

Numerical Methods for the Stress Analysis of Pipe-work Junctions

Jamie P Finlay

A thesis submitted in partial fulfilment of the
requirements of Liverpool John Moores University
for the degree of Doctor of Philosophy

This research programme was carried out
in collaboration with Spromak Ltd. and British Energy Ltd.

May 2004

Abstract

Pipe junctions are a regular feature of piping and pressure vessel systems and are often the subject of multiple loads, acting simultaneously and at irregular intervals. Due to the nature and complexity of the loading, the subject has received a significant amount of study from designers and stress analysts to resolve some of the difficulties in stressing pressure structures.

An extensive finite element (FE) analysis was carried out on 92 reinforced butt-welded pipe junctions manufactured by the collaborating company, Spromak Ltd. After comparing the resulting effective stress factor (ESF) data with ESFs for un-reinforced fabricated tee (UFT) it was concluded that, for the majority of loads, reinforced branch outlets appear better able to contain stresses than their un-reinforced counterparts.

The linear FE study was followed by the inelastic analysis of three reinforced branch junctions. The purpose of the research was to investigate the potential use of such analysis as a tool for estimating the bursting pressure of pipe junctions and satisfying customer requirement for proof of a product's performance under internal pressure. Results obtained showed that small displacement analysis is unsuitable for estimating the bursting pressure of a pipe junction, whilst the large displacement results were similar to those obtained using a hand-calculation. Ultimately, the study concluded that inelastic analysis was too expensive, offering little by way of insight into the problem than could be found by using classical stress analysis techniques.

Following on from the study of reinforced branch outlets, this thesis described work undertaken with British Energy Ltd. to extend their current capability of stress prediction in UFT junctions using a FE based neural network approach. Upon completion of training new neural networks, the PIPET program was tested against new, previously unseen, FE data generated for this study with good results.

The program was further evaluated by comparing the output from PIPET with FE data obtained from reviewed literature. For the pressure load case, a significant proportion of

the data obtained from said literature was within the PIPET predicted stress ranges, with the new version of PIPET tending to calculate slightly lower stresses than the original program. However, whilst the pressure load case comparisons proved useful, the branch bending cases showed less concordance with PIPET's predicted stress ranges.

Acknowledgements

I wish to express my sincere thanks to my supervisors, Dr. Russell G. English and Dr. Glynn Rothwell, for their willing guidance and advice. Thanks are also due to Liverpool John Moores University, in particular Dr. Ian Jenkinson, for financial support over the past three years.

Dr. Mike Fox is an engineer at British Energy in Gloucester and my mentor for the last three years. His thoughts, insight and help in the field of neural networks and the stress analyses of pipe-work have been an inspiration. Thank you.

Thanks to Mr. Kerr Montgomery, for his assistance with respect to design and analysis of reinforced branch outlets.

I sincerely appreciate both the time and knowledge that Dr. Douglas G. Moffat has quite freely given to me on the subject of the stress analysis of pipe-work.

I am grateful for the friendship of Dr. Jeffrey ShangKuan and his wife Shirley, who showed continued support and faith in my ability to complete this work. Thanks also to my family, for their support over the past three years.

For Sarah. Thank you for your encouragement, understanding and love.

Nomenclature

α	: weld angle in the longitudinal plane or neural network learning rate
α_i	: intermediate weld angle
β	: angular increment
π	: mathematical constant pi
σ	: stress
θ	: suffix denoting a circumferential direction
τ	: through thickness parameter
ψ	: weld angle in the transverse plane
ζ	: branch outlet reducing angle
a	: node angle parameter or neural network output
b	: suffix denoting a branch pipe or neural network bias
C	: crotch sectional profile
d, D	: nominal diameter of the branch, run (mm)
d_m, D_m	: mean diameter of a run pipe (mm)
d_i, D_i	: inside diameter of a branch, run pipe (mm)
d_o, D_o	: outside diameter of the branch, run (mm)
ESF	: effective stress factor
f	: neural network activation (transfer) function
F	: flank sectional profile
FE	: finite element
I	: second moment of area for a pipe (mm^4) = $\frac{\pi(D_o^4 - D_i^4)}{64}$
j	: network neuron (node) index
M	: bending or torsion moment (Nmm)
n	: total input passed to an activation (transfer) function
P	: internal pressure (N/mm^2)
p	: pressure or neural network input
r	: suffix denoting a branch/run pipe
S10S	: pipe schedule according to ANSI 36.19-B 36.10
SIF	: stress intensity factor
t, T	: branch, run pipe wall thickness (mm), neural network target
UFT	: un-reinforced fabricated tee
W	: neural network weight
x, y, z	: space coordinates
XS	: pipe schedule according to ANSI 36.19-B 36.10
XXS	: pipe schedule according to ANSI 36.19-B 36.10

CONTENTS

ABSTRACT	I
ACKNOWLEDGEMENTS.....	III
NOMENCLATURE.....	IV
CONTENTS.....	V
1 Introduction	1
2 Pressure Vessel Analysis	3
2.1 INTRODUCTION	3
2.2 SHELL ANALYSIS: THEORY AND PRINCIPALS	4
2.2.1 "Thin" shells.....	4
2.2.2 Application of shell theory to "thick" walled components	5
2.3 A BRIEF NOTE ON DESIGN CODES	7
2.3.1 The Area Replacement Method	9
2.4 ELASTIC ANALYSIS OF PRESSURE VESSELS AND PIPING	10
2.4.1 On stress concentration factors.....	13
2.5 INELASTIC ANALYSIS OF PRESSURE VESSELS AND PIPING	14
2.5.1 Material models.....	15
2.5.2 Multi-axial yield criteria.....	17
2.5.3 Inelastic analysis methods	19
2.5.4 Inelastic analysis using the ASME Section VIII & Section III codes.....	21
2.6 CLOSURE.....	25
3 Elastic Finite Element Analysis Of Reinforced Pipe-work Junctions.....	26
3.1 INTRODUCTION	26
3.2 THE REINFORCED BUTTWELD OUTLET.....	26
3.3 PARAMETRIC MODELLING	29
3.3.1 Scope of work.....	31
3.3.2 Finite element modelling	32
3.3.3 Mesh convergence study.....	34
3.4 PRESENTATION AND DISCUSSION OF RESULTS.....	35
3.4.1 Pressure load case.....	36
3.4.2 Branch out-of-plane bending moment load case, M_{xb}	41
3.4.3 Branch moment load cases M_{yb} and M_{zb}	49
3.4.4 Run pipe moment load cases: M_{xr} , M_{yr} and M_{zr}	51
3.5 CLOSURE	56
4 Inelastic Finite Element Analysis Of Reinforced Pipe-work Junctions.....	58
4.1 INTRODUCTION	58
4.2 MATERIAL DATA	59
4.2.1 Gathering Material Data	59
4.3 FINITE ELEMENT APPROACH	62
4.3.1 Scope	62
4.3.2 Selecting Finite Elements	63
4.3.3 Generating the Model	64
4.3.4 Material Plasticity in ABAQUS.....	67
4.3.5 Geometric Plasticity in ABAQUS	67
4.3.6 Small Vs Large Displacement Analysis.....	68
4.4 DISCUSSION OF RESULTS	70
4.5 CLOSURE	75
5 An Introduction To Neural Networks.....	77
5.1 INTRODUCTION	77
5.2 REAL NEURAL NETWORKS	77
5.3 ARTIFICIAL NEURAL NETWORKS	80

5.3.1	A Brief Historical Overview of Artificial Neural Networks.....	80
5.3.2	The Artificial Neuron	83
5.3.3	Multi-layer Feed-forward Neural Networks	85
5.3.4	Learning Rules.....	87
5.3.5	Multi-layer Feed-forward Neural Networks with Back-Propagation.....	91
5.3.6	Increasing Training Performance.....	93
5.3.7	Eluding Local Minima.....	94
5.3.8	Generalization.....	96
5.4	WHY USE ARTIFICIAL NEURAL NETWORKS?.....	97
5.5	CLOSURE	99
6	Elastic Stress Analysis Of UFT Using Artificial Neural Networks	100
6.1	INTRODUCTION	100
6.2	PIPET: A NEURAL NETWORK APPROACH TO STRESS PREDICTION	100
6.2.1	Scope of Work.....	103
6.3	GENERATING AND PREPARING DATA FOR TRAINING NEURAL NETWORKS.....	106
6.3.1	Introduction.....	106
6.3.2	Finite Element Modelling.....	109
6.4	TRAINING NEURAL NETWORKS USING FE DATA	121
6.4.1	Extracting and Formatting the FE Data	122
6.4.2	Designing Neural Networks.....	125
6.4.3	Training the Networks Using “qckprp”	127
6.5	EVALUATING PIPET’S PERFORMANCE	137
6.5.1	Updating the Weights File	137
6.5.2	Testing PIPET	138
6.5.3	Evaluation Exercise: Pressure Load Case.....	148
6.5.4	Evaluation Exercise: Bending Moment Load Cases	161
6.6	CLOSURE.....	167
7	Conclusions.....	169
7.1	RECOMMENDATIONS FOR FUTURE WORK.....	172
	REFERENCES	174
	BIBLIOGRAPHY	177
	APPENDIX A: BUTTWELD OUTLET BORES AND ASA DIMENSIONS OF SEAMLESS PIPE TO BS1600, ANSI B36.10 AND B3619.....	178
	APPENDIX B: MATLAB SOURCE CODE: ESF STRESS ANALYSIS OF RBO	180
	APPENDIX C: REINFORCED BRANCH OUTLET RANGE OF GEOMETRIES ANALYSED	189
	APPENDIX D: ESF CHARTS FOR REINFORCED BRANCH OUTLETS	193
	APPENDIX E: EFFECTIVE STRESS FACTORS FOR REINFORCED BUTTWELDED BRANCH OUTLETS SUBJECTED TO INTERNAL PRESSURE OR EXTERNAL MOMENT LOADS	197
	APPENDIX F: REINFORCED BRANCH OUTLET PROOF PRESSURE TEST DATA	219
	APPENDIX G: MATERIAL TENSILE TEST DATA	229
	APPENDIX H: FORTRAN AND MATLAB SOURCE CODE: STRESS ANALYSIS OF UFT USING NEURAL NETWORKS	238
	APPENDIX I: UNIX SHELL SCRIPT SOURCE CODE.....	255

1 INTRODUCTION

The primary impetus for conducting this research was to further extend the bounds of knowledge in the field of pressure vessel and piping stress analysis. In particular, the ultimate goal of this work is to add to the body of knowledge pertaining to the stress analysis of 90° welded pipe junctions with and without reinforcement. The objective of this thesis is to present new data to the wider community researching this field and to offer insight to alternative numerical approaches to the stress analysis of such structures.

The work undertaken on the finite element analysis of reinforced branch outlets (RBOs) was during the period 1999 – 2001 for Spromak Ltd., who manufacture such pipe fittings. The research regarding the stress analysis of UFT using neural networks was a collaborative effort involving British Energy Ltd. over three years from 2001 – 2004.

Due to the disparate, yet related, areas of research covered in this work, this thesis is necessarily split into two parts. The decision to do so was based on the philosophy that, although the research topics discussed herein share some common ground, the material may more readily be digested by the reader if it were compartmentalized in some logical fashion. A reader wishing to research the use of neural networks in pipe junction stress analysis can skip the parts of the thesis pertaining to the research on RBOs without risking loss of comprehension and vice versa.

Briefly, the thesis is set out as follows. Each chapter begins with an introduction regarding the content, followed by a presentation and discussion of all findings before the chapter closes with a succinct conclusion.

This chapter sets out the aims and objectives of the thesis and the research it discusses, followed by a short précis of the work carried out in chronological order.

Chapter two introduces pressure vessel analysis theory and provides a survey of previous relevant research in the field of pressure vessel stress analysis (elastic and plastic), with an emphasis on 90° UFT and RBOs.

The finite element modelling and subsequent elastic stress analysis of RBOs is documented in chapter three. The work comprises a study of 92 fittings and presents results in the form of ESFs, which are analysed in the light of ESF data for UFT obtained from open literature.

Chapter four presents work done on the inelastic analysis of 90° RBOs and discusses experimental methods used in obtaining material data and the results of a FE approach to a pressure burst test simulation of 90° RBOs. This chapter effectively brings closure to the research undertaken in the stress analysis of reinforced branch junctions.

Chapter five introduces neural networks, their theory and application along with a discussion of relevant articles published on the subject to date. After laying the foundation and core knowledge required for study of this area, chapter six discusses in detail the methods employed in utilizing this technology for the stress analysis of UFT. After presenting the findings of the neural network study, the chapter concludes with a discussion of the results obtained.

Chapter seven presents an overview of all of the work carried out, the results obtained and concludes with recommendations for further research in the field of pressure piping stress analysis.

Printouts of computer code, drawings and other extraneous material are reserved for the appendices.

2 PRESSURE VESSEL ANALYSIS

2.1 Introduction

For some time it has been understood that when material is removed from a pressure vessel or run pipe in order to accommodate, for example, a branch pipe, the opening in that vessel promotes increased stresses around the edge of the hole. In an effort to compensate for the inherent weakness at the hole, piping engineers may choose to fit a RBO or nozzle that is designed in such a way as to provide material compensation in the area immediately surrounding the hole in the run pipe.

Pipe junctions are a regular feature of piping and pressure vessel systems and are often the subject of multiple loads, acting simultaneously and at irregular intervals. Due to the nature and complexity of the loading, the subject has received a significant amount of study from designers and stress analysts, certainly over the last 50 years, in efforts to resolve some of the difficulties in stressing pressure structures. Many pressure vessel and piping codes have adopted their own approach to the design and analysis of piping branch junctions and their components. However, the underlying principles of all codes are based on the small displacement linear elastic analysis of thin shells.

There are a number of well known and established texts relating to the elastic stress analysis of shells with respect to pressure vessels, including those by Gill¹, Flugge² and Kraus³, therefore, a detailed review will not be presented here. The purpose of this chapter is to give the reader an overview of stress analysis with respect to pressure vessels and to describe the major developments in elastic and inelastic pressure vessel stress analysis that have taken place over the last six decades that are relevant to the work presented in this thesis.

2.2 Shell Analysis: Theory and Principals

2.2.1 “Thin” shells

Thin shells, with respect to pressure vessels, are conventionally thought of as cylinders, cones, spheres and toroids that have radii far greater than their thickness. Typically, a pressure vessel qualifies as a thin shell when the ratio of its radius to thickness, r/t , is greater than or equal to ten¹.

Consider the thin cylinder in fig. 2-1 of length, L , subject to internal pressure, p , with a radius, r , much greater than the thickness, t . The length of the cylinder is such that, at the point of interest, there is no interference with respect to stresses from end effects. If the stress through the thickness of the cylinder is assumed uniform and the radial stresses are negligible, the cylinder can be considered to be in a state of plane stress.

The circumferential and longitudinal stresses, induced by internal pressure, p , are represented in fig. 2-1 and are denoted σ_θ and σ_x respectively. Since no bending stresses are present, these are often referred to as “membrane” stresses.

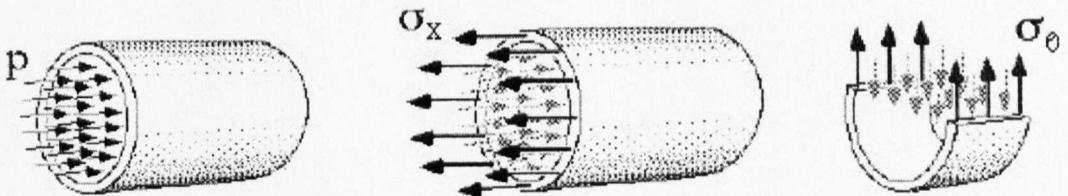


Fig. 2-1: Circumferential and longitudinal stresses in a cylindrical vessel under internal pressure

In order to derive the longitudinal stress, σ_x , the forces acting on the section must be equated:

$$p(\pi r^2) = \sigma_x(2\pi r t) \quad (2.1)$$

Therefore:

$$\sigma_x = \frac{pr}{2t} \quad (2.2)$$

Likewise, the circumferential or hoop stress can be calculated:

$$p2rL = \sigma_\theta 2Lt \quad (2.3)$$

Re-arranging gives:

$$\sigma_\theta = \frac{pr}{t} \quad (2.4)$$

2.2.2 Application of shell theory to “thick” walled components

“Thin” shell analysis, although the basis for the stress analysis of pressure components in piping codes does not satisfy the criteria for a significant number of real-world applications. For a simple “thick” cylinder, removed from any discontinuities, Lamé’s equations may be used to derive stresses (eqns. 2.5, 2.6, 2.7). Fig. 2-2 shows a typical stress distribution in such a cylinder.

$$\sigma_\theta = \frac{p(1 + (b/r)^2)}{((b/a)^2 - 1)} \quad (2.5)$$

$$\sigma_r = \frac{p(1 - (b/r)^2)}{((b/a)^2 - 1)} \quad (2.6)$$

$$\sigma_x = \frac{p}{\left(\left(\frac{b}{a}\right)^2 - 1\right)} \quad (2.7)$$

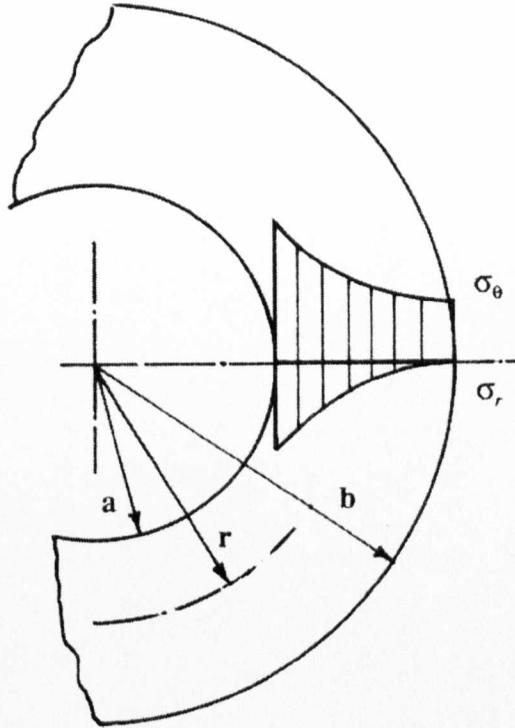


Fig. 2-2: Stress distribution in a “thick” cylinder

For the pressure vessel design standards, analysis of “thick” shells is largely considered too complex, thus the codes revert to use of “thin” shell analysis principles, which can provide useful and accurate solutions up to and including $r/t = 10$. For cylindrical shells that are “moderately thick” a modified “thin” shell theory equation for calculation of the circumferential stress is considered reasonable:

$$\sigma_\theta = \frac{pd_m}{2t} \quad (2.8)$$

Providing the fundamentals of equilibrium, strain displacement compatibility and constitutive relations are adhered to the analysis methodology described in 2.2.1 and

2.2.2 above can be successfully employed in the stressing of simple geometric entities, such as a cylindrical nozzle in a sphere.

However, there are significant limitations. Firstly, despite the simplifying assumptions required in order to apply shell theory, the mathematics remains rather complex and therefore a possible source of error (although computer software from various producers is now available for desktop computers to aid an engineer in this respect).

Perhaps a more significant limitation is that of radius to thickness ratio, which effectively restricts analysis to $r/t = 10$, although some engineers may be inclined to use this technique down to $r/t = 5$ (Gill¹).

Geometric simplifying assumptions are introduced due to the fact that the theory – the mathematics – cannot describe with 100% accuracy effects such as weld details, geometry imperfections and manufacturing tolerances. This, however, is generally considered a problem only if the analyst is concerned with peak stresses or doing detailed stress analysis in the region of a discontinuity or component – which is the case in this study.

Shell analysis is a subject of considerable depth and the purpose of this section was to introduce the underlying principles behind the majority of pressure vessel codes available to the piping designer today and to outline its potential scope and limitations of use. A fuller introduction to the topic is presented in chapter 2 of Pressure Vessel Design: Concepts and Principles edited by Spence and Tooth.⁴

2.3 A Brief Note on Design Codes

Pressure vessel national standards and codes for the design and analysis of pressure vessels, piping and components are largely employed to guard against failure such as bursting, buckling and ratcheting, but leave responsibility for the safety of a design to the designer. Thus, in respect of safety, codes are often no more than guides, and the onus is on designers to satisfy themselves and their customers that a product is safe.

Since the steam engine was introduced in 1769, there has been a drive to legislate (both in the UK and elsewhere) the safe use of pressure vessels, often motivated by incidents in boiler and pressure vessel operation and through a greater understanding of the stresses in such vessels. In the UK, for example, there have been a number of bills passed by the houses of parliament designed to reduce losses, both human and financial, such as the Boiler Explosion Act of 1882 and the Health and Safety at Work Act of 1974.

In the UK and USA the gradual process of reform of working practices and health and safety has run hand-in-hand with the dissemination of knowledge in the field, which has naturally gravitated to the development and refinement of pressure vessel design standards.

Most nation states, including the UK, France, Germany and the USA, all have their own national design codes or standards. The UK has also been engaged with other EU member states in the development of a new European pressure vessel standard – EN13445⁵ – that has been effective since 29th May 2002. In particular, British pressure vessel designers have the EN13445 and PD5500 (formerly BS 5500⁶) codes at their disposal to aid design and analysis activities. The American codes ASME B31 (incorporating B 31.1⁷ and B 31.3⁸), ASME BPVC Sections III⁹, and VIII¹⁰ make provision for the design of power piping, process piping (in particular, petrochemical), nuclear power plant components and pressure vessels respectively.

Both the British (European) and American codes allow for the design of pressure vessels, piping and related components via one of two routes, namely: design by analysis and design by rule. Design by analysis allows the designer to use stress analysis directly in the design of a component, thereby giving a designer some freedom. Design by rule, on the other hand, uses formulae derived from shell analysis (see section 2.2 of this thesis) to calculate primitive shell thickness dimensions and requires complete compliance with the code for design of components. This is the method for design within the main body of PD5500 and involves the calculation of certain geometric criteria used in conjunction with a set of design charts. These charts are derived from data obtained by Leckie and Penny¹¹ in 1963 using shell analysis techniques.

2.3.1 The Area Replacement Method

In general, the American standards require reinforcement by the area replacement method (see fig. 2-3), whilst a similar method is reserved for Annex F in PD5500. The technique essentially provides material compensation around the area of the junction and reinforcement is obtained by designing a fitting, or nozzle as it is commonly referred to in the piping industry, with extra material within pre-prescribed dimensional bounds and according to code guidelines.

Regardless of the standard adopted for design of both reinforced and un-reinforced branch junctions, there has been considerable effort invested in increasing the allowable stresses in pressure vessels due to pressure and moment loads. Researchers have attempted (with a good deal of success) to prove that the existing codes were written in a fundamentally conservative manner. Consequently, so-called stress intensity factors (SIFs) derived for nozzles using the codes were often felt to be too high.

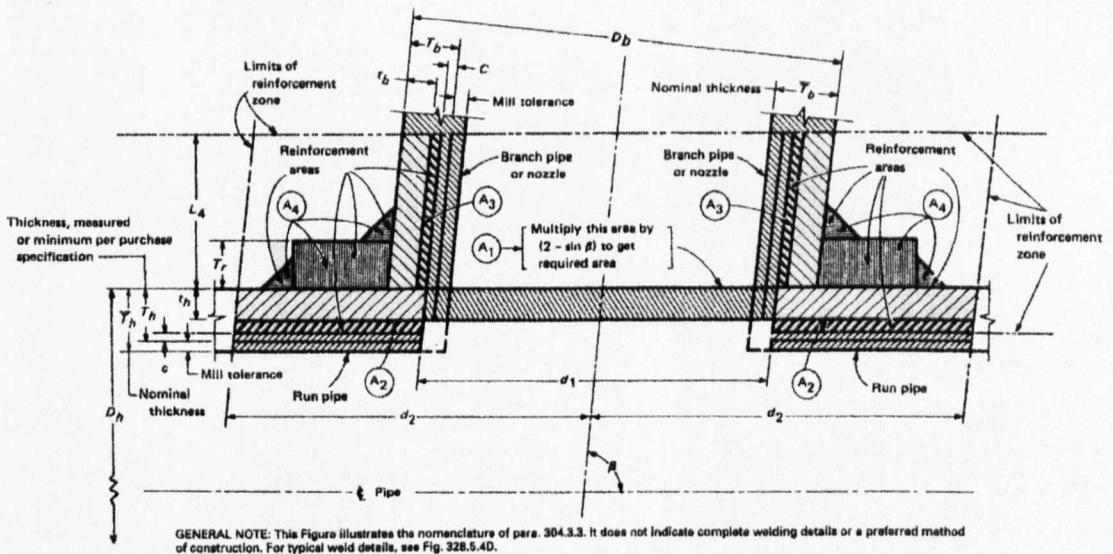


Fig. 2-3: Area replacement nomenclature diagram ASME B 31.3 (1999)

Fig. 2-3 demonstrates area replacement according to ASME B 31.3, paragraph 304.3.3. This is the method to which Spromak Ltd. subscribe in the design of their pipe fittings.

2.4 Elastic Analysis of Pressure Vessels and Piping

Elastic stress analysis is a method for the evaluation of materials and structures based upon mathematically defined relationships between stress and strain. As such it is an idealized model of the behaviour of materials that assumes an initially stress free structure – this is an assumption generally accepted, although ultimately inaccurate. However, elastic analysis has been the de-facto standard for the evaluation of engineering structures since the inception of the theory of elasticity in 1879 by Castigliano.

All engineering metals possess a certain degree of elasticity that is, essentially, a measure of a materials resistance to permanent deformation. Therefore, the elastic analysis of metals is generally linear, readily calculated and necessarily conservative.

Regarding the elastic stress analysis of pipe junctions, most engineers agree that there are nineteen loads that can act at any one time on a system of two intersecting pipes – pressure plus nine individual moment loads and nine forces acting on the limbs of the junction. Lock et al¹² note that the total number of loads can be reduced to seven – pressure plus six moment loads – if the primary concern is with structural integrity. The six moment loads are represented by vectors in fig. 2-4. This is frequently referred to as the “cantilever model”.

