. 4.21 to 4.24.
8) Form the reduced unbalanced force vector $\left\{R^i_{n1}\right\}$ from $\left\{R^i_n\right\}$ by removing element R^i_{n2}, the member corresponding to the control DOF.
9) Increment the iteration number.
10) Use Eq. 4.16 to calculate $\left\{d\bar{\Delta}^i_{n1}\right\}$. When $i=1$ use Eq. 4.17, otherwise use Eq. 4.19 to calculate $\left\{d\bar{\bar{\Delta}}^i_{n1}\right\}$.
11) Calculate incremental load parameter $d\eta^i_n$, if $i=1$ by using Eq. 4.18 otherwise by using Eq. 4.20. Calculate $\eta^i_n = \eta^{i-1}_n + d\eta^i_n$.
12) Calculate the reduced incremental displacement vector $\left\{d\Delta^i_{n1}\right\}$ by using Eq. 4.15.
13) Merge $\left\{d\Delta^i_{n1}\right\}$ into $d\Delta^i_{n2}$ and form the incremental displacement vector $\left\{d\Delta^i_n\right\}$
14) Update vectors and matrices at the element level. This is identical to step 9 of the load control scheme.
15) Calculate the unbalanced load vector $\left\{R^i_n\right\}$ using

$$\left\{R^i_n\right\} = \eta^i_n\left\{P_n\right\} - \left\{F^i_n\right\} \tag{4.25}$$

16) Check convergence criterion (Sec.4.4.3). If it is not satisfied, repeat steps 6 to 15.

17) Repeat steps 1 to 16 until a user defined maximum value of the controlling DOF has been reached.

The above procedure has been implemented under Visual BASIC 6.0 environment (Appendix B).

A numeric example comparing the implementation of displacement control and load control methods is provided in appendix A2.

Chapter 5

Verification Problems

5.1 General

In this chapter, simple examples are presented to assess/demonstrate the performance of the formulation and implementation presented in chapters 3, 4 and appendix B. These examples are sufficiently small so that in most cases they can either be solved or verified by hand calculations.

Analysis procedures available in ABAQUS™ /Standard (Hibbit et. al 2001) were used to compare the results. Suitable elements of this large software have been selected in order to assess the validity and efficiency of the proposed method. A brief description of the theory behind these elements is provided in the following section.

5.2 Finite Elements in ABAQUS

The performance of the element derived in chapter 3 is compared to that of four elements from the ABAQUS library (Hibbit et.al 2001). Each of them has some advantages and some limitations that will be described. These elements are

1) B33: Two node three-dimensional beam with cubic Hermitian interpolation functions.
2) PIPE31: Two-node three-dimensional pipe beam element including shear deformations with independent linear interpolation for the displacement and rotation fields.
3) FRAME3D: Two-node three-dimensional frame element with two potentially plastic hinges at the end nodes.

4) ELBOW31: Two-node three-dimensional pipe including plane distortion modeling capability, with independent linear interpolation of the displacement and rotation fields.

The first two elements are categorized as beam elements. In ABAQUS, a 3D beam element is a one-dimensional line element arbitrarily oriented in a three-dimensional space with a stiffness associated with the deformation of the beam axis. These deformations consist of axial stretch, biaxial curvature change (bending) and torsion. The main advantage of beam elements is that they are geometrically simple to describe and have only a few degrees of freedom. This simplicity is achieved by assuming that the member deformation can be described entirely by displacements that are functions of the position of the section along the beam axis only.

5.2.1 B33 Beam Element

B33 is a general-purpose three-dimensional Euler-Bernoulli beam with two nodes. Longitudinal integration is performed at two Gauss points along the member centreline. The beam interpolation functions for transverse displacement are cubic. It has six degrees of freedom for each node, three displacements and three rotations. In the derivation of this element it is assumed that plane sections initially normal to the beam's axis remain plane and normal to the deformed beam axis. Therefore, the formulation does not incorporate transverse shear deformation effects. A number of cross-sections can be assigned to B33 element in ABAQUS. PIPE section (Sec. 5.2.2) is assigned to this element. The constitutive model used is elastic perfectly plastic (E = 200,000 MPa, ν = 0.3, Fy = 350 MPa). The element can model the spread of plasticity along its centreline when a relatively large numbers of element are used.

5.2.2 PIPE Cross-Section

In order to model a pipe, it is needed to assign the PIPE section (Fig. 5.1) to B33 element. PIPE section is automatically assigned to PIPE31 and ELBOW31 elements too. In this figure, the default integration points are shown for a beam in space. The number of integration points can be increased in order to improve the results. For an ELBOW31 element, the parametric runs in this study indicate that at least twenty integration points around the section and three points across the pipe wall should be defined.

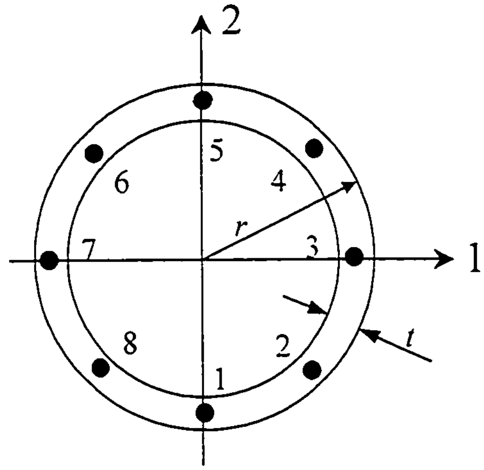

Fig. 5.1 Default Integration Points in ABAQUS for a 3D Pipe Cross-Section

5.2.3 PIPE31 Beam Element

PIPE31 is a Timoshenko hollow circular beam section. This element uses one Gauss point in the longitudinal direction and is therefore able to integrate linear functions exactly. As a result, it needs a finer mesh in modeling a given problem when compared to the B33 element. It includes three displacements and three rotations at each node. In the elastic stage, this element offers additional flexibility associated with transverse shear deformation between the beam centreline and the initial normal to the beam centreline in the elastic range. In the plastic range, numerical experimentations indicated that the effect of shearing stresses due to shearing forces and/or torsion do not affect the magnitude of effective stresses. The default value of the reduced area A_r (Eq. 3.2) is 0.53, which corresponds to a Poisson's ratio of 0.3. The hoop stress caused by internal or external pressure is modelled in this element. PIPE section (Sec. 5.2.2) is internally assigned to this element. The element can capture the spread of plasticity when a finer mesh is used.

5.2.4 FRAME3D Element

FRAME3D provides efficient modeling for design calculations for frame-like structures composed of initially straight slender members. Externally, this element has twelve DOF. FRAME3D formulation includes three additional variables relating to the axial and lateral displacements. These variables are defined at an automatically created node at the middle of the element and the effects of them are taken into account by using static condensation. Plasticity effects are modeled by using nodal internal forces at two potentially plastic hinges lumped at nodes. The element can thus be used for load predictions based on the formation of a mechanism through the formation of plastic hinges.

The formulation of this element is close to what has been introduced in this research with three major differences. These are:

1) FRAME3D is able to approximately account for kinematic hardening. In contrast, the element developed in this research is restricted to elastic perfectly plastic idealization. In principle, a more general formulation that accounts for kinematic hardening could be implemented if desired.
2) FRAME3D does not model shear effects on the yield surface. (i.e. shear stresses do not influence the plastic behaviour of the pipe). This drawback is remedied in this research.
3) The element uses an approximate yield surface in characterizing the behaviour of 3D plastic hinges. By neglecting strain-hardening parameters, the FRAME3D yield surface is

$$\left(\frac{M_1}{M_p}\right)^2+\left(\frac{M_2}{M_p}\right)^2+\left(\frac{T}{T_p}\right)^2+\left(\frac{A}{A_p}\right)^2=1.0 \tag{5.1}$$

Equation 5.1 shows the absence of shear terms in the yield surface function, which will be shown to cause significant errors for spans with high shears (Example 4). In addition, this function yields liberal results for cases including high axial force (Example 2). The element is implicitly assigned a PIPE cross-section (Sec. 5.2.2). It is necessary to input the plastic moment resistance M_p, torsion resistance T_p and axial load resistance A_p. The element yields good results for stocky sections (e.g. D/t < 25). Because of the use of lumped plastic hinges, FRAME3D cannot model the spread of plasticity and cross-section distortions.

5.2.5 ELBOW31 Element

A fundamental assumption adopted in beam element formulation is that the section cannot deform in its own plane. In reality, section collapse may occur and result in weaker response than that predicted by beam theory. Thin-walled pipes typically exhibit much softer bending behaviour than would be predicted by beam theory because the pipe wall readily distorts in the plane of cross-section. This effect, which must generally be considered when designing thin walled pipes (e.g D/t > 60), can be modeled by using a very large number of shell elements or alternatively a smaller number of ELBOW elements. Internal and/or external pressure effects are modeled trough the inclusion of hoop stresses in the formulation. The constitutive model used is very flexible and it can be assigned any type of material behaviour available in the ABAQUS library. The material model used in this research is elastic perfectly plastic for the results to be consistent with those of other elements.

Although ELBOW31 appears to possess only six degrees of freedom at its two nodes, internally it has five DOF at all of its displacement field models. Cross-sectional distortion is represented using up to six Fourier terms to describe the radial and circumferential displacements. Longitudinally, it uses linear interpolation. Hence, it has one Gauss point along the beam.

The PIPE cross-section (Sec. 5.2.2) is internally assigned to this element. Proper modeling of ELBOW31 element necessitates using a small element length and a large number of integration points around the cross-section.

5.3 *Verification Problems*

The finite element developed in chapter 3 will be assigned the identifier P3D2HE, in which P stands for a pipe; 3D denotes a three-dimensional formulation; 2H stands for the two potentially plastic hinges at both nodes and E denotes that the exact yield hyper-surface is used.

In this section, all problems are solved for one cross-section with different loading, spans and support conditions. The section selected is DN90 STD with an outside diameter of 101.6 mm and thickness of 5.74 mm (D/t = 17.7). The section has the following properties, the area A= 1730 mm^2, the moment of inertia I= 1.99×10^6 mm^4, the St.Venant torsional constant J= 3.99×10^6 mm^4. Material properties are the yield stress σ_y = 350 MPa, Poisson's ratio ν= 0.3 and Young modulus E= 200,000 MPa. For the cross-section chosen, the following quantities of interest are calculated.

$$r = (101.6 - 5.74)/2 = 47.93\,mm \qquad p_p = t\sigma_y/(r - t/2) = 44.6\,MPa$$

$$A_p = 2\pi r t\,\sigma_y = 605.0\,KN \qquad V_p = 4rt(\sigma_y/\sqrt{3}) = 222.4\,KN$$

$$M_p = 4r^2 t\sigma_y = 18.5\,KN.m \qquad T_p = 2\pi r^2 t(\sigma_y/\sqrt{3}) = 16.7\,KN.m$$

5.3.1 Problem 1: Bending Moments

The purpose of this example is to assess the validity of the results of P3D2HE element for the pipes subject to bending moments. A large span of 6m is selected to retain a low shear ratio ($V/V_p \approx 0$) throughout the analysis. A vertical load P acting at 4.0 m from the left support (Fig. 5.2) is applied. In this example, a mesh sensitivity analysis is performed in order to determine discretization at which convergence is achieved for each type of ABAQUS element used. The beam is fixed at both ends.

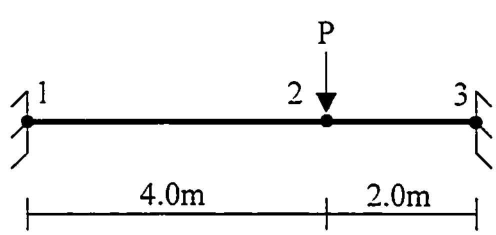

Fig. 5.2 Sketch of Problem 1

The shear ratio is very small ($V/V_p \approx 0$). Therefore, in hand calculation checks shear effects are neglected without introducing any large errors

The hand calculation results for the problem are shown in Fig. 5.3. The first plastic hinge occurs at node 3 when the applied load P reaches $P_1 = 18.5/0.889 = 20.8\,KN$. At this load, the moment at nodes 1 and 2 will be $0.444 \times 20.8 = 9.23\,KNm$ and $0.592 \times 20.8 = 12.3\,KNm$, respectively.

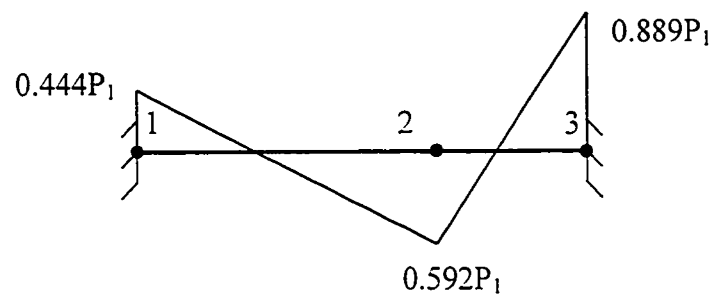

Fig. 5.3 Problem 1: No plastification

After plastification of node 3, the beam acts as a fixed-pinned (Fig. 5.4). The second plastic hinge occurs at node 2 when an additional amount of applied load dP_1 reaches the value of $dP_1 = (18.5 - 12.3)/1.11 = 5.6\ KN$. At this stage, the applied load P_2 is $P_1 + dP_1 = 26.4\,KN$, and the corresponding bending moments at node 1 are $9.23 + 0.592 \times 5.6 = 12.5\ KNm$.

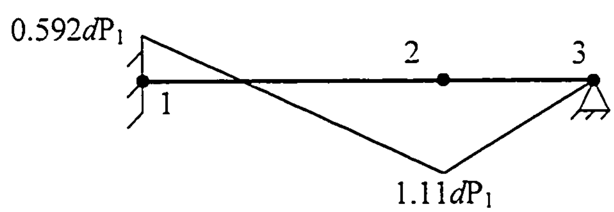

Fig. 5.4 Problem 1: Node 3 Plastified

After plastification of both nodes 2 and 3 occurs, the beam acts as a cantilever (Fig. 5.5). When the additional load dP_2 reaches $(18.5 - 12.5)/4.0 = 1.5\,KN$, node 1 plastifies and a failure mechanism takes place. At this stage, total load $P_3 = P_2 + 1.5 = 27.9\,KN$.

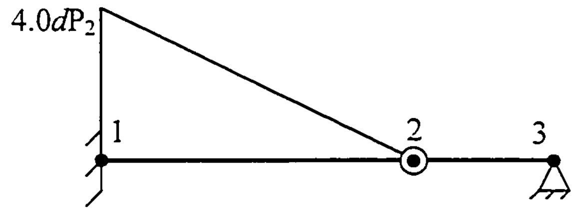

Fig. 5.5 Problem 1: Nodes 2 and 3 plastified

The above three load levels are presented in Fig. 5.6. On the same figure, the applied load P vs. vertical deformation of node 2 is provided. Twenty-four B33 elements predict nearly the same load P, as that based on proposed element (P3D2HE) with only two elements. In this analysis, sixteen integration points around the cross-section are used. The B33 element can capture the gradual yield, while P3D2HE exhibits two kinks corresponding to sudden plastic hinge formation at node 3 and node 2 before forming a mechanism through the plastification of node 1. As expected, the slope of the load deformation relationship of all analysis agrees very well in elastic range. The differences in response become apparent after the formation of the first plastic hinge (P_1= 20.8 KN).

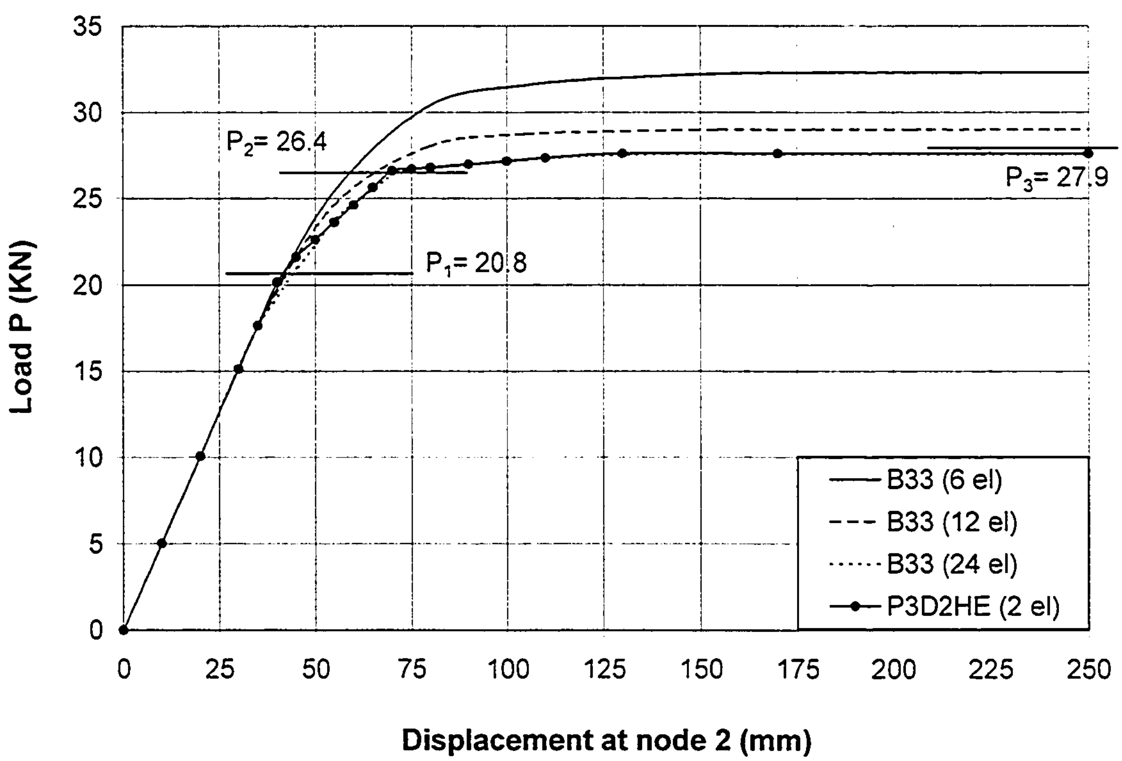

Fig. 5.6 Comparison of Results with ABAQUS B33 for Problem 1

A similar comparison with PIPE31 elements is shown in Fig. 5.7. Sixteen integration points around the cross-section were used. The necessity of forty-eight PIPE31 elements for the solution to converge to the results predicted by P3D2HE is obvious.

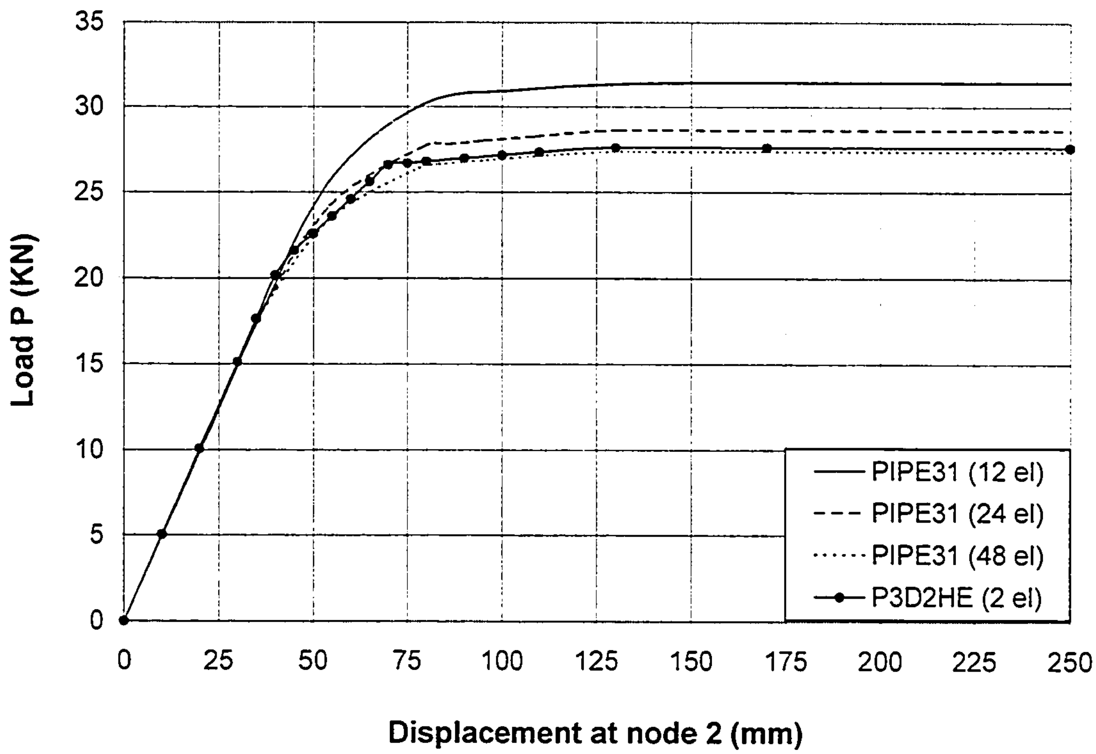

Fig. 5.7 Comparison of Results with ABAQUS PIPE31 for Problem 1

A similar comparison with FRAME3D using two elements is presented in Fig. 5.8. Very close agreement is obtained because the shear ratio (V/V_p) is too small to cause any significant effects. Contrary to the element developed in this study, shear deformation effects are not modelled in FRAME3D. For the problem investigated here, as the ratio of bending to shear rigidity κ (Section 3.5) is negligible, no appreciable difference between the two solutions is noticed. This is not the case for a short span beam (Section 5.3.4). Load level values corresponding to the formation of plastic hinge as calculated by hand almost exactly match those based on FRAME3D element.

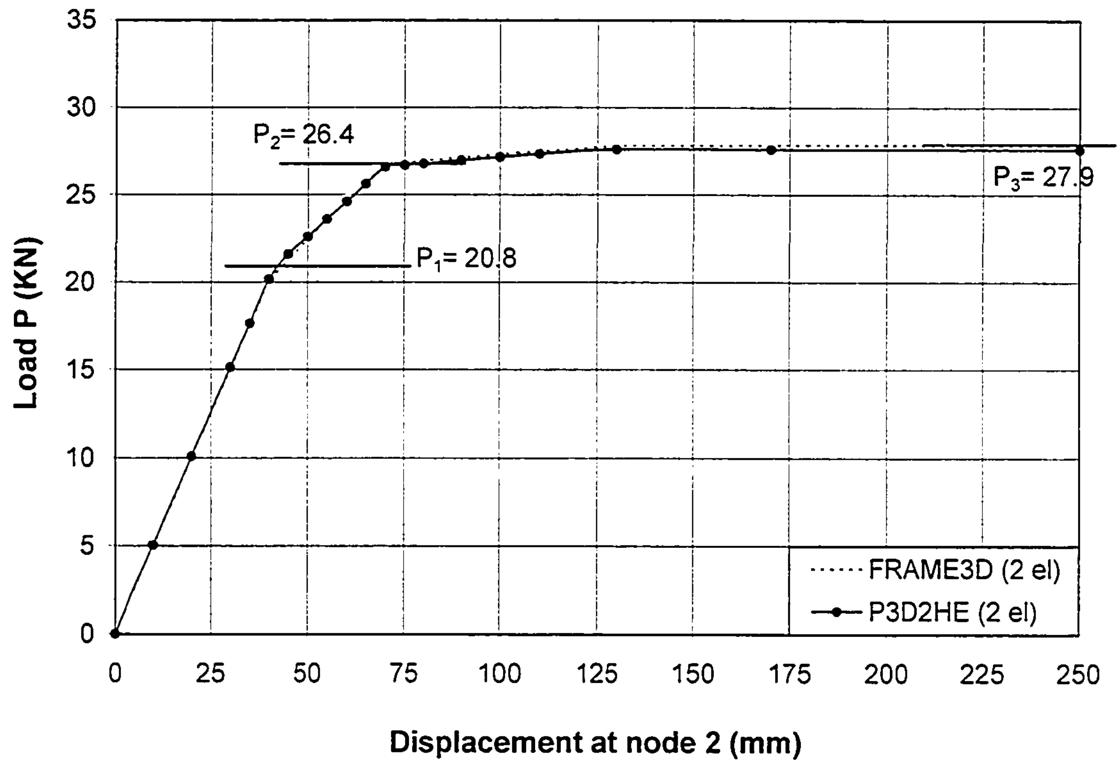

Fig. 5.8 Comparison of Results with ABAQUS FRAME3D for Problem 1

Another comparison between P3D2HE and ELBOW31, with six Fourier series terms to represent the cross-section displacements, twenty integration points around the pipe section and five integration points across the pipe wall, is provided in Fig. 5.9. With ELBOW31, it was necessary to subdivide the beam into 120 elements for the results to converge to the solution. In Fig. 5.9, the dashed line (denoted as ElbowM (120 el)) gives the applied load versus the transverse displacement at node 2 when the analysis included only material nonlinear effects. On the same figure, are the results based on 60 and 180 elements for comparison. Excellent agreement is observed between the ABAQUS solution and that based of the element P3D2HE developed in this study.