Early work in the field of piping analysis using the finite element method included research by Ando et al who presented two papers in 1971¹³ and 1973¹⁴ concerned with branch junctions having diameter ratios 0.84 and 0.51. In 1980, Baldur et al¹⁵ detailed his findings for two branch junctions, $d_m/D_m = 0.13$ and 0.56, subjected to branch in-plane and out-of-plane moment load. However, in modelling the boundary conditions, the authors of^{13,14,15} chose to constrain both ends of the run pipe, therefore the results cannot be directly compared with the work herein, since the cantilever boundary condition is used as illustrated below.

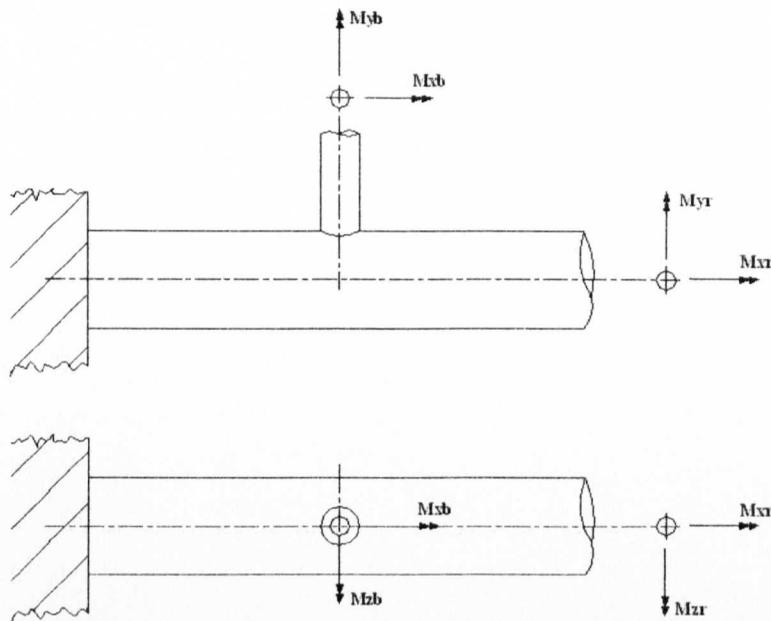


Fig. 2-4 The cantilever model illustrating the six bending moment loads

A combined experimental and finite element study of four branch junctions was presented in 1976 by Gwaltney et al.¹⁶ The study comprised a wide diameter ratio range – 0.114, 0.125, 0.5 and 1.0 – having run pipe to run thickness ratios of 50 and 100. The finite element results for pressure and the six moment load cases were, in the main, concordant with the experimentally derived data.

One year later, in 1977, Bryson et al¹⁷ presented finite element data in a parameter study on stresses in reinforced nozzle-to-cylinder attachments. Seven individually applied loads, including internal pressure and external moments, were considered and, in total, 25 models having diameter ratios in the range $0.08 \leq d/D \leq 0.5$ and cylinder diameter-to-thickness ratios $10 \leq D/T \leq 100$ were analysed. The results were subsequently used in the formulation of new design procedures within the ASME III standard.

In addition to Bryson et al's finite element work¹⁷, analytical work undertaken by Steele et al¹⁸ on branch junctions of $d/D < 0.5$ was used as the foundation for creating new design charts¹⁹ in the ASME III code.

Khan et al²⁰ presented results of a comparative study on the effects of reinforcement on stress fields at the junction of two normally intersecting cylinders having a diameter ratio of 0.67. Four moment load cases comprising in-plane and out-of-plane bending were included. For the moment cases, data was gathered both experimentally, by the use of electrical resistance foil gages, and through three-dimensional FE analysis. The authors were able to include results for internal pressure and their findings were presented in an effort to promote the benefits of reinforcement in branch junctions using a fitting such as those considered in this work. Later Berger et al²¹ conducted finite element work similar to Gwaltney et al¹⁶ to obtain ASME stress indices for a RBO, diameter ratio 0.58.

Lock and his co-workers¹² felt that there was room for improvement and sought to extend the knowledge base for branch junctions having a diameter ratio greater than 0.5. The authors presented a FE stress analysis of an equal diameter intersection subjected to pressure and in-plane branch and run moment loads, the results of which were presented against prior experimental work carried out by Moffat²². The FE¹² and experimental²² results were found to correlate satisfactorily and the authors concluded that the modelling techniques adequately described the system. However, they felt that the finite element model, which did not include the geometry of the weld between the branch and run pipes, could be improved upon (a factor of considerable importance as noted in later work by Moffat and his colleagues²³).

Moffat et al's results²³ included a comprehensive parametric survey of un-reinforced branch junctions although, due to difficulties experienced in modelling the weld profile for the $d/D = 1.0$ combination no FE data were available for this case. However, the authors were able to use experimental data allowing inclusion of ESFs for the $d/D = 1.0$ case in their paper. Moffat and his co-workers concluded that there was room for additional experimental and finite element work to confirm their data.

2.4.1 On stress concentration factors

It is important to define the terminology used in this thesis with respect to stress factors, in particular “stress intensity factor” (SIF) and “effective stress factor” (ESF). In the American codes a SIF is often used with respect to an implied stress factor like those derived from the fatigue analysis work by Markl²⁴ of standard weight four-inch equal tee junctions. This SIF is usually known as an “i – factor” and is calculated via formula using the geometric properties of the run pipe only. Its primary use, cited by Markl, is to allow the piping designer to express the endurance strength of a piping component in terms of that for a straight pipe.

However, the stated purpose of this work on reinforced branch outlets is to determine stress concentration factors, based on the maximum stress intensity in a piping component for a given load. Therefore, an “i – factor” SIF as defined in the American codes cannot readily be used to determine the maximum stress intensity

Until 1991 piping stress analysts had used the term SIF, independent of the codes, to mean the ratio of the maximum Tresca yield stress (the difference between the largest and the smallest principal stress) to the relevant nominal stress.

In Moffat et al’s paper²³, the term “effective stress factor” was proposed as a replacement for “stress intensity factor” as the latter term is used in other fields of solid mechanics with a completely different definition. This new term was defined as being the ratio of maximum Tresca stress to the nominal stress. Such a metric is useful for piping designers since, if one knows the ESF for a given load and the dimensions of the pipe in question, it is a simple task to calculate the maximum stress induced by said load.

There are two widely accepted methods for defining yield criteria in three-dimensional solid mechanics: Tresca and von Mises (see 2.5.2). Although the Tresca yield criterion is often considered a more conservative measure of stress intensity and is mathematically simplistic, the majority of finite element software packages available

today appear to favour the use of the von Mises criterion. This was certainly the case for the packages used by this author for this research.

Thus the definition of ESF used herein is the ratio of the maximum von Mises stress intensity to the relevant nominal stress (see eqn. 2.9), where the nominal stress calculated is dependent on the load under consideration and, for moment loads, whether that load is acting on the branch or run pipe.

$$ESF = \sigma_{\max} / \sigma_{nom} \quad (2.9)$$

For pressure loads the nominal stress is calculated using the equation:

$$\sigma_{nom} = PD_m / 2T \quad (2.10)$$

For moment loads applied to the branch:

$$\sigma_{nom} = Md_o / 2I_b \quad (2.11)$$

For moment loads applied to the run:

$$\sigma_{nom} = MD_o / 2I_r \quad (2.12)$$

Refs.^{12,23} and Gilroy et al²⁵ use these relationships exactly for the calculation of the nominal stresses induced by the loads described above.

2.5 Inelastic Analysis of Pressure Vessels and Piping

Inelasticity or plasticity, is concerned with the response of a material (and implicitly; a structure comprised of this material) beyond the elastic limit. Elasticity is a property of engineering metals that enables it to return to its original shape and dimension upon removal of an applied load. Elastic behaviour in metals is simply linear, whereas inelastic analysis is non-linear.

Unlike elastic analysis (see section 2.4), which simply ignores initial states of stress in a structure, inelastic analysis deals with the issues of residual stresses and the effects of self-weight by suggesting that it has no bearing on the structural strength of the body under consideration.

Inelastic analysis of pressure vessels and piping is often concerned with taking advantage of material excesses built into a structure as a result of a code based or purely elastic analysis and using the results to try and justify either a longer service life, or higher allowable pressures or loads. Inelastic analysis enables a better understanding of the nature of the structure, its potential and limitations.

Since the publication of a report by an ASME Task Group in 1994²⁶ and the subsequent incorporation of their recommendations into Sections III and VIII of the boiler and pressure vessel code (including guidelines on design by analysis using inelastic methods) it is presumed that such analysis is now being undertaken more frequently as an aid to design as well as for the reasons mentioned above.

Explosive failure of pressure vessels was a regular feature of the industry in its infancy. Thankfully, such occurrences are now rare thanks to coherent codes and design practice rules. However, pressure vessels do occasionally fail with potentially grave consequences for both people and the environment.

2.5.1 Material models

When a metal is worked or is subject to a load that takes the material beyond its elastic limit it undergoes the process known as work or strain hardening. In order, therefore, to find the current yield stress the complete history of plastic loading and the corresponding accumulated plastic strain must be considered. Fig. 2-5 illustrates three plasticity models: Non-linear hardening, bi-linear hardening and the simplest idealization of hardening – perfect plasticity.

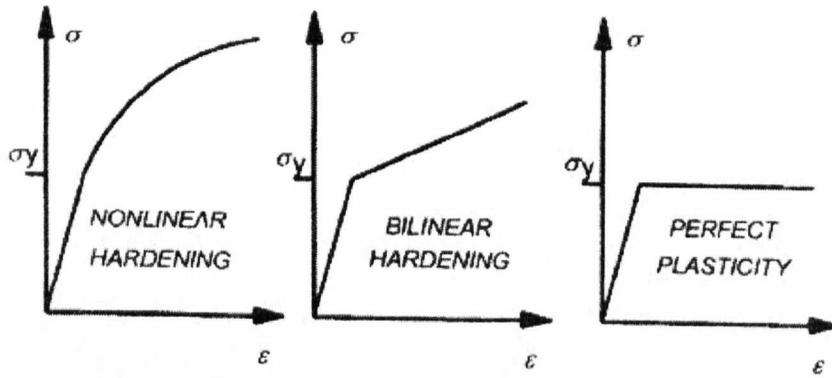


Fig. 2-5: Non-linear material models

On first examination, perfect plasticity and its implied unlimited plastic flow upon reaching first yield, may appear to be unrealistic. However, if one considers the shape of the plastic stress-strain curve for a particular material in fig. 2-6(a) and fig. 2-6(b) it can be appreciated that the shape of the curve depends very much on the range of strain over which the material behaviour is being considered.

For example, the mild steel curve in fig. 2-6(a) demonstrates no significant material hardening in the strain range 0-1 per cent. However, the same material over 0-20 per cent plastic strain displays significant hardening - see fig. 2-6(b).

It follows that in choosing such a model the strain range of interest must be clearly defined and understood. In design, strain of the order of one per cent is of most interest; thus, the model of perfect plasticity can be justifiably used in ensuring reasonably conservative design.

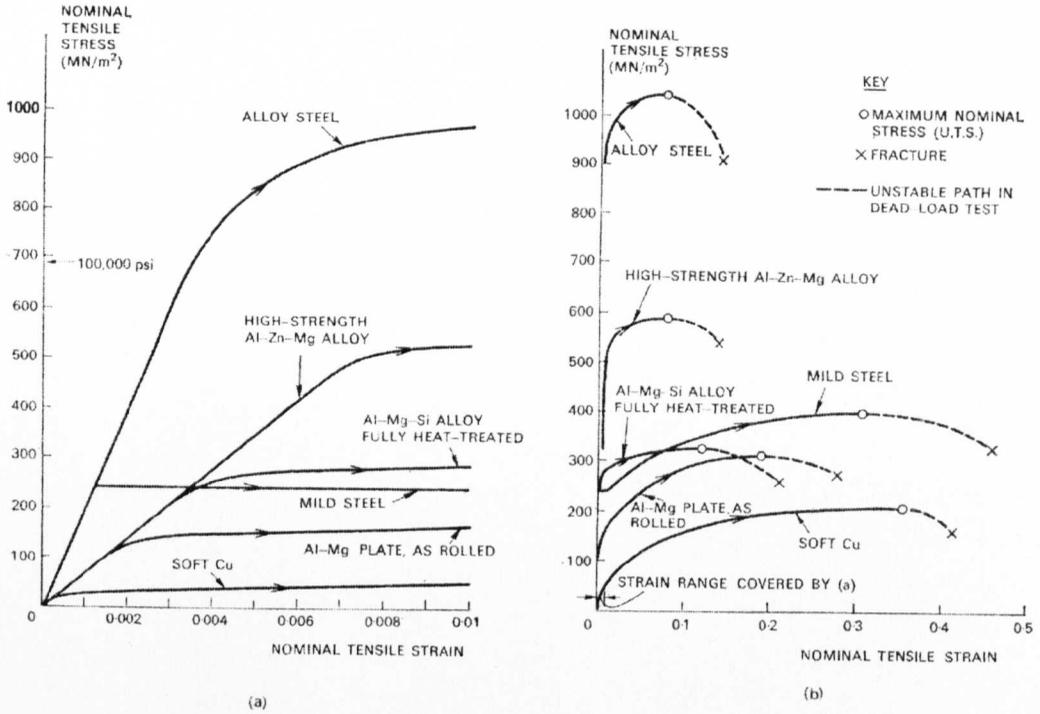


Fig. 2-6(a): Stress-Strain curves for common engineering materials up to 1% strain value. (b): Stress-Strain curves for common engineering materials up to 50% strain value. After Calladine C. R., Plasticity for Engineers, John Wiley & Sons, 1985.

2.5.2 Multi-axial yield criteria

As previously stated in 2.4.1 of this thesis, there are two widely accepted methods for defining yield criteria in three-dimensional solid mechanics: Tresca and von Mises. The Tresca criterion assumes yield is governed by the value of the maximum principal shear, while the von Mises hypothesis assumes the value of the root mean square of the principal shears dominates.

According to Tresca, yield under multi-axial stress occurs when the maximum principal shear reaches a critical value. Assuming a uniaxial stress field and σ_y is yield in tension:

$$\tau_{\max} = \frac{1}{2} \sigma_y \tag{2.13}$$

The von Mises criterion expressed mathematically is:

$$\frac{1}{2} \left[(\sigma_1 - \sigma_2)^2 + (\sigma_2 - \sigma_3)^2 + (\sigma_3 - \sigma_1)^2 \right] \leq \sigma_Y^2 \quad (2.14)$$

In 1931, Taylor and Quinney carefully conducted a number of experiments and sought to establish which of the two yield criteria was closest to the actual yield condition. After plotting the yield loci for Tresca, von Mises and their own data, their results were conclusive: the von Mises theorem correlated much better with their data than the Tresca criterion (see fig. 2-7). Although the von Mises formula is considered more accurate, the Tresca criterion is generally regarded as conservative and is, therefore, widely used in design and in pressure codes the world over.

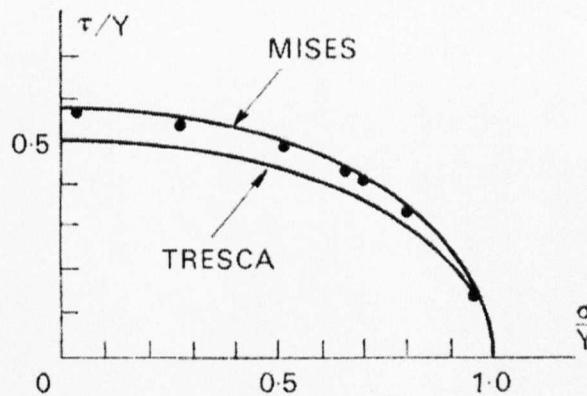


Fig. 2-7: Experiments on combined tension and torsion of a tube of ductile material. After Calladine C. R., *Plasticity for Engineers*, John Wiley & Sons, 1985.

The yield surfaces for the Tresca and von Mises criteria for biaxial loading are shown in fig. 2-8. From this, it is hoped one can appreciate the conservatism of the Tresca condition.

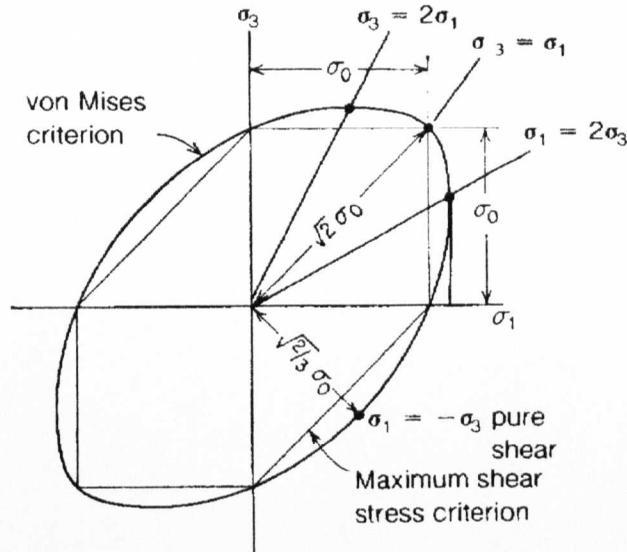


Fig. 2-8: The Tresca and von Mises Yield Surfaces

As was previously alluded to in comment on elastic analysis, the von Mises theory is by far the more prevalent of the two yield theories in commercially available finite element analysis packages and is therefore the criterion used for the plasticity analyses described herein.

2.5.3 Inelastic analysis methods

Inelastic analysis has received much attention over the last few decades from the engineering community, including the pressure vessels and piping industries. Research in the industry has been widespread and although there appears to be little directly relating to the inelastic analysis of reinforced branch outlets, the approach to limit or inelastic analysis in general is well documented.

In particular, developments in methods of inelastic piping analysis were comprehensively documented in 1987 by Boyle and Spence²⁷. The paper presents a thorough survey of inelastic piping analysis, including some 131 references, with an emphasis on simplified methods of analysis and the analysis of piping elbows. More recently, Mackerle²⁸ documented a bibliography listing 670 references including papers

and conference proceedings that were published between 1998 and 2001 on the subject of the use of finite elements in pressure vessel analysis, 140 of which were solely concerned with non-linear, static and dynamic analysis. There is clearly a significant body of ongoing work in this field.

Analysis based on the perfectly plastic material model is often referred to as a limit or limit load analysis. With its “built-in” structural failure mechanism, analysis of this kind is relatively straightforward, does not require extensive material testing or data and is (relatively) computationally inexpensive. Limit analysis forms the basis of design against gross plastic deformation within the American (Appendix 4)¹⁰ and British (Annex A)⁶ codes.

Researchers at the University of Strathclyde have invested a good deal of energy developing linear FE procedures for the inelastic analysis of pressure structures^{29,30}. The use of these procedures within the guidelines of the codes allows designers to take advantages of the higher allowable stress limits allowed under the design by analysis route.

Although the work on inelastic analysis contained herein does not concern use of the perfectly plastic model or limit loads, the author felt that such a significant body of work should not be overlooked. It is included to aid the reader and designer in search of literature on the broader topic of methods for inelastic design of pressure components using the FE method.

The current work, being commercially driven, was undertaken in order to investigate the potential use of FE analysis for the simulated pressure burst testing of RBOs as a cheaper alternative to experimental pressure proof testing in satisfying customer confidence in a particular product. The inelastic analysis of RBOs as described herein is rarely seen in open literature (presumably due to the commercially sensitive nature of the data) and the author has found only one recent reference that mirrors the effort detailed in chapter 4. That is the work by Carino et al at the University of Pavia, Italy in 1996³¹.

Carino and his co-authors analysed three self-reinforcing outlets in an effort to simulate pressure burst tests commonly used in industry. They used a bespoke finite element pre-processor for mesh generation (using 8-noded bricks) and their analysis accounted for both material and geometric non-linearity – a factor of considerable importance as noted by Sanal³² in his two-dimensional finite element work on pipe elbows and dished ends.

Through their *“high-resolution post-processor capability”*, Carino et al were able to predict, with sufficient accuracy, plastic strain trends in the critical flank area of the model assembly. They went on to argue that the method provided a *“powerful tool in view of anticipating proof testing results”* and *“a useful guideline for design purposes in view of a more rational material usage and joint configuration”*.

The results of the work detailed by Carino and his colleagues proved encouraging: the isostrain contours lay on paths previously seen on the actual burst tested laboratory models. In addition, they found that the maximum strain values derived from the FE analyses correlated with their earlier experimental values.

2.5.4 Inelastic analysis using the ASME Section VIII & Section III codes

As has been mentioned in 2.5.2 of this thesis, in 1994 ASME appointed a Task Group to investigate and clarify the definition of primary stress within the code and to provide guidance for analysts and designers wishing to make use of modern stress analysis techniques such as finite element analysis. They concluded, amongst other things, the following:

“Different techniques can be used to demonstrate satisfaction of primary stress such as:

Elastic-plastic with limit load analysis.

Elastic-plastic solutions including work hardening of the material.”

Also:

“Only by employing an inelastic analysis can the designer be assured that an optimum solution has been reached.”

These quotes taken from the Task Group report illustrates ASME’s conviction that the future design of pressure vessels and piping by analysis lies in inelastic stress analysis.

Invoking Appendix 4 of the ASME VIII Division 2¹⁰ code and performing an inelastic analysis allows designers some relaxation in the basic stress limits. Under the ASME guidelines one may choose either of two approaches to inelastic analysis for calculating allowable loads with respect to gross plastic deformation: limit analysis or plastic analysis.

Limit analysis

Limit analysis enables calculation of the limit load of a vessel, based on small deformation theory using the elastic perfectly plastic material model. Appendix 4 of the ASME boiler and pressure vessel code states that:

“The limits on general membrane stress intensity (4-131), local membrane stress intensity (4-132) and primary membrane plus primary bending stress intensity need not be satisfied at a specific location if it can be shown by limit analysis that the specified loadings do not exceed two-thirds of the lower bound collapse load. The yield strength to be used in these calculations is $1.5S_m$ ”

The allowable load, P_a , is therefore:

$$P_a = \frac{2}{3} P_{lim}$$

Where p_{lim} is the calculated limit load.

Plastic analysis

Inelastic analysis is used to determine the plastic load of a vessel. The analysis is based on a model of the actual material stress strain relationship and may assume small or large deformation theory. According to ASME VIII Division 2¹⁰ the allowable load, p_a , is calculated thus:

$$p_a = \frac{2}{3} p_p$$

Where p_{lim} is the calculated limit load.

Twice Elastic Slope Criterion

A fully inelastic finite element analysis is a more difficult method of calculating a plastic load, unlike the limit load analysis a plastic analysis does not have a “built-in” failure mode and there is a little more work required on behalf of the analyst in predicting the pressure at which the vessel may burst or collapse.

To aid the analyst in calculating the plastic load, on completion of the analysis a graphical technique known as the “twice elastic slope” may be employed. A load-deformation relationship obtained from the plastic analysis can be plotted and a line, called the “collapse limit line” drawn at an angle twice that of the elastic portion of the curve. The plastic load is considered the corresponding value, P_ϕ , at the point where the collapse limit line intersects the load-deformation curve. See fig. 2-9.

This method of calculating plastic loads must be used with care, as noted by two prominent authors in the field of stress analysis of pressure vessels – Gerdeen³³ and Moffat³⁴. Gerdeen pointed out that the true collapse load of a pressure structure is often much higher than the calculated twice elastic slope plastic load and that the term “collapse limit line” was a misnomer – he recommended the use of the term “plastic load” instead. He also noted that the area underneath the curve (as in fig. 2-9) should be

representative of work done and was opposed to the use of strain as a metric, since it does not fulfil that requirement.

Moffat and his co-workers at the University of Liverpool suggested³⁴ a 15 times elastic slope method for calculating limit loads using the aforementioned graphical technique. The choice of multiplier was apparently arbitrary, but is said to ensure the deformation curve has reached a plateau – something the twice-elastic slope method does not always achieve.

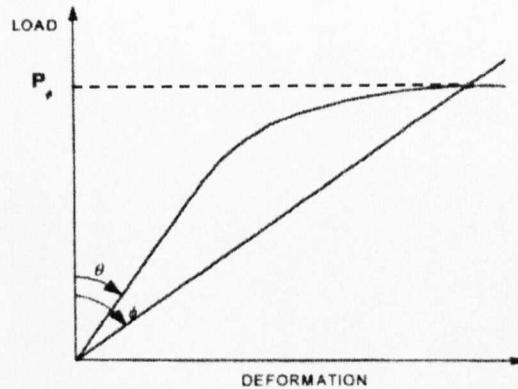


Fig. 2-9: ASME Twice elastic slope criterion for assessment of allowable pressure load

2.6 Closure

This chapter set out to introduce various methods for the elastic and inelastic analysis of pressure vessels, their capabilities, relevance and limitations and began with a brief introduction to shell analysis and the underlying mathematical principles. The reader was then introduced to the design code used by Spromak Ltd. in the design of their fittings.

An historical overview was given on work done in the field of elastic stress analysis of pipe junctions using the finite element method with particular emphasis on deriving ESFs for RBOs – which is discussed further in chapter 3. An explanation of the typical loads experienced in such systems and the method for calculating the ESF values in each case was given.

Inelastic analysis was discussed, explaining possible reasons for using such a technique. Material models – elastic perfectly plastic, bi-linear and full non-linear – were presented as possible options for use in FE stress analysis and was followed by a discussion of the differences between the two multi-axial yield criteria – von Mises and Tresca. Methods for implementing inelastic analysis together with reference to a number of papers on the subject were presented before considering how limit and plastic analyses are interpreted under the ASME Section VIII & Section III codes. The section on inelastic analysis concluded with a brief discussion of the twice and fifteen-times elastic slope methods of calculating plastic loads from full non-linear finite element analysis.

It is hoped that the reader at this point will now be more familiar with the tools available for pressure vessel analysis and have an appreciation of the background against which the following work is set.

3 ELASTIC FINITE ELEMENT ANALYSIS OF REINFORCED PIPE-WORK JUNCTIONS

3.1 Introduction

This chapter describes, in some detail, all aspects concerning the modelling and analysis, in three dimensions, of a reinforced pipe junction. Once the geometry of the junction has been discussed, the reader is guided through the steps taken towards completing modelling and analysis of 92 RBO junctions. Results are presented and discussed with reference to previous relevant work done in this field and, finally, conclusions are drawn with respect to the effectiveness of RBOs compared with UFT.

3.2 The Reinforced Buttweld Outlet

The term “reinforced buttweld outlet”, in this context, describes a forged steel component designed to strengthen the area where circular section branch and run pipes intersect. There are several methods by which a designer can incorporate the extra material necessary to provide reinforcement and there are a number of designs for such fittings.

The type of fitting that this research considers is the welded-on type with a counter bore as shown in fig. 3-1. Essentially, it is manufactured to sit on top of the run pipe and welded into position. Generally, there are two designs for this type of fitting in respect of branch connectivity – with and without a flange. The geometry of both is exactly the same in the area of interest and differs only at the top of the fitting where the former incorporates a flanged bolted connection with the branch pipe and the latter is simply buttwelded to the branch pipe.

Spromak Ltd., were particularly interested in presenting results with respect to their flanged fittings and since the area of interest with respect to stress effects is located at the pipe junction intersection, the flanged butt welded outlet type was modelled.

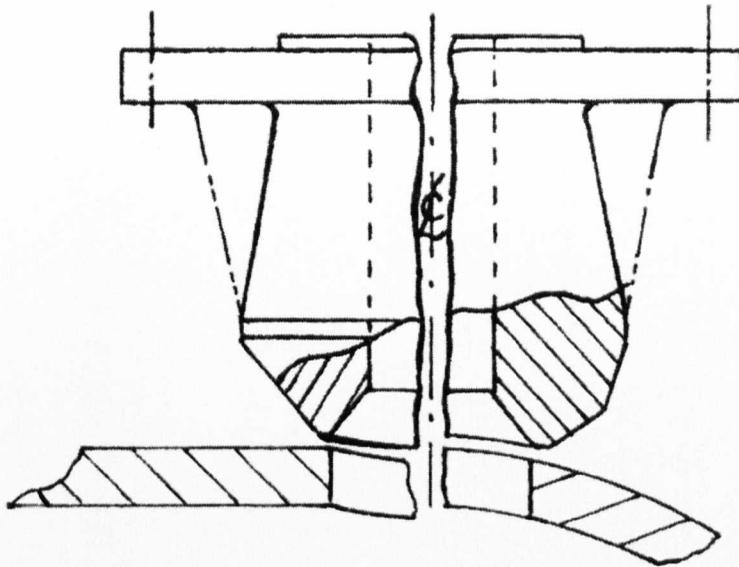


Fig. 3-1: Geometry of a “welded-on” reinforced butt-weld outlet

Both pipe and reinforced branch outlets are available in a number of different “schedules”, a measure of relative strength, according to BS1600 and ANSI B.36.10. Tables for both pipe and RBO bores are in Appendix A. Due to the extensive range of schedules and the numerous permutations of branch to run ratios, the modelling strategy was based around the desire for complete parameterisation of the geometry.

The geometry of a reinforced butt-weld outlet when in the welded-on state is deceptively simple in appearance. Closer inspection reveals that the relationship between the three geometries concerned – the fitting, run pipe and weld – is much more complex (see fig. 3-2) and warrants special attention if accuracy of modelling (and, ultimately, the analysis) is the goal.

The geometry of a weld profile for a butt-welded branch outlet is illustrated in fig. 3-2(a) – which shows two views: longitudinal and transverse. In the longitudinal plane the angle α is, in all cases, $+30^\circ$ from a vertical plane at the point of intersection.

For reducing sizes, i.e. a branch and run combination such that the nominal diameter of the branch is at least two sizes smaller than the nominal diameter of the run, the weld angle, ψ , in the transverse plane is always $+5^\circ$ from vertical. For first reducing sizes, i.e. those branch outlets being one size smaller than the run, ψ is taken as $+1^\circ$ from vertical. For the equal diameter cases ψ was taken to be equal to the branch's reducing angle, ζ . The profile of the weld changes continually through 90° of rotation from crotch to flank positions. The relationship between the weld angle α_β , i.e. the angle α at angle β around the weld, and the angular circumferential position on the run pipe is illustrated by fig. 3-2 and may be described mathematically thus:

$$\alpha_\beta = \alpha_i - \left[\frac{(\alpha - \psi)}{90} \times \beta \right] \quad (3.1)$$

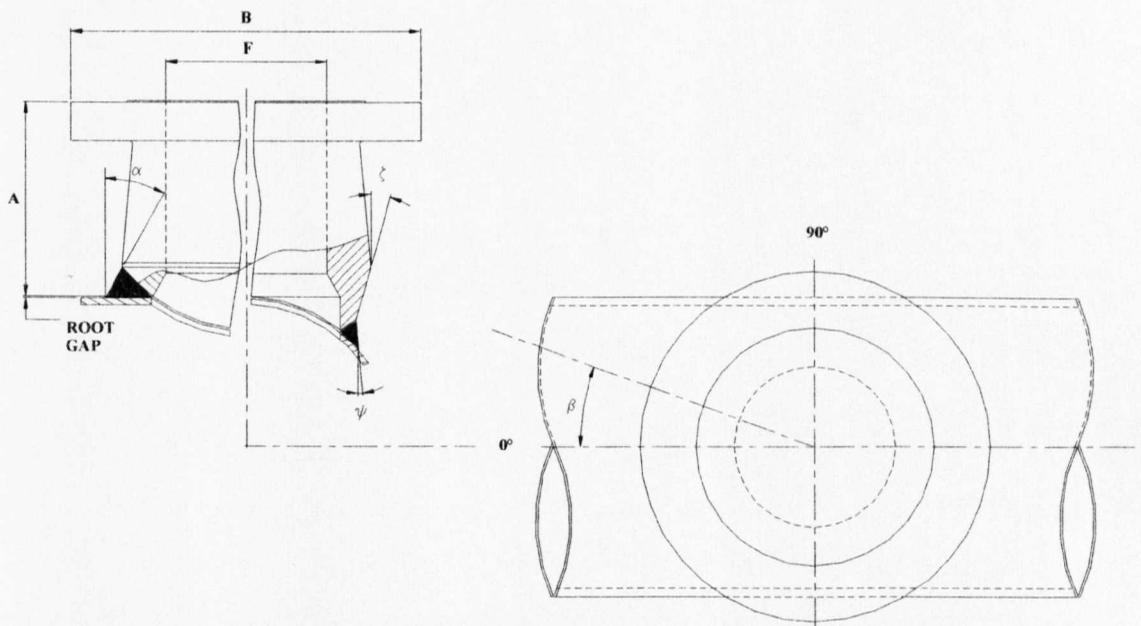


Fig. 3-2: (a) Weld profiles (crotch and flank positions) for a RBO on a cylindrical vessel (b): Relationship of weld angle with position around the circumference of the vessel.

3.3 Parametric Modelling

As previously mentioned, one of the primary goals of the modelling phase was to parameterise the geometry, such that numerous fittings could be readily analysed without the need for working out the intersection profiles by hand each time. Due to the non-symmetric nature of the geometry and in light of the fact that the seven main piping loads experienced by a pipe junction were required to be analysed (see fig. 2-4), the models, of necessity, had to be in three dimensions. Since the intersection profile is complex, parameterisation via use of three dimensional primitives inside a finite element pre-processing package was simply not possible, so an alternative method of modelling using points, lines and surfaces was used together with a number of computer programs to automate the parameterisation process.

Three software packages were used in modelling and analysing the geometry described herein. The programs that generated batch input files describing the geometry, mesh, boundary conditions and loads were written using MATLAB 5.3. Finite element pre and post processing was done using FEMGV 6.1 and the solver used for the finite element analyses was ABAQUS/Standard 5.8-1³⁵. Below is a short description of the function of each software component used.

MATLAB v. 5.3

MATLAB is a technical computing tool widely used in the engineering community. This software was used to program a number of scripts that work together to produce an ASCII file populated with FEMGV keywords. The file is subsequently imported into FEMGV as a batch input file.

There are two methods of input: command line or via a Graphical User Interface. The former prompted the user at the command line in the MATLAB working environment for a filename followed by dimensions taken from manufacturing drawings.

The GUI was produced in order to provide a more intuitive interface for the user, whilst generating the same output as its command line based counterpart. Fig. 3-3 shows the

interface the user is presented with when interacting with the model generator programs. Once a filename and general pipe and branch sizes have been specified, along with material and load data, a window appears with boxes for entering dimensional data for the RBO being analysed. When the data is entered, the user simply selects the “Create History File” button at the bottom of the popup box shown in fig. 3-4 to create the ASCII batch input file used by FEMGV. The MATLAB program code is in Appendix B.

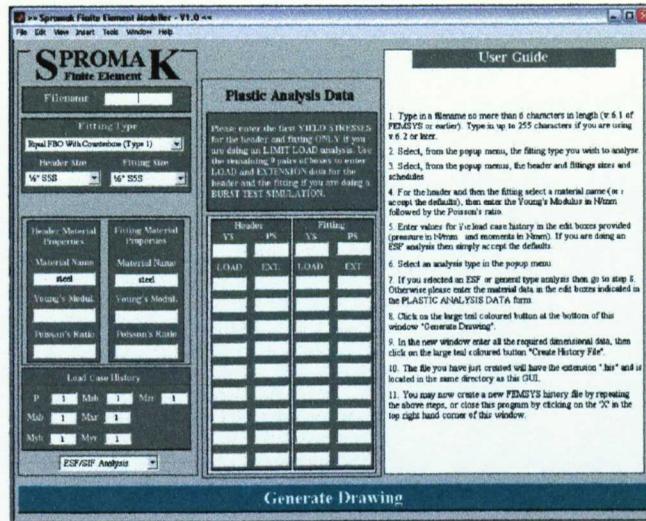


Fig. 3-3: The MATLAB GUI start-up screen

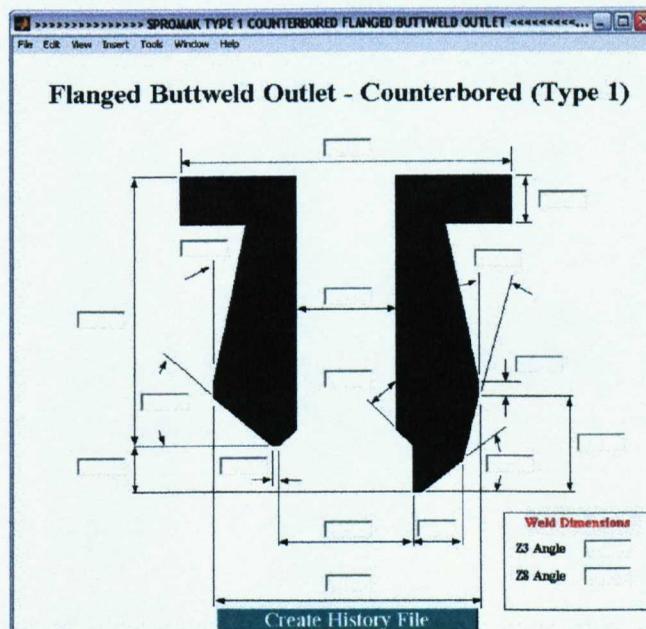


Fig. 3-4: The MATLAB GUI dimensional data input window

FEMGV v. 6.1

The pre and post processing of all finite element models is achieved using the FEMGV package. It processes the batch input files produced by MATLAB and generates the entire model, including geometry, mesh, loads and boundary conditions on-the-fly. Once done, the user has the task of making any necessary refinements to the mesh and then generating the ABAQUS input deck (.inp file) for use within the ABAQUS general-purpose finite element solver – the results are reviewed using the FEMGV post processor, FEMVIEW.

ABAQUS v. 5.8-1

The ABAQUS/Standard v.5.8-1 general-purpose finite element solver is used to process the input files. The results from the analysis are taken from the relevant ABAQUS output file (.fil) and scrutinised in FEMGV's post-processing program.

3.3.1 Scope of work

In total, this work comprises a parametric survey of 92 RBOs and the range of finite element models is defined in Appendix C, tables C1, C2 and C3. In an industrial environment the branch and run sizes are often quoted in terms of their respective nominal bore sizes in inches. Briefly, three different diameter run sizes 8", 10" and 12" were selected and outlets were then chosen and a full range of diameter ratios were analysed for each run size.

The dimensionless ratio d/D in this work is that of the branch nominal bore to the run nominal bore and the wall thickness of a pipe is referred to as its schedule. The parameters of interest, therefore, are d/D and D/T .

3.3.2 Finite element modelling

In order to evaluate the seven loads of primary interest the full geometry was modelled in each case. This is logical in light of the fact that the vast majority of computational effort is expended decomposing the stiffness matrix for the model, and is largely independent of the number of loads in the analysis. It is therefore more cost effective to produce a single complete geometric model including all individual load cases rather than run multiple half models.

The geometry of each model is built up in the following manner. Firstly, for each of the upper and lower weld profiles, 91 three-dimensional space coordinates were calculated for a 90° quarter section of the geometry. This allowed smooth spline lines to be fitted through these points, thus generating reasonably accurate weld profiles. The crotch-curve was generated by intersecting two cylinders of the requisite diameters.

The remainder of the geometry was created using straight lines and arcs to produce surfaces for the inner and outer “skins” of a quarter of the geometry. Each internal surface was then projected onto its outer surface counterpart creating a solid protrusion, shown in fig. 3-5a. Completing a quarter model in this fashion, the full model was generated by mirroring first the quarter then the half models (see fig. 3-5b).



Fig. 3-5(a): 3D Quarter model RBO geometry. (b): 3D Full model RBO geometry

The finite element models were each constructed using 20-node 2nd order (quadratic) reduced integration elements. The ABAQUS code for this element is C3D20R. 20-node

elements have a proven track record for the analysis of pressure vessels and pressure components. Refs.^{12, 23} also Williams³⁶, Moffat and Kirkwood³⁷, Moffat, Mistry and Moore³⁸ have all successfully used these elements for determining elastic stresses in branch pipe junctions.

The cantilever model shown in fig. 2-4 represents the boundary condition adopted for analysis of all seven load-cases. This method for modelling FE boundary conditions has been widely used for branch connections^{23,36}; also Weiss and Joost³⁹ and Mwenifumbo⁴⁰. For the finite element model this means that all nodes at one end of the run pipe are fixed for translation and rotation.

Both refs.^{39,40} pay special attention to locating the nozzle or branch under consideration on the run pipe or vessel at a distance from the fixed end boundary condition that will ensure the results from any analysis are not adversely affected by the presence of this constraint. Ref.³⁹ sets the distance between the fixed end of the run and the outside surface of the branch to a value of at least $15 \times b$, where b is $\sqrt{D_m \times T}$.

Mwenifumbo set the limb length of the run pipe in each case equal to $3.43 \times D_o$ – a factor derived from his stress die away studies for 90° branch pipe connections. Subsequently, Moffat et al²³ adopted Mwenifumbo's length factor, the total run length for each model being $6.86 \times D_o$. The total run length for all analyses described in this thesis was $6.86 \times D_o$.

A unit internal pressure load on all internal surfaces and three individual unit moment loads, on each of the end surfaces of the branch and the run, were applied. The moment loads are transmitted to the branch end surfaces via a rigid body reference node. Both the branch and the run pipe had such a node located at their “free ends” on their central axes. Each node on the surface of either the run or branch end is tied to its rigid body reference node by a rigid beam element. Thus, when a load is applied to the rigid body reference node, it is smoothly transferred to the surface in question. The following assumptions are made in modelling the reinforced buttwelded outlets:

1. One end of the run pipe is constrained as per the cantilever model.
2. The total length of the run is always $6.86 \times D_o$.
3. The branch limb length is always equal to the height of the reinforced butt-weld outlet in the welded state, also known as the stand-out height.
4. The material properties for each complete model are homogeneous and isotropic and obey Hooke's Law.
5. Residual stresses induced by the welding of the branch to the run pipe are neglected, as are the effects of self-weight and temperature.

3.3.3 Mesh convergence study

A mesh convergence study was performed on a 6" flanged nozzle schedule XXS on a 12" schedule XXS run pipe. A quarter of the geometry was modelled, a unit pressure load applied to all internal surfaces and symmetry boundary conditions were applied to the two symmetry faces in the XY and ZY planes. One node was constrained in the global Y direction at the inside surface of the run pipe. Equilibrium was maintained by simulating a "closed end" condition at the "free ends" of both the run and the flange.

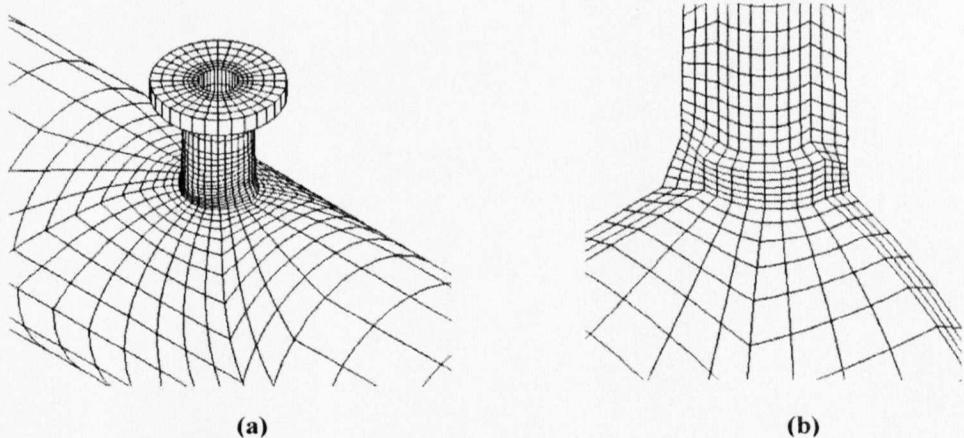


Fig. 3-7: (a) Converged mesh for model FN0995. (b) Converged mesh for model FN0995 – intersection view.

Since the region in which the branch and the run pipe intersect was the particular site of interest, the mesh convergence check focussed on that area. In total six models were

analysed; the mesh considered to have converged is that which has a relative mesh density of 7.2 and three elements through the thickness as shown in figs. 3-7(a) and (b). Von Mises stress results for all six models under internal pressure load is plotted against relative mesh density in fig. 3-8. Relative mesh density is defined here as the density of the mesh in question, divided by the density of the coarsest mesh.

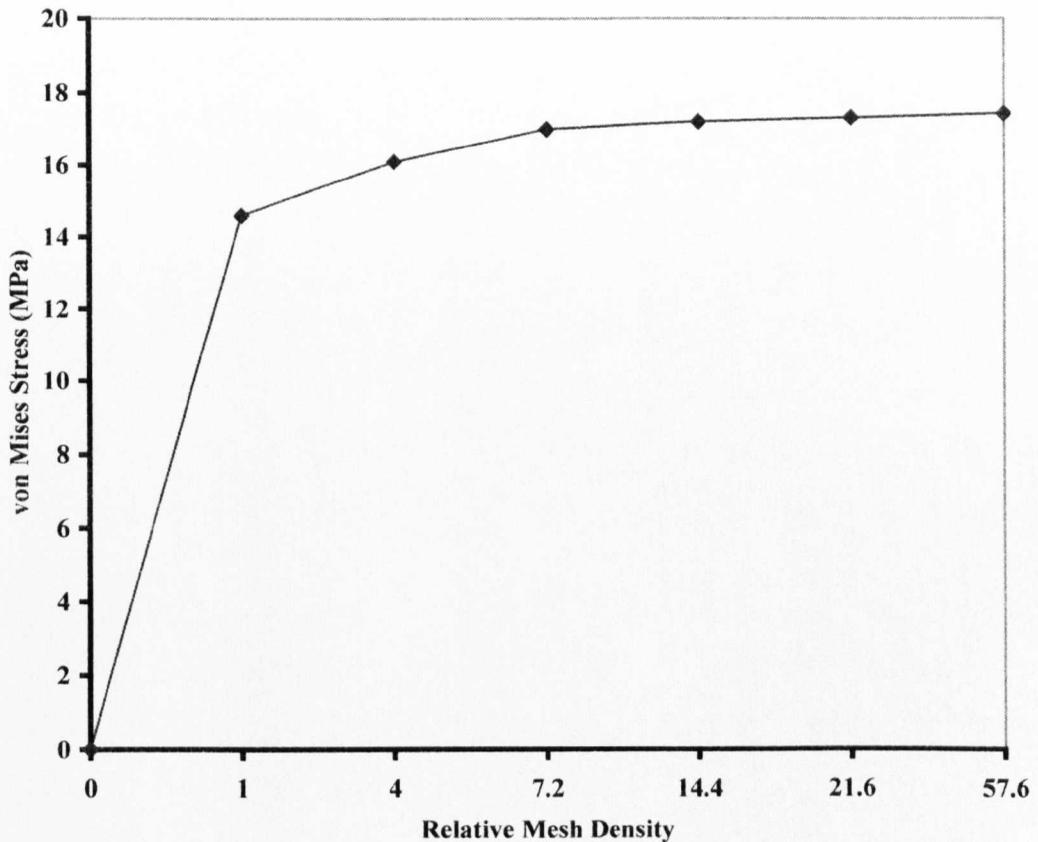


Fig. 3-8: Mesh convergence study: number of elements in crotch area versus von Mises Stress

3.4 Presentation and Discussion of Results

The results herein are solely the product of finite element analysis of reinforced butt-welded 90° branch outlets. The data derived are compared with data published by a number of authors for the analysis of branch connections for the purpose of demonstrating the effectiveness of the reinforcement of fittings described here.

However, the geometries of the branch connections in these papers do differ significantly from the geometry analysed here. In 1968, Money⁴¹ presented results for the stress analysis of UFT based on approximately 90 steel and Araldite photo-elastic, experimental tests. 1975 saw Decock⁴² publish data on 156 specimens using similar experimental methods to Money.

In 1991 Moffat et al²³ presented two sets of FE results – one set for equal branch and run pressure strength $t/T = d/D$, the other for branch and run pipes of equal thickness $t/T = 1.0$. It is the data for $t/T = d/D$ presented in²³ and that of^{41,42}, that is used for benchmarking the current study.

The two load cases of particular interest are internal pressure and branch out-of-plane bending moment, as these are generally regarded as being the most severe loads imposed upon a pipe junction. Results for pressure loading are presented in the first instance and compared with the other available data for a number of D/T ratios. Branch out-of-plane moment loading is then considered in a similar fashion.

It should be emphasized that, the dimensionless ratio d/D is the ratio of the nominal bore of the branch pipe attached to the butt-welded outlet to the nominal diameter of the run. The ratio t/T is that of the thickness of the branch pipe attached to the butt-welded outlet to the thickness of the run.

A full set of ESF versus diameter ratio results are presented in Appendix D for the three different header sizes used in this study for all seven load cases.

3.4.1 Pressure load case

To draw any meaningful conclusions with respect to the true performance of the reinforced branch outlet design it is useful to compare the ESF data presented in fig. 3-9 with that in²³ for the un-reinforced design of pipe junction. In addition, the Money and Decock curves are also considered in the presentation of results. Since the present study has no data for the case $D/T = 40$, it is appropriate only to compare the data for

$D/T \approx 10, 20, 60$ in the range $0.05 \leq d/D \leq 1.0$. In any event, there is very little ESF data available for RBOs in open literature to which the data presented here might be compared.

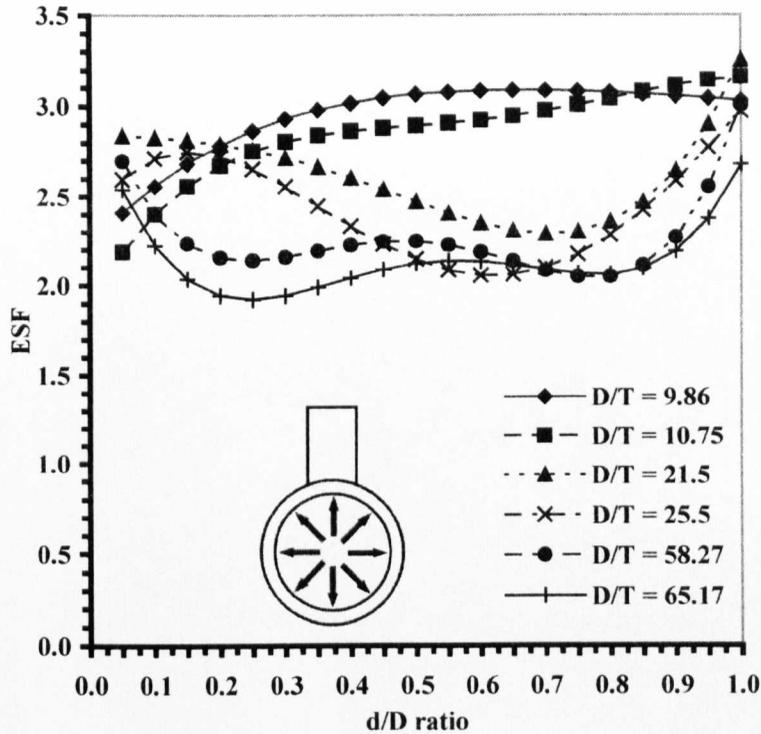


Fig. 3-9: ESFs for six D/T ratios under unit internal pressure.

Fig. 3-9 shows the pressure ESF results for the full range of RBOs. Each “pair” of D/T ratios – those that are numerically closest to one another – have similar curves and ESF values across the full range of diameter ratios. Perhaps the most relevant piece of information is that, in all cases, the value of ESF lies between 2.0 and 3.25. When compared to the range of UFT ESF values ($2.5 \leq ESF \leq 5.4$) in²³ for similar D/T ratios, the data suggests that the reinforcement is performing according to the design intent and minimising the stress induced by the load case that is of primary concern.

The perceived strength of the reinforced fittings is further enhanced when one considers fig. 3-9 in light of the hypothesis that, very small branches are analogous to a large flat plate with a hole subjected to 2:1 biaxial stressing with an associated ESF of 2.5^{23} .

Another piece of information that may be deduced from fig. 3-9 is that there appears to be no obvious relationship between increasing diameter ratio and ESF. Unlike the curves for UFT under internal pressure presented in²³, there is no common relationship between D/T and d/D ratios and no neat extrapolation to the theoretical ESF of 2.5. Hence, any attempt to derive credible parametric formulae for the anticipation of ESFs using conventional techniques would be unlikely to be successful.