On the same plot, the result of an analysis including both material and geometric non-linearity is provided. The spread of plasticity along the member length as well as the cross-sectional distortion are modelled in this analysis. The response shows a descending branch characteristic of pipe behaviour (Mohareb et. al. 2001) after the peak load value.

This type of behaviour cannot be modelled by any of the other elements. Therefore, for short segments of pipe, Elbow elements provide the most accurate results.

On an AMD Athlon™ 950 MHz computer with 516 MB RAM, the analysis based on 120 ELBOW31 elements including material nonlinearity took 65 seconds of running time while the analysis based on the proposed element took only 12 seconds. Adding an automatic control parameter algorithm can reduce the calculation time even further. For a larger problem, the difference in computing time is expected to grow significantly.

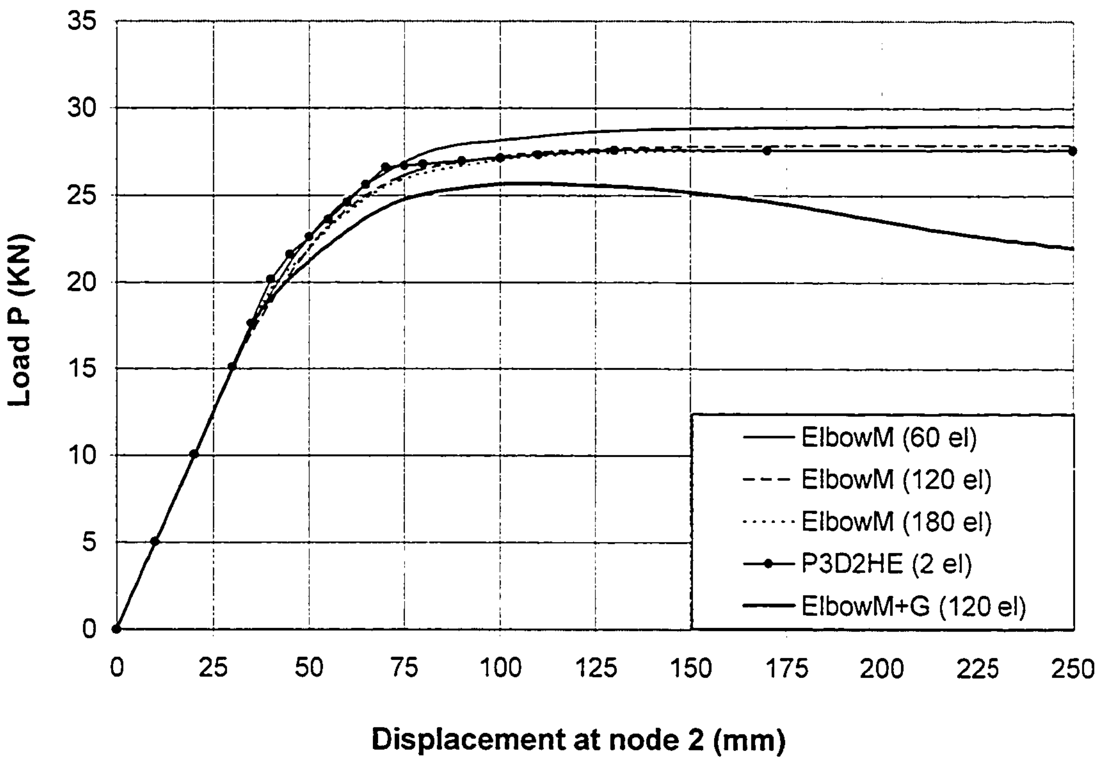

Fig. 5.9 Comparison of Results with ABAQUS ELBOW31 for Problem 1

For a large pipeline in the order of a few hundred meters or more, the use of Elbow elements is clearly impractical. In these cases, simplified methods such as the use of P3D2HE element would provide a more realistic alternative.

5.3.2 Problem 2: Axial Force and Bending Moments

The purpose of this example is to test the validity of the results for problems including combinations of bending moments and axial force. A 6m span pipe fixed at one end, and free to rotate at the other end is subject to a transverse load 0.02P and a horizontal load P acting at 4.0m from the left support (Fig. 5.10). Both loads are monotonically increased until failure is reached.

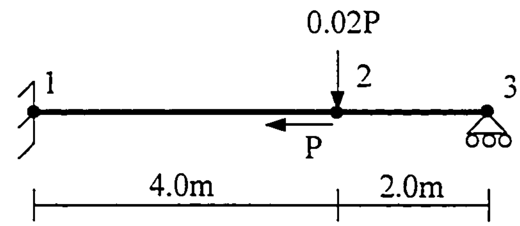

Fig. 5.10 Sketch of Problem 2

The applied load P vs. vertical displacement of node 2 is plotted in Fig. 5.11. In generating this plot, forty-eight PIPE31 elements, twenty-four B33 elements, one hundred twenty ELBOW31 elements and two FRAME3D elements were used. Finer meshes in all cases were found not to change the results.

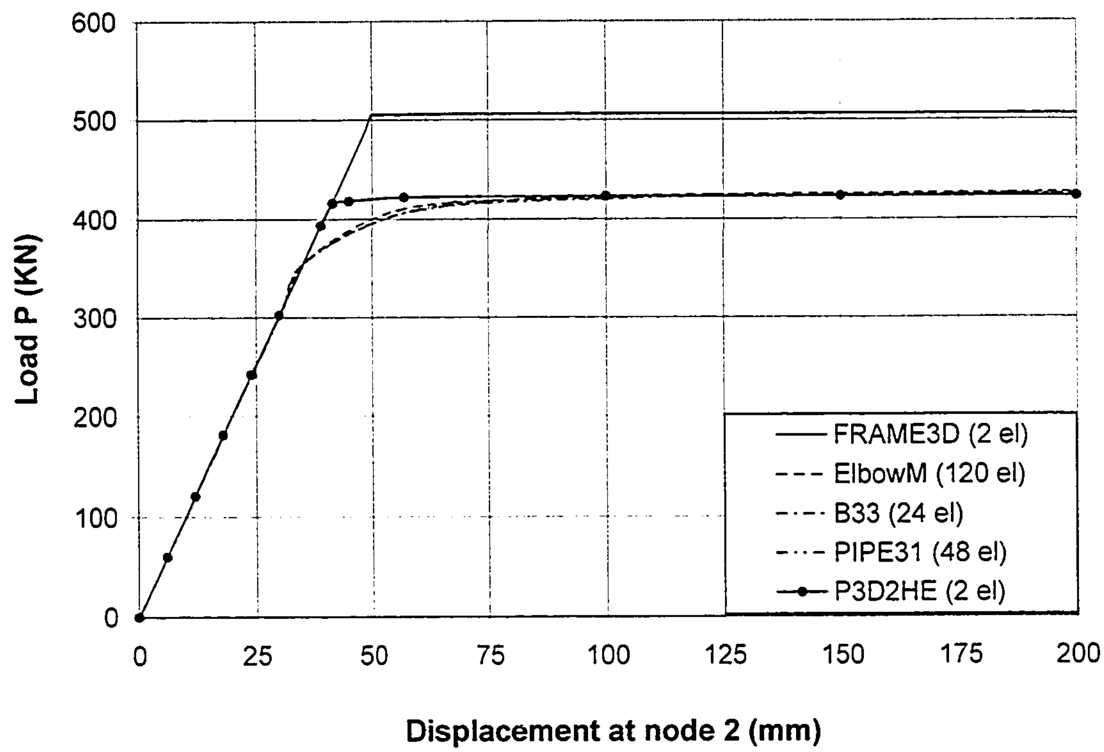

Fig. 5.11 Comparison of Results with ABAQUS for Problem 2

A few interesting points are noticed in Fig. 5.11. Firstly, very close agreement of ELBOW31, B33 and PIPI31 analysis results with those based on the proposed element P3D2HE is apparent. Secondly, there is some disagreement between the results of B33, PIPE31 and ELBOW31 elements with P3D2HE at the formation of the plastic hinge, which is due to the lumped plasticity assumption adopted in the formulation of P3D2HE. Since the gradual spread of plasticity cannot be captured by the proposed element, a multi-linear Load-Displacement relationship is exhibited while the other analyses demonstrate a smoother behaviour. Thirdly, in the case of FRAME3D element, the peak load P is significantly overestimated. This discrepancy is caused by the yield surface definition adopted in this element. For a FRAME3D element under combination of bending moments and axial force, the assumed yield surface adopted takes the form

$$\left(\frac{M}{M_p}\right)^2 + \left(\frac{A}{A_p}\right)^2 = 1.0 \tag{5.2}$$

while the pipe exact yield surface, used in P3D2HE formulation, is

$$\left(\frac{M}{M_p}\right)^2 - \cos^2\left(\frac{\pi}{2}\frac{A}{A_p}\right) = 0 \tag{5.3}$$

A comparison between the two interaction surfaces is provided in Fig. 5.12 for a DN90 STD section (recall $M_p = 18.5\,KNm$ and $A_p = 605\,KN$). By assuming an axial force $A = 410KN$ ($A/A_p = 0.678$), Eq. 5.2 and Eq. 5.3 result into a predicted moment $M = 13.6KNm$ and $M = 8.9\,KNm$, respectively. This means that the yield surface for a FRAME3D element overestimates the bending capacity by about 53% in this case.

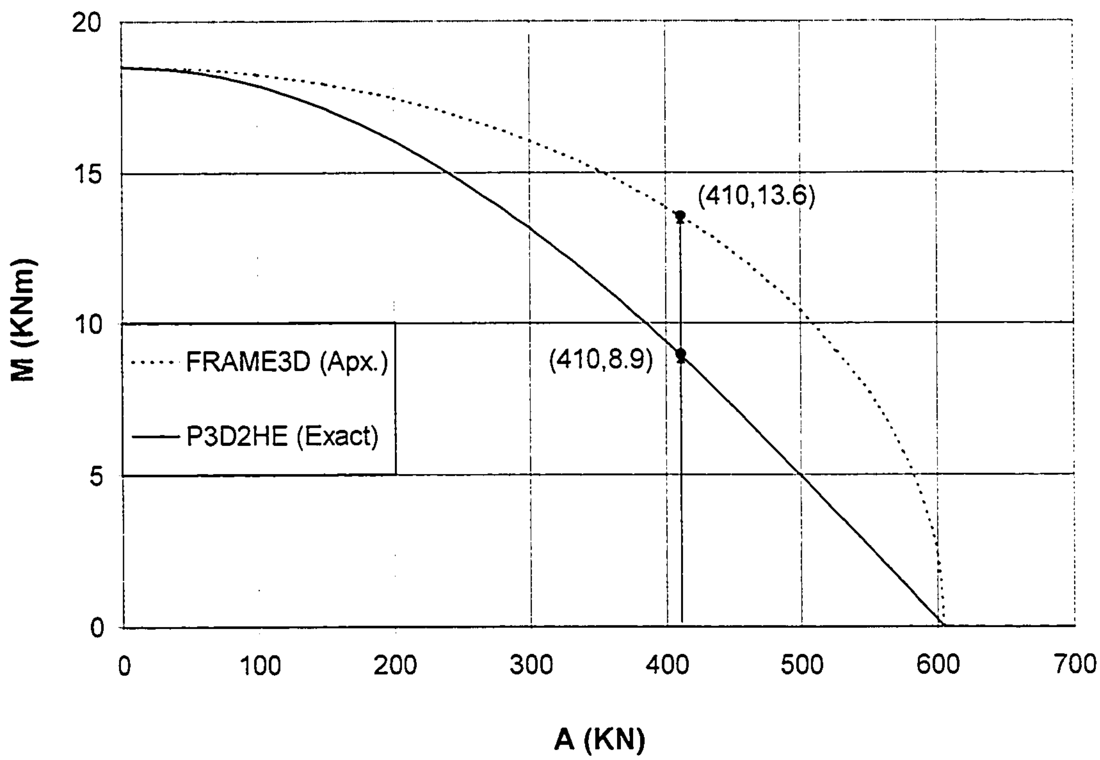

Fig. 5.12 ABAQUS FRAME3D and Exact Yield Surfaces for a DN90 STD Section

5.3.3 Problem 3: Torsion and Bending Moments

The purpose of this example is to assess the validity of the results of P3D2HE for combination of bending moments and torsion. A 6m span pipe fixed at both ends is subject to a vertical load P and a concentrated torque T=1000P (KNmm) acting at 4m from the left support (Fig. 5.13).

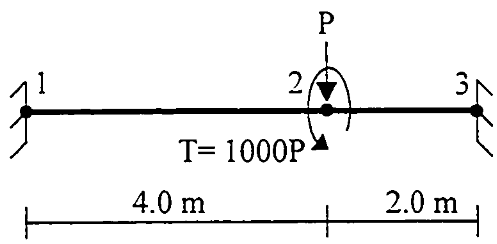

Fig. 5.13 Sketch of Problem 3

This beam can be analysed by hand before the formation of a plastic hinge. In this analysis the effects of shear deformations are neglected because the shear ratio (V/V_p) is negligible.

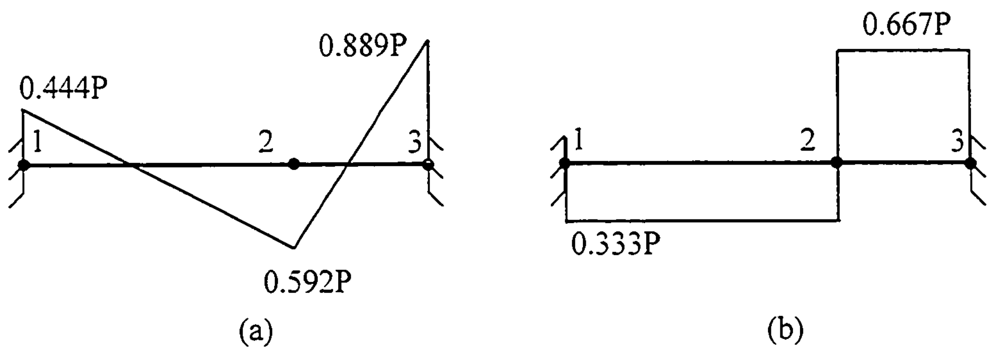

Fig. 5.14 Problem 3: No Plastification

(a) Bending Moments Diagram; (b) Torsion Diagram

The yield surface function including bending moment and torsion is $(M/M_p)^2 + (T/T_p)^2 = 1.0$. From Fig. 5.14, the first plastic hinge is expected to occur at node 3. For this node by substituting $M = 0.889\mathrm{P}$, $T = 0.667\mathrm{P}$ and recalling that for a DN90 STD section $M_p = 18.5\,KNm$ and $T_p = 16.7\,KN.m$, results into $\mathrm{P} = 16.0\,KN$. This load level is marked on Fig. 5.15.

After the formation of a plastic hinge at node 3, it is tedious to trace the forces in this hinge by hand, since the stress ratios (T/T_p) and (M/M_p) at this node do not remain constant although they satisfy the yield condition equilibrium.

The applied load P vs. the vertical displacement of node 2 is plotted in Fig. 5.15. In generating this plot, forty-eight PIPE31, forty-eight B33, one hundred twenty ELBOW31 and two FRAME3D elements have been used. The results of B33 and PIPE31 nearly coincide.

An important point in Fig. 5.15 is the very good agreement of FRAME3D and the proposed element. This is because both solutions are based on the same yield surface function for the case of bending moment and torsion ($M_r^2 + T_r^2 = 1$). The small difference between the two solutions is due to the fact that FRAME3D neglects shear force effects associated with bending deformation. It is of interest to note that in this case, B33, PIPE31 and ELBOW31 elements give close results in good agreement with the proposed element.

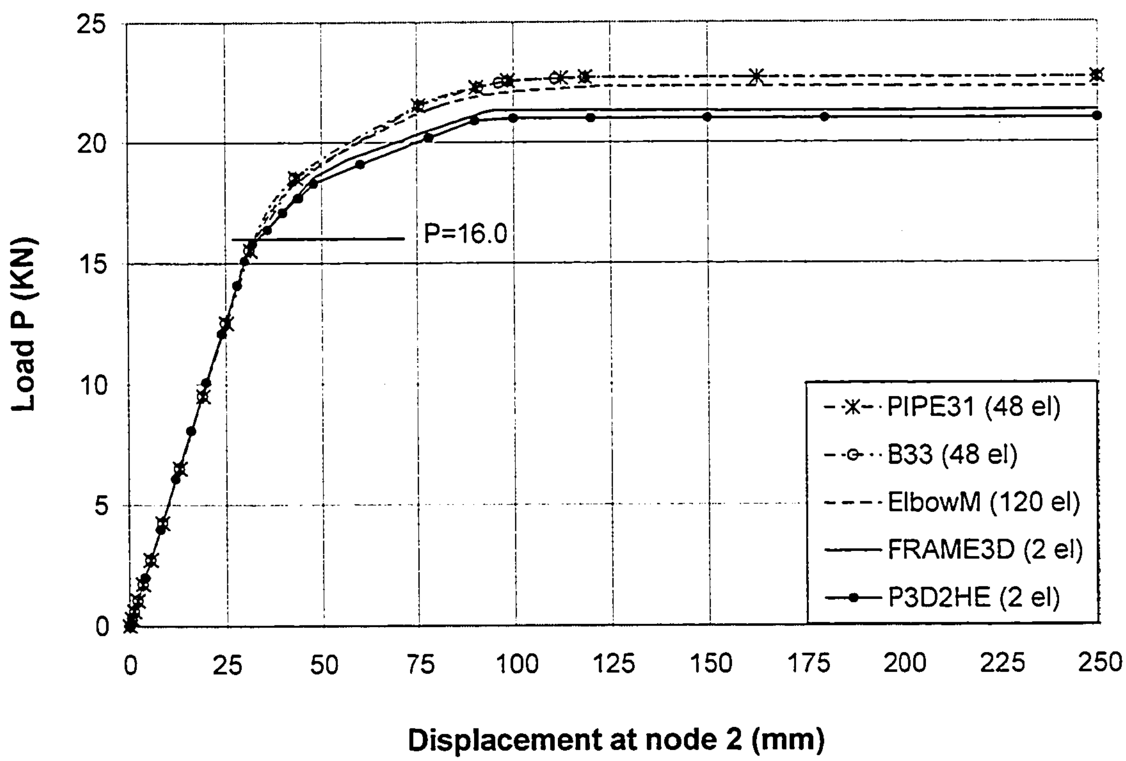

Fig. 5.15 Comparison of Results with ABAQUS for Problem 3

5.3.4 Problem 4: Shear and Bending Moments

The purpose of this example is to assess the validity of the results in the case of combinations of bending moment and shear. A 1.0m span pipe fixed at both ends is subject to a vertical load P at 0.8m from the left support (Fig. 5.16).

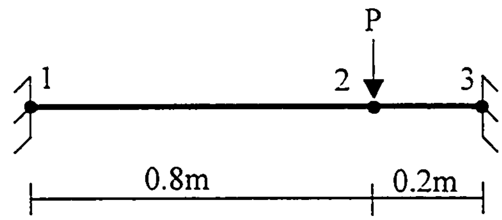

Fig. 5.16 Sketch of Problem 4

The bending moment and shear force diagrams induced by a force $P = P_1$ for this beam before the formation of plastic hinge are shown in Fig. 5.17.

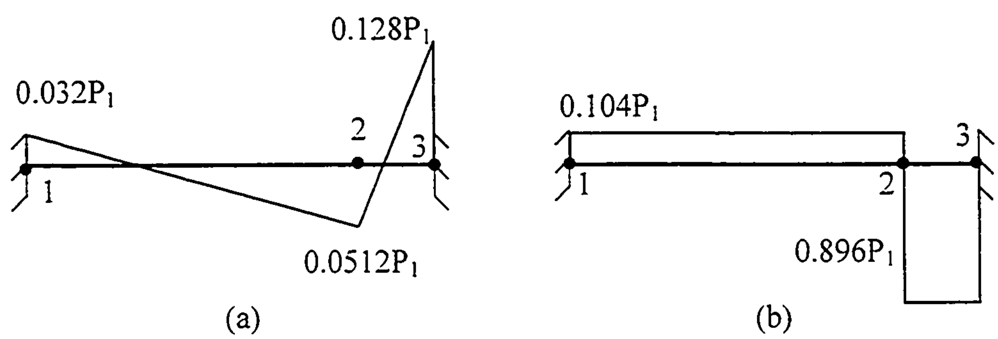

Fig. 5.17 Problem 4: No Plastification

(a) Bending Moments Diagram; (b) Shear Force Diagram

The yield condition for bending moments and shear force combination is $(M/M_p)^2 + (V/V_p)^2 = 1$. Recalling that for Fig. 5.17 $M_p = 18.5\,KNm$ and $V_p = 222.4\,KN$ it is obvious that node 3 plastifies first. Therefore, by substituting M= 0.128P_1 and V= 0.896P_1 for node 3 into the yield function, P_1 is obtained as 126.9 *KN*. At this load stage, the corresponding bending moments at node 1 and node 2 are 4.06 *KNm* and 6.47 *KNm*, respectively. The shear forces at node 1 and node 2 are 13.2 *KN* and 113.7 *KN*, respectively.

Again, after formation of a plastic hinge at node 3, it is difficult to trace the shear force and bending moment using hand calculation. At this node, the moment ratio reduces and shear ratio increases while satisfying the yield condition. This type of behaviour is subsequently <u>neglected</u> to find an <u>upper bound</u> for the corresponding force P of the plastification of node 2. This value of applied load P is an upper bound because the bending moments reduction at plastified node 3 should be redistributed to other nodes. This causes the actual moments at nodes 1 and 2 be higher than the values computed here. By plastification of node 3, the structure behaves as a fixed-pinned beam. The corresponding bending moment due to an additional load dP_1 are illustrated in Fig. 5.18.

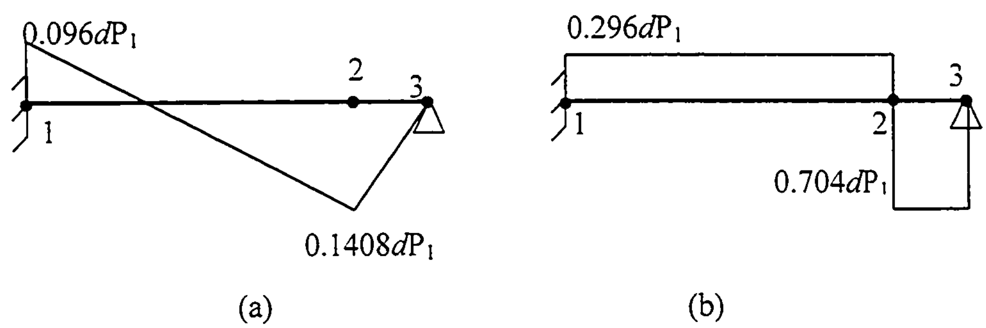

Fig. 5.18 Problem 4: Node 3 is Plastified

(a) Bending Moments Diagram; (b) Shear Force Diagram

The next plastic hinge occurs at node 2. The additional load necessary to form the hinge dP is calculated as

$$\left(\frac{6.47+0.1408dP_1}{18.5}\right)^2+\left(\frac{113.7+0.704dP_1}{222.4}\right)^2=1.0 \Rightarrow dP_1=51.1\ KN$$

Therefore an upper bound for applied load P_2 necessary to form plastic hinge at node 2 is $P_1+dP_1=178.0\ KN$. The exact value of this load (including the effect of changing

V/V_p ratio) as given by the program, is 172.8 *KN*. At this stage, the moment and shear at node 1 are 8.96 *KNm* and 28.3 *KN*, respectively.

By using the same method, an upper bound value for the failure load can be found. At this stage of loading, the structure starts acting as a cantilever beam (Fig. 5.20).

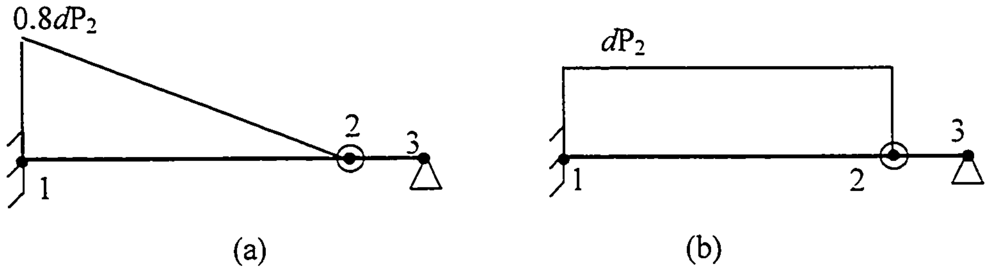

Fig. 5.19 Problem 4: Node 2 and 3 are Plastified

(a) Bending Moments Diagram; (b) Shear Force Diagram

The additional load needed to form the last plastic hinge is calculated from

$$\left(\frac{8.96+0.8dP_2}{18.5}\right)^2+\left(\frac{28.3+dP_2}{222.4}\right)^2=1.0\Rightarrow dP_2=11.5\ KN$$

Therefore an upper bound for failure load is $P_2+dP_2=189.5\ KN$. The exact value of failure load as predicted by the program is 183.0 *KN*.

Comparing results using ABAQUS elements are shown in Fig. 5.20. In generating this figure, 2 FRAME3D, 60 PIPE31, 60 B33 elements and 120 ELBOW31 are used.

Two important points are noticeable in Fig. 5.20: a) Firstly, the initial slope based on ABAQUS PIPE31 and ELBOW31 analyses match that of P3D2HE. This is due to inclusion of shear deformation effects in the elastic stiffness matrix of ABAQUS elements and P3D2HE. ABAQUS B33 and FRAME3D show a stiffer behaviour in the elastic range because they neglect the effect of shear deformations. By neglecting shear deformations, the analysis based on P3D2HE yields the same results of FRAME3D and B33. b) Secondly, only the proposed element nearly agrees with the upper limits for plastification points. The other four elements over estimate the loads at which plastification occur. This is due to the fact that the shear effects are neglected in the definition of the yield surface.

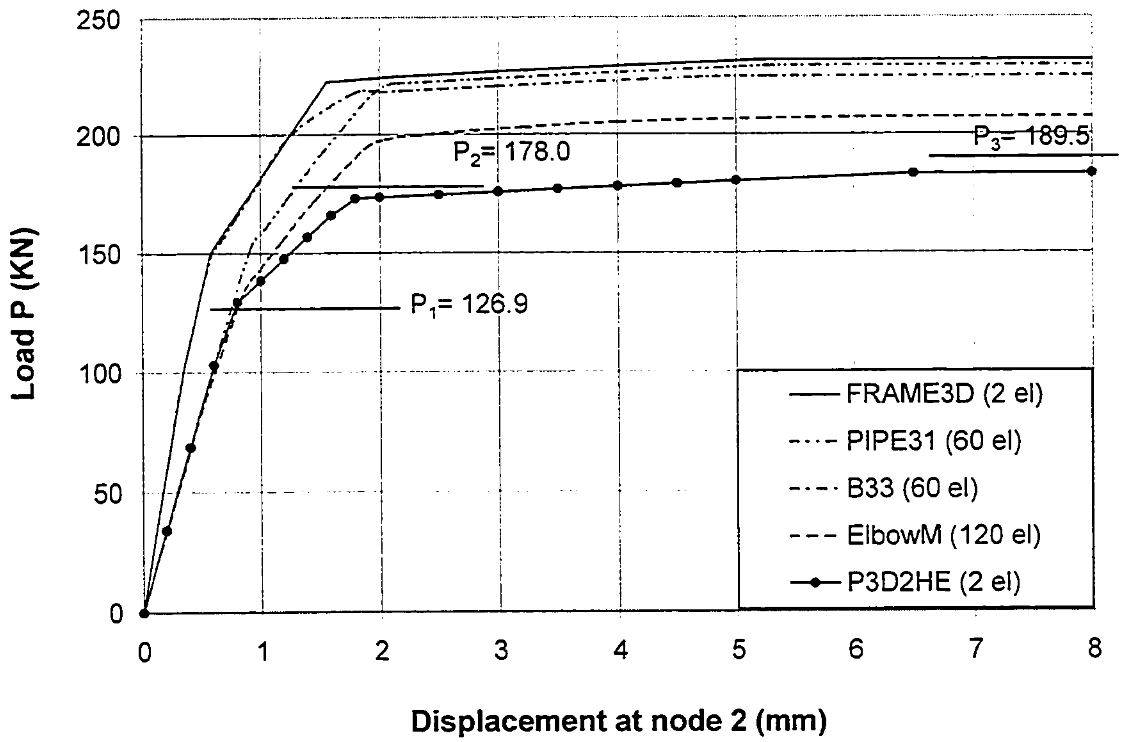

Fig. 5.20 Comparison of Results with ABAQUS Elements for Problem 4

5.3.5 Problem 5: Internal Pressure and Bending Moments

The purpose of this example is to assess the validity of the results of P3D2HE in the case of bending moments acting on a pressurized pipe. A 6m span pipe fixed at both ends is subject to a vertical load P at 4.0m from the left support (Fig. 5.21). This pipe is pressurized up to 30 *MPa*.

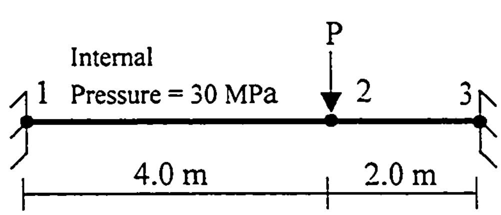

Fig. 5.21 Sketch of Problem 5

By neglecting the shear effects from Eqs. 2.4 and 2.5 the yield surface function will be $M_r^2 - (1-0.75p_r^2)\cos^2(0.25\pi p_r/\sqrt{1-0.75p_r^2}) = 0$. By substituting the pressure stress ratio $p_r = 30.0/44.6 = 0.67$ into the yield surface function, the moment ratio is found as $M_r = 0.650$. Knowing that node 3 plastifies first (Fig. 5.3) and neglecting shear force effects, load P_1 corresponding to the plastification of node 3 is found as $P_1 = 13.5KN$.

Using the same methodology as example 1, the approximate loads (neglecting shear effects) corresponding to plastification of nodes 2 and 1 are calculated as 17.1*KN* and 18.1*KN*, respectively.

Only ABAQUS PIPE31 and ELBOW31 elements incorporate the effect of internal or external pressure. Therefore, the comparison is performed with these two elements (Fig. 5.22). By using 120 ELBOW31 and 120 PIPE31 elements, the results in Fig. 5.22 were generated. In spite of the differences in modeling capabilities caused by spread of plasticity and cross sectional distortion, a very good agreement among the three solutions is obtained.

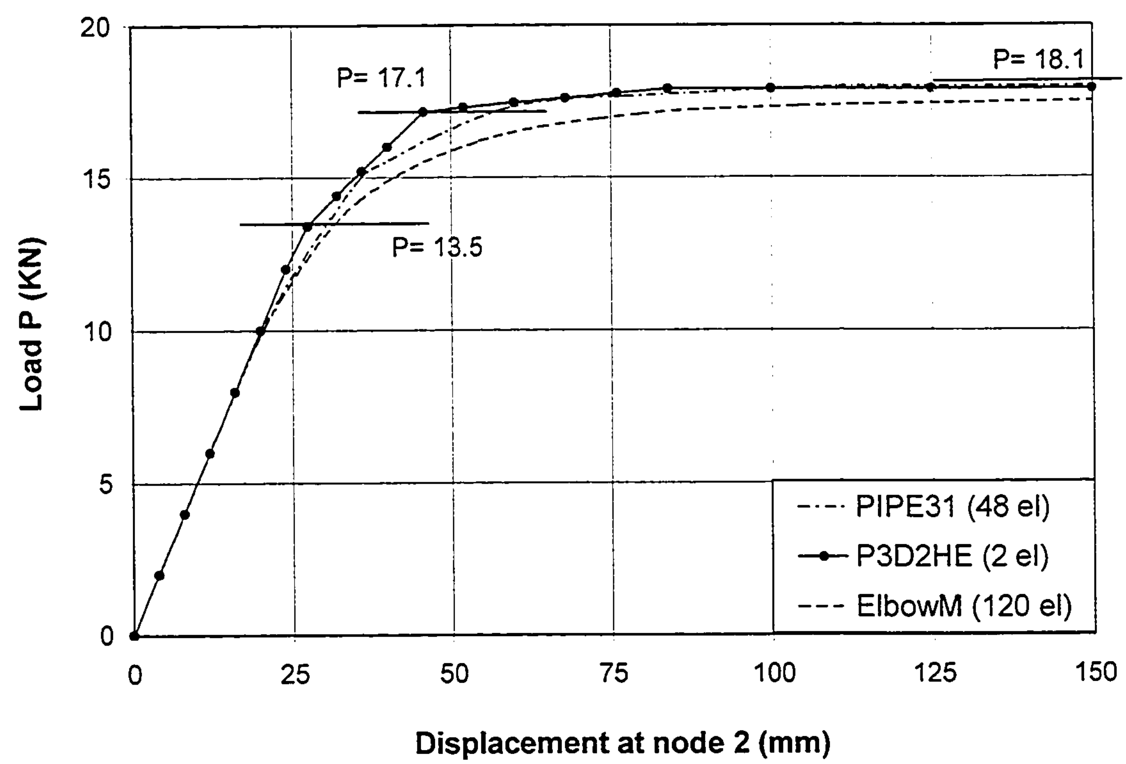

Fig. 5.22 Comparison of Results with ABAQUS Elements for Problem 5

A comparison of the load deformation responses of DN90 STD pipe under internal pressure ratios varying between p_r =0.0 and p_r =0.95 are illustrated in Fig. 5.23. This figure is generated for a fix-fix 6.0m span pipe. A concentrated load P is acting at 4.0m from the left support.

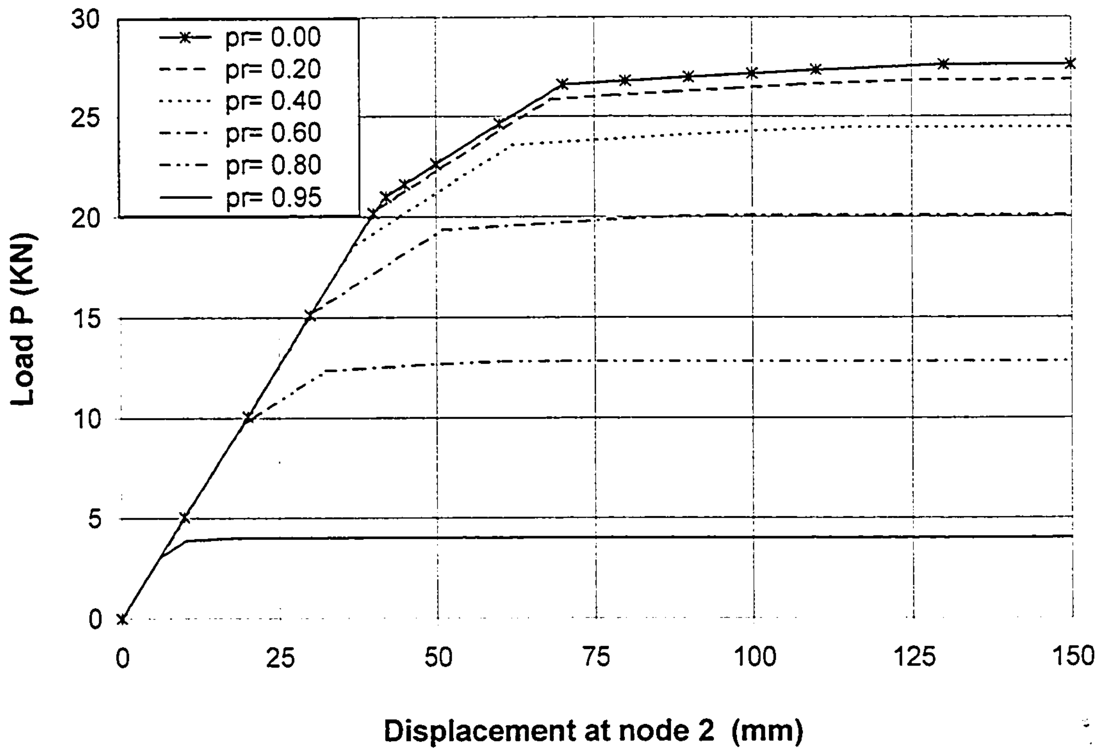

Fig. 5.23 Pipes under Various Pressure Ratios for a Pipe Section DN90 STD

As the pressure ratio increases, the peak applied load P attained decreases. This reflects the loss of bending capacity of the pipe section due to internal pressure.

5.3.6 Problem 6: A 3D Pipeline Problem

The purpose of this example is to illustrate the ability of the model to analyze 3D pipeline (Fig. 5.24) problems. The system of pipes is subject to a vertical load P at the middle of expansion loop and a distributed load w= 400 N/m. Load w is caused by self-weight of the pipe and the fluid it conveys and is assumed constant throughout the analysis. Load P varies from zero to an ultimate value.

In this problem, the pipe system is un-pressurized and the shear ratio is low. Therefore, ABAQUS FRAME3D should yield good results with relatively few elements while other ABAQUS elements would necessitate a finer mesh (problem 1). For this reason, the comparison is performed only with FRAME3D element (Fig. 5.25). Again, the small differences between the results are due to shear effects that are neglected in FRAME3D formulation.

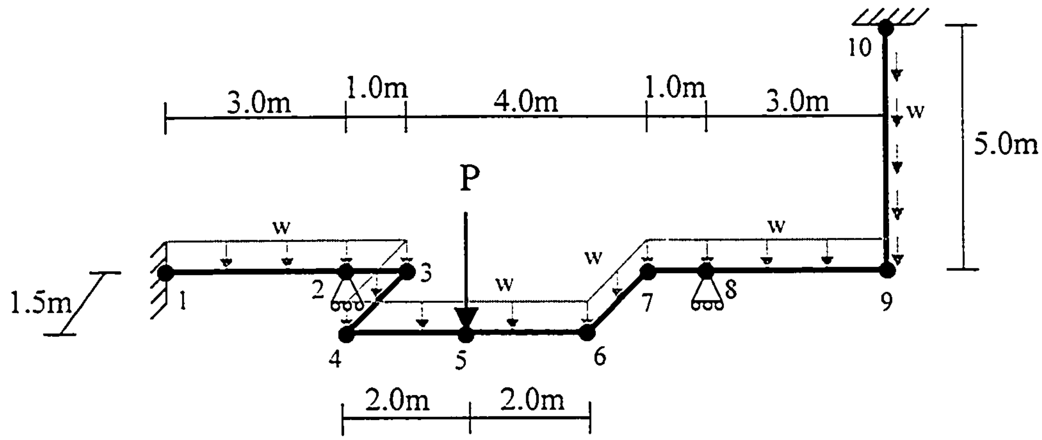

Fig. 5.24 Sketch of Problem 6

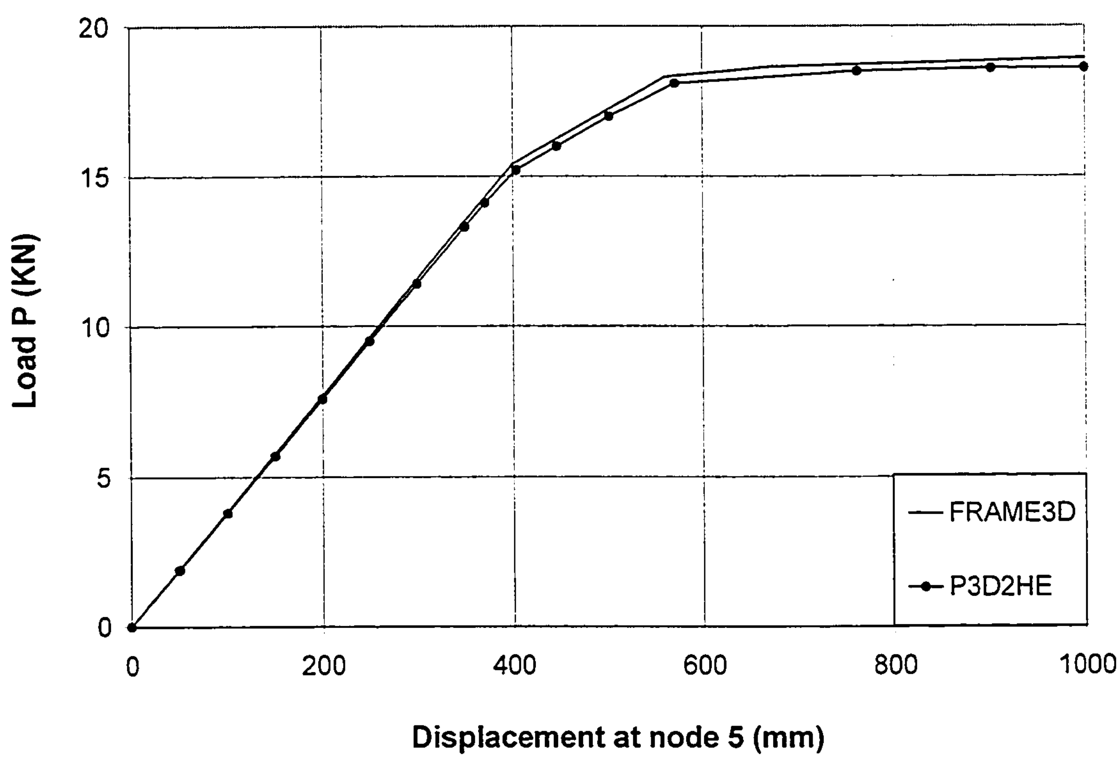

Fig. 5.25 Comparison of results with ABAQUS FRAME3D for problem 6

Chapter 6

Conclusions and Recommendations

An element that captures the plastic behaviour of pipelines was developed. The element is able to model the behaviour of a 3D pipe system. A comparative study against other well-established finite element solutions is summarized in Table 6.1.

Table 6.1: Comparison of Features for Various Finite Elements

Features	P3D2HE	FRAME3D	PIPE31	ELBOW31	B33
Models Geometric nonlinearity	No, can be added	Yes	Yes	Yes	Yes
Models Material nonlinearity	Yes	Yes	Yes	Yes	Yes
Models spread of plasticity	No, can be added	No	Yes	Yes	Yes
Models cross-section distortion	No	No	No	Yes	No
Models Internal/ External Pressure	Yes	No	Yes	Yes	No
Models material hardening	No, can be approximated	Yes	Yes	Yes	Yes
Longitudinal mesh needed for convergence	Coarse	Coarse	Fine	Very fine	Medium
Models shear effects in elastic range	Yes	No	Yes	Yes	No
Models shear effects in plastic range	Yes	No	No	Yes	No
Ability to model long pipelines	High	High	Low	Very low	Low
Uses exact yield surface	Yes	No	No	No	No
Models variation of pressure	No, can be added	No	Yes	Yes	No
Suitable for stocky pipes	Yes	No	Yes	Yes	No
Suitable for thin pipes	No	No	No	Yes	No
Models detailed behaviour of short pipe segment	No	No	No	Yes	No

Based on this study, the principal conclusions are:

1) The developed element yields very good prediction of the failure load of a piping system with very few elements, based on material plasticity.
2) The element is based on the exact yield surface for pipe sections and accounts for axial force, bi-axial shear, bi-axial bending moments, twisting moment and internal and/or external pressure, in characterizing the behaviour of 3D plastic hinges. This makes the element superior to other elements using other approximate yield surfaces.
3) The element models shear deformation effects both in the elastic and plastic ranges. This causes the element to yield accurate results for short span pipes under high concentrated loads when compared to other solutions based on beam elements.
4) The element models the effects of internal and external pressure into the analysis.
5) In conjunction with the displacement control method adopted in this research, the developed element can capture the point on the load deformation path at which a mechanism forms without any numerical difficulties.
6) The element is most suitable for pipeline analysis than commonly used elastic models and it is more in line with newer code methodologies for design of pipelines (CAN/CSA-Z66.2 1996), which allow the designer to benefit from the additional pipeline capacity by deforming into the plastic range.
7) Although other elements (e.g. ABAQUS ELBOW element) yield results, which are more accurate due to geometric nonlinearity effects, this accuracy is attained by significantly increasing the degrees of freedom and the computational effort. Therefore, they are impractical to use in common design problems. The element developed in this study (P3D2HE) provides a practical alternative for those problems.

The principal limitations of the developed element are:

1) The element does not model the spread of plasticity, although this feature can be added using a procedure similar to that devised by Attalla et al. (1994).

2) The element does not model geometric nonlinearity. This feature can be added by adding a geometric stiffness matrix $[K^g]$ to the tangent stiffness matrix (Chen et. al. 1995 and McGuire et. al. 2000). In this case, the incremental loads are related to incremental displacements through the relation

$$[K^e + K^m + K^g]\{d\Delta\} = \{dF\} \tag{6.1}$$

3) The element does not model cross-section distortion. Therefore, it is suitable for analyzing only stocky sections (i.e. $D/t \leq 25$). This effect can be modelled by using shell or ELBOW elements.
4) For members under distributed load, the user needs to define the location of nodes coinciding with the potential plastic hinges based on engineering judgment and experience. However, the outcome of the analysis has been shown not to be sensitive to the assumed location for the plastic hinge (Chen and Atsuta 1976).
5) The element does not model the variation of the pressure along the pipe centreline. Pressure variations along the pipeline can be easily added.