Figs. 3-10(a) – 3-10(c) present comparisons between the ESF results for reinforced branch outlets versus the UFT ESF data^{23,41,42}. Fig. 3-10(a) represents data for $D/T = 10$, the two curves for the current FE data being $D/T = 9.86$ and 10.75 . For these relatively thick-walled run pipe and branch combinations, ESF values are generally low (less than 4.75 in all cases). However, the reinforced fittings perform particularly well, with the majority of d/D cases having ESFs under 3.0.

Generally, the $D/T = 9.86$ case shows higher ESF values than for $D/T = 10.75$. When comparing the former of these two with the Decock, Moffat et al and Money curves, and in summary of fig. 3-10(a), the following is observed for ESF values for the range $0.05 \leq d/D \leq 1.0$. Comparing the Decock and $D/T = 9.86$ curves at $d/D = 0.05$ there is a 10% difference in ESF, with the former performing better than the RBO. However, the performance difference at $d/D = 0.2$ is reduced to 1% and at $d/D = 1.0$ the margin has risen steadily to a 24% performance advantage for the reinforced fitting over its unreinforced counterpart.

For the Moffat et al versus $D/T = 9.86$ case, at $d/D = 0.05$ the reinforced branch outlet shows a 10% advantage over the UFT model. ESF values rise sharply to $d/D = 0.5$ where the difference is 30% and increases further to 36% at $d/D = 1.0$.

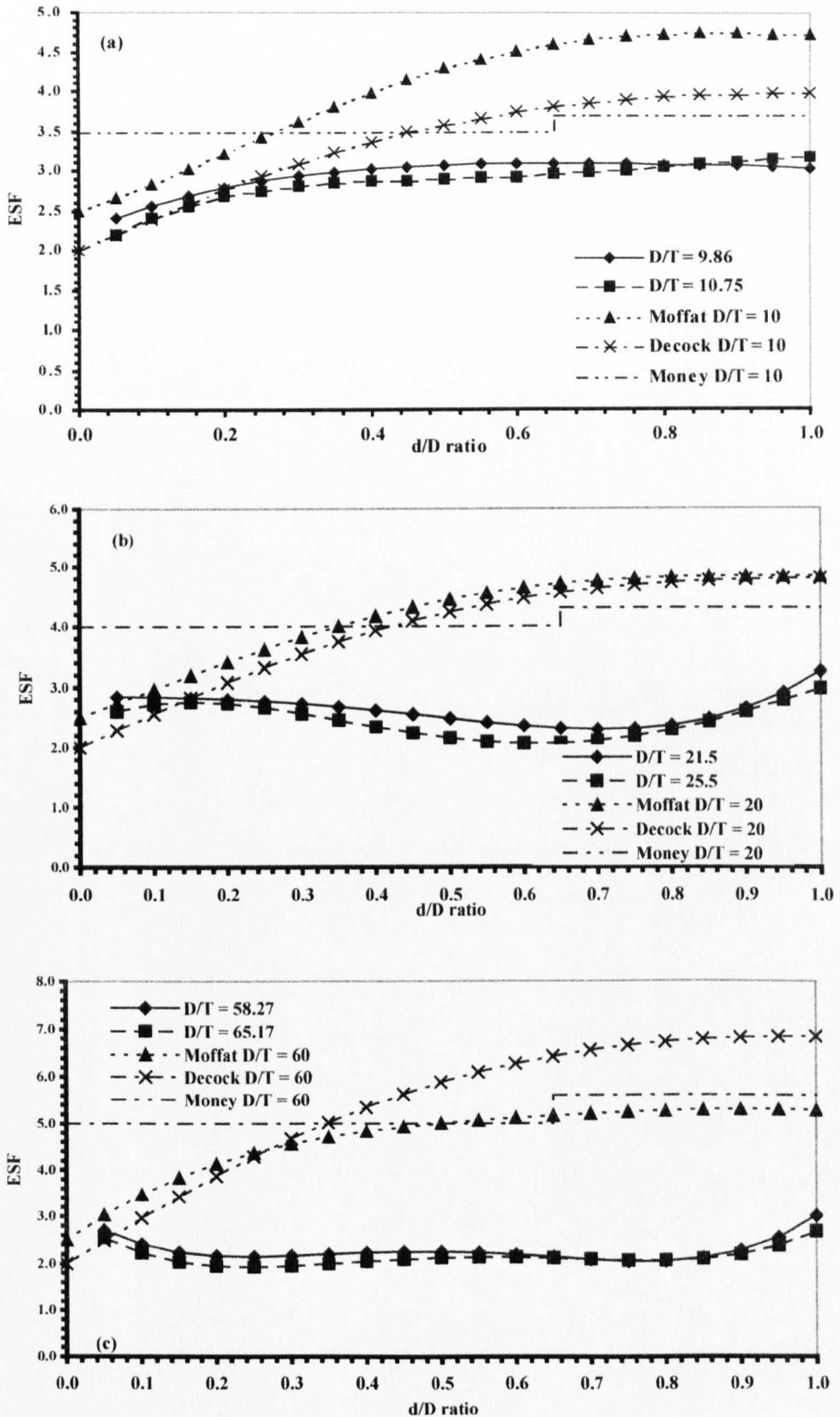


Fig. 3-10: Comparison of pressure ESF data for (a) $D/T = 10$ (b) $D/T = 20$ (c) $D/T = 60$.

When comparing Money's data with the results from this study, it is apparent that there is a reversal of trends with respect of the previous two comparisons. Money's data at $d/D = 0.05$ is 31% higher than the current data and this figure falls steadily to 12% at $d/D = 0.65$. There is a dramatic step increase in ESF at $d/D = 0.7$ due to the construction of the Money curve and the difference in ESF increases to 17% creeping to 18% by $d/D = 1.0$ in the reinforced components favour.

Fig. 3-10(b) draws the same comparisons for $D/T = 20$ as fig. 3-10(a) does for $D/T = 10$, with the current FE data having D/T ratios of 21.5 and 25.5. The results presented in fig. 3-10(b) for $D/T = 21.5$ vary between 2.29 at $d/D = 0.7$ and 3.25 at $d/D = 1.0$. For geometries in the range $0.05 \leq d/D \leq 0.7$ there is only a gradual decrease in ESF from 2.84 to 2.29 respectively, representing stable and consistent stress containment for this range of diameter ratios.

In comparing the results for $D/T = 21.5$ with those presented by the aforementioned authors, it is apparent that for the vast majority of d/D cases the reinforced branch outlet outperforms, by considerable margins, the UFT design. The notable exceptions to this rule being at $d/D \leq 0.1$ compared with the Decock and Moffat et al curves. However, the ESFs for these diameter ratios are all very low and therefore the numerical differences are not of appreciable significance.

For diameter ratios greater than 0.1 the reinforced branch outlet begins to show its superiority under pressure as the other three benchmarks rise steadily in ESF value. The current data show a contrary and steady decrease in ESF up to $d/D = 0.7$. At this point, the curve begins an upward trend and, from having an ESF approximately 50% lower than the other UFT curves the ESF value at unit diameter ratio is 33%, 32% and 24% lower than the Moffat et al, Decock and Money curves respectively.

The $D/T = 60$ data for the reinforced junctions presented in fig. 3-10(c) lie, in almost every single case, significantly lower than their UFT counterparts as predicted by Decock, Moffat et al and Money. The curves representing $D/T = 58.27$ and 65.17 show little variation in ESF between $d/D = 0.15$ and 0.85 and it is only at the

extremities of these two curves that the ESF values increase from their average of 2.2 to 2.5 at the lower end of the diameter ratio spectrum and towards 3.0 at unity. Clearly, for the thin walled assemblies, the reinforcement is excellent and is obviously key in keeping stress levels relatively low in comparison with the UFT tee junctions.

3.4.2 Branch out-of-plane bending moment load case, M_{xb}

Branch out-of-plane bending can result in high stresses in the area of the weld and run intersection, in particular for thin-wall assemblies. Generally, ESF increases with diameter ratio, although for the thin-walled assemblies a steep rise in ESF up to diameter ratio of approximately 0.7 gives way to a sharp drop, as diameter ratio tends to unity.

The diameter to thickness ratios 58.27 and 65.17 in fig. 3-11 illustrate this effect and indicate that the extent of the shell bending induced by the moment load is dramatically increased with reducing wall thickness of the run pipe and increasing diameter ratio. However, the relatively flat profiles of four out of six of the reinforced branch outlet curves in fig. 3-11 suggest that, for the heavier weight assemblies, the reinforcement provided is more than adequate to contain the stresses. Additionally, for the heavy-walled junctions, the geometry of these structures in the transverse plane is such that the discontinuity effect is minimised.

As one would expect, and as with the pressure load case, each “pair” of D/T ratios – those that are numerically closest to one another – have, once again, similar curves and values for ESF across the full d/D range. For the heavier walled junctions, the values for ESF lie between 1.0 and 5.0. However, the lighter walled assemblies $D/T = 58.27$ and 65.17 ESF values rise sharply and are much higher than the other four sets of data due to extensive shell bending.

The curves for $D/T = 58.27$ and 65.17 show a maximum ESF, of 22.04 and 25.37 respectively to occur at a diameter ratio of approximately 0.7, after which the ESF values fall away sharply to 8.63 and 10.71 respectively at a diameter ratio of 1.0. It is

postulated here that this is due to the effect of the geometric discontinuity where the weld meets the run pipe at $d/D \approx 0.7$ and the contrasting stress smoothing effect of the weld for junctions of equal diameter. This effect had been noted and investigated by Wordsworth⁴³, and later Schneider⁴⁴, Woods and Rodabaugh⁴⁵.

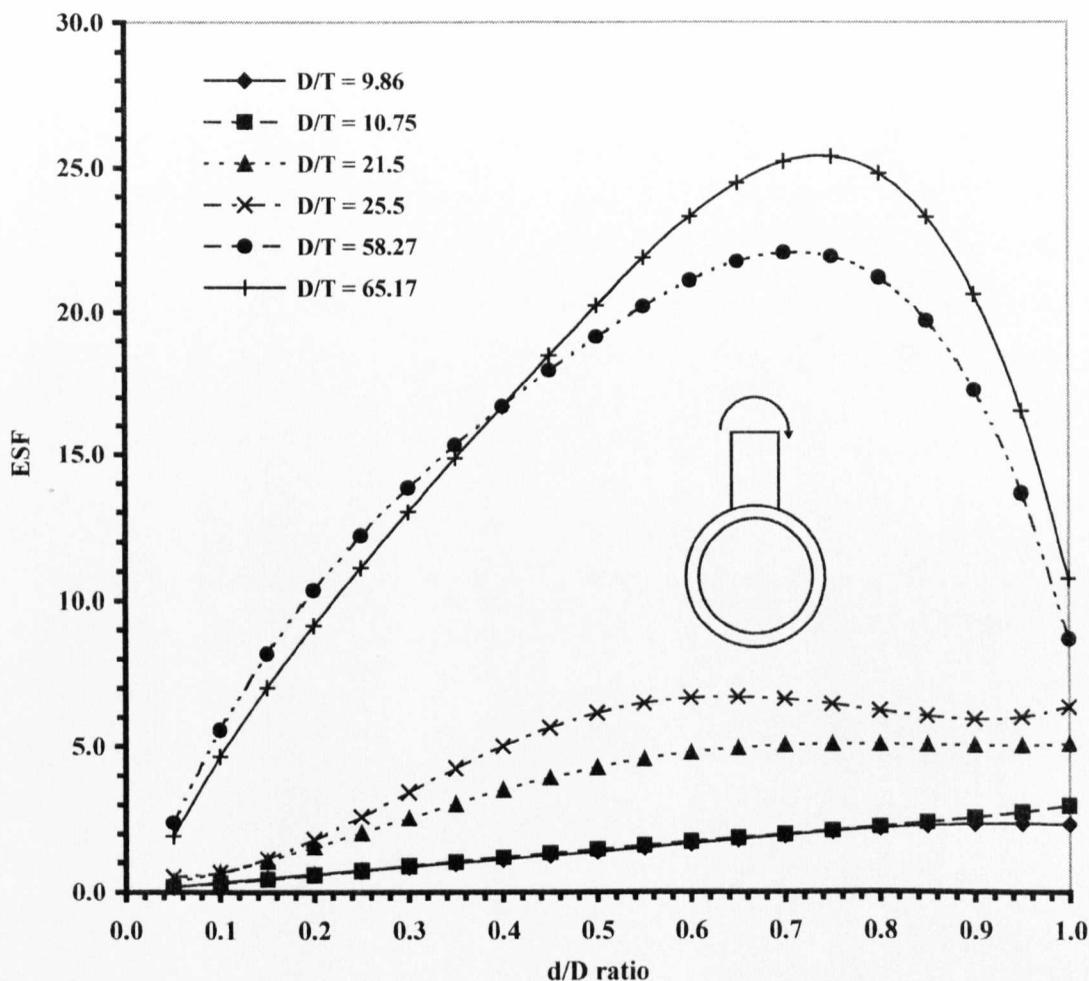


Fig. 3-11: ESFs for six D/T ratios subject to unit branch out-of-plane bending moment load.

The implication of inaccurate stress intensity factors possibly induced by geometry discontinuities in pipe junctions of certain ratios in the ASME design codes, was of concern to analysts and became the subject of a Welding Research Council Bulletin published by Rodabaugh in 1987⁴⁶. In the same year Woods⁴⁷ produced further evidence of potential non-conservative stress concentration factors, within the American codes for the branch out-of-plane bending moment load case, in his paper detailing

results from reverse bending fatigue tests on four reducing fabricated tee geometries. The data produced supported the modification of ASME SIF equations – equations that had been in use since Markl's original work on flexibility analysis was published in 1955⁴⁸.

In comparing the D/T ratios in isolation with other sources of data, there is only one that is sufficient for any meaningful comparisons to be drawn, that is in the data presented by Moffat et al²³. It should be pointed out that the data in²³ cannot be reasonably compared to the data in this research for ESF values below 1.0. This is because, although the RBOs may indeed yield ESFs numerically lower than 1.0 in the immediate vicinity of the junction thanks to the extent of reinforcement, by definition ESFs in the branch of a UFT must be equal to or greater than unity. Therefore, these comparisons shall only be valid for diameter ratios having an ESF greater than 1.0.

Fig. 3-12(a) shows results for $D/T = 10$ from Moffat et al²³ versus $D/T = 9.86$ and $D/T = 10.75$. For these relatively thick-walled assemblies the reinforcement proves more than adequate when compared to the UFT curve. Both FE curves rise with an almost constant gradient from $d/D = 0.35$ to 1.0, with ESF values increasing from 0.98 to 2.32 ($D/T = 9.86$) and from 1.03 to 2.96 for $D/T = 10.75$. This compares to the UFT $D/T = 10$ case where ESF rises from 1.55 to 3.57 across the same diameter range.

Fig. 3-12(b) shows that, for $D/T = 20$ (21.5 and 25.5 for the current FE data), the effectiveness of reinforcement for this load case is contentious. Perhaps surprisingly, the UFT junction of $D/T = 20$ returns a lower ESF than the reinforced junctions $D/T = 21.5$ in the range $0.15 \leq d/D \leq 0.65$. At $d/D = 0.65$ Moffat et al's curve and the $D/T = 21.5$ curve intersect and the ESF values for the latter level off at approximately 5.0. However, Moffat et al's curve continues upward and reaches a final peak ESF value of 6.41 at unit diameter ratio – some 21% higher than its reinforced counterpart.

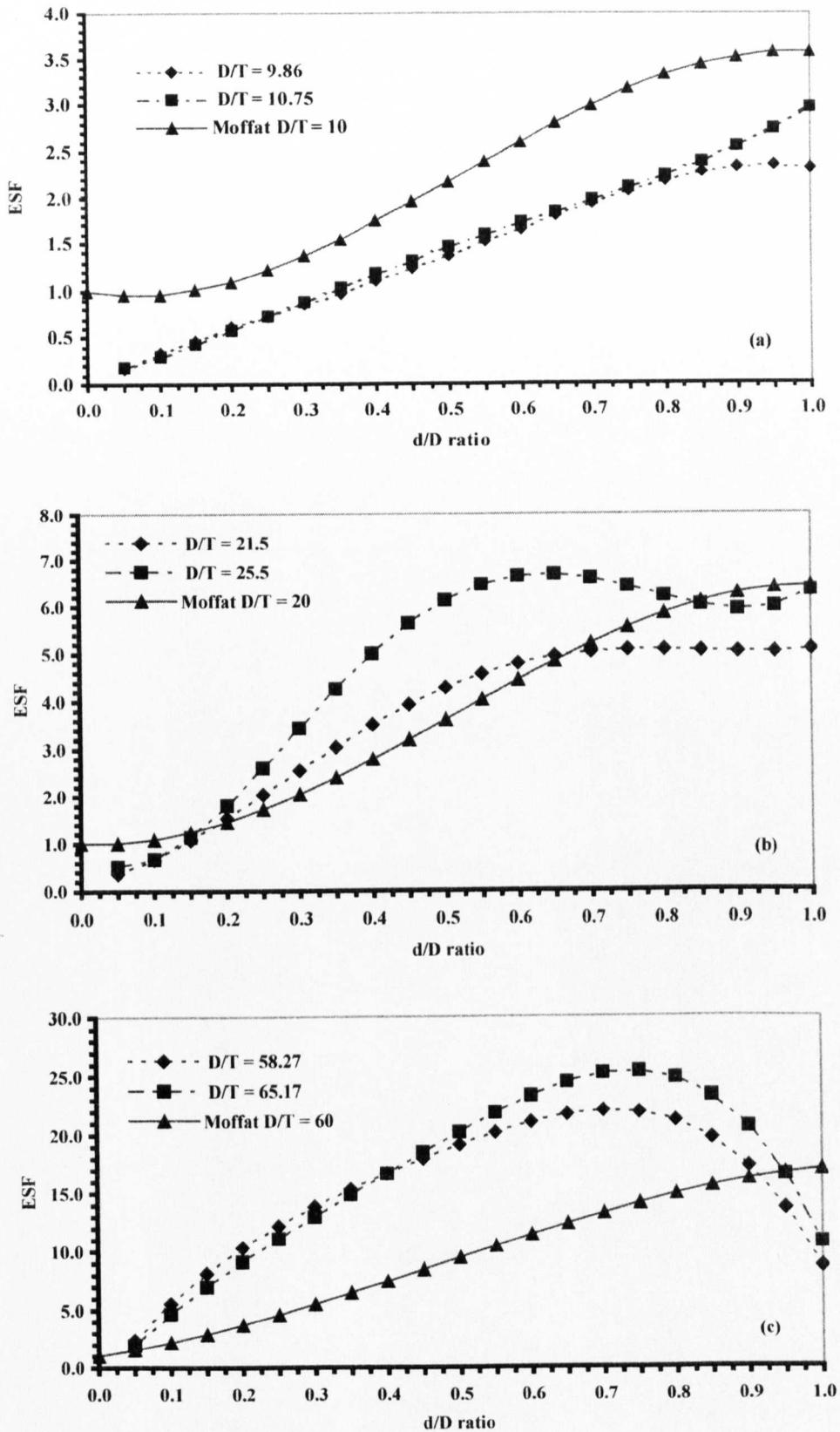


Fig. 3-12: Comparison of branch out-of-plane ESF data for (a) $D/T = 10$ (b) $D/T = 20$ (c) $D/T = 60$.

An even greater advantage of the UFT over the reinforced junction $D/T = 25.5$ is demonstrated by the latter only dipping marginally below Moffat et al's $D/T = 20$ curve at $d/D = 0.85$ and remaining only fractionally below at equals diameter.

The $D/T = 60$ data shown in fig. 3-12(c) is quite dissimilar to the previous two D/T cases. For both $D/T = 58.27$ and $D/T = 65.17$ models the value of ESF increases sharply, peaking at a d/D ratio of approximately 0.7 to fall away in dramatic fashion as unity is approached. At $d/D = 0.7$ the $D/T = 58.27$ case exceeds the predicted value for a plain pipe junction $D/T = 60$ by 39.9% and at $d/D = 0.75$, $D/T = 65.17$ there is a 44.5% increase in ESF over Moffat et al's figure. This data, in the light of that for $D/T \approx 10$ and $D/T \approx 20$, tends to support the theory that for thicker walled assemblies the effect of the weld on shell bending is minimised due to the lesser extent of geometric discontinuity.

Fig. 3-12(c) suggests that the thin walled structures are susceptible to excessive shell bending in the run pipe as the moment is transferred through the discontinuity at the weld/run intersection.

From figs. D1, D2 and D3 in Appendix D it is apparent that, of all the moment load cases, out-of-plane bending on the branch is by far the worst a piping intersection will have to withstand and, for very thin-walled assemblies, this load may induce very high stresses. Schneider⁴⁴ states:

"... after reviewing much of the SIF data the parameter appeared to reach a maximum value at r/R of between 0.7 and 0.8".

Some consideration was given to the subject of the presence of a peak in stress factors for fabricated branch outlets for out-of-plane branch moment load in a report by Woods and Rodabaugh⁴⁵, which was written to provide ASME and ANSI with an experimental basis for improving of the codes with respect to this anomaly. The authors presented a hypothesis, first postulated by Wordsworth⁴³ that the variation in stress for branch out-of-plane bending could be rationalised by regarding the way in which the branch load is transferred to the run pipe. The supposition is that, for $d/D = 1.0$, the welded junction

presents less of a geometric discontinuity to the applied load and there is, therefore, no stress raiser as such. For a reducing branch outlet with a d/D ratio of, say, 0.7 there is a significant geometric discontinuity and therefore the extent of shell bending is greater. Figs. 3-13(a) and (b) help illustrate this more clearly.

Where $d/D \ll 1.0$, the branch is too small to impose any significant load upon the run pipe thus, for small d/D ratios one would not expect to see high stresses, despite the obvious and large geometric discontinuity. At $d/D = 1.0$ the obvious discontinuity between the weld and run has largely been removed.

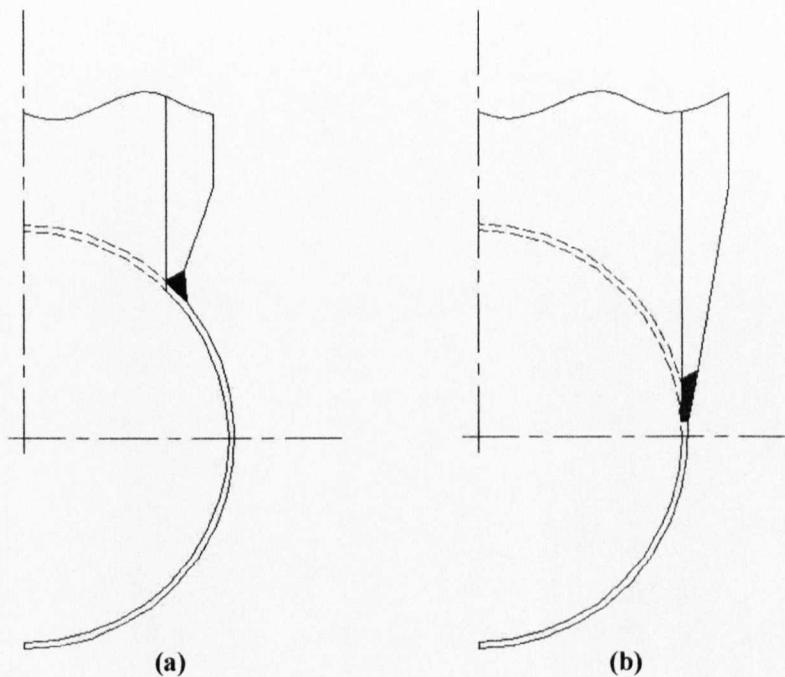


Fig. 3-13: (a) Transverse plane $d/D = 0.7$. (b) Transverse plane $d/D = 1.0$

In 1997, Chen and Wu⁴⁹ published their results for the low cycle bending fatigue testing of branch junctions. They conducted a total of twenty-six experiments on four UFT specimens, twelve butt-welded reinforced branch outlets and ten reinforced contour insert fittings. The authors produced tabular data for stress concentration factors for all twenty-six experiments.

The twelve RBOs were manufactured according to a design by WFI. These results are of particular interest as this recent experimental data represented something that could be readily compared with the current FE data for Spromak's design. However, the twelve branch assemblies are effectively reduced to five, as just five different diameter ratios were considered. Chen and Wu considered their results in light of stress factors calculated using the ASME III, B31.1 and B31.3 design codes.

The ASME III stress index for out-of-plane branch bending moment load, C_{2b} , can be equated to ESF:

$$ESF = C_{2b} \quad (3.2)$$

In their paper⁴⁹ the authors presented a table for comparison of the experimentally measured stress index C_{2b} with ASME III code calculated C_{2b} values. Since more than one experiment for each size of fitting was done, the experimental data have been averaged so that a direct comparison with this study's FE data and ASME III values can be made. It is worth noting that Chen and Wu's experimental data presented in fig. 3-14 closely resembles the current FE ESF curve, both showing a peak at $d/D = 0.75$.

The ASME III values correlate closely with the current FE data up to $d/D = 0.75$, after which the two curves diverge significantly. The ASME III curve suggests that ESF will continue to rise with increasing diameter ratio – much like the Moffat et al²³ data for this load case. However, both the experimental results⁴⁹ and the work presented here shows that, for reinforced branch outlets, the maximum ESF is reached at a diameter ratio of between 0.70 and 0.75. This is also supported by the experimental work of Schneider⁴⁴ and Woods⁴⁵.