Based on the limitations described, the following recommendations are proposed as possible future research

1) Including the geometric nonlinear effects into the formulation.
2) Modeling of the variation of pressure along the pipeline.
3) Using a distributed plasticity instead of lumped plasticity formulation.

References

Attalla, M. R., Derielein, G. G. and McGuire, W. (1994), "Spread of Plasticity: A Quasi-Plastic-Elastic Approach", ASCE, *J. Struct. Engrg.* , Vol. 120(8), 2451-2473

Bathe, K. J. and Cimento, A. P., "Some Practical Procedures for the solution of nonlinear Finite Element Equations" *J. Computer. Meth. App. Mech. In Engrg.*, Vol. 22, 231-277

Bathe, K. J. and Dvorkin, E. N., (1983), " On the Automatic Solution of Nonlinear Finite Element Equations", *J. of Computers & Structures*, Vol. 17(5-6), 871-879, Pub. Pergamon, Great Britain

Bergan, P. G. and Clough R. W., (1972), "Convergence Criteria for iterative Process", *J. AIAA*, Vol. 10(8), 1107-1108

Blevins, R. D., (1984), " Formulas for Natural Frequency and Mode Shapes", *Pub. Malabar, Fla*: Kreiger R.E.

Boresi, P. and Sidebottom, O. M., (1985), "Advanced Mechanics of Material", 4th edition, *Pub. New York: Wiley*

Boukamp, J. G., and Stephen, R. M. (1973). "Large Diameter under Combined Loading" ASCE, *Transp. Engrg. J.*, 99(3), 521-536.

Bruneau, M. , Chia, M. and Whitaker, A., (1998), "Ductile Design of Steel Structures", *Pub. McGraw-Hill, NY.*

CAN/CSA S16 (2001). "Limit State Design of Steel Structures", Canadian Standards Association, Rexdale, Ont.

CAN/CSA- Z66.2 (1996), Oil and Gas Pipeline Systems, Canadian Standards Association. Rexdale, Ont.

Chen, W. and Atsuta, T. (1972). "Interaction Equations for Biaxially Loaded Sections." ASCE, *J. Struct. Engrg.*, Vol. 98(5), 1035-1051

Chen, W. F, and Atsuta, T. (1976), " Theory of beam-column, 1, in-Plane Behaviour and Design", *Pub. McGraw-Hill Int.*, pp 513

Chen, W. F, and Atsuta, T. (1972), " Interaction Equations for Biaxially Loaded Sections.", ASCE, *J. of Struct. Division*, Vol. 98(5), 1035-1051

Chen, W. F., Sohal, I. (1995), "Plastic Design and Second-Order Analysis of Steel Frames", Pub. Springer-Verlag.

Det Norske Veritas. (1996). "Rules for Submarine Pipelines." Oslo, Norway.

Fellipa, D., (2000), "http://caswww.colorado.edu/Felippa.d/FelippaHome.d/Home.html"

Gerald, S. and Becker, H., (1957), "Handbook of Structural Stability, Part III-Buckling of Curved Plates and Shell", National Advisory Committee for Aeronautics, Tech. Note No. 3783

Glenn, A. M, Steven, J. F., (1970), " Elastic-Plastic Analysis of Frameworks", ASCE, *J. Struct. Division*, Vol. 96(5), 931-946

Hibbit, Karlsson and Sorensen, Inc. (2001), "ABAQUS/Standard User's Manual", Version 6.2, Pawtucket, USA.

Hibbit, Karlsson and Sorensen, Inc. (2001), "ABAQUS theory Manual", Version 6.2, Pawtucket, USA.

Kardestuncer, H., (1974), "Elementary Matrix Analysis of Structures", McGraw-Hill Book Company

Kassimali, A., (1999), " Matrix Analysis of Structures", Brooks/ Cole publishing Company

McGuire, W., G., Ziemian, R. H., and Ronald D. (2000), "Matrix Structural Analysis", Second edition, John Wiley & Sons, Inc.

Mohareb, M. (2001), "Exact yield hyper-surface for thin pipes", *Int. J. of pressure vessels and piping*, Vol. 78(7), 507-514

Mohareb, M., Kulak, G. L., Elwi, A., Murray, D. W. (2001), "Testing And Analysis of Steel Pipe Segments", ASCE, *J. Trans. Engrg.* Vol. 127(5), 408-417

Mohareb, M. (2002), "Plastic Interaction Relations for Pipe sections." *ASCE, J. Engrg. Mech.*, Vol. 128(1), 112-120

Mohareb. M. (2003), "Plastic Resistance of Pipe Sections- an Upper Bound Solution.", ASCE., *J. Struct. Engrg.* , to appear Vol. 129(1)

Mohareb, M. and Murray, D. W. (1999). "Mobilization of Fully Plastic Moment Capcity for Pressurized Pipes Subjected to Axial Loading." ASME, *J. Offshore Mech. Arct. Engrg.*, Vol. 121(4), 237-241

Morris, G. A., Fenves, S. J. , (1970), "Elastic-Plastic Analysis of Frameworks", ASCE, *J. of Struct. Division*, Vol. 96 (5), 931-946

Morris, G. and Fenves, S. (1969). "Approximation Yield Surface Equations", ASCE, *J. Engrg. Mech Division*, Vol. 95(4): 937-954

Obaia, K. H., Elwi, A., Kolak, G. L. (1991). "Inelastic Transverse Shear Capacity of large fabricated steel tubes." *Struct. Engrg. Rep. No. 170, Dept. of Civ. Engrg.*, University of Alberta, Edmonton, Canada.

Orbison, J. G. (1982), "Yield Surface Applications in Non-Linear Steel Frame Analysis", *J. of Computer Meth. in App. Mech. and Engrg.*, Vol. 33, 557-573

Ozkan, I. F. (2002), " Plastic Interaction Relations for Hollow Structural Steel Sections", M.A. Sc. Thesis, University of Ottawa

Ozkan, I. F. and Mohareb, M. (2002), " Testing of Steel Pipes under Bending, Twist and Shear", ASCE, *J. Struct. Engrg.*, Under review

Pillai, S. U., and Ellis, J. S. (1971). "Hollow tubular Beam-Column in Biaxial Bending." ASCE, *J. Struct. Engrg.*, 97 (5), 1399-1406

Porter, F. L. and Powel, G. H. (1971), "Static and Dynamic Analysis of Inelastic Frame Structures", Report No. EERC 71-3 Earthquake engineering research centre, Univ. of California, Berkeley

Powel, G. H. (1986), " 3D Beam-Column Element with Generalized Plastic Hinges", ASCE, *J. of Engrg. Mech.*, Vol. 112 (7), 627-641

Prion, H. and Birkemoe, P. (1998). "Experimental Behaviour of Unstiffened Fabricated Tubular Steel Beam-Column." *University of Toronto, Dept. of Civ. Engrg. Pub. No. 88-03*, Toronto, Ont., Canada

Ramm, E. (1981), "Strategies of Tracing Response near Limit Point. In Nonlinear Finite Element Analysis in Structural Mechanics " Edited by Wunderlich et. al., 63-89, Springer-Verlag, New York

Sherman, D. R. (1976), "Test of Circular Steel Tubes in Bending", ASCE, *J. of Struct. Division*, Vol. 102(11), 2181-2195

Timoshenko S. and Goodier J. (1987). Theory of Elasticity, 3rd Ed., McGraw-Hill, New York.

Yamaki, H. (1976). "Experimental on the post-buckling of the circular cylinder Shell Under Torsion. " *Proc. IUTAM, Buckling of structures., Springer-Verlag*, Heidelberg, 312-330

Appendix A1

Derivatives of the Yield Surface Equations

Equation 2.15 shows the necessity of finding nine partial derivatives of interaction relations. Because of problems with orthogonalilty between the axial force and the internal or external pressure, which causes that pressure cannot be treated as a separate degree of freedom, and the complexity of analysis based on a seven dimensional yield surface, it was decided to hold the pressure constant during the analysis. However, the yield hyper-surface equations are still functions of pressure.

$$\frac{\partial \Phi_1}{\partial F_2} = \frac{2}{V_p} \frac{F_{r2}}{\tau_r^2} \tag{A1.2}$$

$$\frac{\partial \Phi_1}{\partial F_3} = \frac{2}{V_p} \frac{F_{r3}}{\tau_r^2} \tag{A1.3}$$

$$\frac{\partial \Phi_1}{\partial M_1} = \frac{1}{T_p} \frac{\pi}{2\tau_r} \sin\left(\pi \frac{M_{r1}}{\tau_r}\right) \tag{A1.4}$$

$$\frac{\partial \Phi_1}{\partial \tau} = -\frac{\sqrt{3}}{\sigma_y \tau_r^3}\left[2F_{r2}^2 + 2F_{r3}^2 + \frac{\pi}{2} M_{r1} \tau_r \sin\left(\pi \frac{M_{r1}}{\tau_r}\right)\right] \tag{A1.5}$$

$$\frac{\partial \Phi_2}{\partial F_1} = \frac{1}{4} \frac{\gamma \pi}{A_p} \sin\left(2\pi \frac{\mu}{\gamma}\right) \tag{A1.6}$$

$$\frac{\partial \Phi_2}{\partial M_2} = \frac{2}{M_p} M_{r2} \tag{A1.7}$$

$$\frac{\partial \Phi_2}{\partial M_3} = \frac{2}{M_p} M_{r3} \tag{A1.8}$$

$$\frac{\partial \Phi_2}{\partial p} = 0 \tag{A1.9}$$

$$\frac{\partial \Phi_2}{\partial \tau} = \tau_r \frac{\sqrt{3}}{\sigma_y}\left[1 + \cos\left(2\pi \frac{\mu}{\gamma}\right) + \frac{\pi\mu}{\gamma}\sin\left(2\pi \frac{\mu}{\gamma}\right)\right] \tag{A1.10}$$

in which

$$\mu = F_{r1} - \frac{1}{2} p_r$$

$$\gamma = \sqrt{4 - 3p_r^2 - 4\tau_r^2}$$

Example

A DN90 STD pipe section has a 101.6 mm outside diameter and 5.74 mm thickness, is subjected to the following loads. Assume σ_y=350 Mpa and E =200,000 Mpa. Calculate the normal to the yield surface at this point.

$F_1 = 200\,KN$ $\quad F_2 = 30\,KN$ $\quad F_3 = 90\,KN$

$M_1 = 6.5\,KNm$ $\quad M_2 = 12.0\,KNm$ $\quad M_3 = 7.0\,KNm$

$p = 5.0\,Mpa$

Solution:

a) Calculate plastic resistances in absence of other stress resultants (Under expression of Eq. 2.4 and 2.5):

$$A_p = 2\pi rt\,\sigma_y = 2\pi \times 47.93 \times 5.74 \times 350 \times 10^{-3} = 605.0\,KN$$

$$V_p = 4rt(\sigma_y / \sqrt{3}) = 4 \times 47.93 \times 5.74 \times \left(350 / \sqrt{3}\right) \times 10^{-3} = 222.4\,KN$$

$$M_p = 4r^2 t\sigma_y = 4 \times (47.93)^2 \times 5.74 \times 350 \times 10^{-6} = 18.5\,KN.m$$

$$T_p = 2\pi r^2 t(\sigma_y / \sqrt{3}) = 2\pi \times (47.93)^2 \times 5.74 \times \left(350 / \sqrt{3}\right) \times 10^{-6} = 16.7\,KN.m$$

$$p_p = t\sigma_y / (r - t/2) = 5.74 \times 350 / (47.93 - 2.87) = 44.6\,MPa$$

b) Calculate the ratio of the applied load to plastic resistance

$$F_{r1} = F_1 / A_p = 200/605.0 = 0.331$$

$$F_{r2} = F_2 / V_p = 30/222.4 = 0.135$$

$$F_{r3} = F_3 / V_p = 90/222.4 = 0.405$$

$$M_{r1} = M_1 / T_p = 6.5/16.7 = 0.389$$

$$M_{r2} = M_2 / M_p = 12/18.5 = 0.649$$

$$M_{r3} = M_3 / M_p = 7/18.5 = 0.378$$

$$p_r = p / p_p = 5/44.6 = 0.112$$

c) Check the plastic status of pipe under applied loads

Find τ from the first yield hyper-surface equation (Eq. 2.4)

$\Phi_1(20.0, 15.0, 4.5, \tau) = 0 \rightarrow \tau = 102.6$ Mpa and $\tau_r = 0.508$

Substituting τ and other loads into the second equation of yield hyper-surface and calculate corresponding stress resultant (Eq. 2.5)

$\Phi_2(200.0, 12, 7, 5, 102.6) = -0.001 \approx 0$ $\Rightarrow$ The point is on the yield surface.

d) Check the applicability limits

$$F_{r2}^2 + F_{r3}^2 = (0.135)^2 + (0.405)^2 = 0.182$$

$\cos^2(\pi M_{r1}/2) = \cos^2(\pi \times 0.389/2) = 0.931 > 0.182 \Rightarrow$ Eq. 2.6 is met

$F_{r2}^2 + F_{r3}^2 = 0.182 \le \tau_r^2 = 0.258 \Rightarrow$ Eq. 2.7 is met

$$2\sqrt{\frac{1-\tau_r^2}{3}} = 0.995$$

$-0.995 \le 0.119 \le 0.995 \Rightarrow$ Eq. 2.8 is met.

$$\frac{1}{2}p_r - \sqrt{1 - \frac{3}{4}p_r^2 - \tau_r^2} = 0.5 \times 0.112 - \sqrt{1 - .75 \times 0.112^2 - 0.508^2} = -0.800$$

$$\frac{1}{2}p_r + \sqrt{1-\frac{3}{4}p_r^2 - \tau_r^2} = 0.5\times 0.112 + \sqrt{1-.75\times 0.112^2 - 0.508^2} = 0.912$$

$$-0.800 \le F_{r1} = 0.331 \le 0.912 \qquad \Rightarrow \qquad \text{Eq. 2.9 is met.}$$

e) Calculate the normal to the yield surface {G}

$$\mu = F_{r1} - p_r/2 = 0.331 - 0.112/2 = 0.275$$

$$\gamma = \sqrt{4-3p_r^2 - 4\tau_r^2} = \sqrt{4-3\times 0.112^2 - 4\times 0.508^2} = 1.712$$

From Eq. A1.6

$$\left.\frac{\partial \Phi_1}{\partial F_1}\right|_{200} = \frac{1}{4\times 605.0}\times 1.712\pi \sin(2\pi \times 0.275/1.712) = 1.881\times 10^{-3}\ \text{KN}^{-1}$$

From Eq. A1.7

$$\left.\frac{\partial \Phi_1}{\partial F_2}\right|_{30} = \frac{2}{222.4}\times\frac{0.135}{0.508^2} = 4.704\times 10^{-3}\ \text{KN}^{-1}$$

From Eq. A1.3

$$\left.\frac{\partial \Phi_1}{\partial F_3}\right|_{90} = \frac{2}{222.4}\times\frac{0.405}{0.508^2} = 14.11\times 10^{-3}\ \text{KN}^{-1}$$

From Eq. A1.4

$$\left.\frac{\partial \Phi_1}{\partial M_1}\right|_{6.5} = \frac{1}{16.7}\times\frac{\pi}{2\times 0.508}\sin\left(\pi\frac{0.389}{0.508}\right) = 0.1243\ \text{KN}^{-1}\text{m}^{-1}$$

From Eq. A1.7

$$\left.\frac{\partial \Phi_2}{\partial M_2}\right|_{12} = \frac{2}{18.5}\times 0.549 = 0.0593\ \text{KN}^{-1}\text{m}^{-1}$$

From Eq. A1.8

$$\left.\frac{\partial \Phi_2}{\partial M_3}\right|_7 = \frac{2}{18.5} \times 0.378 = 0.0408 \ \ \mathrm{KN^{-1}m^{-1}}$$

From Eq. A1.5

$$\left.\frac{\partial \Phi_1}{\partial \tau}\right|_{102.6} = -\frac{\sqrt{3}}{350 \times 0.508^3} \times$$

$$\left[2 \times 0.135^2 + 2 \times 0.405^2 + \frac{\pi}{2} 0.389 \times 0.508 \ \sin\left(\pi \frac{0.389}{0.508}\right)\right] = -0.0216 \ \ \mathrm{Mpa^{-1}}$$

From Eq. A1.10

$$\left.\frac{\partial \Phi_2}{\partial \tau}\right|_{102.6} = 0.508 \times \frac{\sqrt{3}}{350} \times$$

$$\left[1 + \cos\left(2\pi \frac{0.275}{1.712}\right) + \frac{0.275\pi}{1.712} \sin\left(2\pi \frac{0.275}{1.712}\right)\right] = 4.926 \times 10^{-3} \ \ \mathrm{Mpa^{-1}}$$

By using equation 2.15 the vector {G} is calculated as

$$\{G\} = \left\{ 8.196 \times 10^{-3} \ \ 4.704 \times 10^{-3} \ \ 14.11 \times 10^{-3} \ \ 0.1243 \ \ 0.260 \times 10^{-3} \ \ 0.179 \times 10^{-3} \ \ 0 \right\}$$

This is while the first three terms have the unit of KN^{-1} and the next three have the unit of $KN^{-1}m^{-1}$.

Appendix A2

Displacement Control vs. Load Control

The equilibrium equation to be solved is $[K]\{d\Delta\} = \eta\{P\}$. Where $[K]$ is the stiffness matrix, $\{d\Delta\}$ is the incremental displacement vector, $\{P\}$ is the unbalanced load vector and η is the incremental load factor. The stiffness matrix is symmetric with positive terms on the diagonal. These two conditions are relaxed here to show the generality of the algorithm.

Load Control Method:

Assume that matrix $[K]$ and vector $\{P\}$ are given as the following. Find incremental displacement vector corresponding to $\eta = 0.5$.

$$[K] = \begin{bmatrix} 1 & 2 & 3 & 5 & 6 \\ 2 & -1 & 0 & 1 & 1 \\ 3 & 0 & 1 & 3 & 7 \\ 5 & 1 & 3 & 1 & 5 \\ 3 & 2 & 7 & 5 & 4 \end{bmatrix} \qquad \{P\} = \begin{Bmatrix} 30 \\ 5 \\ 29 \\ 21 \\ 22 \end{Bmatrix}$$

Solution:

The equation that should be solved is

$$\begin{bmatrix} 1 & 2 & 3 & 5 & 6 \\ 2 & -1 & 0 & 1 & 1 \\ 3 & 0 & 1 & 3 & 7 \\ 5 & 1 & 3 & 1 & 5 \\ 3 & 2 & 7 & 5 & 4 \end{bmatrix} \times \begin{Bmatrix} d\Delta_1 \\ d\Delta_2 \\ d\Delta_3 \\ d\Delta_4 \\ d\Delta_5 \end{Bmatrix} = 0.5 \begin{Bmatrix} 30 \\ 5 \\ 29 \\ 21 \\ 22 \end{Bmatrix} \tag{A2.1}$$

The incremental displacement vector is as

$$\{d\Delta\} = 0.5 \times \begin{bmatrix} 1 & 2 & 3 & 5 & 6 \\ 2 & -1 & 0 & 1 & 1 \\ 3 & 0 & 1 & 3 & 7 \\ 5 & 1 & 3 & 1 & 5 \\ 3 & 2 & 7 & 5 & 4 \end{bmatrix}^{-1} \times \begin{Bmatrix} 30 \\ 5 \\ 29 \\ 21 \\ 22 \end{Bmatrix} = \begin{Bmatrix} 0.5 \\ 1.0 \\ -0.5 \\ 1.0 \\ 1.5 \end{Bmatrix} \tag{A2.2}$$

It is important to note that $[K]$ is for the whole structure thus it is non-singular after imposing boundary conditions.

Displacement Control Method:

Find load incremental factor η corresponding to $d\Delta_2 = 1.0$ (second row in Eq. A2.2).

Solution:

The system of equations that should be resolved is

$$\begin{bmatrix} 1 & 2 & 3 & 5 & 6 \\ 2 & -1 & 0 & 1 & 1 \\ 3 & 0 & 1 & 3 & 7 \\ 5 & 1 & 3 & 1 & 5 \\ 3 & 2 & 7 & 5 & 4 \end{bmatrix} \times \begin{Bmatrix} d\Delta_1 \\ 1.0 \\ d\Delta_3 \\ d\Delta_4 \\ d\Delta_5 \end{Bmatrix} = \eta \begin{Bmatrix} 30 \\ 5 \\ 29 \\ 21 \\ 22 \end{Bmatrix} \tag{A2.3}$$

By removing the second row and column of the stiffness matrix $[K]$ and the second row of the load vector $\{P\}$ corresponding to $d\Delta_2$ matrix $[K_{11}]$ and vector $\{P_1\}$ are found, respectively. Recall from chapter four (Eqs. 4.16 and 4.17) that $[K_{11}]\{d\bar{\Delta}_1\} = \{P_1\}$ and $[K_{11}]\{d\bar{\bar{\Delta}}_1\} = -[K_{12}] \cdot d\Delta_2$. Therefore

$$\left\{d\bar{\Delta}_1\right\}=\begin{bmatrix}1 & 3 & 5 & 6\\ 3 & 1 & 3 & 7\\ 5 & 3 & 1 & 5\\ 3 & 7 & 5 & 4\end{bmatrix}^{-1}\begin{Bmatrix}30\\ 29\\ 21\\ 22\end{Bmatrix}=\begin{Bmatrix}-3.875\\ 3.313\\ -3.313\\ 6.750\end{Bmatrix} \tag{A2.4}$$

$$\left\{d\bar{\bar{\Delta}}_1\right\}=-1.0\times\begin{bmatrix}1 & 3 & 5 & 6\\ 3 & 1 & 3 & 7\\ 5 & 3 & 1 & 5\\ 3 & 7 & 5 & 4\end{bmatrix}^{-1}\begin{Bmatrix}2\\ 0\\ 1\\ 2\end{Bmatrix}=\begin{Bmatrix}2.438\\ -2.156\\ 2.656\\ -1.875\end{Bmatrix} \tag{A2.5}$$

The load incremental factor is found as

$$\eta=\frac{\langle K_{21}\rangle\left\{d\bar{\bar{\Delta}}_1\right\}+K_{22}\times d\Delta_2}{P_2-\{K_{21}\}\{d\bar{\Delta}_1\}}=\frac{\langle 2\quad 0\quad 1\quad 1\rangle\begin{Bmatrix}2.438\\ -2.156\\ 2.656\\ -1.875\end{Bmatrix}+(-1)\times(1.0)}{5-\langle 2\quad 0\quad 1\quad 1\rangle\begin{Bmatrix}-3.875\\ 3.313\\ -3.313\\ 6.750\end{Bmatrix}} \tag{A2.6}$$

$$=\frac{4.6562}{9.3126}=0.5$$

Other incremental displacement vector components are found as

$$\eta\cdot d\bar{\Delta}_1+d\bar{\bar{\Delta}}_1=0.5\begin{Bmatrix}-3.875\\ 3.313\\ -3.131\\ 6.750\end{Bmatrix}+\begin{Bmatrix}2.438\\ -2.156\\ 2.656\\ -1.875\end{Bmatrix}=\begin{Bmatrix}0.5\\ -0.5\\ 1.0\\ 1.5\end{Bmatrix} \tag{A2.7}$$

which is identical to that found in Eq. A2.2.