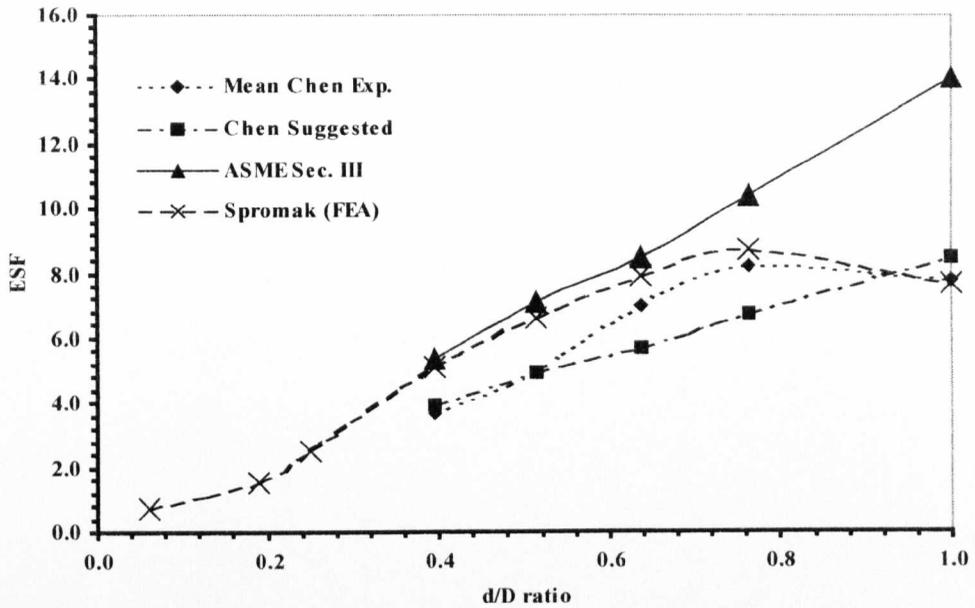


Fig. 3-14: Comparison of branch out-of-plane bending moment ESF data with the ASME III code and Chen et al experimental data

Table 3-1 presents the data shown in fig. 3-14 along with the percentage difference between the current FE data, ASME III and the Chen and Wu's experimental results.

d/D	FE	ASME III	% Difference	Chen & Wu	% Difference
0.06	0.75	-	-	-	-
0.19	1.52	-	-	-	-
0.25	2.54	-	-	-	-
0.38	5.14	5.36	-4.1	3.73	+27.4
0.50	6.64	7.12	-7.2	4.96	+25.3
0.63	7.93	8.45	-6.6	7.01	+11.6
0.75	8.69	10.37	-19.3	8.21	+5.5
1.00	7.66	14.03	-83.2	7.78	-1.6

Table 3-1: Comparative ESF values for a limited range of reinforced branch outlets

In summary, for branch out-of-plane bending moment load case, it can reasonably be concluded that, for heavy walled sections, the effect of this load is well contained perhaps due to a more equal distribution of strength at the junction between the run pipe

and the component coupled with a smaller geometry discontinuity after the diameter ratio exceeds 0.5. In thinner walled assemblies the discontinuity is much more apparent and the relative stiffness of the welded base of the reinforced branch outlet means that flexibility is reduced and shell bending is exacerbated by the difference in relative strength of the branch and run pipes.

3.4.3 Branch moment load cases M_{yb} and M_{zb}

Branch torsion load, M_{yb}

For a RBO, torsion load of the branch is insignificant, when compared to branch out-of-plane bending or pressure. However, it is worth noting that the reinforcement in this case ensures that, as diameter ratio increases, ESF remains resolutely low. Fig. 3-15 shows just how well the RBO fares compared with its un-reinforced cousin when the diameter ratio increases beyond $d/D = 0.5$.

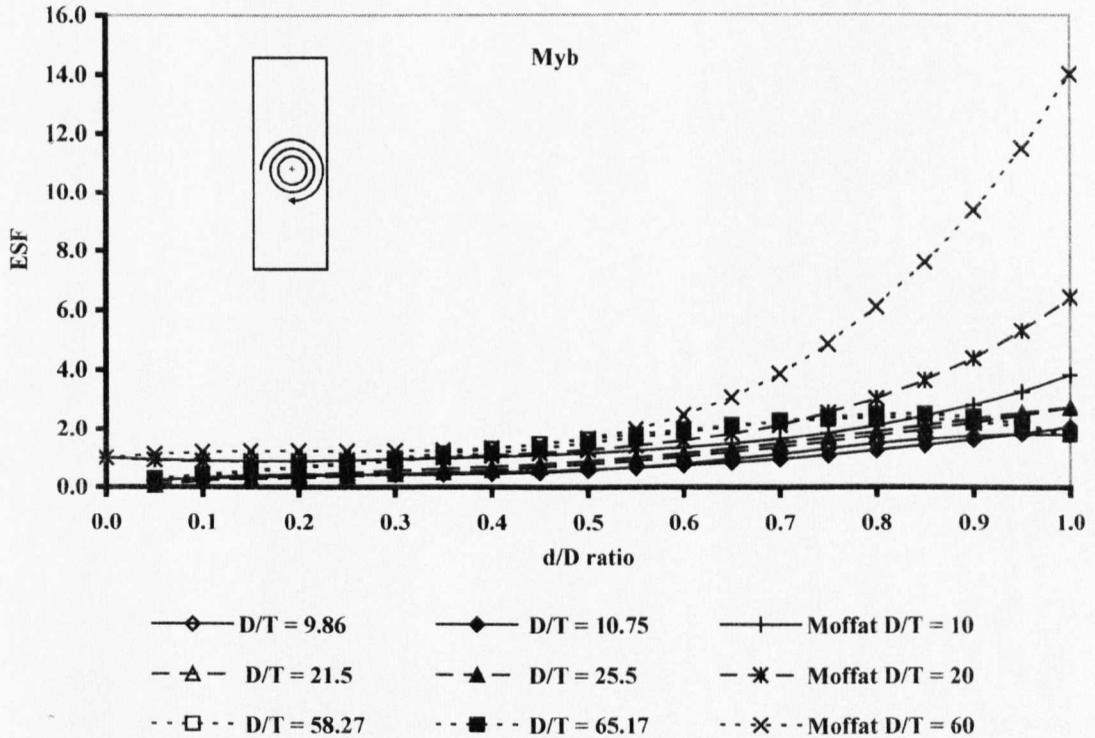


Fig. 3-15: Comparison of branch torsion ESFs

The difference is most notable for the thinner walled $D/T = 20$ and, in particular, the $D/T = 60$ series. For $D/T = 20$ a 34% difference in ESF in the reinforced components favour rises to 57% (and an ESF of 6.46) at $d/D = 1.0$. Although similar differences can be seen for Moffat et al²³ $D/T = 10$ versus $D/T = 9.86$, the overall magnitude of the ESFs are much more modest. The $D/T = 60$ comparison sees a 10% larger ESF from $d/D = 0.55$ to 667% at unit diameter ratio, with the un-reinforced junction having the higher values.

For this load case, it is apparent that a reinforced junction copes very well, with ESF values no higher (and in a good deal of cases much lower) than 2.75. In the light of a theoretical minimum ESF of 1.0 for this case, these results are even more encouraging. For UFT junctions, stresses rise sharply and steadily with increasing diameter ratio. After $d/D = 0.5$, and especially for the thinner walled junctions, stresses rise sharply. These higher ESFs should not be ignored by designers choosing an un-reinforced design.

Branch in-plane bending moment load, M_{zb}

Observations with respect to this load case show that, overall, there is very little to choose between the two methods of joining pipes. The most notable differences appear, once more, when comparing the thinner walled examples. For $D/T = 60$, the UFT junctions perform admirably, with a 31% lower ESF than the reinforced equivalent (an ESF of 2.06 versus 2.97). This advantage begins to erode from that point on until $d/D = 0.65$, when the reinforced junction design starts to demonstrate superiority; the difference in ESF swings in its favour and continues to do so to $d/D = 1.0$ – the UFT having an ESF nearly five times that of the RBO equivalent at that point.

For $D/T = 20$, there is no such advantage displayed by the UFT junctions, the ESFs for the RBOs are some 30% to 40% lower for the majority of comparable cases. Reflecting the typically sharp rising values for UFT junctions in the higher diameter ratio range, the UFT equals junction is 78% in excess of the reinforced equivalent.

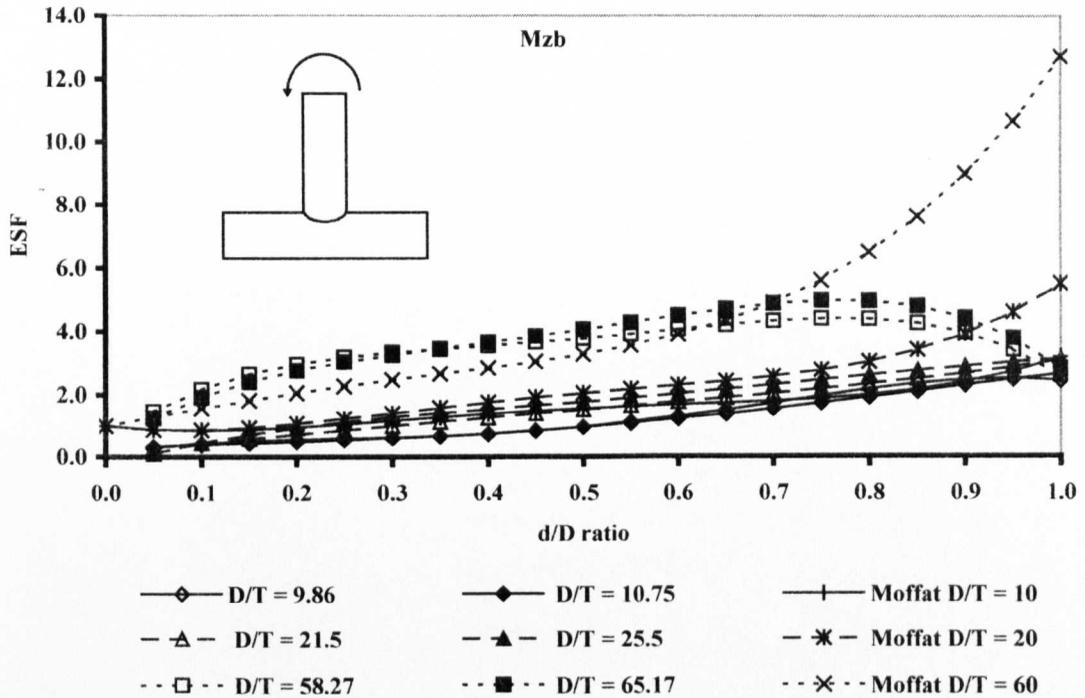


Fig. 3-16: Comparison of branch in-plane bending bending moment load ESFs

Considering the D/T ratio of 10, the reader will note that, once more, the ESF values are quite low – even at equals size. Reinforced junctions of this weight show that they are superior when $0.55 \leq d/D \leq 0.7$ and for diameter ratios in excess of 0.9. In between, UFT proves more effective, although the reasons for this are somewhat obscure.

3.4.4 Run pipe moment load cases: M_{xr} , M_{yr} and M_{zr}

Of the seven load cases imposed upon a piping junction, the run pipe moment loads are the least significant in terms of the magnitude of induced stress – at least as far as reinforced branch outlets are concerned. Nonetheless, the conscientious engineer should give these load cases careful consideration. To give an idea of the scales involved when considering these loads in terms of ESF, not a single reinforced branch outlet case in the data presented below has an ESF greater than 2.8. The same cannot be said for the un-

reinforced junctions, which display familiar trends when considering ESFs for the larger diameter ratios.

Run pipe torsion load, M_{xr}

Torsion of the run pipe is, according to the data presented here, the more likely of the three run moment loads to yield the highest ESFs. Moffat et al²³ suggest that the load case is, for very small diameter ratios, equivalent to a large flat plate with a hole subjected to shear and, as such, has an ESF of 2.0. Consider the following for $D/T = 60$: the reinforced junction starts at $d/D = 0.05$ with an ESF of 2.59. As diameter ratio increase towards 0.5, the ESF also climbs peaking at 2.79 ($d/D = 0.45$). ESF then begins a downward trend towards $d/D = 1.0$ and an ESF of 1.73. These results are encouraging, since it is apparent that, to a large degree, the stresses are well contained, even for this thin-walled assembly.

When these results are compared to UFT, there is a clear performance advantage in the reinforced junctions favour. The former ESF values, whilst always higher than the RBO's, increase dramatically for diameter ratios greater than 0.5, illustrated in fig. 3-17. For $D/T = 10$ and $D/T = 20$, this is not perhaps quite as pronounced, however, there is a distinct advantage displayed by the RBO data over UFT in these cases.

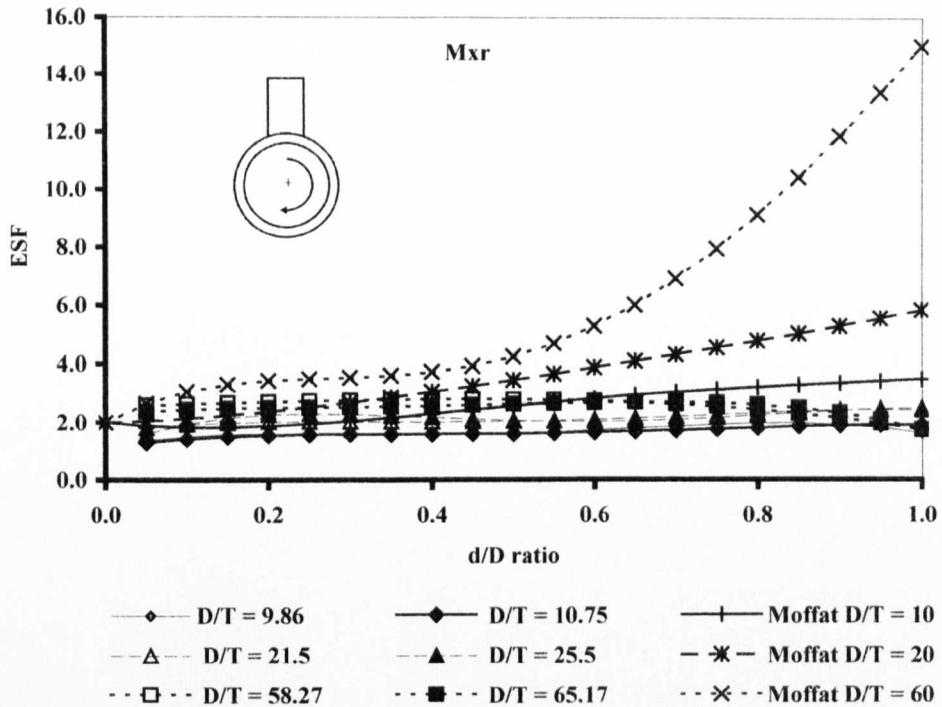


Fig. 3-17: Comparison of run pipe torsion ESFs

Run pipe out-of-plane bending load, Myr

In contrast to the run torsion load case, analysis of the results for out-of-plane bending of the run pipe shows that un-reinforced fabricated tee is marginally better under this load for D/T ratios of 10 and 20 and for diameter ratios less than 0.4. For $D/T = 60$ up to a diameter ratio of 0.45, UFT outperforms the reinforced alternative by significant margins – between 6% and 27%. As has been a feature of all comparisons drawn thus far between the two designs, the performance of the RBO only starts to show at diameter ratios in the order of 0.5. For $D/T = 10$ and for $0.45 \leq d/D \leq 0.85$ this advantage in strength is only modest, hovering around 3%. However, beyond 0.85 the RBO stretches the lead to 33% at a diameter ratio of 1.0.

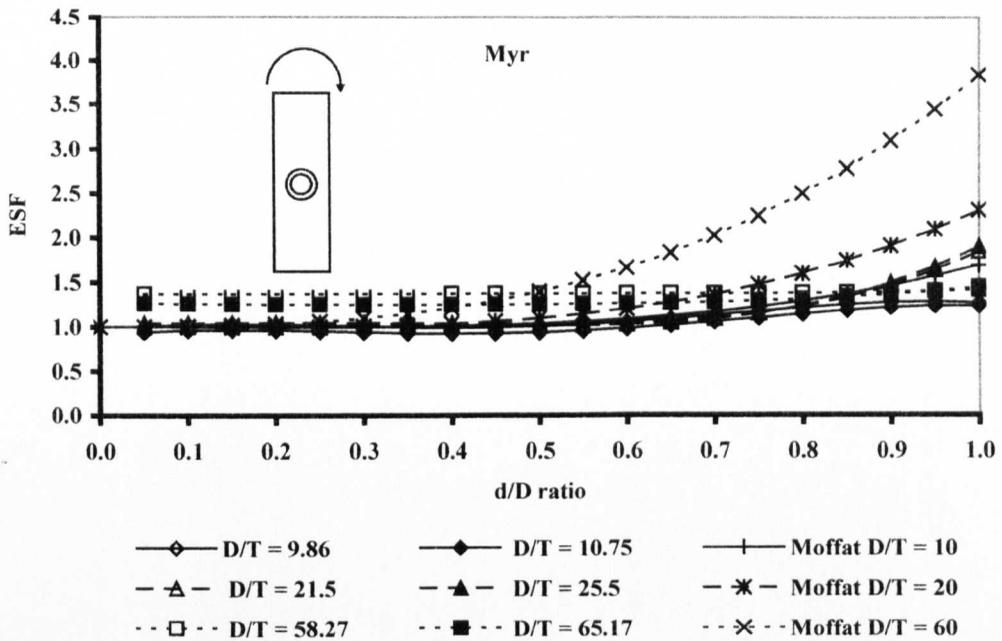


Fig. 3-18: Comparison of run pipe out-of-plane bending moment load ESFs

For the $D/T = 20$ case, the RBO demonstrates superiority from $d/D = 0.55$ and continues to $d/D = 1.0$, where the UFT ESFs are some 25% higher. Like the $D/T = 20$ case for thinner walled junctions at $D/T = 60$, the same trend is apparent – at $d/D = 0.55$ there is a 10% advantage to the reinforced design, which rockets to 172% for $d/D = 1.0$. This highlights the reinforced fittings ability to contain stresses induced by this load case for thin-walled assemblies, where the extra material reinforcement is obviously required.

Run pipe in-plane bending load, M_{zr}

Of all the load cases considered thus far, run pipe in-plane bending is by far the clearest example of a reinforced fitting's ability to contain stresses compared to un-reinforced fabricated tee. Fig. 3-19 shows that, reinforced junctions for all D/T ratios are practically uniform across the diameter ratio range at approximately 2.0. UFT however, does not exhibit such behaviour. As with the previous six load cases UFT displays a tendency to increase in ESF dramatically with increasing diameter ratio – particularly for the thinner junctions. $D/T = 60$ is clearly much worse at containing stresses than

it's reinforced cousin and whilst the $D/T = 20$ UFT junctions manage to keep ESF below 5.0 it is still over $2\frac{1}{2}$ times that of the RBO at $d/D = 1.0$.

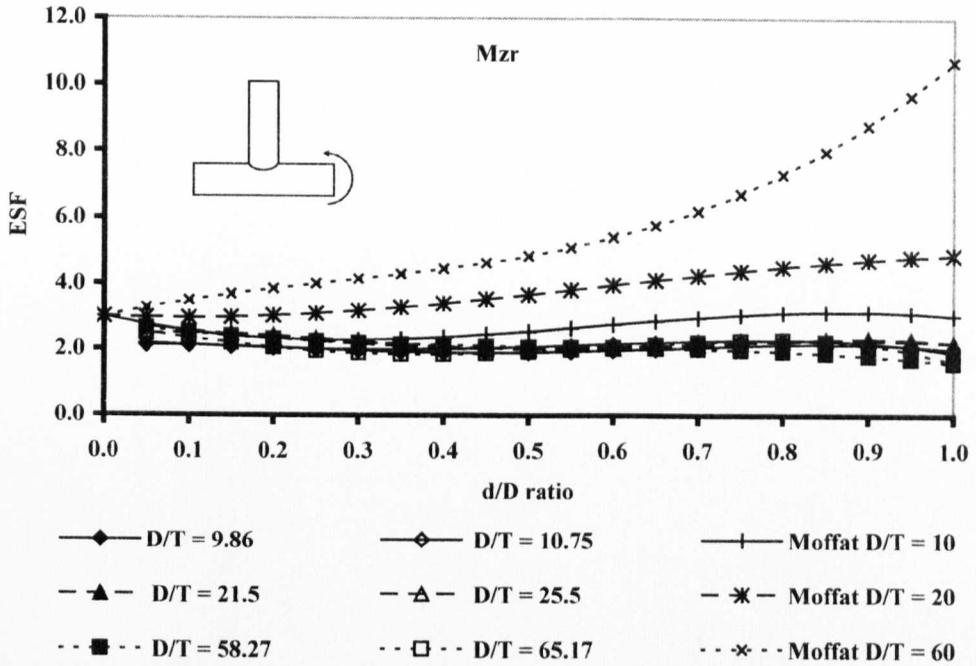


Fig. 3-19: Comparison of run pipe in-plane bending moment load ESFs

3.5 Closure

An extensive finite element analysis was carried out into 92 reinforced butt-welded pipe junctions with the aid of bespoke programs written by the author. The resulting ESF data was compared with the only other readily available source of data of a similar scale, that presented by Moffat et al²³ for a series of un-reinforced fabricated tees.

The study showed some trends that might be unwelcome if one is designing a piping structure using UFT: and that is, with the exception of pressure and run pipe out-of-plane bending moment, all load case comparisons reveal that ESF increases dramatically with increasing diameter ratio. This is particularly noticeable for thin-walled junctions. RBOs appear better able to contain stresses for diameter ratios exceeding 0.5 and proves to be a capable design when compared with UFT,

The reinforcement for the thinner junctions in the vast majority of d/D ratios is more than adequate for internal pressure load. It can be seen from the results documented in Appendix D (see also a paper by the current author included in Appendix E), that the current design ensures that ESF values for all diameter ratios and all values of D/T analysed herein remain between 1.8 and 3.5. In addition, when this design is compared with the Money, Moffat et al and Decock data for comparable D/T ratios it is clear that the reinforced branch outlet consistently outperforms UFT under internal pressure conditions. Indeed, it may be that for heavier weight junctions, the reinforcement provided under the ASME B31.3 design code is more than adequate and, perhaps, more than necessary.

The current study showed that, for branch out-of-plane bending moment loading, the extent of shell bending in the run pipe increases with decreasing wall thickness. For very thin walled junctions ($D/T < 20$) the extent of shell bending in the run pipe is so large that the stresses are greatly magnified at approximately $d/D \approx 0.7$. This phenomenon, first investigated by Wordsworth⁴³ and subsequently Schneider⁴⁴, Woods and Rodabaugh⁴⁵, was also discovered in the experimental data of Chen and Wu⁴⁹. The

fact that data from the latter closely resembles the ESF results presented here goes some way towards validating the current work.

For the remaining branch bending moment loads, M_{yb} and M_{zb} , this research has shown that in nearly all cases the reinforced fitting performs well under such conditions. This performance is enhanced when viewed in the light of the performance of the unreinforced fabricated tee ESFs when diameter ratios exceed 0.5 and, in particular when pipe walls are comparatively thin.

The results from all run moment load cases further confirmed the reinforced design's ability to deal with the stresses induced by these loads. Run pipe moment loads do not receive as much attention in open literature as pressure and branch out-of-plane bending, perhaps justifiably. However, the effects of these loads on piping systems design cannot be overlooked if overall structural integrity in any given situation is of concern.

4 INELASTIC FINITE ELEMENT ANALYSIS OF REINFORCED PIPE-WORK JUNCTIONS

4.1 Introduction

This chapter describes a study done on the inelastic analysis of reinforced branch outlets and in particular simulated pressure burst testing. Although it is often argued that analysis of this type is of academic interest only and not much use as a design tool, there is some valuable information that can be gleaned from what is largely considered to be – even today – a costly exercise.

Motivation for the inclusion of the work herein primarily stems from the desire of the sponsoring company, Spromak Ltd., to use the method in satisfying a customer's requirement for further proof of their product's performance, should the following pilot study prove successful. In addition, due to the scarcity of openly available literature on reinforced branch fittings it is hoped that this study can add to the body of knowledge on the subject.

The objective of this chapter is to further expand the pool of knowledge in this field and to examine the possibility of using this method of analysing such components alongside the more traditional approaches of experimental pressure testing and hand calculations with a view to its commercial use. Despite concerns previously mentioned, it is hoped that burst test simulation can be shown to be a useful exercise, if not for the purpose of design.

This chapter presents a detailed outline of steps taken to perform non-linear pressure burst test simulation, using the finite element method, of three 90° reinforced branch outlet junctions. The results are discussed in the light of similar work undertaken in the

pressure vessel and piping industry and results are compared to actual burst test data for those components analysed herein.

4.2 Material Data

In order to conduct a satisfactory inelastic or plastic analysis of a structure one must, at the very least, be in receipt of the appropriate material data. The finite element work is based on a full description of the material behaviour for both the reinforced outlet and the run pipe material in the elastic-plastic domain.

Since the analyses are to be compared to real burst test data obtained from an independent third party the ideal situation would have been to have either material tensile test pieces (or material data) from the original test assemblies. Unfortunately, this was not the case and data was instead obtained from material of the same grade previously used in the burst test assemblies. One must, therefore, exercise care when assessing results from these analyses, as in order to do a like-for-like analysis, one should, ideally, be using material data gathered from tests carried out on the same virgin material used in fabricating the test assemblies.

However, simulation and reality are often dissimilar in many respects and counted amongst the assumptions in this study are the potential differences between the actual material used in the experiments and that data used in these analyses.

4.2.1 Gathering Material Data

The run pipe and branch fitting materials used in the original tests were ASTM A106B and A105N respectively. The specified and actual tensile strengths, proof test pressures and results along with schematics and other relevant data for each of the test assemblies used in the original burst tests are documented in Appendix F.

As was stated above, the original material data was unavailable and in order to effectively simulate a burst test procedure one must have full description of the material

behaviour beyond the elastic limit. For this reason, a series of tensile tests were done on samples of the A106B and A105N materials, which were of the same grade and specification as the original material used in the test assemblies.

The tensile test pieces were machined from virgin material in both longitudinal and circumferential orientations. The A105N tensile test pieces were taken from three different batches. However, just one batch of the A106B pipe material was available for analysis at that time. Six individual tests were done for the A105N material and two for the A106B material.

The tests were carried out on a Lloyd LR30K tensile and compressive testing machine and the test pieces were machined, as per the Hounsfield specification and according to the sketch shown in fig. 4-1. Each had a cross sectional area of 20.02 square millimetres and a gauge length of 27 millimetres. The tests were conducted using a 30kN load cell.