Appendix B

Program Listing and Programmer's Guide

B.1 General

The developed program, PIPLAST v1.0, is object-oriented base and is written under MS Visual BASIC 6.0. This program includes one from as a user interface, three modules (Analysis, Initializing and Matrix) and three classes (Element, Property and Joint). This appendix provides a guide for programmers who are familiar with Visual Basic.

In the program listing provided, procedures are sorted alphabetically. Their name and job (if necessary) are

Main Form..99

Initializing Module..110

Analysis Module...112

- Private Sub BCImpose, imposes boundary conditions....................................112
- Public Sub DispControl, Displacement control method....................................112
- Private Sub GlobalForces, Constructs global force vectors..............................114
- Private Sub globalK, Constructs global stiffness matrix...................................115
- Private Faction UnBalLoad, checks the existence of unbalanced load.................115

Matrix Module...116

- Private Function Dimension, Returns dimension of an array..............................116
- Private function Gauss, Guess method..116
- Public Function MatAdd, adds two matrices..117
- Public Function Matdet, Determinates a matrix..118
- Public Function MatInv, Inverts a matrix...118
- Public Function MatMul, multiplies two matrices...118
- Public Function MatMulNum, multiplies a matrix by number.............................119
- Public Function MatMulVec, multiplies a matrix by a vector.............................119
- Public Function MatSysSolve, solves a system of equations and etc....................120

- Public Function MatTrans, transposes a matrix....................................120
- Private Function Pivot, matrix pivoting....................................121

Element Class....................................122

- Private Function ActLoadInc, acceptable amount of load increment....................................124
- Private Function ArcSin....................................124
- Private Sub ChkJntMech, checks joint mechanism....................................125
- Private Sub Create, initializes an element object....................................125
- Private Sub ElasticStiffness....................................125
- Private Function FindRatio, fraction of load crosses the YS....................................126
- Private Sub IntForces, Calculates internal force vector....................................127
- Private Sub PlasticFlow, Plastic flow rule....................................127
- Private Sub PlasticReduction, Plastic reduction stiffness matrix....................................128
- Private Function Stress, calculates stress resultant....................................129
- Private Sub Tangent Stiffness....................................130
- Private Sub TransformMatrix....................................130
- Public Sub Update, updates an object....................................131
- Private Sub WtLoad, Weight load....................................131

Joint Class....................................132

Property Class....................................134

B.2 Main form

B.2.1 Fields

When the program starts, the form in Fig. B.1 is displayed to the user. Various field names as they named in the program are shown. The function for each of these fields is:

cmbDOF: The degree of freedom number of the controlling parameter

cmbLoad: The load to be selected among the externally applied nodal loads

cmbNode: Node of the controlling displacement parameter

cmdAnal: Starts analysis

cmdDraw: Plots results stored in the output file (After termination of the analysis) cmdEnd: Quits the program

cmdRefresh: Erase the plotting window

cmdSelect: Selects the input file to be read

Picture1: The drawing box

txtDisp: Shows the value of controlling displacement parameter at the current iteration

txtFile: The name of input file

txtLoad: Shows the value of base load at the current iteration

txtMaxDisp: Value of the upper limit of the controlling parameter for drawing

txtMaxLoad: Value of the upper limit of applied load in plot

txtPath: The directory of input file

txtValue: User specified amount of the controlling displacement for each increment

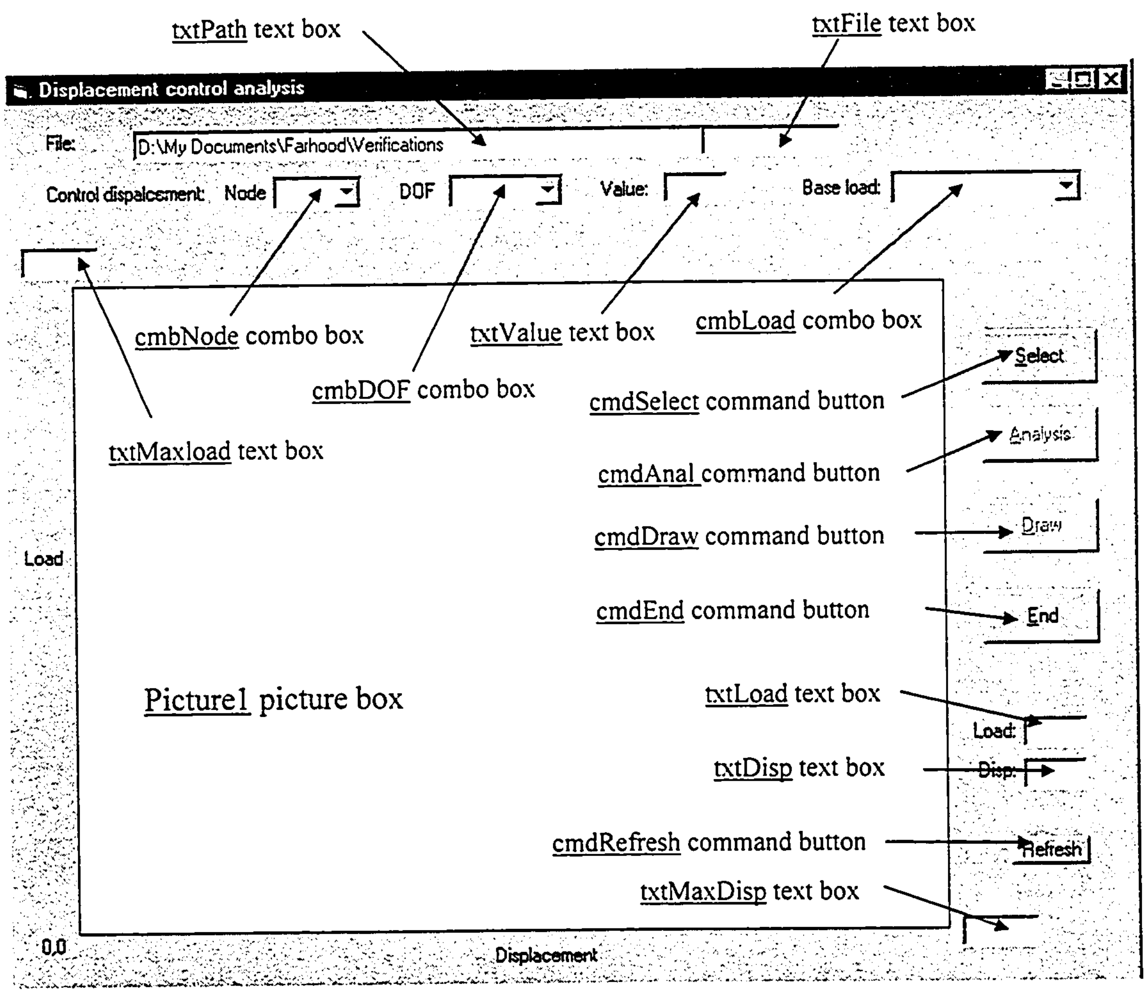

Fig. B.1 Names of the form Fields

B.2.2 Code List

```
VERSION 5.00
'*****************************************************************************
'* This is a part of PIPLAST Version 1.0 written by Farhood Nowzartash.
'* Copyright © by Farhood Nowzartash 2002
'*****************************************************************************

'=============== The following is the displaying parameters of the form
Begin VB.Form frmMain
   Caption         =   "Displacement control analysis"
   ClientHeight    =   8130
   ClientLeft      =   60
   ClientTop       =   345
   ClientWidth     =   10650
   LinkTopic       =   "Form1"
   ScaleHeight     =   143.404
   ScaleMode       =   6  'Millimeter
   ScaleWidth      =   187.854
   StartUpPosition =   3  'Windows Default
   Begin VB.CommandButton cmdRefesh
      Caption         =   "Refresh"
      Height          =   255
      Left            =   9480
      TabIndex        =   26
      TabStop         =   0   'False
      Top             =   6840
      Width           =   735
   End
   Begin VB.TextBox txtMaxDisp
      Height          =   285
      Left            =   9000
      TabIndex        =   23
      Top             =   7560
      Width           =   735
   End
   Begin VB.TextBox txtMaxLoad
      Height          =   285
      Left            =   120
      TabIndex        =   22
      Top             =   1320
      Width           =   735
   End
   Begin VB.CommandButton cmdDraw
      Caption         =   "&Draw"
      Height          =   495
      Left            =   9240
      TabIndex        =   21
      Top             =   3720
      Width           =   1095
   End
   Begin VB.TextBox txtLoad
      Height          =   285
      Left            =   9600
      TabIndex        =   18
      TabStop         =   0   'False
      Top             =   5730
      Width           =   615
   End
   Begin VB.TextBox txtDisp
      Height          =   285
      Left            =   9600
      TabIndex        =   17
      TabStop         =   0   'False
      Top             =   6120
      Width           =   600
```

```
End
Begin VB.ComboBox cmbLoad
   Height          =   315
   Left            =   8400
   Style           =   2  'Dropdown List
   TabIndex        =   14
   Top             =   675
   Width           =   1815
End
Begin VB.ComboBox cmbDOF
   Height          =   315
   Left            =   4200
   Style           =   2  'Dropdown List
   TabIndex        =   12
   Top             =   680
   Width           =   1095
End
Begin VB.ComboBox cmbNode
   Height          =   315
   ItemData        =   "start.frx":0000
   Left            =   2520
   List            =   "start.frx":0002
   Style           =   2  'Dropdown List
   TabIndex        =   11
   Top             =   680
   Width           =   855
End
Begin VB.TextBox txtValue
   Height          =   285
   Left            =   6240
   TabIndex        =   10
   Top             =   680
   Width           =   615
End
Begin VB.CommandButton cmdSelect
   Caption         =   "&Select"
   Height          =   495
   Left            =   9240
   TabIndex        =   8
   Top             =   2160
   Width           =   1095
End
Begin VB.TextBox txtFile
   Height          =   285
   Left            =   6600
   TabIndex        =   5
   Top             =   240
   Width           =   1335
End
Begin VB.TextBox txtPath
   Height          =   285
   Left            =   1200
   TabIndex        =   4
   Top             =   240
   Width           =   5415
End
Begin VB.CommandButton cmdEnd
   Caption         =   "&End"
   Height          =   495
   Left            =   9240
   TabIndex        =   2
   Top             =   4560
   Width           =   1095
End
Begin VB.PictureBox Picture1
   Appearance      =   0  'Flat
   AutoSize        =   -1  'True
   BackColor       =   &H80000005&
   ClipControls    =   0   'False
   ForeColor       =   &H80000008&
   Height          =   6000
```

```
      Left             =    600
      Negotiate        =    -1  'True
      ScaleHeight      =    -65000
      ScaleLeft        =    -5
      ScaleMode        =    0  'User
      ScaleTop         =    60000
      ScaleWidth       =    127.413
      TabIndex         =    1
      Top              =    1680
      Width            =    8280
   End
   Begin VB.CommandButton cmdAnal
      Caption          =    "&Analysis"
      Enabled          =    0   'False
      Height           =    495
      Left             =    9240
      TabIndex         =    0
      Top              =    2880
      Width            =    1095
   End
   Begin VB.Label Label11
      Caption          =    "Displacement"
      Height           =    255
      Left             =    4560
      TabIndex         =    25
      Top              =    7800
      Width            =    975
   End
   Begin VB.Label Label10
      Caption          =    "Load"
      Height           =    255
      Left             =    120
      TabIndex         =    24
      Top              =    4080
      Width            =    375
   End
   Begin VB.Label lblStatus
      Height           =    255
      Left             =    9000
      TabIndex         =    20
      Top              =    5400
      Width            =    1095
   End
   Begin VB.Label Label8
      Caption          =    "0,0"
      Height           =    255
      Left             =    240
      TabIndex         =    19
      Top              =    7680
      Width            =    255
   End
   Begin VB.Label Label7
      Caption          =    "Load:"
      Height           =    255
      Left             =    9120
      TabIndex         =    16
      Top              =    5760
      Width            =    495
   End
   Begin VB.Label Label6
      Caption          =    " Disp:"
      Height           =    255
      Left             =    9120
      TabIndex         =    15
      Top              =    6120
      Width            =    495
   End
   Begin VB.Label Label5
      Caption          =    "Base load:"
      Height           =    255
      Left             =    7560
```

```
      TabIndex        =   13
      Top             =   720
      Width           =   855
   End
   Begin VB.Label Label4
      Caption         =   "Value:"
      Height          =   255
      Left            =   5640
      TabIndex        =   9
      Top             =   720
      Width           =   615
   End
   Begin VB.Label Label3
      Caption         =   "DOF"
      Height          =   255
      Left            =   3720
      TabIndex        =   7
      Top             =   720
      Width           =   375
   End
   Begin VB.Label Label2
      Caption         =   "Control dispalcement:    Node"
      Height          =   255
      Left            =   360
      TabIndex        =   6
      Top             =   720
      Width           =   2175
   End
   Begin VB.Label Label1
      Caption         =   "File:"
      Height          =   255
      Left            =   360
      TabIndex        =   3
      Top             =   240
      Width           =   615
   End
End
Attribute VB_Name = "frmMain"
Attribute VB_GlobalNameSpace = False
Attribute VB_Creatable = False
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False

'==================== End of displaying parameters
Private Sub cmdAnal_Click()
'*****************************************************************************
'* Analysis button was clicked. Starting point of analysis
'*****************************************************************************
  Dim a As Double, b As Integer

  If cmbLoad.Text = "" Then
     MsgBox "Please choose a base load for calculation."
     Exit Sub
  End If

  If txtMaxDisp.Text = "" Then
     MsgBox "Please enter the maximum displacemnt value."
     Exit Sub
  Else
     Picture1.ScaleLeft = 0
     Picture1.ScaleWidth = Val(txtMaxDisp.Text)
  End If

  a = Val(txtValue.Text)                                'Inc. disp. value
  b = (Val(cmbNode.Text) - 1) * DOF + _                 'Location of the control
                          cmbDOF.ListIndex + 1          'parameter in the stiffness
                                                        'matrix
   If a <> 0 Then
    lblStatus.Caption = "Processing ..."
    lblStatus.Refresh
    Call DispControl(a, b)
```

```
    cmdSelect.Enabled = True
    cmdAnal.Enabled = False
    txtFile.Enabled = True
    txtPath.Enabled = True
    txtValue.Enabled = False
    cmbNode.Enabled = False
    cmbDOF.Enabled = False
    cmdDraw.Enabled = True
    cmbLoad.Enabled = False
  Else
    MsgBox "Control displacement must have a value.", vbCritical
  End If
  Close #1: Close #2
  lblStatus.Caption = "Maximum:"
End Sub

Private Sub cmdDraw_Click()
'*****************************************************************************
'* Draw button was clicked. Plots a Load vs. displacement plot
'*****************************************************************************
  Dim i As Integer
  Dim a As Single, b As Single

  Dim outfile As String

   If txtMaxLoad.Text = "" Or txtMaxDisp.Text = "" Then
     MsgBox "Both maximum load and displacemnt limits must be specified.", vbCritical
     Exit Sub
   End If

   With Picture1
      .ScaleTop = Val(txtMaxLoad.Text)
      .ScaleHeight = -Val(txtMaxLoad.Text)
      .ScaleLeft = 0
      .ScaleWidth = Val(txtMaxDisp.Text)
  End With

  outfile$ = txtPath.Text + "\" + txtFile.Text + "." + "out"
  Open outfile$ For Input As #1
  Do
     Input #1, a, b
     Picture1.PSet (b, a)
  Loop Until EOF(1)
  Close #1
End Sub

Private Sub cmdRefesh_Click()
'*****************************************************************************
'* Refresh button was clicked. Refreshes the drawing box.
'*****************************************************************************
   Picture1.Refresh
End Sub

Private Sub cmdSelect_Click()
'*****************************************************************************
'* Select button was clicked. Opens and reads the input file.
'* Initializing starts here.
'*****************************************************************************
  Dim inFile As String, outfile As String
  Dim t As String, t2 As String

  On Error GoTo Errhandler

  inFile$ = txtPath.Text + "\" + txtFile.Text + "." + "in"
  outfile$ = txtPath.Text + "\" + txtFile.Text + "." + "out"

  Open inFile$ For Input As #1
  Open outfile$ For Output As #2

  Call OmitCmnt("SYSTEM")
  Input #1, nNode, nElm, nProp
```

```
 If nNode = 0 Or nElm = 0 Or nProp = 0 Then
    MsgBox "Error: 00"
    End
 End If

 cmdSelect.Enabled = False
 cmdAnal.Enabled = True
 txtFile.Enabled = False
 txtPath.Enabled = False
 cmbNode.Enabled = True
 cmbDOF.Enabled = True
 cmbLoad.Enabled = True
 txtValue.Enabled = True

 cmbNode.Clear
 For i = 1 To nNode
   cmbNode.AddItem i
 Next

 Call OmitCmnt("CONTROL")
 Input #1, t: cmbNode.ListIndex = Val(t) - 1
 Input #1, t: cmbDOF.ListIndex = Val(t) - 1
 Input #1, t: txtValue.Text = t
 Input #1, t: txtMaxDisp.Text = t
 Input #1, t: txtMaxLoad.Text = t
 On Error GoTo 0

 Call Initialize
 Exit Sub

Errhandler:
  If Err.Number = 53 Or Err.Number = 76 Then
      msg = "The file (or path) you entered can not be found."
      Button = vbOKOnly + vbCritical
      Title = "Error: Missing input file"
      MsgBox msg, Button, Title
      Exit Sub
  Else
     MsgBox "Error no." & Err.Number
     End
  End If
End Sub

Private Sub Form_Load()
'*****************************************************************************
'* The first event that runs.
'* It does just some simple variable initiation.
'*****************************************************************************
   cmbDOF.AddItem "u (Fx)"
   cmbDOF.AddItem "v (Fy)"
   cmbDOF.AddItem "w (Fz)"
   cmbDOF.AddItem "Rx (Mx)"
   cmbDOF.AddItem "Ry (My)"
   cmbDOF.AddItem "Rz (Mz)"
   txtPath.Text = CurDir()
End Sub

Private Sub txtFile_Change()
'*****************************************************************************
'* User has altered file name field.
'*****************************************************************************
   cmbNode.Enabled = False
   cmbDOF.Enabled = False
   txtValue.Enabled = False
   cmdDraw.Enabled = False
End Sub

Private Sub txtPath_Change()
'*****************************************************************************
'* Use has altered directory name field.
```

```
'*********************************************************************************
    cmbNode.Enabled = False
    cmbDOF.Enabled = False
    txtValue.Enabled = False
    cmdDraw.Enabled = False
End Sub

Private Sub cmdEnd_Click()
'*********************************************************************************
'* End button was clicked. Ends the program.
'*********************************************************************************
    End
End Sub
```

B.2.3 Example

The program when it is running the problem 1 of chapter 5 is shown in Fig. B.2. During analysis, Select and Draw buttons are deactivated. The final stage of solving this problem is shown in Fig. B.3. After analysis, Analysis button is deactivated.

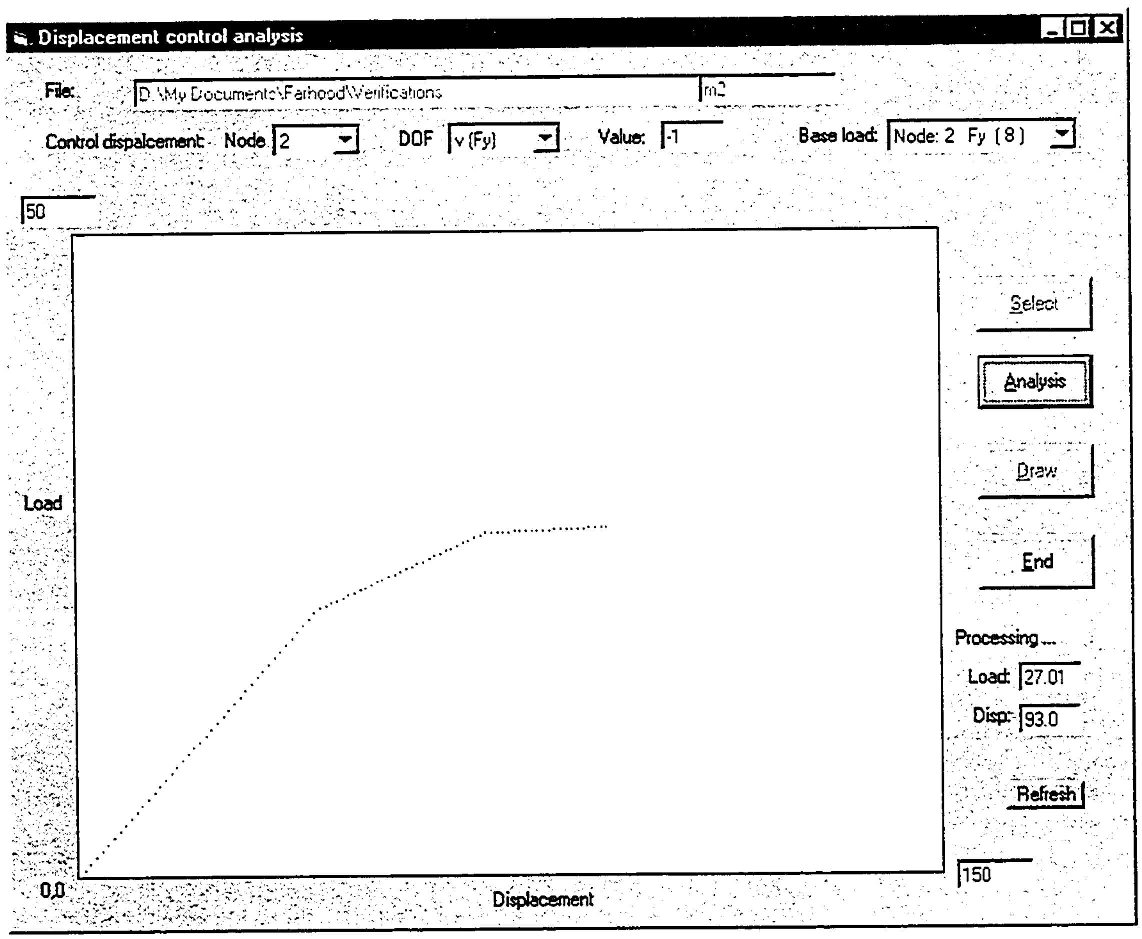

Fig. B.2 Program intermediate screen for problem 1

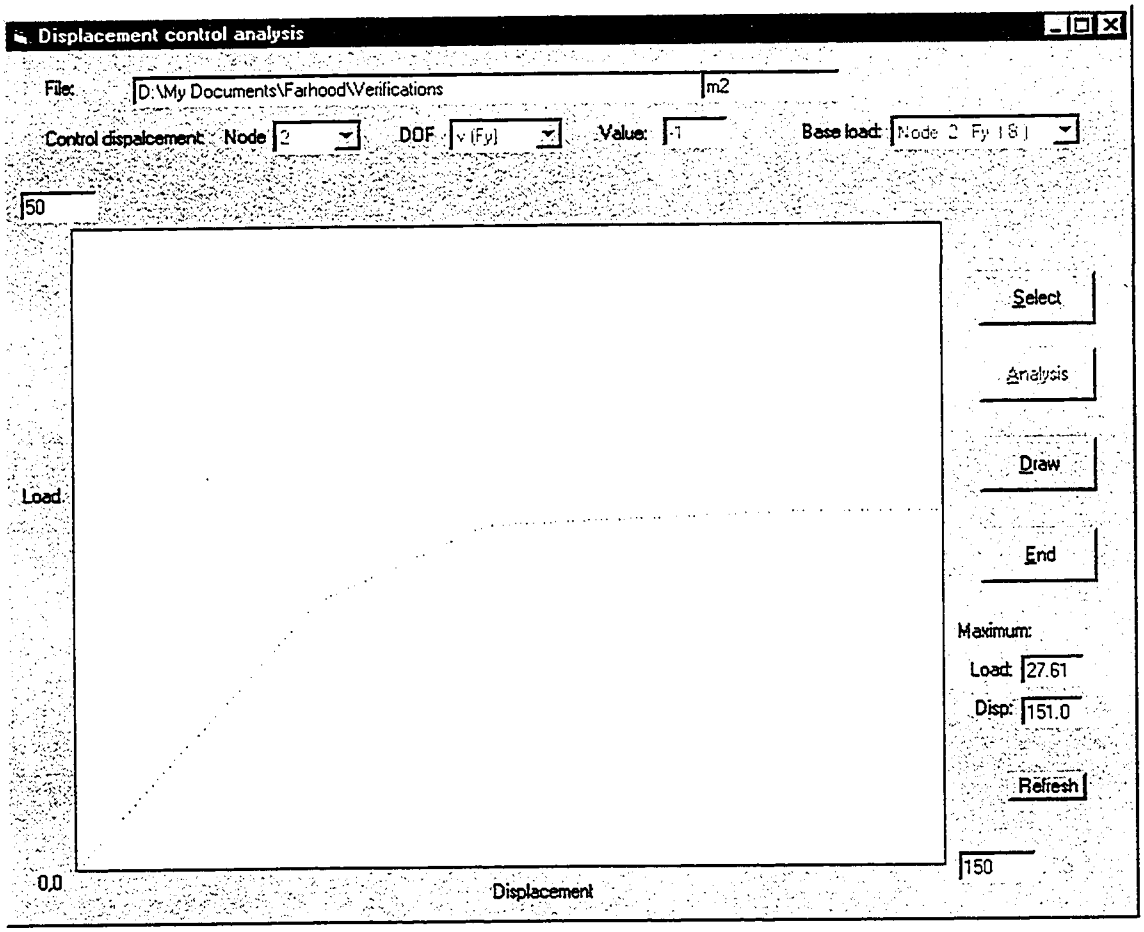

Fig. B.3 Program screen after problem 1 was solved

B.3 Sample Input file

PIPLAST accepts a free format for the input file. Every line starting with ' is a comment and is omitted. The input file for problem 1 of chapter five is as follows:

```
'          Number of
' nodes   elements    properties
System
3,  2,  1


'
' 1st line Displacement : Node, DOF, value (global coordinate), |Max|
' 2nd line load: |Max| (unit / 1000)
CONTROL
2, 2, -1, 150
50

'Corx  Cory  Corz   restraints              "No choice for ID"
Joints
0,     0, 0, 111111
4000, 0, 0, 001010
6000, 0, 0, 111111


'E        Nu   radius      THK   Fy    wt/len
Property
200000,  .3,   50.8, 5.74, 350,  0


'nodei   nodej     propid       "No choice for ID"
Elements
1, 2,    1
2, 3,    1
3, 4,    1


' Load ratio entry (Based on global coordinate)
'ID      Fx    Fy    Fz    Mx    My    Mz
JLOADS
2, 0,-1,0,0,0,0


'Internal pressure in the pipe
PRESSURE
0
```

B.4 Modules Code List

All necessary comments are provided in the code. When applicable, the relevant sections of the thesis are referred.