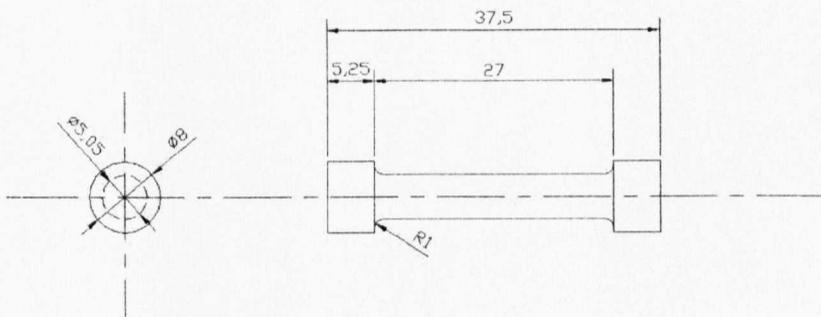


Fig. 4-1: A Sketch of the Material Test Specimen Geometry

Upon completion of the experiments, one is left with a set of charts and data in load versus extension form (see Appendix G). The data for the longitudinal and circumferential test pieces were averaged to produce a single set of load versus extension values. The mean data is subsequently converted into nominal stresses and strains, then converted once more into a suitable format for use in the ABAQUS finite element solver as described in 4.3.4.

If one has the nominal stress-strain data (such as those derived from material tensile tests) they may be converted into values of true stress and true strain using equations 4.1 and 4.2 respectively.

$$\sigma_t = \sigma_{nom} (1 + e_{nom}) \quad (4.1)$$

$$\varepsilon = \ln(1 + e_{nom}) - \frac{\sigma_t}{E} \quad (4.2)$$

The averaged post-yield load-extension values and the corresponding true stress-strain data derived from testing the A105N and A106B materials are shown below in tables 4-1 and 4-2.

Ext (mm)	Load (N)	Nom. Stress	Nom. Strain	True Stress	True Strain	Plastic Strain
0.94	6776.54	3.39E+08	0.03	3.51E+08	0.03	0.0000
1.09	6391.99	3.20E+08	0.04	3.33E+08	0.04	0.0380
1.50	7114.53	3.56E+08	0.06	3.75E+08	0.05	0.0522
2.00	8069.83	4.03E+08	0.07	4.33E+08	0.07	0.0693
3.00	9074.49	4.54E+08	0.11	5.04E+08	0.11	0.1029
4.00	9490.69	4.75E+08	0.15	5.45E+08	0.14	0.1355
5.00	9710.43	4.86E+08	0.19	5.75E+08	0.17	0.1671
6.00	9608.01	4.80E+08	0.22	5.87E+08	0.20	0.1978
7.00	8984.17	4.49E+08	0.26	5.66E+08	0.23	0.2277
8.36	5837.50	2.92E+08	0.31	3.82E+08	0.27	0.2679

Table 4-1: Stress-Strain data for A105N (RBO material)

Ext (mm)	Load (N)	Nom. Stress	Nom. Strain	True Stress	True Strain	Plastic Strain
0.95	6.47E+03	3.23E+08	0.04	3.35E+08	0.03	0.0000
1.03	6.45E+03	3.23E+08	0.04	3.35E+08	0.04	0.0358
1.50	7.68E+03	3.84E+08	0.06	4.05E+08	0.05	0.0521
2.00	8.66E+03	4.33E+08	0.07	4.65E+08	0.07	0.0692
3.00	9.72E+03	4.86E+08	0.11	5.40E+08	0.11	0.1027
4.00	1.02E+04	5.10E+08	0.15	5.86E+08	0.14	0.1353
5.00	1.04E+04	5.18E+08	0.19	6.14E+08	0.17	0.1669
6.00	1.03E+04	5.17E+08	0.22	6.32E+08	0.20	0.1976
7.00	1.01E+04	5.04E+08	0.26	6.34E+08	0.23	0.2274
8.00	8.92E+03	4.46E+08	0.30	5.78E+08	0.26	0.2567
8.84	6.42E+03	3.21E+08	0.33	4.26E+08	0.28	0.2811

Table 4-2: Stress-Strain data for A106B (run pipe material)

4.3 Finite Element Approach

4.3.1 Scope

The purpose of this study was to determine the simulated bursting pressures for three reinforced branch outlet junctions in an attempt to assess the effectiveness of the finite element method in successfully predicting burst test pressures. The following analyses consist of three finite element models each subjected to increasing internal pressure in an attempt to simulate an internal pressure burst test. The range of geometries included in this work include the following:

18 x 6 S30/S40 Flanged Buttwelded Outlet (FBO)

3 x 1 STD/STD Flanged Nipple Outlet (FNO)

16 x 1 XS/STD FNO

The cases above were analysed using true stress and plastic strain material data (see Table 4-3) gathered by experiment, described in section 4.2.1. It was assumed that the materials were isotropic and that the weld was of the same homogeneous material as the branch fitting.

ASTM A106B (RUN) MATERIAL		ASTM A105N (RBO) MATERIAL	
True Stress (MPa)	Plastic Strain (mm)	True Stress (MPa)	Plastic Strain (mm)
334.85	0.0000	350.62	0.0000
405.12	0.0521	375.49	0.0522
465.33	0.0692	433.38	0.0693
540.04	0.1027	504.14	0.1029
585.94	0.1353	544.84	0.1355
613.78	0.1669	575.43	0.1671
631.94	0.1976	587.16	0.1978
634.20	0.2274	565.67	0.2277
577.90	0.2567	470.96	0.2572
		446.93	0.2855

Table 4-3: Material Data Used in Finite Element Analyses

4.3.2 Selecting Finite Elements

The finite element models described herein were constructed from solid hexahedral type elements, commonly known as bricks, such as those in fig. 4-2. Selection of the finite elements was achieved by comparing the finite element results, based on one geometry – the 18x6 S30/S40 FBO, for both small and large displacement analyses, to the actual burst pressure obtained by experiment. The element found to converge to a pressure closest to the actual bursting pressure was subsequently used for the remaining elastic-plastic analyses.

In total, five element types were tested including the 20-node 2nd order C3D20 and C3D20R bricks; the 8-node 1st order (linear) C3D8, C3D8R bricks and the linear hybrid brick, C3D8H. In each case, mesh density was varied and each element type was evaluated with respect to final bursting pressure (compared to the actual pressure) and time taken to complete the analysis.

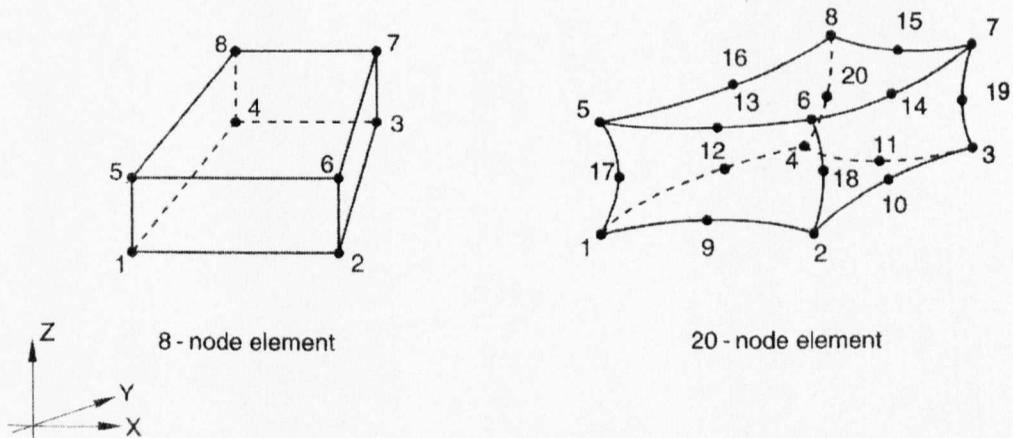


Fig. 4-2: 8 and 20-node ABAQUS finite elements

With the quadratic elements proving too expensive, in computational terms, the 8-node linear elements were eventually selected, as they offered efficiency without compromising accuracy. Additionally, when the material behaviour becomes almost incompressible, such as in elastic-perfectly plastic analysis, C3D8 elements prevent

mesh locking due to its constant volume strain characteristic throughout the element. The final mesh considered to have converged successfully is shown in fig. 4-3.

Although C3D8 elements were selected for the FE analyses done here, there is no implication or inference that this is the best or only element that may successfully be employed in analysis of this type. Indeed Plummer⁵⁰ used a reduced integration variant of the C3D8 element – a small departure from the element used here.

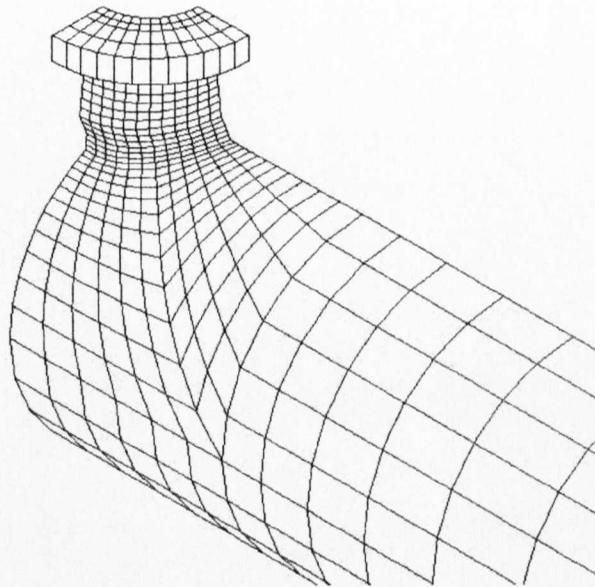


Fig. 4-3: Converged mesh model for an 18x6 FBO (Model FB1991 ($D/D=0.3$, $D/T=41.1$, $T/T=0.64$))

4.3.3 Generating the Model

Since the primary goal of the modelling phase for the elastic analysis of reinforced branch outlets had earlier been achieved, namely parameterisation of the geometry, generation of the three-dimensional finite element models was a relatively simple task.

Modification of the original MATLAB program and GUI was done to allow the user to input additional and essential material data (other than the Moduli of Elasticity and Poisson's Ratios) and the magnitude of the load (see fig. 4-4). The material data is input

as load-extension value pairs, since it was envisaged that one would have such data to hand following tensile testing of the materials concerned. The data are subsequently converted into plastic stress-strain data by the program.

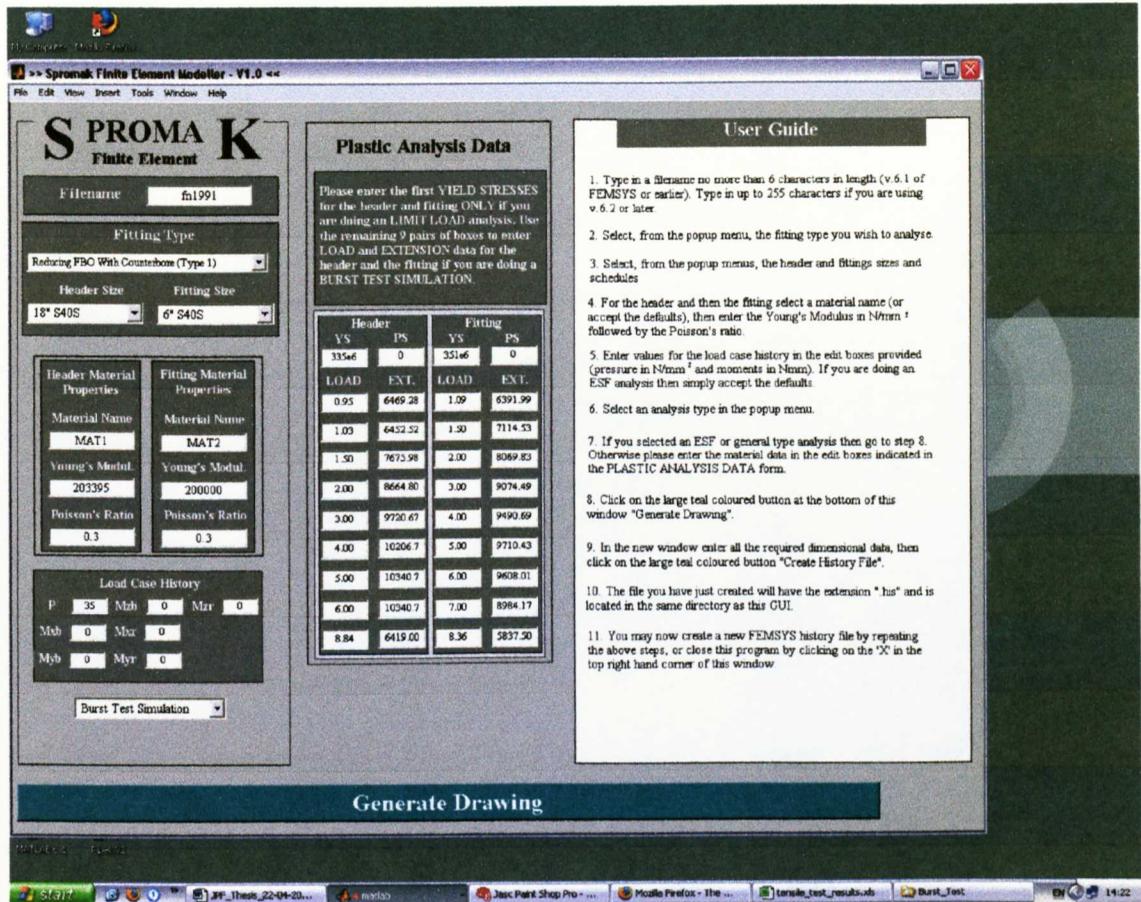


Fig. 4-4: Inelastic analysis using the MATLAB GUI

Since the load case under consideration was pressure only, a quarter finite element model was used for the analysis in each case. In order to constrain the model effectively, symmetry boundary conditions were applied to those surfaces on the run pipe axis ($z=0$ plane) and on the branch pipe axis ($x=0$ plane). One node was constrained in the y -direction to prevent rigid body movement. Each FE model had two elements through the thickness with 12 elements around the circumference of the weld.

Equilibrium pressure loads applied to the top surface of the flange and "free" end of the run pipe, simulating a closed-end system, were calculated according to eqn. 4.3.

$$\sigma = \frac{P R_i^2}{R_o^2 - R_i^2} \quad (4.3)$$

In order to ensure that the FE analysis would fail at some stage, each model had an internal pressure applied that was approximately 15% in excess of the experimentally obtained proof test pressure. The pressure was applied to the model incrementally, using a built-in algorithm within the ABAQUS FE solver software. Initially, one is required to specify the size of the first load increment, which in the cases herein was 10% of the total pressure. Upon completion of the initial increment, ABAQUS automatically calculates the size of each, subsequent, load increase thereafter. Within each of the load increments, ABAQUS tries to find a converged solution, each attempt is known as an iteration. If a solution is not found within 16 iterations, the current increment is stopped and the load increment size is reduced by 25% of its previous value and the program attempts to find a converged solution. This is known as a “cutback” and ABAQUS allows a total of five cutbacks per increment before abandoning the analysis. The following lines are taken from a sample ABAQUS input deck:

```
*STEP, NLGEOM, INC=50
*STATIC
0.1, 1.0
```

The above code tells the solver to start a new load step on a large displacement analysis with a maximum of 50 increments and an initial step of 10% of the total load value. The final piece of data above tells ABAQUS that the total time for the analysis is 1.0; which has no physical meaning. In this instance, the proportion of the applied load is equal to the current step time, so at a load step time of 0.25, 25% of the total load is applied. Thus, in order to calculate the pressure at which the component fails one simply multiplies the time at the last successful increment by the total load applied. All of the FE final burst pressures in this study were calculated in this way.

4.3.4 Material Plasticity in ABAQUS

For isotropic yielding the von Mises yield surface with associated plastic flow is used within ABAQUS. The values of the uniaxial yield stress together with the equivalent plastic strain are tabulated in the *PLASTIC option, an example of which is shown below:

```
*PLASTIC
 334.8500, 0.0000000
 405.1200, 0.0521000
 465.3300, 0.0692000
 540.0400, 0.1027000
 585.9400, 0.1353000
 613.7800, 0.1669000
 631.9400, 0.1976000
 634.2000, 0.2274000
 577.9000, 0.2567000
```

In the above snippet, the *PLASTIC option is followed by two columns of data. The left-hand column contains the true stresses and the right-hand the plastic strains. Note that the first value in column one represents the yield stress and has an associated plastic strain value that is always zero.

ABAQUS approximates the smooth stress-strain curve by interpolating linearly between data points. Since there is no limit on the number of data points one can include under the *PLASTIC option, it is possible to obtain a good approximation of the true material behaviour.

4.3.5 Geometric Plasticity in ABAQUS

In certain circumstances, where very large displacements are anticipated, it may not be sufficient to simply model the material behaviour and geometric non-linearity must also be accounted for. In such cases ABAQUS allows a user to specify either a “small” or “large” displacement analysis, by omission or inclusion respectively of the NLGEOM argument in the *STEP card in the input deck. The two types of analysis differ in the respect that a large displacement analysis considers the effect of the changing geometry

of the structure with deformation, allowing elements to become distorted; a small displacement analysis does not.

A previous study of the simulated burst testing of reinforced branch outlets by Plummer⁵⁰ was undertaken using small displacement analysis. However, for this type of analysis, one would expect a large deformation of the structure, thus a large displacement analysis is considered necessary in order to accurately reflect the real situation, as acknowledged by the authors of recent, reviewed literature^{31 32}.

Due to the range of geometries to be investigated being limited to just three, however, both small and large displacement analyses were done for each model to determine the implications of running one type over the other. These results were then compared to both the actual bursting pressure recorded in the independent experimental tests and the computed proof test pressure in each case.

4.3.6 Small Vs Large Displacement Analysis

Fig. 4-5 shows the results for model FB1991 ($D/D=0.3$, $D/T=41.1$, $T/T=0.64$), for both small and large displacement analysis using the experimentally obtained material data. SD and LD in the legend indicate the analysis type (small or large displacement).

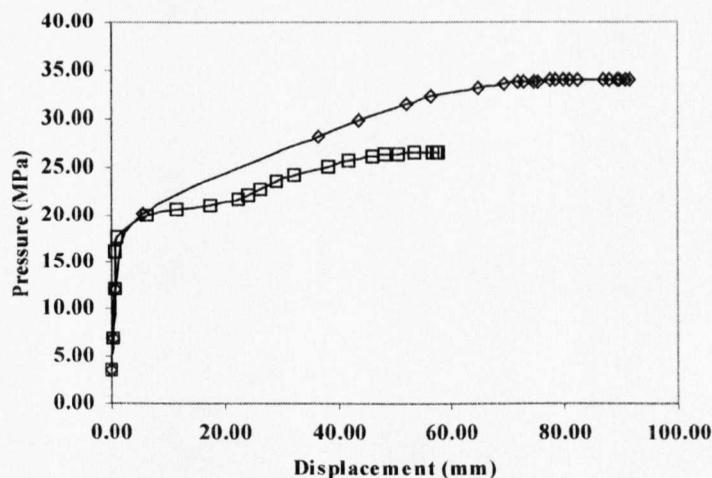


Fig. 4-5: A comparison of small and large displacement analysis methods

As can be appreciated from fig. 4-5, the eventual pressure at failure achieved using a large displacement analysis is significantly lower than for its small displacement counterpart.

The actual burst pressure recorded for this case was 30.06MPa, which compares to predicted burst pressure of 33.95MPa and 26.60MPa for small and large displacement analyses respectively. Obviously, it would be better to underestimate, rather than overestimate final bursting pressures, since the latter could be potentially dangerous.

It is argued, that an analysis such as the type considered in this chapter is of little use, since one can obtain a reasonable estimate of the pressure at burst with a simple hand calculation (taken from ANSI B16.9 – 1986) as per eqn. 4.4:

$$P_c = \frac{2T\sigma}{D} \quad (4.4)$$

Where D and T are the inside diameter and thickness of the run pipe respectively, σ is the specified run pipe tensile strength and P_c is the computed proof test pressure.

If one has the actual tensile strength for the fitting material, σ_{act} , an *adjusted proof test pressure* is calculated according to eqn. 4.5:

$$P_a = P_c \frac{\sigma_{act}}{\sigma} \quad (4.5)$$

ANSI B16.9 requires that the minimum acceptable internal pressures used to test the fitting are:

- Assembly must withstand 110% of the computed pressure, P_c
- Pressure prior to rupture must be at least equal to the adjusted proof test pressure, P_a

Or:

- If the pipe ruptures or if sufficient pressure to rupture any part of the assembly cannot be attained, the test pressure is acceptable if a final test pressure is at least 105% of the adjusted proof test pressure.

The reader is referred to Appendix F for full details of the material data and calculated and adjusted proof test pressures used in the actual pressure proof tests.

In each case, the actual tensile strength data were available. For case FB1991 ($d/D=0.3$, $D/T=41.1$, $t/T=0.64$), the calculated and adjusted proof pressures are 20.13MPa and 24.62MPa respectively. Recall the FE results of 33.95MPa and 26.60MPa for small and large displacements and the experimentally obtained final bursting pressure 30.06MPa.

Note that, both small and large displacement predictions are higher than both the computed and adjusted pressures calculated using eqns. 4.4 and 4.5 respectively. In particular, the small displacement analysis result is above both calculated pressures and the actual burst pressure of the vessel, which is why small displacement analysis should not be employed for a pressure vessel proof or burst test simulation. Thus, the current author concurs with Carino³¹ and Sanal³², that large displacement analysis is necessary for the conservative (if not accurate) solution of this type of problem.

4.4 Discussion of Results

The following results are solely the product of finite element analysis of three reinforced butt-welded 90° branch outlets and are compared with data obtained from actual burst test experiments performed in accordance with the ANSI B16.9 standard. Table 4-4 presents the results obtained for three flanged butt-weld geometries and six finite element models were completed in total.

The FE results from all three models demonstrate a similar trend in that only the large displacement final pressure value was below the experimental burst pressure in each

case. Therefore, a small displacement analysis using the material data obtained in this study is wholly unsuitable.

	18x6 S30/S40S [FB1991]	3x1 STD/STD FNO [FN0254]	16x1 XS/STD FNO [FN0723]
Run pipe inside diam., D (mm)	457.20	88.90	406.40
Run pipe wall thickness, T (mm)	11.13	5.49	12.70
Calc. Proof pressure, P_c (MPa)	20.13	51.06	25.85
Adj. Proof pressure, P_a (MPa)	24.62	60.24	30.50
Actual burst pressure P (MPa)	30.06	86.18	40.75
SD (MPa)	33.95	87.50	43.85
LD (MPa)	26.60	69.70	34.12

Table 4-4: Burst test simulation results

The above results reaffirm Boyle and Spence's²⁷ hypothesis that inelastic analyses can be expensive and are only as good as the material model. Indeed, even when one has access to samples and accurate tensile test data the material model will, in all likelihood, be incomplete. Assuming the material experimental tests were done correctly and that there were no errors in transcribing and converting the load-extension data obtained, one could argue that the differences (between the actual and simulated FE burst pressures) are largely a function of the material properties.

No two batches of the same material are likely to have the same characteristics, qualities and properties after being subjected to a forming process. Having an inaccurate material model is compounded by the assembly process, when the branch fitting and run pipe will have been welded together altering the material properties further in the region of the heat affected zone; these changes are largely unknown. The difficulty associated with doing an accurate analysis, in terms of modelling the material, suggests that using the FE method (in full non-linear mode) as a tool for increasing pressure limits for vessels of this type is ill-advised. Therefore, if one undertakes such an analysis, it must be accepted that the solution is probably not going to yield highly accurate results.

For these analyses, material samples were taken in the longitudinal and circumferential directions and the resultant data subsequently averaged. The author assumed isotropic material behaviour in both the run pipe and branch outlet fitting material. Depending on

the forming process used for the run pipe (which is where failure occurs) it is likely that the material is stronger in one direction than the other. In order to account for this particular property of the run pipe material, an anisotropic analysis could have been done and is, perhaps, a possible option for future study in this area.

The purpose of this pilot study was to investigate the potential use of inelastic FE analysis as a tool for estimating the bursting pressures of pipe junctions with reinforced branch outlets, satisfying the commercial goal of proving fitness for purpose without extensive physical testing. Assuming an engineer does not have easy access to the specific material data pertaining to a given case, and taking the above results into consideration, the very best outcome one could hope for, is a predicted bursting pressure similar to that which would be obtained using eqn. 4.4.

Figs. 4-6 and 4-7 show plots taken from model FB1991 ($d/D=0.3$, $D/T=41.1$, $t/T=0.64$) for a large displacement analysis.