B.4.1 Initializing Module

```
Attribute VB_Name = "Initializing"
'******************************************************************************
'* This is a part of PIPLAST Version 1.0 written by Farhood Nowzartash.
'* Copyright © by Farhood Nowzartash 2002
'******************************************************************************
Option Explicit
Public Sub Initialize()
'******************************************************************************
'* Initializes the matrices.
'******************************************************************************
  Dim i As Integer, n As Integer, t As String

 '============== Dynamic objects memmory allocation
  ReDim elms(1 To nElm)
  ReDim jnts(1 To nNode)
  ReDim props(1 To nProp)

 '============== Dynamic matrices memory allocation
  ReDim gK(1 To nNode * DOF, 1 To nNode * DOF)
  ReDim gD(1 To nNode * DOF)
  ReDim gDelD(1 To nNode * DOF)
  ReDim gP(1 To nNode * DOF)
  ReDim gF(1 To nNode * DOF)
  ReDim NodeYld(1 To 2, 1 To nNode)       'Yield status of elements ends
                                          'connected to a node

 '============== Reading JOINT block from input file
  Call OmitCmnt("JOINTS")
  For i = 1 To nNode
     Input #1, t: jnts(i).CorX = Val(t)
     Input #1, t: jnts(i).CorY = Val(t)
     Input #1, t: jnts(i).CorZ = Val(t)
     Input #1, t: jnts(i).Restraint = t
  Next

 '============== Reading PROPERTY block from input file
  Call OmitCmnt("PROPERTY")
  For i = 1 To nProp
     Input #1, t: props(i).Elasticity = Val(t)
     Input #1, t: props(i).Nu = Val(t)
     Input #1, t: props(i).Radius = Val(t)
     Input #1, t: props(i).Thickness = Val(t)
     Input #1, t: props(i).UltStr = Val(t)
     Input #1, t: props(i).WtPLen = Val(t)
  Next i

 '============== Reading ELEMENT block from input file
  Call OmitCmnt("ELEMENTS")
  For i = 1 To nElm
     elms(i).ElmID = i
     Input #1, t: elms(i).nodei = Val(t)
     Input #1, t: elms(i).nodej = Val(t)
     Input #1, t: elms(i).Property = Val(t)
  Next i
```

```
'============== Reading joint load block from input file
 Call OmitCmnt("JLOADS")
 Do
    Input #1, n
    If n = 0 Then Exit Do
    Input #1, t: jnts(n).Forces(1) = Val(t)
    Input #1, t: jnts(n).Forces(2) = Val(t)
    Input #1, t: jnts(n).Forces(3) = Val(t)
    Input #1, t: jnts(n).Forces(4) = Val(t)
    Input #1, t: jnts(n).Forces(5) = Val(t)
    Input #1, t: jnts(n).Forces(6) = Val(t)

    frmMain.cmbLoad.Clear
    For i = 1 To 6
      If jnts(n).Forces(i) <> 0 Then
         Select Case i
            Case 1
              t = "Fx"
            Case 2
              t = "Fy"
            Case 3
              t = "Fz"
            Case 4
              t = "Mx"
            Case 5
              t = "My"
            Case 6
              t = "Mz"
          End Select
         frmMain.cmbLoad.AddItem "Node: " & n & "    " _
                                  & t & "  (" & Str((n - 1) * DOF + i) & " )"
      End If
    Next
 Loop
'============== Reading PRESSURE block from input file
 Call OmitCmnt("PRESSURE")
 Input #1, t
 For i = 1 To nElm
    elms(i).PressValue = Val(t)
 Next

'======= Initializing global applid load vector
 For n = 1 To nNode
   For i = 1 To DOF
       gP((n - 1) * DOF + i) = gP((n - 1) * DOF + i) + jnts(n).Forces(i)

       If Mid(jnts(n).Restraint, i, 1) = "1" Then
           gP((n - 1) * DOF + i) = 0        ' No load alloawable on supports
       End If
   Next i
 Next n
End Sub


Public Sub OmitCmnt(block$)
'**********************************************************
'* Makes it possible to use comments in input file.
'**********************************************************
 Dim tmp$, job$

 Line Input #1, tmp$
 Do: Line Input #1, tmp$: Loop While Mid$(tmp$, 1, 1) = "'" Or tmp$ = ""

 job$ = tmp$
 If UCase$(job$) <> block$ Then
   MsgBox ("Error in " + block$ + " block.")
 End If
End Sub
```

B.4.2 Analysis Module

```
Attribute VB_Name = "Analysis"
'*****************************************************************************
'* This is a part of PIPLAST Version 1.0 written by Farhood Nowzartash.
'* Copyright © by Farhood Nowzartash 2002
'*****************************************************************************

'*****************************************************************************
'*  Declaration of public variables.
'*  This variables are shared in the whole program.
'*****************************************************************************
Option Explicit
Public Const DOF = 6                 ' No of DOF per node
Public Const Pi As Double = 3.14159265358979

Public gK() As Double                ' Global stiffness matrix
Public gP() As Double                ' Global applied force
Public gF() As Double                ' Global internal force
Public gR() As Double                ' Global unbalance force
Public gD() As Double                ' Global displacements
Public gDelD() As Double             ' Global incremental displacements
Public nNode As Integer              ' Number of nodes
Public nElm As Integer               '   "       Elements
Public nProp As Integer              '   "       properties
Public elms() As New Element         ' Element objects
Public jnts() As New Joint           ' Joint objects
Public props() As New Property       ' Property objects
'
Private Sub BCImpose()
'*****************************************************************************
'* Imposes Boundary conditions.
'*
'*  Input : Nothing
'* Output : Nothing (Changes on a Public array)
'*****************************************************************************
  Dim i As Integer, j As Integer, a As Integer

  For i = 1 To nNode
     For j = 1 To DOF
       If Mid$(jnts(i).Restraint, j, 1) = "1" Then
          For a = 1 To nNode * DOF
             gK((i - 1) * DOF + j, a) = 0: gK(a, (i - 1) * DOF + j) = 0
          Next a
          gK((i - 1) * DOF + j, (i - 1) * DOF + j) = 1: gR((i - 1) * DOF + j) = 0
       End If
     Next j
  Next i
End Sub

Public Sub DispControl(CtrlDisp As Double, CtrlId As Integer)
'*****************************************************************************
'* Analyzes the structure by using displacement control (DC) method.
'* For DC scheme see section 4.5 of the thesis.
'*
'*  Inputs:
'*        CtrlDisp : Incremental value of control displacement parameter
'*        CtrlId   : Control parameter DOF according to the whole system
'*  Output:
'*        A drawing and a file. (BLoadID vs. CtrlDisp)
'*****************************************************************************
  Dim i As Integer, j As Integer                  'general use variables
  Dim ii As Integer, jj As Integer                '  "       "       "
  Dim time As Single
  Dim DrawFlg As Boolean                          'Drawing need flag
  Dim duI() As Double, duII() As Double           'displacement components
  Dim gP1() As Double, gR1()
  Dim dEta As Double                              'incremental load factor
```

```
Dim BLoadID As Integer                          'base load ID
Dim K11() As Double, K12() As Double            'partitioned stiffness
Dim K21() As Double, K22 As Double              'matrix parts
Dim delta As Double                             'value of control parameter

ReDim K11(1 To nNode * DOF - 1, 1 To nNode * DOF - 1)
ReDim K12(1 To nNode * DOF - 1)
ReDim K21(1 To 1, 1 To nNode * DOF - 1)
ReDim gP1(1 To nNode * DOF - 1)                 'reduced load vector
ReDim gR1(1 To nNode * DOF - 1)                 'reduced unbalanced load
delta = 0                                       'Initial value

'============ Extract BloadID from the Form
i = InStr(frmMain.cmbLoad.Text, "(")
j = InStr(frmMain.cmbLoad.Text, ")")
BLoadID = Val(Mid(frmMain.cmbLoad.Text, i + 1, j - 1))

'============ set the scale of drawing box
If frmMain.txtMaxLoad.Text <> "" Then
   DrawFlg = True
   With frmMain
       .Picture1.ScaleTop = Val(.txtMaxLoad.Text)
       .Picture1.ScaleHeight = -Val(.txtMaxLoad.Text)
   End With
End If

'============== Displacement control main loop
Do
   Call globalK
   Call GlobalForces
   Call BCImpose

   '============ Check the exsistance of unbalanced load

   If Not UnBalLoad() Then                  ' System is in equilibrium.
      delta = CtrlDisp
      If DrawFlg Then
         frmMain.Picture1.PSet (Abs(gD(CtrlId)), Abs(gF(BLoadID) / 1000))
      End If

      Debug.Print Format(Abs(gF(BLoadID)), "#,##0"), Format(Abs(gD(CtrlId)), "#0.00")
      Print #2, Format(Abs(gF(BLoadID) / 1000), "##0.0"), _
                                    Format(Abs(gD(CtrlId)), "##.0")
   Else
                                            ' There is an unbalanced load.
      delta = 0
   End If

   '============== Calculate load factor and displacement componnets
   GoSub Partition

   duI = MatSysSolv(K11, gP1)
   duII = MatSysSolv(K11, MatAdd(gR1, MatMulNum(K12, -delta)))

   dEta = -gR(CtrlId) + MatMulVec(K21, duII)(1) + K22 * delta
   dEta = dEta / (gP(CtrlId) - MatMulVec(K21, duI)(1))
   ii = 0

   '============== Calculate displacement vector
   For i = 1 To nNode * DOF
      If Not i = CtrlId Then
        ii = ii + 1
        gDelD(i) = dEta * duI(ii) + duII(ii)
      End If
   Next i
   gDelD(CtrlId) = delta

   For i = 1 To nNode * DOF
      gD(i) = gD(i) + gDelD(i)
   Next I
```

```
    '============== Update elements matrices
     For i = 1 To nElm: elms(i).Update: Next i

    '============== Refresh informations shown on the Form
     If Timer > time + 0.1 Then
        frmMain.txtLoad.Text = Format(Abs(gF(BLoadID)) / 1000, "#,###.0#")
        frmMain.txtLoad.Refresh
        frmMain.txtDisp.Text = Format(Abs(gD(CtrlId)), "#,###.0#")
        frmMain.txtDisp.Refresh
        time = Timer
     End If
  Loop Until Abs(gD(CtrlId)) > frmMain.Picture1.ScaleWidth
  Debug.Print "End of analysis"
  Exit Sub

'=========== Extracting reduced matrices for DC method.
Partition:
  ii = 0
  For i = 1 To nNode * DOF
    jj = 0
    If i <> CtrlId Then ii = ii + 1
    For j = 1 To nNode * DOF
       If j <> CtrlId Then jj = jj + 1
       If i <> CtrlId And j <> CtrlId Then
          K11(ii, jj) = gK(i, j)
       ElseIf i = CtrlId And j <> CtrlId Then
          K21(1, jj) = gK(CtrlId, j)
       ElseIf j = CtrlId And i <> CtrlId Then
          K12(ii) = gK(i, CtrlId)
          gP1(ii) = gP(i)
          gR1(ii) = gR(i)
       End If
    Next j
  Next i
  K22 = gK(CtrlId, CtrlId)
  Return
End Sub

Private Sub GlobalForces()
'*****************************************************************************
'* Constructs global force matrices. (gF() and gR())
'*
'*  Input : Nothing
'* Output : Nothing ( Changes on a Public array)
'*****************************************************************************
  Dim i As Integer, n As Integer
  Dim fn As Integer, sn As Integer          ' 1st & 2nd nodes ID

  Dim gtF() As Double                       'temporary matrix
  ReDim gtF(1 To DOF * nNode)               'this kind of definition is needed
                                            'because of VB restrictions
  For i = 1 To DOF * nNode
    gF(i) = 0: gtF(i) = 0
  Next

 '================ Extract elements internal forces
   For n = 1 To nElm
     fn = elms(n).nodei
     sn = elms(n).nodej
     For i = 1 To DOF
        gF((fn - 1) * DOF + i) = gF((fn - 1) * DOF + i) + elms(n).InternalForce(i)
        gF((sn - 1) * DOF + i) = gF((sn - 1) * DOF + i) + _
                                               elms(n).InternalForce(DOF + i)
         gtF((fn - 1) * DOF + i) = gtF((fn - 1) * DOF + i) + elms(n).TotalForce(i)
        gtF((sn - 1) * DOF + i) = gtF((sn - 1) * DOF + i) + elms(n).TotalForce(DOF + i)
     Next i
  Next n

  '================= Caculate unbalaced load vector
  gR = MatAdd(gtF, MatMulNum(gF, -1))
End Sub
```

```
Private Sub globalK()
'*****************************************************************************
'* Assembles the global stiffness matrix.
'*
'*  Input : Nothing
'* Output : Nothing ( Changes on a Public array)
'*****************************************************************************
  Dim n As Integer, a As Integer, b As Integer
  Dim i As Integer, j As Integer
  Dim eK() As Double                               ' pointer to an element
                                                   ' stiffness matrix
  ReDim gK(1 To nNode * DOF, 1 To nNode * DOF)     ' dynamically memory assignment

  For n = 1 To nElm                                ' No of elements
     eK = elms(n).TangentK
     a = elms(n).nodei - 1: b = elms(n).nodej - 1
     For i = 1 To DOF
        For j = 1 To DOF
           AddEqual gK(DOF * a + i, DOF * a + j), eK(1, i, j)
           AddEqual gK(DOF * a + i, DOF * b + j), eK(2, i, j)
           AddEqual gK(DOF * b + i, DOF * a + j), eK(3, i, j)
           AddEqual gK(DOF * b + i, DOF * b + j), eK(4, i, j)
         Next j
     Next i
  Next n
  Erase eK
End Sub

Private Function UnBalLoad() As Boolean
'*****************************************************************************
'* Check the existance of unbalanced load.
'* For convergence criteria see section 4.4.3 of the thesis.
'*
'*  Input : Nothing
'* Output : True if unbalance load is higher than the acceptable tolerance
'*****************************************************************************
  Dim i As Integer, j As Integer, tmp As Double        'general use variables
  Dim Tolerance As Double
  Dim max As Double                                    'largest memeber of unbalanced
                                                       'load vector
  Tolerance = 0.0001

  max = Abs(gP(1))
  For i = 2 To nNode * DOF
     If Abs(gP(i)) > max Then max = Abs(gP(i))
  Next

  For i = 1 To nNode * DOF
     tmp = tmp + Abs(gR(i) / max)
  Next i

  If tmp / (nNode * DOF) > Tolerance Then
     UnBalLoad = True
     Exit Function
  End If
  UnBalLoad = False
End Function

Public Sub AddEqual(num1 As Variant, num2 As Variant)
'*****************************************************************************
'* Increases the first number by the amount of the second one.
'*****************************************************************************
  num1 = num1 + num2
End Sub
```

B.4.3 Matrix Module

```
Attribute VB_Name = "matrix"
'******************************************************************************
'* This is a part of PIPLAST Version 1.0 written by Farhood Nowzartash.
'* Copyright © by Farhood Nowzartash 2002
'******************************************************************************
Option Explicit
Public MatDone As Boolean                    'No error detected
Private row() As Integer                     'used for pivoting

Private Function Dimension(Mat, dir As Integer)
'******************************************************************************
'* Find the dimension of a matrix
'*
'*   Input: Mat() Given matrix
'*          dir   asked coordinate
'*  Output: matrix dimension
'******************************************************************************
On Error GoTo exceed

  Dimension = UBound(Mat, dir)
  Exit Function

exceed:
   Dimension = 1
End Function

Private Function Gauss(ByVal Mat, opt As String, Optional ByVal Vec) As Double()
'******************************************************************************
'* Gauss method to solve a system or invert a matrix.
'*
'*   Input: Mat() Given Matrix [Coeficient matrix]
'*          opt   Operation flag
'*                  INV -> Invert given matrix
'*                  SLV -> Solve given equation system
'*                  DET -> Find determinant
'*          Vec() [Known values]        [Mat]*{X}={Vec}
'*  Output: Inverted matrix or {X}
'******************************************************************************
  Dim i As Integer, j As Integer, k As Integer
  Dim ub As Integer, tmp
  Dim rt As Double, Inv() As Double, ans() As Double
  Dim det(1 To 1) As Double                  ' Just to be consistant with
                                             ' the definition of the function
  MatDone = False
  det(1) = 1: opt = UCase(opt)

  ub = Dimension(Mat, 1)
  If opt = "INV" Then
     ReDim Inv(1 To ub, 1 To ub)
     For i = 1 To ub: Inv(row(i), i) = 1: Next ' Unit matrix
  End If

  For k = 1 To ub                               ' make Upper triangle
     For i = k + 1 To ub
       If Mat(row(k), k) = 0 Then
            If opt = "DET" Then
               det(1) = 0                       ' Determinant = zero
               MatDone = True: Gauss = det
               Exit Function
            End If
            MatDone = False                  ' try to fix it by changing rows
            Exit Function
       End If
       rt = Mat(row(i), k) / Mat(row(k), k)
       If rt <> 0 Then
          For j = k To ub
```

```
                Mat(row(i), j) = Mat(row(i), j) - rt * Mat(row(k), j)
             Next
          End If
          If UCase(opt) = "INV" Then
             For j = 1 To ub
                Inv(row(i), j) = Inv(row(i), j) - rt * Inv(row(k), j)
             Next
          End If
          If Not IsMissing(Vec) Then
             Vec(row(i)) = Vec(row(i)) - rt * Vec(row(k))
          End If
       Next i
    Next k

   If opt = "DET" Then
       For i = 1 To ub                                    ' find determinant
         det(1) = det(1) * Mat(row(i), i)
       Next
       Gauss = det: MatDone = True
       Exit Function
   ElseIf opt = "INV" Then
      If det(1) = 0 Then
          MsgBox "Matrix is singular", 16, "Matrix inversion"
          MatDone = False
          Exit Function
      End If
      For k = ub To 1 Step -1                        'Org Mat -> one diagonal
         For i = k - 1 To 1 Step -1
            rt = Mat(row(i), k) / Mat(row(k), k)
            For j = ub To 1 Step -1
               Inv(row(i), j) = Inv(row(i), j) - rt * Inv(row(k), j)
            Next j
         Next i
      Next k
      For i = 1 To ub                                'Org Mat -> unit vector
         For j = 1 To ub
            Inv(row(i), j) = Inv(row(i), j) / Mat(row(i), i)
         Next j
      Next i
      MatDone = True: Gauss = Inv
   ElseIf opt = "SLV" Then
      ReDim ans(1 To ub)
      For i = ub To 1 Step -1
         rt = 0
         For j = ub To i + 1 Step -1
            rt = rt + Mat(row(i), j) * ans(j)
         Next j
         ans(i) = (Vec(row(i)) - rt) / Mat(row(i), i)
      Next i
      MatDone = True: Gauss = ans
   End If
End Function

Public Function MatAdd(Mat1, Mat2) As Double()
'****************************************************************
'* Add two matrices together.
'*
'*   Input: Mat1() The first given matrix
'*          Mat2() The second matrix
'*  Output: Result matrix.
'****************************************************************
  Dim i As Integer, j As Integer
  Dim ub1 As Integer, ub2 As Integer
  Dim ans() As Double
  MatDone = False
  ub1 = Dimension(Mat1, 1): ub2 = Dimension(Mat1, 2)

  If ub1 <> Dimension(Mat2, 1) Or ub2 <> Dimension(Mat2, 2) Then
     MsgBox "Two matrix with different dimensions.", 16, "Matrix addition"
     Exit Function
  End If
```

```
  If ub2 = 1 Then
     ReDim ans(1 To ub1)
     For i = 1 To ub1
        ans(i) = Mat1(i) + Mat2(i)
     Next
  Else
     ReDim ans(1 To ub1, 1 To ub2)
     For i = 1 To ub1
        For j = 1 To ub2
          ans(i, j) = Mat1(i, j) + Mat2(i, j)
        Next j
      Next i
  End If

  MatDone = True: MatAdd = ans
End Function

Public Function MatDet(Mat) As Double
'******************************************************************************
'* Find determinant of a matrix by using Gauss method.
'*
'*   Input: Mat() Given matrix
'*  Output: Determinant of Mat()
'******************************************************************************
  Dim i As Integer, ub As Integer

  MatDone = False
  If Dimension(Mat, 1) <> Dimension(Mat, 2) Then
     MsgBox "Matrix must have equal dimension", 16, "Matrix uppertriangle"
     Exit Function
  End If

  ub = Dimension(Mat, 1)
  ReDim row(1 To ub)
  For i = 1 To ub: row(i) = i: Next i           ' No pivoting

  MatDet = Gauss(Mat, "det")(1)
End Function

Public Function MatInv(Mat) As Double()
'******************************************************************************
'* Find inverse of a matrix by using Gauss method.
'*
'*   Input: Mat() Given matrix
'*  Output: Inverted Mat()
'******************************************************************************
  Dim ub As Integer, i As Integer

  MatDone = False
  If Dimension(Mat, 1) <> Dimension(Mat, 2) Then
     MsgBox "Matrix must have equal dimension", 16, "Matrix uppertriangle"
     Exit Function
  End If
  ub = Dimension(Mat, 1)

  ReDim row(1 To ub)
  For i = 1 To ub: row(i) = i: Next i               ' No pivoting

  MatInv = Gauss(Mat, "inv")
End Function

Public Function MatMul(ByVal Mat1, ByVal Mat2) As Double()
'******************************************************************************
'* Multiply two matrices.
'*
'*   Input: Mat1() The first given matrix
'*          Mat2() The second matrix
'*  Output: Result matrix.
'******************************************************************************
  Dim i As Integer, j As Integer, k As Integer
  Dim ub1 As Integer, ub2 As Integer
```

```
  Dim ans() As Double
  MatDone = False

  ub1 = Dimension(Mat1, 1): ub2 = Dimension(Mat2, 2)

  If Dimension(Mat1, 2) <> Dimension(Mat2, 1) Then
      MsgBox "Mistmatch dimensions.", 16, "Matrix multiplication"
      Exit Function
  End If

  If ub2 = 1 Then
     ReDim ans(1 To ub1)
     For i = 1 To ub1
        For k = 1 To Dimension(Mat2, 1)
             ans(i) = ans(i) + Mat1(i, k) * Mat2(k)
        Next k
     Next i
  Else
     ReDim ans(1 To ub1, 1 To ub2)
     For i = 1 To ub1
        For j = 1 To ub2
          For k = 1 To Dimension(Mat2, 1)
             ans(i, j) = ans(i, j) + Mat1(i, k) * Mat2(k, j)
          Next k
        Next j
     Next i
  End If

  MatDone = True: MatMul = ans
End Function

Public Function MatMulNum(ByVal Mat, Num As Double) As Double()
'*****************************************************************************
'* Multiply a matrix by a number
'*
'*   Input: Mat() Given matrix
'*          Num   The number
'*  Output: Result matrix.
'*****************************************************************************
  Dim i As Integer, j As Integer
  Dim ub1 As Integer, ub2 As Integer
  Dim ans() As Double
  MatDone = False
  ub1 = Dimension(Mat, 1): ub2 = Dimension(Mat, 2)

  If ub2 = 1 Then
     ReDim ans(1 To ub1)
     For i = 1 To ub1
        ans(i) = Mat(i) * Num
     Next i
  Else
     ReDim ans(1 To ub1, 1 To ub2)
     For i = 1 To ub1
        For j = 1 To ub2
           ans(i, j) = Mat(i, j) * Num
        Next j
     Next i
  End If
  MatDone = True: MatMulNum = ans
End Function


Public Function MatMulVec(Mat, Vec) As Double()
'**************************************************************
'* Multiply a matrix and a vector.
'*
'*   Input: Mat() The given matrix
'*          Vec() The given vector
'*  Output: Result matrix.
'**************************************************************
   Dim i As Integer, j As Integer, k As Integer
```

```
   Dim ub1 As Integer, ub2 As Integer
   Dim ans() As Double
   MatDone = False

   ub1 = Dimension(Mat, 1): ub2 = 1

   If Dimension(Mat, 2) <> Dimension(Vec, 1) Then
      MsgBox "Mistmatch dimensions.", 16, "Matrix multiplication"
      Exit Function
   End If
   ReDim ans(1 To ub1)

   For i = 1 To ub1
      For k = 1 To Dimension(Vec, 1)
         ans(i) = ans(i) + Mat(i, k) * Vec(k)
      Next k
   Next i

   MatDone = True: MatMulVec = ans
End Function


Public Function MatSysSolv(Mat, Xmat) As Double()
'*******************************************************************************
'* Find inverse of a matrix by using Gauss method.
'*
'*   Input: Mat() Given matrix
'*  Output: Inverted Mat()
'*******************************************************************************
  Dim ub As Integer, i As Integer

  MatDone = False
  If Dimension(Mat, 1) <> Dimension(Mat, 2) Then
     MsgBox "Matrix must have equal dimension", 16, "Equation System solver"
     Exit Function
  End If
  ub = Dimension(Mat, 1)

  If MatDet(Mat) = 0 Then
      MsgBox "Cooficient matrix is singular.", 16, "Equation System solver"
      Exit Function
  Else
     Call Pivot(Mat)
     MatSysSolv = Gauss(Mat, "slv", Xmat)

     If Not MatDone Then
        For i = 1 To ub: row(i) = i: Next i
        MatSysSolv = Gauss(Mat, "slv", Xmat)
        If Not MatDone Then
             MsgBox "Matrix is singular", 16, "Equation System Solver"
        End If
     End If
  End If
End Function


Public Function MatTrans(ByVal Mat) As Double()
'*******************************************************************************
'* Treansposes a matrix.
'*
'*   Input: Mat() Given matrix
'*  Output: Result matrix.
'*******************************************************************************
  Dim i As Integer, j As Integer
  Dim ub1 As Integer, ub2 As Integer
  Dim ans() As Double
  MatDone = False

  ub1 = Dimension(Mat, 1): ub2 = Dimension(Mat, 2)
  ReDim ans(1 To ub2, 1 To ub1)
```

```
  For i = 1 To ub1
     For j = 1 To ub2
        ans(j, i) = Mat(i, j)
     Next j
  Next i

  MatDone = True: MatTrans = ans
End Function


Private Sub Pivot(Mat)
'*******************************************************************************
'* Puts the largest array members on main diagonal.
'*     - Useful to reduce errors in Gauss iteration.
'*
'*     - Input  : Given matrix
'*     - Output : Nothing
'*                It changes Row(): A pointer to show
'*                the rows location of pivoted matrix.
'*******************************************************************************
  Dim i As Integer, j As Integer, sel As Integer
  Dim ub As Integer
  Dim slct() As Boolean, satisfied As Boolean
  Dim buf As Double, tmp As Double

  ub = Dimension(Mat, 1)

  ReDim row(1 To ub): ReDim slct(1 To ub)

  For j = 1 To ub
     buf = 0
     For i = 1 To ub
        If Abs(Mat(i, j)) > buf And Not slct(i) And Mat(j, j) <> 0 Then
           sel = i: buf = Abs(Mat(i, j))
        End If
     Next i
     If sel Then row(j) = sel: slct(sel) = True: sel = 0
  Next j

  For i = 1 To ub
     If row(i) = 0 Then
        For j = 1 To ub
          If Not slct(j) Then row(i) = j
        Next j
     End If
  Next i

  j = 0
  Do                                 ' Check the main diagonal elements
     j = j + 1                       ' to be not zero.
     For i = 1 To ub
        If Mat(row(i), i) = 0 Then
           sel = row(i): row(i) = row(i - 1): row(i - 1) = sel
           satisfied = False
           Exit For
        End If
        satisfied = True
     Next
     If j > 2 * ub Then              ' Can not satisfy. Forget about pivoting
        For i = 1 To ub: row(i) = i: Next i
        Exit Do
     End If
  Loop Until satisfied
End Sub
```

B.5 Classes Code List

All necessary comments are provided in the code. When applicable, the relevant sections of the thesis are referred.

B.5.1 Element Class

```
VERSION 1.0 CLASS
'****************************************************************************
'* This is a part of PIPLAST Version 1.0 written by Farhood Nowzartash.
'* Copyright © by Farhood Nowzartash 2002
'****************************************************************************

BEGIN
  MultiUse = -1  'True
  Persistable = 0  'NotPersistable
  DataBindingBehavior = 0  'vbNone
  DataSourceBehavior  = 0  'vbNone
  MTSTransactionMode  = 0  'NotAnMTSObject
END
Attribute VB_Name = "Element"
Attribute VB_GlobalNameSpace = False
Attribute VB_Creatable = True
Attribute VB_PredeclaredId = False
Attribute VB_Exposed = False

Option Explicit
'****************************************************************************
'* Declaration of class level variables
'* This variables are shared in this class
'****************************************************************************
 Private Id As Integer                           ' Element ID
 Private PropId As Integer
 Private Fnode As Integer, Snode As Integer      ' 1st & 2nd node ID
 Private eL As Double                            ' Element lenght
 Private Pressure As Double                      ' Differential pressure

 Private Tau(1 To 2) As Double                   ' Circumferential shear stress ratio

 Private YieldFlg(1 To 2) As Boolean             ' last itaration plastic situation
 Private YldFlgChg(1 To 2) As Boolean            ' Plastic situation changed?
 Private YSCut(1 To 2) As Boolean                ' Stress resultant breached YS?

 Private Ke(1 To 2 * DOF, 1 To 2 * DOF) As Double ' Elastic stiffness
 Private Km() As Double                          ' Plastic reduction matrix

 Private Kt(1 To 4, 1 To DOF, 1 To DOF) As Double ' Tangent plastic Stiffness
                                                   (Global coordinate)
 Private Ktl() As Double                         ' same as [Kt] with local coordinate

 Private Ti(1 To 2 * DOF, 1 To 2 * DOF) As Double ' Transformation matrix
 Private eG(1 To 2 * DOF, 1 To 2) As Double      ' Normal to the YS

 Private dF() As Double                          ' Force increment
 Private tF() As Double                          ' Initial total force
 Private IntForce(1 To 2 * DOF) As Double       ' Intern. forces in the last iteration

 Private dD() As Double                          ' Displacement increment
 Private tD(1 To 2 * DOF) As Double              ' Total displacemnet
```

```
'*******************************************************************************
'* Property inlet ports. Changes/Set to the local variables
'*
'*      ElmId: Sets the element identification number
'*      nodei: Sets the element first node number
'*      nodej: Sets the element second node number
'* PressValue: Sets the internal/external pressure of the pipe
'*   Property: Sets the element property number
'*******************************************************************************
Friend Property Let ElmID(a As Integer)
  Id = a
End Property

Friend Property Let nodei(ByVal a As Integer)
   Fnode = a
   Call Create
End Property

Friend Property Let nodej(ByVal a As Integer)
   Snode = a
   Call Create
End Property

Friend Property Let PressValue(a As Double)
   Pressure = a
End Property

Friend Property Let Property(ByVal a As Integer)
   PropId = a
   Call Create
End Property

'*******************************************************************************
'* Property outlet ports. Makes variables accessible to the rest of the program.
'*
'* InternalForce: Returns the internal force vector based on the GCS
'*         nodei: Returns the element first node number
'*         nodej: Returns the element second node number
'*      TangentK: Returns the element stiffness matrix
'*    TotalForce: Returns total load vector based on GCS
'*******************************************************************************
Friend Property Get InternalForce()
   InternalForce = MatMul(MatTrans(Ti), IntForce)
End Property

Friend Property Get nodei() As Integer
   nodei = Fnode
End Property

Friend Property Get nodej() As Integer
   nodej = Snode
End Property

Friend Property Get TangentK() As Double()
   TangentK = Kt
End Property

Friend Property Get TotalForce()
   TotalForce = MatMul(MatTrans(Ti), tF)
End Property

Friend Property Get Yielded(node) As Boolean
     Yielded = YieldFlg(node)
End Property
```

```
Private Function ArcSin(x As Double) As Double
'******************************************************************************
'* Arc sinus function
'******************************************************************************
  If x = 1 Then
     ArcSin = Pi / 2
  ElseIf x = -1 Then
     ArcSin = -Pi / 2
  Else
     ArcSin = Atn(x / Sqr(-x * x + 1))
  End If
End Function


Private Function ActLoadInc(node As Integer) As Double ' node 1 = node i ; 2 = j
'******************************************************************************
'*    Determine the situation of the stress in a node and yield surface.
'*    If the point passes the yield surface returns it to the surface.
'*    If the point was already on YS the tangent to the surface cause
'*  an offset from YS. This routin returns it again to the surface.
'*
'*  newStrs : Stress ratios in new iteration
'*  oldStrs : Stress ratios after finishing last iteration
'*      tol : Acceptable tolerance for a stress point to be considered on YS
'*
'*   See section 4.4.2 and load control algorithm (Sec. 4.4.1) of the thesis.
'******************************************************************************
  Dim newStrs(1 To DOF) As Double
  Dim oldStrs(1 To DOF) As Double
  Dim Strs As Double
  Dim tol As Double
  Dim i As Integer, tmp As Integer
  tol = 0.00001

   For i = 1 To DOF
     tmp = DOF * (node - 1) + i
     oldStrs(i) = Abs(IntForce(tmp) / props(PropId).UltResist(i))
   Next
   For i = 1 To DOF
     tmp = DOF * (node - 1) + i
     newStrs(i) = Abs(tF(tmp) / props(PropId).UltResist(i))
   Next

   Strs = Stress(newStrs, node)
'   If Strs > 1 Then Stop                          ' added for debuging

   If Strs < 1 - tol Then                          'Elastic
      If YieldFlg(node) Then YldFlgChg(node) = True
      YieldFlg(node) = False
      ActLoadInc = 1#
   ElseIf Strs > 1 - tol And Strs < 1 + tol Then   'On the yield surface
      If Not YieldFlg(node) Then YldFlgChg(node) = True
      YieldFlg(node) = True
      ActLoadInc = 1#
   ElseIf Strs > 1 + tol Then                   ' Passed yield surface and _
      If YieldFlg(node) Then                    ' already on YS.
          Erase oldStrs                         ' Return the point to the YS _
                                                ' through radius. Put oldStrs=0
          ActLoadInc = FindRatio(oldStrs, newStrs, node)
      Else                                      ' Just passed the YS.
          ActLoadInc = FindRatio(oldStrs, newStrs, node)
          If Not YieldFlg(node) Then YldFlgChg(node) = True
          YieldFlg(node) = True
          YSCut(node) = True
      End If
  End If
End Function
```

```
Private Sub ChkJntMech(NodeId As Integer, endx As Integer)
'*************************************************************************
'* Prevents joint mechanism. See chapter 3.4 of the thesis.
'*
'* If all elements end connected to a joint plastify the last one can not
'* be considered as a plastified end. Otherwise a zero member on the main
'* diagonal of stiffness matrix will appear.
'**************************************************************************
  Dim i As Integer, j As Integer
  Dim tmp As String

  tmp = jnts(NodeId).Element(Id)
  If tmp = "NG" Then                      'All other elements end conneceted
     YieldFlg(endx) = False               'to this joint were plastified?
  End If
End Sub


Private Sub Create()
'**************************************************************************
'*  Initializes the object. (called just once)
'**************************************************************************
  Dim dX, dY, dZ                                    'element projections in GCS

  If Fnode * Snode * PropId = 0 Then Exit Sub       'All must be defined (>0)
  dX = jnts(Snode).CorX - jnts(Fnode).CorX
  dY = jnts(Snode).CorY - jnts(Fnode).CorY
  dZ = jnts(Snode).CorZ - jnts(Fnode).CorZ
  eL = Sqr(dX ^ 2 + dY ^ 2 + dZ ^ 2)

  Call ElasticStiffness
  Call TransformMatrix(dX / eL, dY / eL, dZ / eL)  'Called by direction cosines
  Call WtLoad(ArcSin(dY / eL))

  jnts(Fnode).Element(Id) = "i"
  jnts(Snode).Element(Id) = "j"

 '============== Dynamic memory allocation
  ReDim Km(1 To 2 * DOF, 1 To 2 * DOF)              '[Km] is ZERO at the begining
  ReDim dF(1 To 2 * DOF)
  ReDim tF(1 To 2 * DOF)
  ReDim tFi(1 To 2 * DOF)
  Call TangentStiffness
End Sub


Private Sub ElasticStiffness()
'**************************************************************************
'* Constructs the elastic stiffness matrix of the element
'* This matrix includes shear deformations.
'* See section 3.5 of the thesis for details.
'**************************************************************************
  Dim i As Integer, j As Integer
  Dim eI As Double, eA As Double, Nu As Double
  Dim eAr As Double, kapa As Double

  eI = props(PropId).Inertia
  eA = props(PropId).Area
  Nu = props(PropId).Nu
  eAr = (2 * (1 + Nu) / (4 + 3 * Nu)) * eA          ' Reduced area
  kapa = 24 * (1 + Nu) * eI / (eL ^ 2 * eAr)

  Ke(1, 1) = eA / eL
  Ke(1, 7) = -eA / eL
  Ke(2, 2) = 12 * eI / eL ^ 3
  Ke(2, 6) = 6 * eI / eL ^ 2
  Ke(2, 8) = -12 * eI / eL ^ 3
  Ke(2, 12) = 6 * eI / eL ^ 2
  Ke(3, 3) = 12 * eI / eL ^ 3
  Ke(3, 5) = -6 * eI / eL ^ 2
  Ke(3, 9) = -12 * eI / eL ^ 3
```

```
  Ke(3, 11) = -6 * eI / eL ^ 2
  Ke(4, 4) = eI / (1 + Nu) / eL
  Ke(4, 10) = -eI / (1 + Nu) / eL
  Ke(5, 5) = (4 + kapa) * eI / eL
  Ke(5, 9) = 6 * eI / eL ^ 2
  Ke(5, 11) = (2 - kapa) * eI / eL
  Ke(6, 6) = (4 + kapa) * eI / eL
  Ke(6, 8) = -6 * eI / eL ^ 2
  Ke(6, 12) = (2 - kapa) * eI / eL
  Ke(7, 7) = eA / eL
  Ke(8, 8) = 12 * eI / eL ^ 3
  Ke(8, 12) = -6 * eI / eL ^ 2
  Ke(9, 9) = 12 * eI / eL ^ 3
  Ke(9, 11) = 6 * eI / eL ^ 2
  Ke(10, 10) = eI / (1 + Nu) / eL
  Ke(11, 11) = (4 + kapa) * eI / eL
  Ke(12, 12) = (4 + kapa) * eI / eL

 '================= Fills the lower portion of the stiffness matrix
  For j = 2 To 2 * DOF
     For i = 1 To j
        Ke(j, i) = Ke(i, j)
     Next i
  Next j
  For i = 1 To 2 * DOF
     For j = 1 To 2 * DOF
        Ke(i, j) = props(PropId).Elasticity * Ke(i, j) / (1 + kapa)
     Next j
  Next i
End Sub


Private Function FindRatio(ByVal corl, ByVal coru, node As Integer) As Double
'*****************************************************************************
'*  Finds a ratio that if be multiplied by the main stresses the stress
'*  resultant will be laid to the YS without changing the ratio
'*  between them. (Same stiffness)
'*  Uses secant root finding method. See section 4.4.2 of the thesis.
'*****************************************************************************
  Dim Rou As Double, Rol As Double, Ror As Double
  Dim i As Integer
  Dim corr(1 To DOF) As Double
  Dim tol As Double
  Dim stru As Double, strl As Double, strr As Double
  Dim CorlBak() As Double, CoruBak() As Double

  tol = 0.0000001                ' cor = stress along each cordinate
  Rol = 0: Rou = 1               ' cor(1) = shear stress along x and ...

  CorlBak = corl
  CoruBak = coru

  Do
     stru = Stress(coru, node): strl = Stress(corl, node)
     Ror = Rou - (stru - 1) * (Rol - Rou) / (strl - stru)
     For i = 1 To DOF
        corr(i) = Ror * (CoruBak(i) - CorlBak(i)) + CorlBak(i)
     Next i
     strr = Stress(corr, node)

     If Sgn(strr - 1) = Sgn(strl - 1) Then
        corl = corr
        Rol = Ror
     ElseIf Sgn(strr - 1) = Sgn(stru - 1) Then
        coru = corr
        Rou = Ror
     End If
   Loop Until Abs(strr - 1) < tol
   FindRatio = Ror
End Function
```

```
Private Sub IntForces()
'*****************************************************************************
'* Calculates Internal nodal forces.
'* See section 4.4.2 of the thesis.
'*****************************************************************************
  Dim i As Integer
  Dim AccRatio(1 To 2) As Double
  Dim Ro As Double

  Erase eG: Erase YSCut
  AccRatio(1) = ActLoadInc(1)
  AccRatio(2) = ActLoadInc(2)


  If YSCut(1) Or YSCut(2) Then                    ' One end crossed the YS
     If Not (YieldFlg(1) And YieldFlg(2)) Then    ' Other node is still elastic
         If AccRatio(1) > AccRatio(2) Then Ro = AccRatio(2) Else Ro = AccRatio(1)
          For i = 1 To 2 * DOF
             IntForce(i) = tF(i) - (1 - Ro) * dF(i)
          Next
      Else                                        ' Other node was plastic already
          If YSCut(1) Then
              For i = 1 To DOF
                 IntForce(i) = tF(i) - (1 - AccRatio(1)) * dF(i)
                 IntForce(DOF + i) = AccRatio(2) * tF(DOF + i)
              Next i
          ElseIf YSCut(2) Then
              For i = 1 To DOF
                  IntForce(i) = AccRatio(1) * tF(i)
                  IntForce(DOF + i) = tF(DOF + i) - (1 - AccRatio(2)) * dF(DOF + i)
              Next i
          End If
      End If
  Else                          ' No transition (Both nodes are elastic or plastic)
      For i = 1 To DOF
         IntForce(i) = AccRatio(1) * tF(i)
         IntForce(DOF + i) = AccRatio(2) * tF(DOF + i)
      Next
  End If
End Sub


Private Sub PlasticReduction()
'*****************************************************************************
'* Constructs plastic reduction matrix [Km].
'* See section 3.6.2 of the thesis for details.
'*****************************************************************************
  Dim KeG() As Double                            ' all have temporary usage
  Dim GKeG() As Double
  Dim GKe() As Double
  Dim tmp() As Double

  GKeG = MatMul(MatMul(MatTrans(eG), Ke), eG)
  If GKeG(1, 1) = 0 And GKeG(2, 2) = 0 Then       ' Still Elastic
     ReDim Km(1 To 2 * DOF, 1 To 2 * DOF)
     Exit Sub
  End If
  KeG = MatMul(Ke, eG)
  GKe = MatMul(MatTrans(eG), Ke)

  If GKeG(2, 2) = 0 Then                  ' Just node i is plastic
     Km = MatMulNum(MatMul(KeG, GKe), -1 / GKeG(1, 1))
  ElseIf GKeG(1, 1) = 0 Then              ' Just node j is plastic
     Km = MatMulNum(MatMul(KeG, GKe), -1 / GKeG(2, 2))
  Else                                    ' Both nodes are plastic
     tmp = MatMul(MatMul(KeG, MatInv(GKeG)), GKe)
     Km = MatMulNum(tmp, -1)
  End If
  Erase GKeG, KeG, tmp
End Sub
```

```
Private Sub PlasticFlow()
'*****************************************************************************
'*  Constructs Gradiant to the surface matrix [G] which is used to find
'*  tangent to yield surface if the node is on the YS.
'*  See section 3.6.2 and appendix A1 of the thesis
'*****************************************************************************
  Dim Vp As Double, Ap As Double, Mp As Double, Tp As Double
  Dim Vr2 As Double, Vr3 As Double, Ar As Double
  Dim Mr2 As Double, Mr3 As Double, Tr As Double
  Dim Pr As Double
  Dim mu As Double, gama As Double, Fy As Double
  Dim dif_p2t As Double, dif_p1t As Double

  ' Tau() containts Tau_r, which is the circumferential shear stress ratio

  Ap = props(PropId).UltResist(1)
  Vp = props(PropId).UltResist(2)
  Tp = props(PropId).UltResist(4)
  Mp = props(PropId).UltResist(5)
  Pr = Pressure / props(PropId).UltResist(7)
  Fy = props(PropId).UltStr

  If YieldFlg(1) Then
       Ar = IntForce(1) / Ap
      Vr2 = IntForce(2) / Vp: Vr3 = IntForce(3) / Vp
       Tr = IntForce(4) / Tp
      Mr2 = IntForce(5) / Mp: Mr3 = IntForce(6) / Mp

       gama = Sqr(4 - 3 * Pr ^ 2 - 4 * Tau(1) ^ 2)
       mu = IntForce(1) / Ap - Pr / 2

       dif_p1t = -2 * (Vr2 ^ 2 + Vr3 ^ 2) / Tau(1) ^ 3 - _
                 0.5 * Pi * Tr * Sin(Pi * Tr / Tau(1)) / Tau(1) ^ 2
       dif_p2t = Tau(1) * (1 + Cos(2 * Pi * mu / gama) + _
                 Pi * mu * Sin(2 * Pi * mu / gama) / gama)

       eG(1, 1) = -0.25 * gama * Pi * Sin(2 * Pi * mu / gama) / Ap
       eG(1, 1) = eG(1, 1) * dif_p1t / dif_p2t
       eG(2, 1) = 2 * Vr2 / (Tau(1) ^ 2 * Vp)
       eG(3, 1) = 2 * Vr3 / (Tau(1) ^ 2 * Vp)
       eG(4, 1) = Pi * Sin(Pi * Tr / Tau(1)) / (2 * Tau(1) * Tp)
       eG(5, 1) = -(2 * Mr2 / Mp) * dif_p1t / dif_p2t
       eG(6, 1) = -(2 * Mr3 / Mp) * dif_p1t / dif_p2t
  End If

  If YieldFlg(2) Then
       Ar = IntForce(DOF + 1) / Ap
      Vr2 = IntForce(DOF + 2) / Vp: Vr3 = IntForce(DOF + 3) / Vp
       Tr = IntForce(DOF + 4) / Tp
      Mr2 = IntForce(DOF + 5) / Mp: Mr3 = IntForce(DOF + 6) / Mp

      gama = Sqr(4 - 3 * Pr ^ 2 - 4 * Tau(2) ^ 2)
      mu = IntForce(DOF + 1) / Ap - Pr / 2

      dif_p1t = -2 * (Vr2 ^ 2 + Vr3 ^ 2) / Tau(2) ^ 3 - _
                0.5 * Pi * Tr * Sin(Pi * Tr / Tau(2)) / Tau(2) ^ 2
      dif_p2t = Tau(2) * (1 + Cos(2 * Pi * mu / gama) + _
                Pi * mu * Sin(2 * Pi * mu / gama) / gama)

      eG(DOF + 1, 2) = -0.25 * gama * Pi * Sin(2 * Pi * mu / gama) / Ap
      eG(DOF + 1, 2) = eG(DOF + 1, 2) * dif_p1t / dif_p2t
      eG(DOF + 2, 2) = 2 * Vr2 / (Tau(2) ^ 2 * Vp)
      eG(DOF + 3, 2) = 2 * Vr3 / (Tau(2) ^ 2 * Vp)
      eG(DOF + 4, 2) = Pi * Sin(Pi * Tr / Tau(2)) / (2 * Tau(2) * Tp)
      eG(DOF + 5, 2) = -(2 * Mr2 / Mp) * dif_p1t / dif_p2t
      eG(DOF + 6, 2) = -(2 * Mr3 / Mp) * dif_p1t / dif_p2t
   End If
 End Sub
```

```
Private Function Stress(ByVal Strs, node As Integer) As Double
'*****************************************************************************
'* Calculates the resultant stress by having main stresses.
'* See section 2.4.2 of the thesis.
'*****************************************************************************
  Dim tol As Double
  Dim Vr As Double, Tr As Double
  Dim Mr As Double, Pr As Double, Ar As Double
  Dim xl As Double, xr As Double, xu As Double
  Dim fl As Double, Fr As Double, fu As Double
  Dim gama As Double

  tol = 0.000001

  Ar = Strs(1)                                          ' Axial force
  Vr = Sqr(Strs(2) ^ 2 + Strs(3) ^ 2)                   ' Shear stress
  Tr = Strs(4)                                          ' Torsion
  Mr = Sqr(Strs(5) ^ 2 + Strs(6) ^ 2)                   ' Bending moments
  Pr = Pressure / props(PropId).UltResist(7)            ' Pressure

  If Vr = 0 Then                                        ' No shear
     Tau(node) = Tr
  ElseIf Tr = 0 Then                                    ' No torsion
     Tau(node) = Vr
  Else
                                               ' Both shear and torsion
     '============== Secant root finding method to find Tau
     If Vr > Tr Then xl = Vr Else xl = Tr
     xu = 2#
     fl = Vr / xl - Cos(Pi * Tr / 2 / xl)
     fu = Vr / xu - Cos(Pi * Tr / 2 / xu)

     Do
        xr = xu - fu * (xl - xu) / (fl - fu)
        Fr = Vr / xr - Cos(Pi * Tr / 2 / xr)
        If Sgn(Fr) = Sgn(fl) Then
            fl = Fr: xl = xr
        Else
            fu = Fr: xu = xr
        End If
     Loop Until Abs(Fr) < tol

     If xr > 1 Then Stress = xr: Exit Function    ' Shear ratio is more than unity
     Tau(node) = xr
  End If

 '=========== Applicability limits check

  If Tau(node) > 1 Then
      MsgBox "Error: Large shear stress.", vbCritical, "Stress calculating"
      End
  End If

  gama = Sqr(4 - 3 * Pr ^ 2 - 4 * Tau(node) ^ 2)

  If Pr < -2 * Sqr((1 - Tau(node) ^ 2) / 3) Or _
          Pr > 2 * Sqr((1 - Tau(node) ^ 2) / 3) Then
      MsgBox "Error: Large axial stress.", vbCritical, "Stress calculating"
      End
  End If

  If Ar < (-Pr - gama) / 2 Or Ar > (Pr + gama) / 2 Then
      MsgBox "Error: Large axial stress.", vbCritical, "Stress calculating"
      End
  End If

  Stress = 1 + Mr ^ 2 - (gama ^ 2 / 4) * (Cos(Pi * (Ar - Pr / 2) / gama)) ^ 2

End Function
```

```
Private Sub TangentStiffness()
'*******************************************************************************
'* Constructs stiffness matrix of the element
'*                                   1 | 2
'* and devides Kt into 4 parts       - - -
'*                                   3 | 4
'* See section 3.6.3 of the thesis
'*******************************************************************************
  Dim i As Integer, j As Integer, m As Integer, n As Integer
  Dim Ktmp() As Double

  Ktl = MatAdd(Km, Ke)

  Ktmp = MatMul(MatMul(MatTrans(Ti), Ktl), Ti)

  For n = 0 To 1                    ' build stiffness matrix for use
    For m = 1 To 2                  ' in global coordinate system
      For i = 1 To DOF
        For j = 1 To DOF
           Kt(m + n * 2, i, j) = Ktmp(i + n * DOF, j + (m - 1) * DOF)
        Next j
      Next i
    Next m
  Next n

  Erase Ktmp
End Sub


Private Sub TransformMatrix(r1X As Double, r1Y As Double, r1Z As Double)
'*******************************************************************************
'* Constructs the transform matrix of the element
'* See section 3.7 of the thesis for details.
'*******************************************************************************
  Dim mag As Double, i As Integer, j As Integer, n As Integer

  mag = Sqr(r1X ^ 2 + r1Z ^ 2)

  If mag = 0 Then                       ' Element is along axis Y
     Ti(1, 2) = r1Y
     Ti(2, 1) = -r1Y
     Ti(3, 3) = 1
  Else
     Ti(1, 1) = r1X
     Ti(1, 2) = r1Y
     Ti(1, 3) = r1Z

     Ti(2, 1) = -r1X * r1Y / mag
     Ti(2, 2) = mag
     Ti(2, 3) = -r1Y * r1Z / mag

     Ti(3, 1) = -r1Z / mag
     Ti(3, 2) = 0
     Ti(3, 3) = r1X / mag
  End If

  For n = 1 To 3
     For i = 1 To 3
         For j = 1 To 3
            Ti(n * 3 + i, n * 3 + j) = Ti(i, j)
         Next j
      Next i
  Next n
End Sub
```

```
Public Sub Update()
'*****************************************************************************
'* Update the element status. This is the only class.method of "element" class
'*
'* Material Or/And Geometry nonlinearity can be done.
'*****************************************************************************
  Dim i As Integer, Dtmp(1 To 2 * DOF) As Double

  YldFlgChg(1) = False                    ' Initial values.
  YldFlgChg(2) = False

  For i = 1 To DOF                        ' Extract related nodal displacements
     Dtmp(i) = gDelD((Fnode - 1) * DOF + i)
     Dtmp(DOF + i) = gDelD((Snode - 1) * DOF + i)
  Next

  dD = MatMul(Ti, Dtmp)                   ' Transform to LCS
  dF = MatMul(Ktl, dD)                    ' Based on the previous iteration
  tF = MatAdd(IntForce, dF)               ' Initial total force

  Call IntForces

  Call ChkJntMech(nodei, 1)
  Call ChkJntMech(nodej, 2)
  If YldFlgChg(1) Or YldFlgChg(2) Then
      Call PlasticFlow
      Call PlasticReduction
  End If

  Call TangentStiffness
End Sub


Private Sub WtLoad(alpha As Double)
'*************************************************************************
'* Calculates global FEM and FES based on weight of unit length of the pipe
'*
'* Gravity is in -Y direction (Not optional). Therefore the fix end forces
'* are just functions of the angle between XZ plane and the member.
'*
'*  Input:
'*          Alpha: The angle between element and XZ plane
'*************************************************************************
  Dim lFEF(1 To 12) As Double
  Dim gFEF() As Double
  Dim wt As Double, i As Integer

  wt = props(PropId).WtPLen

  lFEF(1) = wt * eL / 2 * Sin(alpha)
  lFEF(2) = wt * eL / 2 * Cos(alpha)
  lFEF(6) = wt * eL ^ 2 / 12 * Cos(alpha)

  lFEF(DOF + 1) = wt * eL / 2 * Sin(alpha)
  lFEF(DOF + 2) = wt * eL / 2 * Cos(alpha)
  lFEF(DOF + 6) = -wt * eL ^ 2 / 12 * Cos(alpha)

  gFEF = MatMul(Ti, lFEF)

  For i = 1 To DOF
     gP((Fnode - 1) * DOF + i) = gP((Fnode - 1) * DOF + i) + gFEF(i)
     gP((Snode - 1) * DOF + i) = gP((Snode - 1) * DOF + i) + gFEF(i + DOF)
  Next i

End Sub
```