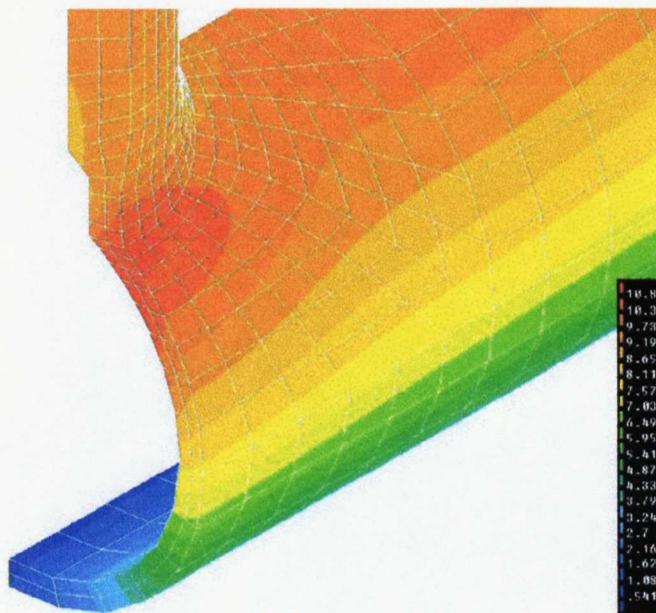


Fig. 4-6: Displacement contour plot model FB1991 ($d/D=0.3$, $D/T=41.1$, $t/T=0.64$), large displacement analysis

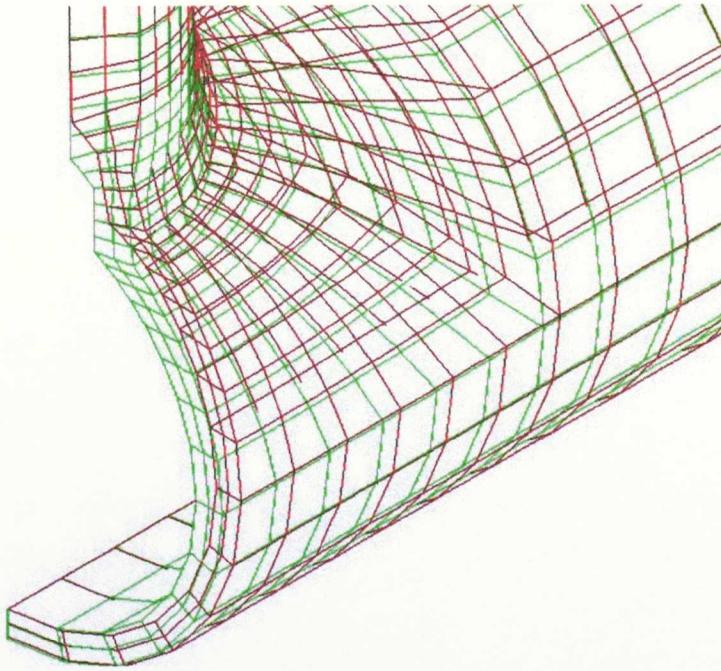


Fig. 4-7: Displaced shape plot model FB1991 ($d/D=0.3$, $D/T=41.1$, $t/T=0.64$), large displacement analysis

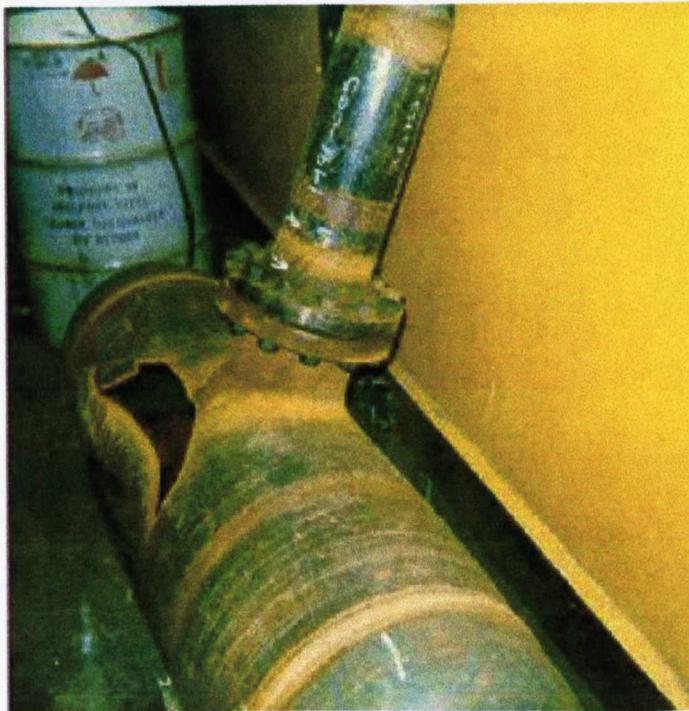


Fig. 4-8: Experimental proof testing of a reinforced branch outlet

Fig. 4-6 is a displacement contour plot taken at the increment prior to bursting. It shows the maximum displacement at a point in the flank of the junction, just below the weld toe. Fig. 4-7 shows the displaced shape of the vessel at the same increment. Both plots suggest that the rupture path would lie in the flank of the junction and appear to show some agreement with that of the experimental test shown in fig. 4-8.

Carino et al³¹, also noted that good correlation was found between the experimental and FE results in respect of the rupture path of the structure:

“the isostrain curves suggest a rupture path that was showed to be in agreement with the experimental test proofs performed in recent years...”

Carino et al were able to predict, with sufficient accuracy, plastic strain trends in the critical flank area of the model assembly. They went on to argue that the method provided a *“powerful tool in view of anticipating proof testing results”* and *“a useful guideline for design purposes in view of a more rational material usage and joint configuration”*.

Despite Carino's³¹ enthusiastic report of his study in this field and the encouraging results from the large displacement analyses, the fact remains that a simple hand calculation will produce a reasonable estimate of the proof bursting pressure without the need for lengthy FE modelling and analysis. In addition, the failure point on the vessel should not really be surprising, since the discontinuity at the toe of the weld where it meets the run pipe will always serve as a stress raiser.

In any event, the purpose of pressure vessel design is to ensure that bursting is improbable, if not impossible, therefore inelastic analyses of this kind are unlikely to prove useful in design or for satisfying a customer's requirement for pressure proof testing.

4.5 Closure

This study set out to investigate the potential use of inelastic FE analysis as a tool for estimating the bursting pressure of reinforced branch outlet assemblies to satisfy customer requirement for proof of a products performance. In addition, it was hoped that the work would further expand the body of knowledge on the subject.

The experimental testing procedure for the run pipe and branch outlet materials was described and an account was given with respect to inclusion of such data into a finite element model for inelastic analysis. Assumptions included: the weld material was the same as the branch outlet and that all materials were isotropic and homogeneous. Additionally, the material tested was not from the same batch as that used in the original pipe and reinforced outlets. A brief description of how the existing FE pre-processing programs were adapted, to allow automated generation of FE models within the FEMGV environment, was given.

The differences between small and large displacement analysis were discussed before presenting the FE results alongside results obtained from hand calculations in table 4-4. The FE results obtained tend to reaffirm Boyle and Spence's²⁷ hypothesis that inelastic analysis can be expensive and is only as good as the material model.

A complete and comprehensive material model, required in order to accurately reproduce a pressure burst test, is unlikely to be available to an engineer performing a finite element analysis. This is largely due to the unknown properties of the materials after undergoing the forming and welding processes. In addition, insufficient information is available regarding the extent of any anisotropy present in the run pipe and branch outlet materials.

Results obtained from the small displacement analyses proved that such a method is unsuitable for estimating the bursting pressure of a pipe junction. The difference between the large displacement FE results and corresponding hand calculations were, it is felt, not significant enough to warrant the cost of a finite element analysis, especially

since the results obtained are in no way surprising or offer further insight from a design or stress analysis perspective in respect of pressure vessel behaviour under such conditions.

5 AN INTRODUCTION TO NEURAL NETWORKS

5.1 Introduction

This chapter's primary goal is to briefly introduce artificial neural networks, their underlying principles and inner workings with a view to using the technique in solution of the problem posed in chapter 6: to evaluate stresses in un-reinforced fabricated tees based on finite element analyses and interpolate between them.

An overview of the human neuron is presented in an effort to elucidate the reader on the theoretical background to the artificial neurons used in this research. This is followed by a short review of artificial neural networks with reference to historic landmarks in this relatively new field of science.

Artificial neural networks are subsequently discussed in more detail. In particular, the feed-forward multi-layer perceptron with back-propagation learning algorithm is singled out as a likely candidate for the problem described above. Finally, an attempt is made to justify selection of the neural network route in solving the current problem before concluding.

5.2 Real Neural Networks

Biological neurons that exist in the brains of humans and animals are extremely complex and not at all homogeneous. That is to say that the structures and behaviours of such cellular matter are considerably different from one neuron to the next. It would be both impossible and inappropriate therefore to describe each "variety", thus the discussion herein is of a stereotypical neuron model. It is hoped that what follows enables the reader to gain a fundamental, albeit basic, understanding of real neural networks and their workings.

Naturally occurring neural networks, such as those within the human brain are, perhaps, the most complex systems known. The human brain has approximately 10^{11} individual neurons, each of which forms part of a larger local structure that processes specific information. Although neural operations are relatively slow when compared with digital computers - 10^{-3} s versus 10^{-9} s respectively - the massively interconnected state of a biological neuron and sheer number of neurons ensures that the net processing power of this structure is far more efficient and agile than any man-made device.

Fig. 5-1 illustrates a biological neuron, as understood today. Neurons communicate with one another by transmitting electrical signals (known as spikes or action potentials) along a long, thin fibre known as an axon. An axon is characterized with few branches and it carries the action potential from the neuron to the axon terminals or collaterals.

Upon reaching the axon terminals, the action potential mobilizes a number of molecules called vesicles containing a substance known as a neurotransmitter. Thus, the migrating vesicles effectively convert the action potential from an electrical to a chemical signal. At the pre-synaptic membrane, the neurotransmitter is released across the synaptic cleft and is chemically bound to receptor sites on the corresponding postsynaptic membrane. An electrochemical reaction takes place, the result of which is to polarize the postsynaptic membrane on a local scale.

The postsynaptic membrane has a negative charge when in a state of equilibrium. The postsynaptic potential (PSP) generated at the postsynaptic membrane has the effect of either raising or further lowering the PSP and are thus known as excitatory PSPs (EPSP) or inhibitory PSPs (IPSP) respectively.

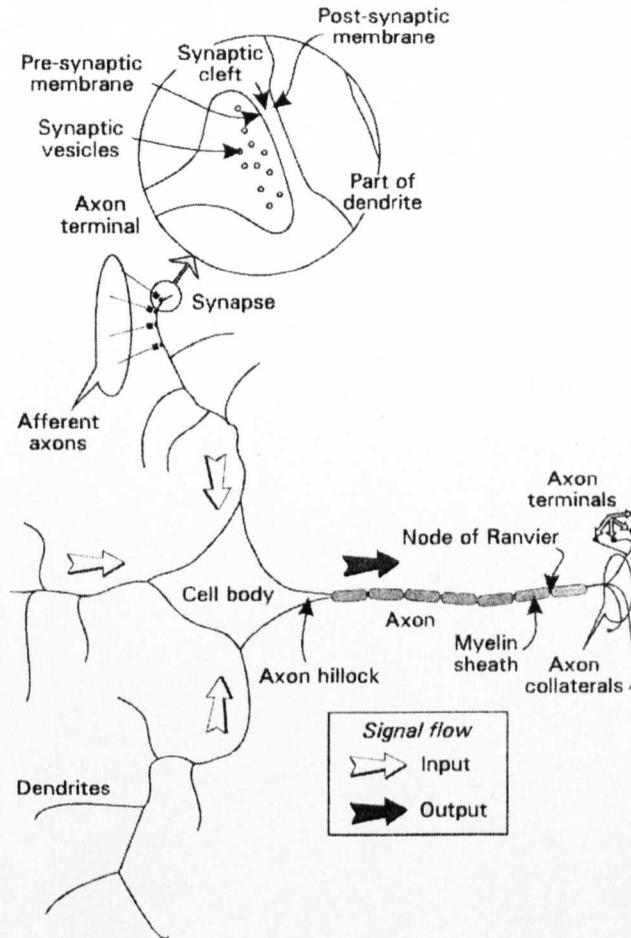


Fig. 5-1: A Biological Neuron. After Gurney⁵¹

The EPSPs and IPSPs from receptor sites at many thousands of axon terminals make their way along fibres known as dendrites to the cell body and are, therefore, considered inputs to the neuron. PSPs arriving at the axon hillock are integrated, the result of which is to affect the membrane potential of the cell. If the summed PSPs exceed some threshold, an action potential is generated and this output signal is then transmitted to other neurons via the axon and associated terminals as before. If the potential threshold is not exceeded there is no net output from that neuron.

So far, the processes that run continuously in the human brain have been discussed, although an explanation of how the brain actually learns has not yet been forthcoming. This is now redressed. The process of learning in humans is a function of experience and, since humans experience things continually during their lifetime, the brain

(neurons) must be able to adapt or develop. The ability to learn from or change according to our environment and experiences is known as “plasticity”⁵¹.

How does experience translate to learning, and thus knowledge, through the mechanism of plasticity? Most have heard the adage “[doing something] is like riding a bike”. This statement suggests that, perhaps, riding a bike is easy, but the underlying implication is that, once learned riding a bike becomes second nature and is never forgotten. Such experiences are said to be “hard-coded” within the brain and there are many examples of such instances. However, not all experiences get hard-coded and human beings have the ability to forget.

Generally, if a certain task is learned, for example, how to integrate a mathematical function, it is easily forgotten unless the procedure is repeated and frequently used. Thus, the act of learning and of repetition is thought to stimulate dendritic growth between neurons, strengthening both neural connectivity and, ultimately, knowledge.

The field of neuroscience is an extensive one and further relevant texts for engineers with brief introductions to the biological basis for artificial neurons are those of Gurney⁵¹, Haykin⁵² and, to a lesser extent, Tsoukalas et al⁵³.

5.3 Artificial Neural Networks

5.3.1 A Brief Historical Overview of Artificial Neural Networks

This section represents a very brief summary of the most significant events in the field of neural networks directly relating to the content of chapters five and six of this thesis. It is not a full or complete bibliographical reference spanning the fields of neuroscience and neural networks and, should such background material be required, the reader is referred to Haykin⁵², chapter 1.9.

Artificial neurons and artificial neural networks (ANNs) are of course based on their biological counterparts. However, as one might expect, ANNs represent a highly simplified model of the workings of the human brain. The first artificial neuron developed by McCulloch & Pitts is known as the Threshold Logic Unit or TLU. Based on human physiology, the TLU model was a quantum leap forward in their fields of research and culminated in their seminal paper of 1943⁵⁴. Their combined expertise in the fields of neurophysiology and theoretical mathematics opened up new frontiers in computation now known as neural networks and artificial intelligence.

In 1949 Hebb⁵⁵ published his book describing a physiological rule for synaptic modification, that was only picked up by the engineering community in 1956, when Rochester et al⁵⁶ tested a neural theory, based on Hebb's "postulate of learning" using a computer simulation. Also in 1956, Uttley's work⁵⁷ on neural networks with modifiable synapses proved that classification of simple binary sets was possible.

By 1958, Rosenblatt⁵⁸ described a new method for supervised learning in neural networks using his perceptron rule. Shortly afterwards, in 1960, Widrow and Hoff⁵⁹, used the delta learning rule for training neurons known as ADALINEs and subsequently were able to train one of the first multiple layer networks, known as a MADALINE.

In a significant breakthrough, some 26 years after its inception, the delta rule was generalised and the resultant algorithm popularised after publication of a paper by Rumelhart et al⁶⁰ in 1986. The back-propagation algorithm subsequently became the most popular method for training supervised multi-layer neural networks.

Lippmann⁶¹ in 1987 conducted a thorough review of the state of computing with neural networks at that time. He discussed six different network topologies, one of which was the feed-forward back-propagation network. Since back-propagation was in its infancy (as were many other algorithms) the author cited several limitations on the practicality of the algorithm that were largely overcome by the time Hush and Horn⁶² published their article in 1993. Hush and Horn reviewed the developments since 1987 and up to 1993 in neural networks with respect to those theorems that were most immature during

the period of Lippmann's writing – including the multi-layer feed-forward network with back-propagation.

In 1988, Irie and Miyake⁶³ demonstrated that a two-layer feed-forward network (one hidden-layer) could approximate any arbitrary function, given definitive input bounds. Hornik later confirmed this in 1989⁶⁴.

Throughout the late eighties and nineties, researchers continued to attempt to refine the feed-forward back-propagation network and, generally, the focus has been on network performance and learning rates. The original delta rule, introduced by Widrow and Hoff⁵⁹, has been modified countless times in a search for ever-faster algorithms.

One algorithm of particular note is that of Marquardt, which was first presented in 1963⁶⁵. This algorithm is noteworthy due to its relative efficacy and accuracy in computing function approximation problems when used within back-propagation networks having less than a few hundred weights.

Hagan and Menhaj⁶⁶ published their findings for the Marquardt algorithm used in multi-layer feed-forward networks for four different function approximation problems. They found that the algorithm successfully converged to an optimal solution when other algorithms, such as the conjugate gradient and variable learning rate, did not. In addition, the rate of convergence was orders of magnitude greater than the other algorithms tested.

The field of neural networks continues to enjoy a resurgence and attention from researchers having disparate spheres of expertise. Such is the flexibility and power of the method; it is difficult to imagine their fall from grace, back into the obscurity that was prevalent during the 1970s.

5.3.2 The Artificial Neuron

Artificial neural networks usually consist of a number of simple processing nodes that are connected in such a way as to map input data to a desired output. The elementary units of such a network are the inputs, weights, neurons (nodes), and outputs. The input data, sometimes supplied to the network in a normalized form, are usually series of real numbers and/or integers and relate to specific variables identified as parameters of the current problem to be solved.

Inputs are connected to either hidden layer nodes or output nodes via weights. These weights convey the output signals of their inputs to nodes in the network. Depending on the nature (mathematical sign) of the weight, the signal from the associated input is either excitatory (positive weights) or inhibitory (negative weights). The relative strength of a weight is measured by the magnitude of its absolute value.

Neurons are the processing units in the network and their function is (commonly) to integrate the inputs and weights using a prescribed function and then calculate an output signal according to an activation rule. The purpose of the integrator is to sum the products of the inputs (denoted by the letter p) and weights (W). To this resultant, a bias (b) is added, which is analogous to an input of unit value. The net combined input of the neuron is passed to an activation (transfer) function, f . This is usually a sigmoid, linear or hard limit (step) function.

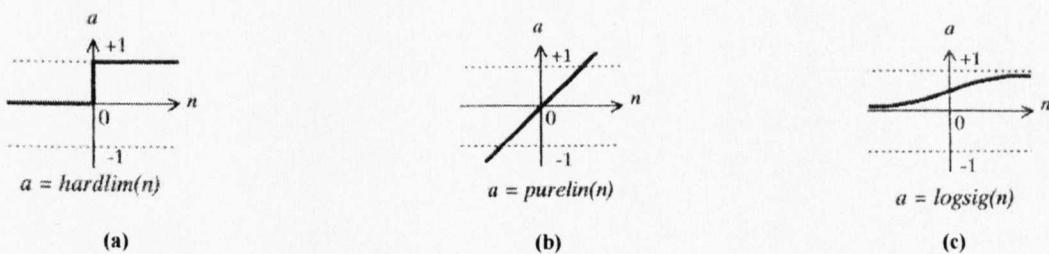


Fig. 5-2: Artificial Neuron Activation Functions (a): Hard Limit (b): Linear (c): Sigmoid. After Demuth et al.⁶⁷

A hard limit activation function is restricted to Boolean operations and, as such, can only output (denoted by “a”) an activation of zero or one. Linear transfer functions cannot perform non-linear computations, but often find use in the output layer of networks designed to solve non-linear problems, since their activation value is unlimited. Sigmoid functions are commonly used in neural networks required to solve non-linear problems of every kind and address the limitations of both the hard limit and linear activation functions and typically pass output values between 0 and +1 or -1 and +1.

Consider an artificial neuron such as that represented schematically in fig. 5-3.

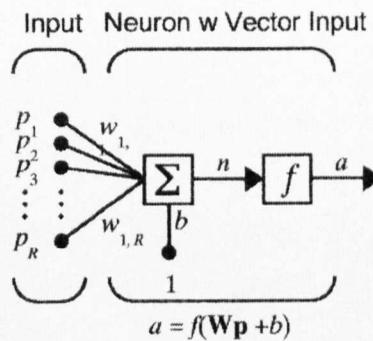


Fig. 5-3: A Schematic Model of an Artificial Neuron. After Demuth et al.⁶⁷

The inputs (p_R), on the left-hand side of the diagram, are multiplied by the weights ($W_{1,R}$) and summed:

$$n = \sum_{n=1}^R (p_n W_n) \quad (5.1)$$

After summing the weighted inputs a bias, b , of constant unit value is applied to the total. The total input to the transfer function is thus:

$$n = Wp + b \quad (5.2)$$

Eqn. 5.2 can also be expressed as per 5.3 if the bias is considered a weight attached to a node having unit activation:

$$n = \sum_{n=0}^R (p_n W_n) \quad (5.3)$$

The final output, a , is the sum of the weighted inputs, plus the bias applied to a prescribed transfer function, f . Thus, the output expressed in terms of the inputs, weights, bias and transfer function is:

$$a = f(W_p + b) \quad (5.4)$$

5.3.3 Multi-layer Feed-forward Neural Networks

Neural networks come in many different configurations, the most ubiquitous and popular of which is, perhaps, the feed-forward network. Firstly, a description of a feed-forward network is presented along with some of the associated terminology.

A single neuron is not of much use and, indeed, this is not analogous to the massively parallel nature of billions of interconnected biological neurons, previously described. A much more likely network would consist of many neurons in a layer, such as that shown in fig. 5-4.

Calculating the number of neurons required to solve a given problem in a multi-layer feed-forward network is impractical if not impossible. It is common to see networks that have relatively few inputs connected to many neurons. The more neurons available in a given network, the more powerful it is. However, having vast numbers of neurons in a network is not always a guarantee of successfully solving a problem, as shall be seen.

Masters⁶⁸ advises that the only sure way of determining the optimal number of neurons is by trial and error. The network architect must start training a network with a number of neurons that he feels sure will not be able to solve the problem. Once the network has

been “trained” and the performance noted, the architect must then increase the number of neurons by a small amount and repeat this process until the error is, either, acceptably low or no significant improvement on performance is seen.

The structure of a feed-forward neural network is often considered in terms of its layers. For example, consider fig. 5-4. On the left-hand side of the diagram there are a number of inputs that are considered to be fully connected, i.e. every node in the previous layer is connected to each of the nodes in the next layer.

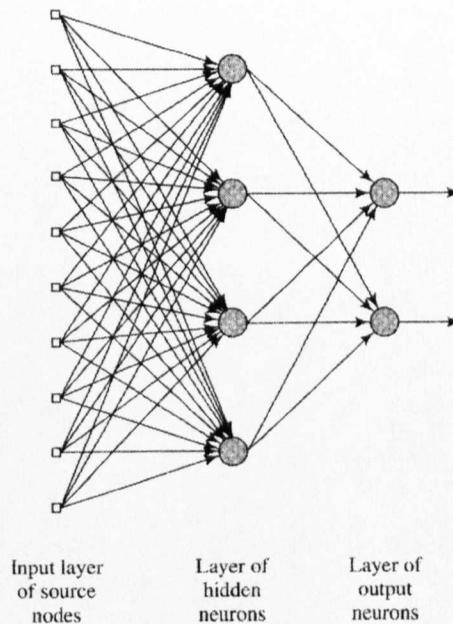


Fig. 5-4: A Neural Network with One Hidden Layer of Neurons. After Haykin⁵²

The inputs (generally not considered to be a layer as such, since no computation takes place) feed into neurons structured in a so-called “hidden layer”. The term “hidden layer” suggests that nodes within this layer do not receive direct inputs from or transmit outputs to the environment. The “output layer” is that which takes its inputs from a preceding hidden layer of neurons and outputs its result to the environment. The notion of information propagating through layers from the inputs through to the outputs (strictly in this direction only) gives this structure the name “feed-forward network”.

The number of hidden layers is not limited to one and it is possible to have multiple hidden layers of neurons. However, it has been proven^{60, 61} that, if the inputs to the function one is trying to approximate have definitive bounds, a network with a single hidden-layer can successfully learn it. Since the inputs to the problem discussed in chapter 6 are considered to qualify as having definite bounds, the multiple hidden layer network model is not pursued here.

5.3.4 Learning Rules

There are a number of learning rules available to the neural architect that may be formulated such that they constitute a learning algorithm. Learning algorithms differ from each other by the way in which the weights in a network are adjusted. The rule of particular interest herein is the error-correction rule which, in straightforward terms, aims to minimize the error of outputs generated by the network with respect to given targets by adjustment of the synaptic weights.

The error correction rule infers that a network is “taught” or “supervised”, since the network is provided with a set of input/output patterns and it is hoped that the network will eventually successfully replicate the given patterns – within a specified tolerance. Once training has been successfully completed, the network is left to deduce outputs from previously unseen inputs within the problem domain. This training paradigm is known as supervised learning and is used extensively with feed-forward networks.

Under the supervised learning regime, when inputs are processed via their associated weights and attached neurons, the output is compared directly with a target output. The difference between the two is known as the error and it is desirable to minimize this quantity as much as is practical.

Gradient Descent

Consider the function $y(x)$, plotted on a graph as shown in fig. 5-5.

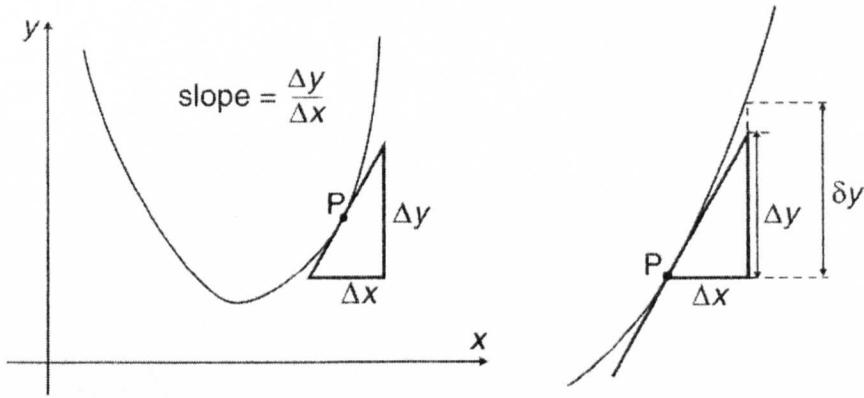


Fig. 5-5: Finding the Minimum of a Function. After Gurney⁵¹

If the changes in y due to x are considered small $\delta_y \approx \Delta_y$, it can be shown that the gradient of the slope $\frac{\Delta y}{\Delta x}$ can be evaluated in terms of x :

$$\Delta x = -\alpha \left(\frac{dy}{dx} \right)^2 \quad (5.5)$$

Where the learning rate, $-\alpha$ is greater than zero but small enough that $\delta_y \approx \Delta_y$ still holds true.

If $-\alpha$ is a small non-zero value, it is assured that the direction of descent is always towards the minimum of the function. In addition, since the direction of descent is beyond doubt, convergence to a minimum (at least locally) is a virtual certainty. The technique of iteratively calculating the slope of a function towards a function minimum is known as *gradient descent*.