B.5.2 Joint Class

```
VERSION 1.0 CLASS
'*******************************************************************************
'* This is a part of PIPLAST Version 1.0 written by Farhood Nowzartash.
'* Copyright © by Farhood Nowzartash 2002
'*******************************************************************************

BEGIN
  MultiUse = -1  'True
  Persistable = 0  'NotPersistable
  DataBindingBehavior = 0  'vbNone
  DataSourceBehavior  = 0  'vbNone
  MTSTransactionMode  = 0  'NotAnMTSObject
END
Attribute VB_Name = "Joint"
Attribute VB_GlobalNameSpace = False
Attribute VB_Creatable = True
Attribute VB_PredeclaredId = False
Attribute VB_Exposed = False
Option Explicit

Private coordinate(1 To 3) As Single ' Three coordinates of a node
Private Res As String * DOF          ' Restraints of a node
Private Elements() As String * 4     ' Elements connected to a node
Private JForce(1 To 6) As Double     ' Fx, Fy, Fz, Mx, My, Mz

'*******************************************************************************
'* Property inlets and outlets
'*
'*******************************************************************************

Friend Property Let CorX(ByVal a As Single)
   coordinate(1) = a
End Property

Friend Property Let CorY(ByVal a As Single)
   coordinate(2) = a
End Property

Friend Property Let CorZ(ByVal a As Single)
   coordinate(3) = a
End Property

Friend Property Get CorX() As Single
   CorX = coordinate(1)
End Property

Friend Property Get CorY() As Single
   CorY = coordinate(2)
End Property

Friend Property Get CorZ() As Single
   CorZ = coordinate(3)
End Property

Friend Property Let Restraint(ByVal a As String)
  Res = a
End Property

Friend Property Get Restraint() As String
  Restraint = Res
End Property

Friend Property Let Forces(i As Integer, ByVal a As Double)
  JForce(i) = a
End Property
```

```
Friend Property Get Forces(i As Integer) As Double
   Forces = JForce(i)
End Property

Friend Property Let Element(ElmID As Integer, endx As String)
   Dim tmp As Integer
   tmp = UBound(Elements)

   ReDim Preserve Elements(1 To tmp + 1)

   Elements(tmp) = Str(ElmID) + endx

End Property


Friend Property Get Element(ElmID As Integer) As String
'*****************************************************************************
'* Checks that ElmId is the last joint which is plastified or not. If yes it
'* returns "NG" which means this element end is not allowed to considered
'* plastified.
'*****************************************************************************
   Dim i As Integer
   Dim con As Integer

   If Res <> "000000" Then Exit Property                          ' It's a support

   For i = 1 To UBound(Elements) - 1
      If ElmID <> Val(Mid(Elements(i), 1, 3)) Then
         If Mid(Elements(i), 4, 1) = "i" Then
             If Not elms(i).Yielded(1) Then con = con + 1      'Plastic ?
         Else
             If Not elms(i).Yielded(2) Then con = con + 1
         End If
      End If
   Next i

   If con = 0 Then Element = "NG"

End Property


Private Sub Class_Initialize()
   Res = String(DOF, "0")
   ReDim Elements(1 To 1)
End Sub
```

B.5.3 Property class

```
VERSION 1.0 CLASS
'*****************************************************************************
'* This is a part of PIPLAST Version 1.0 written by Farhood Nowzartash.
'* Copyright © by Farhood Nowzartash 2002
'*****************************************************************************
BEGIN
  MultiUse = -1  'True
  Persistable = 0  'NotPersistable
  DataBindingBehavior = 0  'vbNone
  DataSourceBehavior  = 0  'vbNone
  MTSTransactionMode  = 0  'NotAnMTSObject
END
Attribute VB_Name = "Property"
Attribute VB_GlobalNameSpace = False
Attribute VB_Creatable = True
Attribute VB_PredeclaredId = False
Attribute VB_Exposed = False
Option Explicit

Private Rad As Double                  ' Redius of pipe section
Private THK As Double                  ' Thickness of pipe section
Private Weight As Double               ' Weight per unit length
Private eE As Double                   ' Young of modulus
Private Poisson As Double              ' Poisson's ratio
Private eA As Double                   ' Cross-Section area
Private eI As Double                   ' Cross-section moment of inertia
Private YStress As Double              ' Yield stress
Private Resist(1 To 7) As Double       ' Ap,Vp,Vp,Tp,Mp,Mp,Pp

'*****************************************************************************
'* Property inlets and outlets
'*
'*****************************************************************************

Friend Property Let Radius(ByVal a As Double)
   Rad = a
   Call resistance
End Property

Friend Property Let Thickness(ByVal a As Double)
   THK = a
   Call resistance
End Property

Friend Property Let Elasticity(ByVal a As Double)
   eE = a
   Call resistance
End Property

Friend Property Let Nu(ByVal a As Double)
   Poisson = a
   Call resistance
End Property

Friend Property Let UltStr(ByVal a As Double)
   YStress = a
   Call resistance
End Property

Friend Property Let WtPLen(ByVal a As Double)
   Weight = a
End Property

Friend Property Get UltStr() As Double
   UltStr = YStress
End Property
```

```
Friend Property Get Elasticity() As Double
    Elasticity = eE
End Property

Friend Property Get Nu() As Double
   Nu = Poisson
End Property

Friend Property Get Area() As Double
  Area = eA
End Property

Friend Property Get Inertia() As Double
   Inertia = eI
End Property

Friend Property Get WtPLen() As Double
   WtPLen = Weight
End Property

Friend Property Get UltResist()
   UltResist = Resist
End Property


Private Sub resistance()
'********************************************************************************
'* Calculates the pipe section resistances
'********************************************************************************

  If Rad * THK * eE * Poisson * YStress = 0 Then Exit Sub
  Rad = Rad - THK / 2                    ' Mid-surface radius

  eA = 2 * Pi * Rad * THK
  eI = 0.25 * Pi * Rad * THK * (4 * Rad ^ 2 + THK ^ 2)
  Resist(1) = Area * YStress                                  'Ap
  Resist(2) = 4 * Rad * THK * YStress / Sqr(3)                'Vp
  Resist(3) = Resist(2)
  Resist(4) = 2 * Pi * Rad ^ 2 * THK * YStress / Sqr(3)       'Tp
  Resist(5) = 4 * Rad ^ 2 * THK * YStress                     'Mp
  Resist(6) = Resist(5)
  Resist(7) = YStress * THK / (Rad - THK / 2)                 'Pp

End Sub
```

B.6 Input Files

The input file of problem 1 of chapter 5 was presented in section B.3. Other input files for chapter 5 problems are presented in this section.

B.6.1 Problem 2

```
'          Number of
' nodes   elements    properties
System
3,  2,  1

'
' 1st line Displacement : Node, DOF, value (global coordinate), |Max|
' 2nd line load: |Max| (unit / 1000)
CONTROL
2, 2, -3, 100
500

'Corx  Cory  Corz   restraints              "No choice for ID"
Joints
0,    0, 0, 111111
4000, 0, 0, 000000
6000, 0, 0, 011010

'E        Nu   radius     THK   Fy    wt/len
Property
200000,  .3,   50.8, 5.74, 350,  0

'nodei   nodej     propid       "No choice for ID"
Elements
1, 2,     1
2, 3,     1


' Load ratio entry (Based on global coordinate)
'ID      Fx     Fy     Fz     Mx     My     Mz
JLOADS
2, -1.0,-0.02,0,0,0,0


'Internal pressure in the pipe
PRESSURE
0
```

B.6.2 Problem 3

```
'          Number of
' nodes   elements    properties
System
3,  2,  1

'
' 1st line Displacement : Node, DOF, value (global coordinate), |Max|
' 2nd line load: |Max| (unit / 1000)
CONTROL
2, 2, -2, 200
50


'Corx  Cory  Corz   restraints              "No choice for ID"
Joints
```

```
0,      0, 0, 111111
4000,  0, 0, 000000
6000,  0, 0, 111111

'E        Nu   radius      THK   Fy    wt/len
Property
200000,  .3,   50.8, 5.74, 350,  0

'nodei   nodej     propid       "No choice for ID"
Elements
1, 2,    1
2, 3,    1
3, 4,    1

' Load ratio entry (Based on global coordinate)
'ID      Fx    Fy    Fz    Mx    My    Mz
JLOADS
2, 0,-1,0,-1000,0,0


'Internal pressure in the pipe
PRESSURE
0
```

B.6.3 Problem 4

```
'          Number of
' nodes   elements    properties
System
3,  2,  1

'
' 1st line Displacement : Node, DOF, value (global coordinate), |Max|
' 2nd line load: |Max| (unit / 1000)
CONTROL
2, 2, -.1, 5
200

'Corx  Cory  Corz   restraints              "No choice for ID"
Joints
0,     0, 0, 111111
800,  0, 0, 001110
1000, 0, 0, 111111


'E        Nu   radius      THK   Fy    wt/len
Property
200000,  .3,   50.8, 5.74, 350,  0


'nodei   nodej     propid       "No choice for ID"
Elements
1, 2,    1
2, 3,    1
3, 4,    1


' Load ratio entry (Based on global coordinate)
'ID      Fx    Fy    Fz    Mx    My    Mz
JLOADS
2, 0,-1,0,0,0,0

'Internal pressure in the pipe
PRESSURE
0
```

B.6.4 Problem 5

```
'          Number of
' nodes   elements    properties
System
3,  2,  1

' 1st line Displacement : Node, DOF, value (global coordinate), |Max|
' 2nd line load: |Max| (unit / 1000)
CONTROL
2, 2, -2, 150
20

'Corx  Cory  Corz   restraints              "No choice for ID"
Joints
0,     0, 0, 111111
4000, 0, 0, 000000
6000, 0, 0, 011111

'E        Nu   Oradius     THK  Fy    wt/len
Property
200000,  .3,   50.8, 5.74, 350,  0

'nodei   nodej     propid       "No choice for ID"
Elements
1, 2,    1
2, 3,    1
3, 4,    1

' Load ratio entry (Based on global coordinate)
'ID      Fx    Fy    Fz    Mx    My    Mz
JLOADS
2, 0,-1,0,0,0,0

'Internal pressure in the pipe
PRESSURE
30
```

B.6.5 Problem 6

```
'          Number of
' nodes   elements    properties
System
10,  9,  1

'
' 1st line Displacement : Node, DOF, value (global coordinate), |Max|
' 2nd line load: |Max| (unit / 1000)
CONTROL
5, 2, -1, 1000
20

'Corx  Cory  Corz   restraints              "No choice for ID"
Joints
    0,     0,     0, 111111
 3000,     0,     0, 010000
 4000,     0,     0, 000000
 4000,     0, 1500, 000000
 6000,     0, 1500, 000000
 8000,     0, 1500, 000000
 8000,     0,     0, 000000
 9000,     0,     0, 010000
12000,     0,     0, 000000
12000, 5000,     0, 111111

'E        Nu   radius      THK   Fy    wt/len                    ' USE N & mm as units
Property
200000,  .3,   50.8, 5.74, 350,  0.4
```

```
'nodei   nodej     propid        "No choice for ID"
Elements
1, 2,    1
2, 3,    1
3, 4,    1
4, 5,    1
5, 6,    1
6, 7,    1
7, 8,    1
8, 9,    1
9, 10,   1

' Load ratio entry (Based on global coordinate)
'ID      Fx     Fy     Fz     Mx     My     Mz
JLOADS
5, 0,-1,0,0,0,0

'Internal pressure in the pipe
PRESSURE
0
```

B.7 Output File of Problem 1

The following shows the output file of problem 1 when the controlling parameter is set to –5 mm.

```
Force(KN)        v Disp. of
                 Node 2 (mm)
---------      ----------------
 0.0               0.0
 2.5               5.0
 5.0              10.0
 7.5              15.0
10.0              20.0
12.5              25.0
15.0              30.0
17.5              35.0
20.0              40.0
21.4              45.0
22.4              50.0
23.4              55.0
24.4              60.0
25.4              65.0
26.4              70.0
26.7              75.0
26.8              80.0
26.9              85.0
27.0              90.0
27.1              95.0
27.2             100.0
27.2             105.0
27.3             110.0
27.4             115.0
27.5             120.0
27.6             125.0
27.6             130.0
27.6             135.0
27.6             140.0
27.6             145.0
27.6             150.0
```