Complex functions are, however, more likely to have multiple variables and one would write:

$$y = y(x_1, x_2 \dots x_n) \quad (5.6)$$

Eqn. 5.5 can be expressed in terms of multiple variables by using partial differentiation. Using this method the gradient of each variable is calculated and, instead of eqn. 5.5, one may now write the following equation for each of the variables, x_i :

$$\Delta x_i = -\alpha \frac{\partial y}{\partial x_i} \quad (5.7)$$

The error on output of a given neuron is a function of the weights: $E = E(w_1, w_2, \dots, w_i)$. If the function has more than one variable (more than 1 weight), one can apply eqn. 5.7 to express the error in terms of the weights:

$$\Delta w_i = -\alpha \frac{\partial E}{\partial w_i} \quad (5.8)$$

As was mentioned earlier in 5.3.4, the error in a network can be expressed in terms of the difference between a target and output value pair under the supervised learning paradigm. The original artificial neuron model developed by McCulloch and Pitts⁵⁴, the threshold logic unit (TLU), and Rosenblatt's⁵⁸ perceptron use the linear step activation function of fig. 5-2(a). The step function, however, has been shown to be unsuitable for non-linear function minimization due to its Boolean nature. Therefore, in this case, the error can only be expressed in terms of a rather carefully selected target vector and a corresponding activation vector:

$$E = \frac{1}{N} \sum_{p=1}^N \frac{1}{2} (t^p - a^p)^2 \quad (5.9)$$

Where the superscript, p , indicates the current input pattern, t , is the target vector and, a , is the activation output.

The Delta Rule

The Delta rule allows incremental updating of the weights, thus the error gradient is estimated sequentially in a least mean square sense and is not the “true” error gradient as derived by batch mode training using the gradient descent method. The Delta rule, expressed in terms of the target and activation output, is:

$$\Delta w_i = \alpha (t^p - a^p) x_i^p \quad (5.10)$$

In order to do non-linear function approximation a “smooth” function such as the sigmoid of fig. 5-2(c) is required. This activation function is differentiable and can be incorporated into the Delta rule:

$$\Delta w_i = \alpha \sigma'(a) (t^p - y^p) x_i^p \quad (5.11)$$

The sigma term in eqn. 5.11, denoted by the $\sigma'(a)$ symbol, is the differential of the slope of the sigmoid and directly affects the magnitude of the error and thus the rate of change of the weights.

In the case of multiple nodes in a layer, eqn. 5-11 can be altered to include the node index as the error is summed across all nodes. First, one must calculate the square of the summed error across all nodes for pattern, p:

$$e^p = \sum_{j=1}^M (t_j^p - y_j^p)^2 \quad (5.12)$$

Where j is the node index. The weight of the jth node may now be calculated:

$$\Delta w_{ij} = \alpha \sigma'(a_j) (t_j^p - y_j^p) x_{ji}^p = \alpha \sigma'(a_j) \delta \cdot x_{ji}^p \quad (5.13)$$

The delta rule, with a differentiable activation function, represented a significant step in the search to find an algorithm that could be generalized to allow multiple layered feed-forward networks to be successfully trained.

5.3.5 Multi-layer Feed-forward Neural Networks with Back-Propagation

The back-propagation algorithms substantial advantage over, say the Boolean perceptron or the TLU, is that it does not require prior knowledge of the pattern space, thus one is not required to input more information than is absolutely necessary for direct solution of the problem.

The groundwork for adjusting the weights for multiple layers of nodes was laid in the previous section. Eqns. 5.12, 5.13 are used to calculate the weights of the output layer nodes for sequential mode training, where the weights are adjusted presentation of each pattern, as per the delta rule. The next problem is: how might the weights in the hidden layer be adjusted? The answer to this question lies, once more, in being able to generalize a quantity.

The solution to the problem of generalizing eqn. 5.13, such that it could be applied to the adjustment of hidden layer weights was first conceived by Werbos⁶⁹ in 1974 and subsequently popularized by Rumelhart⁶⁰ in 1986. The heart of the back-propagation algorithm then, lies in the ability to rationalize the influence that a given input from a hidden layer node has on the output layer node(s) and how the output node(s) affects the error.

Recall the delta learning rule as eqn. 5.13. If one examines the equation carefully, it can be seen that the only term specific to the output node is the difference between the target and output (the delta term). Therefore, if the delta term can be re-written with respect to hidden nodes, it is possible to formulate a rule for updating the hidden layer weights. The relationship between a hidden node and an output node to which it is connected via a weight must be examined in order to determine such a rule. Gurney⁵¹ presents the final key to the back-propagation algorithm:

“The contribution that [the output node] makes towards the error is, of course, expressed in the δ for that node δ^j . The influence that [the hidden layer node has on the output node] is given by the weight w_{jk} . The required interaction between these two factors is captured by combining them multiplicatively as $\delta^j w_{jk}$.”

If the weight, k , is connected to more than one output node, j , summing the products over all j :

$$\delta^k = \sum_{j \in I_k} \delta^j w_{jk} \quad (5.14)$$

Where I_k is the set of output neurons taking input from the hidden neuron, k .

If the delta term is augmented to include the slope of the sigmoid and using eqn. 5.14, eqn. 5.13 may be rewritten in terms of any node (hidden or output), k :

$$\Delta w_{ki} = \alpha \delta^k x_{ki}^p \quad (5.15)$$

For updating the weights in the hidden layer, the delta term is:

$$\delta^k = \sigma'(a_k) \sum_{j \in I_k} \delta^j w_{jk} \quad (5.16)$$

For updating the weights in the output layer, the delta term is:

$$\delta^k = \sigma'(a_k) (t_k^p - y_k^p) \quad (5.17)$$

Now that the equations to enable updating of the weights in a multi-layer neural network have been defined, it is possible to define the structure of the algorithm.

```
Initialise weights and biases
do
  for each input vector
    1. Present the vector to the inputs
    2. Evaluate outputs of the hidden layer nodes
    3. Evaluate outputs of the output layer nodes using result from (2)
    4. Evaluate the  $\delta S$  on the outputs
    5. Apply eqns. 5.15/5.17 to the output nodes and update the weights
    6. Apply eqns. 5.15/5.16 to the hidden nodes and update the weights
  end
until the error meets a specified goal
```

Inside the “for” loop, the first three steps are considered to be the forward pass through the network. Steps 4-6 represent the backward pass, since data is flowing from the right-hand side of the network to the left.

5.3.6 Increasing Training Performance

When one trains a neural network in the manner described in the previous section, training times can be extremely long. The time taken for a network to converge on a solution depends on so many variables; some of which are easier to improve upon than others. In truth, there is a fair amount of experimentation called for when designing a network and with this experimentation comes experience. Knowledge of the problem domain and prior experience in designing networks are qualities not to be underestimated. There are a number of techniques that will improve performance of training with the back-propagation algorithm, some of which are now discussed.

Previously, sequential updating of the synaptic weights has been described in this thesis and, indeed, the original back-propagation algorithm was conceived in these terms. This method of training can be advantageous in terms of computational speed, thanks to its iterative nature and, in addition, it does not require large amounts of storage space.

Evidence exists to show that selection of the sigmoid activation function can play a role in reducing the number of training iterations to convergence. LeCun⁷⁰ recommends use of the hyperbolic tangent function instead of the logistic sigmoid.

According to Haykin⁵², selection of suitable initial weights and threshold values plays an important role in the success (or otherwise) of a network. LeCun⁷¹ proposes all initial weights should be drawn from a set of uniformly distributed numbers with a mean of zero and a variance equal to the reciprocal of the number of synaptic connections on the neuron.

Normalizing inputs and targets is another method that can be employed in reducing training times, although targets can often be left alone if one uses linear neurons in the output layer. Normalization of the inputs is done primarily to ensure that the weight vectors do not zigzag the error landscape, thereby increasing the time to convergence on a minimum significantly. If more performance enhancing heuristics, together with their theoretical backgrounds and proofs are desired, the reader is directed to Haykin's comprehensive book⁵².

5.3.7 Eluding Local Minima

In complex non-linear functions, local minima abound. Avoiding these is paramount if one is to find the true, global minimum of the function. There are a number of methods for doing so, some of which are highlighted here. Take the function shown in fig. 5-6.

From fig. 5-6, one can deduce that a gradient descent based algorithm of the type discussed already in this chapter, will seek to follow the negative of the slope, thereby arriving at a minimum. It is hoped that it is obvious why this is not a good thing in the case of fig. 5-6: the algorithm will not find the global minimum at all, rather it will find a local minimum and stop searching.

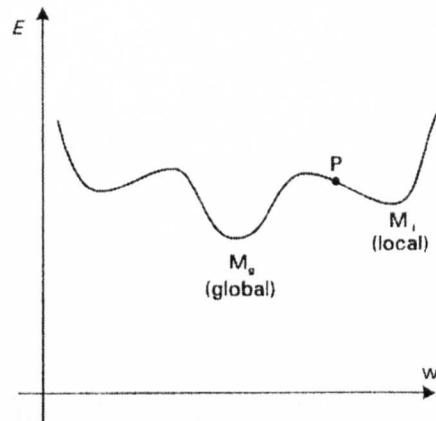


Fig. 5-6: Eluding Local Minima. After Gurney⁵¹

One simple method for eluding such minima is to perform the occasional “uphill” search. This occurs naturally as a part of the sequential method for updating weights and, although it is not able to compute the true error value, this is one of the advantages of training networks in this way. Searching “uphill” is also a feature of algorithms that include a “momentum” term, such as the Marquardt algorithm that was discussed earlier.

Momentum can be considered an additional term in the learning rule that takes into account recent historical gradient estimates. Algorithms with such a term included speed up learning when the gradient has a history of the same sign (direction) and therefore, the cumulative effect of the momentum term makes larger changes to the weight vector, thus speeding up learning. Conversely, when the gradient has a history of changing sign, the cumulative effect tends to slow down learning and thus, smaller weight change updates are made.

Annealing is an often-employed technique for avoiding the local minima of a function. The mechanical process of heating a metal inspired annealing with respect of neural networks, whereby, upon heating, the atoms in a metal start to shake violently, coming to an orderly resting state if gradually cooled.

Consider a ball at rest on a surface like that shown in fig. 5-6. If the surface is shaken with considerable force, the ball may come to rest at any point on the surface. If the ball

falls into any of the shallower minima and if the strength of shaking is reduced with each iteration, the chances of finding the global minimum are relatively high, one would imagine. Also, if the ball falls into the global minimum first time, successive shaking will not move the ball from its present resting state, due to the decreasing strength of the perturbation.

In practice, this method is often combined with an algorithm, such as the Marquardt, that can quickly find a local minimum. Thus, simulated annealing is used to find the initial weights and once this is achieved, the learning algorithm is employed to rapidly converge on a local minimum. Upon finding one, the weights are once more annealed to try to shake out of the minimum. If it succeeds, the process is repeated until it cannot escape such a minimum; this is then assumed to be a global minimum.

Finally, one can start from a number of different random points on the error landscape, thereby ensuring that the algorithm searches in different directions on the surface. The chances of finding a global minimum are, therefore, much greater. Effectively this entails running an algorithm with random starting weights, completing a training “session” and then repeating the process several times, each time with a different set of weights (starting position on the error surface). In fact, Masters⁶⁸ points out that to train a network only once and assume that the minimum of the function has been located is a grave mistake.

5.3.8 Generalization

Generalization with respect to neural network training is, effectively, the ability of a network to reproduce the correct outputs for the majority of input vectors in a test set that was not used during training of the network. In order for the network to stand a chance of achieving a generalized state, the training data must adequately represent the problem domain.

In addition, the network must not have an excess of neurons in either the hidden or output layers in order to avoid “overfitting”. Overfitting (see fig. 5-7) is a condition that

results from giving the network too much freedom (too many resources i.e. neurons) resulting in a network that essentially pays too much attention to the intricacies of the training data and disregards the underlying trend. Thus, after training, when a new input vector is presented the network it tries to find a suitable output value from those that were learned as part of the training process, rather than interpolating to a new output value. The network has effectively acquired a good memory rather than an ability to generalize.

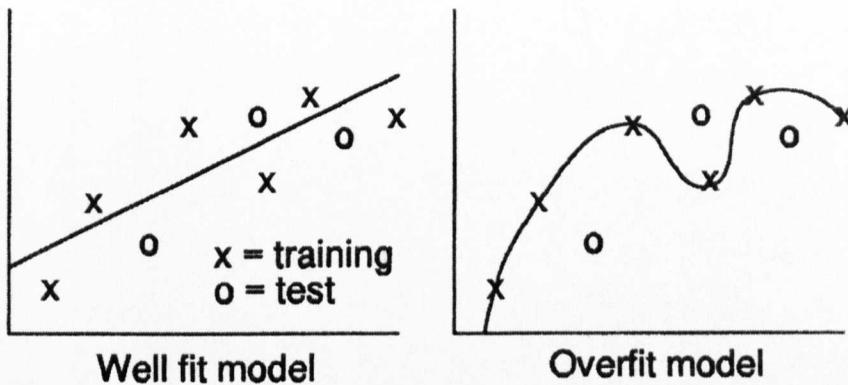


Fig. 5-7: Generalization and Overfitting. After Masters⁶⁸

Masters⁶⁸ places a good deal of emphasis on avoiding the above situations by suggesting that the neural architect should ensure that the chosen training set adequately reflects the problem. He states that the number of training samples should be large in comparison to the number of neurons and that, if in doubt, one should start with a neuron count that will be insufficient to train the network and then increasing their numbers in an iterative manner. In this way, the analyst is assured of building the optimum network.

5.4 Why Use Artificial Neural Networks?

Traditional mathematical techniques are often rendered inadequate when confronted with non-linear problems. Neural networks however, are capable of learning those tenuous, and often unidentifiable, threads or patterns within data that enables the non-linear mapping of inputs to outputs. The ability to learn means that neural networks are

able to adapt to their environment: they are often able to recognize patterns, even when the data presented is fuzzy or incomplete, and subsequently produce valid output.

They are robust in the sense that, should part of a network lose connectivity, the remainder will continue to function, thus a partial loss of function is not catastrophic in the same way it would be for conventional algorithms or programs.

Finally, when presented with inputs from outside the known problem domain, a neural network will try to extrapolate. In addition, the accuracy of output will decay in a graceful manner from the outer boundary of its programmed domain such that there can be some level of confidence in the results, even at its furthest reaches.

The problem posed in chapter 6 is to evaluate stresses in un-reinforced fabricated tees based on results from a number of finite element analyses and interpolate between them. Although the geometry parameter space is well defined, with reasonably regularly spaced data, the nodal coordinate data space is not.

The meshes of any two finite element models are necessarily different – they are unlikely to share a regular grid of nodal coordinates. Therefore, any method intended as an interpolation tool, must be able to do so smoothly between points spaced irregularly with respect to a position parameter and that vary from case to case. Neural networks fit this bill.

5.5 Closure

This chapter introduced the reader to the basic principles of neural networks, their origins and some of the theory behind them, in particular with respect to the multi-layer, feed-forward, back-propagation algorithm. Major contributions in this field since inception to the present day were highlighted in an effort to show some of the advances made and the potential power of such models.

A large proportion of the chapter dealt with the workings of the back-propagation algorithm, its relative strengths and weaknesses. Hopefully, what is conveyed is that neural networks offer a potential route to the solution of difficult, non-linear problems such as that posed in chapter 6.

In truth there is much that is not known about the workings of neural networks and interpreting results obtained via this route can be something of a challenge in itself! Nonetheless, it remains a fact that well trained neural networks often produce excellent results for difficult problems.

6 ELASTIC STRESS ANALYSIS OF UFT USING ARTIFICIAL NEURAL NETWORKS

6.1 Introduction

This chapter describes work undertaken with British Energy Ltd. to extend their current capability of stress prediction and estimation in un-reinforced fabricated tee junctions using a neural network approach.

Firstly, an overview of the current extent of British Energy's neural network program PIPET is presented before the scope of the current work is laid out. The tools used during this work are then discussed outlining the role of each in the project.

Presentation and validation of the FE results is subsequently discussed followed by an explanation of the methods used to utilize the data and implement it in British Energy's neural network stress prediction program. Results from the newly updated PIPET program are then presented and discussed with reference to previous relevant work available in open literature. Finally, conclusions are drawn with respect to the accuracy and usefulness of a neural network based rapid stress prediction system.

6.2 PIPET: A Neural Network Approach to Stress Prediction

Since 1996, British Energy have been using a neural network based program – PIPET – for the rapid evaluation of stresses in welded UFT pipe junctions. This technique was born from the desire to be able to quickly compute stresses in un-cracked pipe junctions to aid decision-making and risk analysis with respect to their physical counterparts in several operational plants.

In the PIPET system, assessment of the stress levels in un-cracked piping junctions is done without solution of the governing material equations, instead they are found by interpolation of data from a number of linear elastic finite element reference solutions. PIPET enables a rapid analysis of stress levels in a given junction, the results of which are then used, in conjunction with other data, to determine whether a more costly and lengthy FE analysis study and/or other analytical analysis should be done; which might involve complex loading and/or meshes to accurately describe the situation.

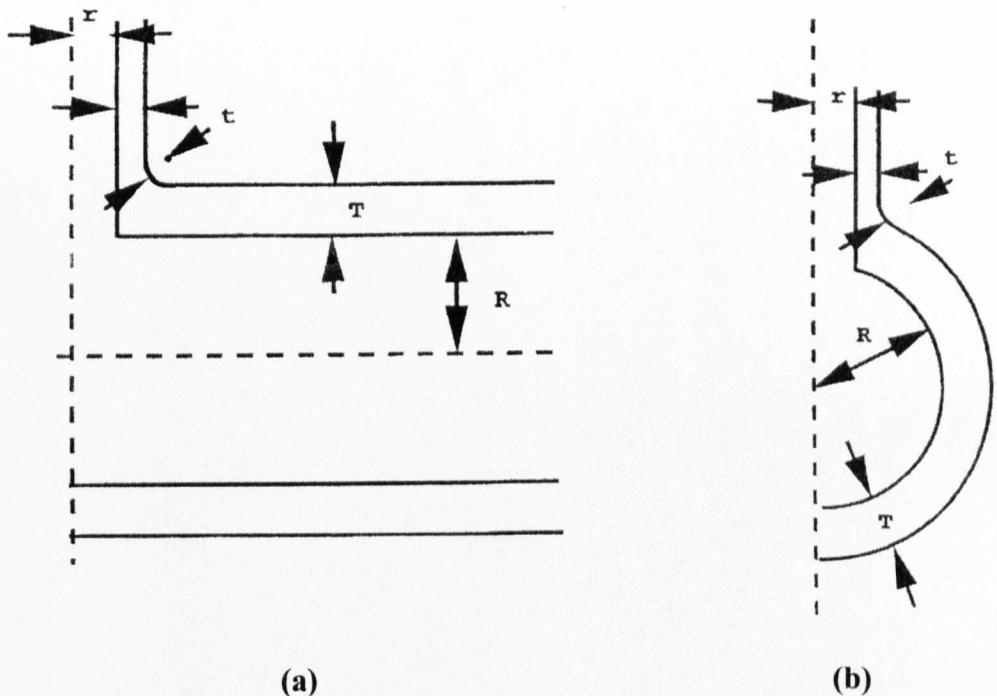


Fig. 6-1(a): Crotch section UFT profile (b): Flank section profile

The original system used data from 68 linear elastic FE analyses to estimate stresses in the crotch (longitudinal) and flank (transverse) branch junction planes (see fig. 6-1). The specified geometry parameter limits of the system, according to the range of FE analyses done, were as follows:

$$0.1 \leq r/R \leq 1.0$$

$$0.025 \leq T/R \leq 0.25$$

$$r/R \leq t/T \leq (2.0 - r/R)$$

PIPET is able to compute stresses, based on the aforementioned FE data in both crotch and flank section planes and has a positional range of -0.5 to 1.5 as shown in fig. 6-2. The indexing angle is known as the position parameter, a .

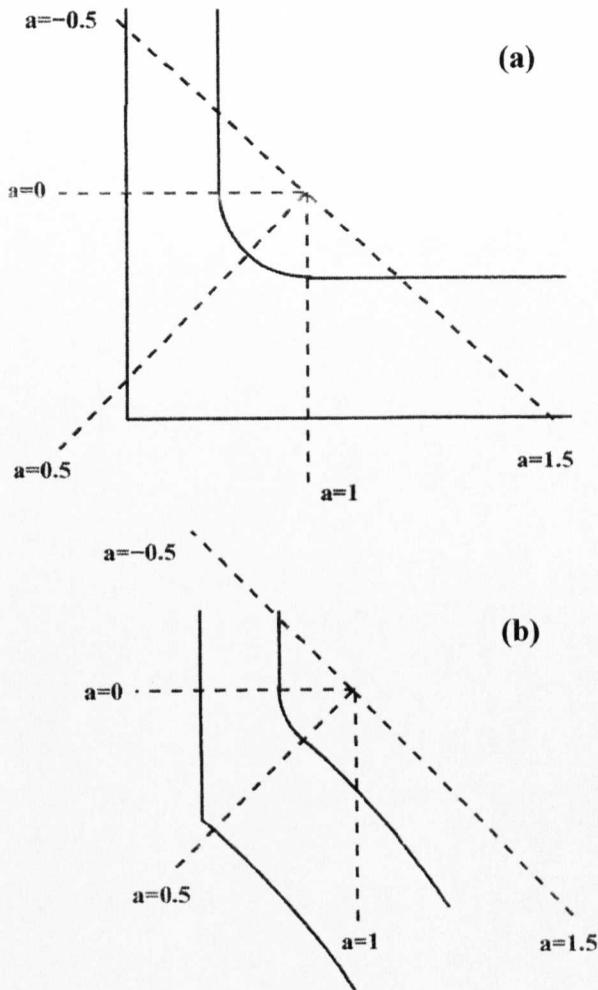


Fig. 6-2: Definition of position parameter for (a) Crotch Profile (b) Flank Profile

The actual cases originally analysed by British Energy were:

$$r/R=0.1, T/R=0.025, 0.09, 0.17, 0.25$$

$$t/T=0.1, 0.36, 0.68, 1.0, 1.32, 1.64, 1.9$$

$$r/R=0.4, T/R=0.025, 0.09, 0.17, 0.25$$

$$t/T=0.4, 0.7, 1.0, 1.3, 1.6$$

$r/R=0.7$, $T/R=0.025, 0.09, 0.17, 0.25$

$t/T=0.7, 1.0, 1.3$

$r/R=1.0$, $T/R=0.025, 0.09, 0.17, 0.25$

$t/T=0.95$ (flank results only), 1.0 (crotch results only)

The original three-dimensional FE analyses were done for geometries conforming to the profile in fig. 6-1, where r is the branch pipe inside radius, t is the branch wall thickness, R is the run pipe inside radius and T is the run pipe wall thickness. The radius of the weld is a circular arc considered to be equal to the branch pipe thickness and the intersection between the inside surfaces of the pipes has a sharp corner.

Quarter models of the actual geometries were used to create the FE models using the PATRAN pre-processing package in order to reduce the computational overhead. ABAQUS was the FE solver used for the analyses. As a result, a combination of anti-symmetric and symmetric boundary conditions were necessarily used to constrain the models – a method of constraining pipe junction FE models that may produce slightly different results from those obtained with the more usual cantilever approach.

Fox and Mitchell⁷² showed that PIPET was capable of producing reasonably accurate results even for geometries that were not exactly as per fig. 6-1. Consequentially, PIPET should provide engineers with a more accurate estimate of stresses than traditional hand calculations. An additional benefit of using neural networks for the interpolation is that they are often able to produce meaningful results beyond their original specified and intended domains, since they tend to decay “gracefully” unlike other numerical interpolation methods.

6.2.1 Scope of Work

The project mandate, as agreed with British Energy, consisted of the following:

1. Produce a mesh pre-processor, similar to that used in generating the models for the work described in chapter 3 of this thesis, using the MATLAB programming language; allowing automated model and mesh generation within the FE analysis pre-processor – FEMGV.
2. Perform a series of FE analyses, verifying the procedures and programs implemented by comparison with previous British Energy FE solutions that were used to derive weights for the original PIPET program.
3. Upon satisfactory validation of the new FE data complete the FE phase, extending the geometry parameter domain beyond PIPET's original capability by supplementing the geometry matrix with additional cases.
4. Analyse all seven primary load conditions for a branch junction, i.e. pressure plus three branch and three run moment load cases.
5. Verify new FE data as far as possible.
6. Investigate and optimise neural network methods for improving PIPET interpolation of the FE results, reducing the predicted stress range uncertainty.

In total, this work comprised a parametric survey of 108 UFT FE models, 97 of which were used to train the neural networks as defined in table 6-1. The 11 geometries listed in table 6-2 are those FE jobs used to evaluate the performance of the new weights used in PIPET, once training was complete.

The dimensionless parameters r/R , T/R and t/T in this work refer to the branch pipe to run pipe inside radius; run pipe wall thickness to run pipe inside radius and branch pipe to run pipe thickness ratios respectively.

r/R	T/R	t/T	r/R	T/R	t/T
0.05	0.025	0.05	0.40	0.25	0.30
		0.50			0.70
		1.00			1.00
		1.40			1.30
	0.17	0.05			1.60
		0.50		0.50	0.30
		1.00			0.70
		1.40			1.00
	0.50	0.05			1.30
		0.50			1.60
		1.00	0.70	0.025	0.60
		1.40			1.00
0.10	0.025	0.10			1.30
		0.36		0.09	0.60
		0.68			1.00
		1.00			1.30
		1.32		0.17	0.60
	0.09	0.10			1.00
		0.36			1.30
		0.68		0.25	0.60
		1.00			1.00
		1.32			1.30
	0.17	0.10		0.50	0.60
		0.36			1.00
		0.68			1.30
		1.00	0.85	0.025	0.70
		1.32			1.00
	0.25	0.10			1.30
		0.36		0.09	0.70
		0.68			1.00
		1.00			1.30
		1.32		0.17	0.70
	0.50	0.10			1.00
		0.36			1.30
		0.68		0.25	0.70
		1.00			1.00
		1.32			1.30
0.40	0.025	0.30		0.50	0.70
		0.70			1.00
		1.00			1.30
		1.30	1.00	0.025	0.95
		1.60		0.09	0.95
	0.09	0.30		0.17	0.95
		0.70		0.25	0.95
		1.00		0.50	0.95
		1.30		0.03	1.00
		1.60		0.09	1.00
	0.17	0.30		0.17	1.00
		0.70		0.25	1.00
		1.00		0.50	1.00
		1.30			
		1.60			

Table 6-1: Geometries analysed for training PIPET

r/R	T/R	t/T
0.20	0.17	1.00
0.20	0.40	0.40
0.30	0.05	0.25
0.30	0.09	0.60
0.50	0.30	0.70
0.50	0.50	1.20
0.60	0.20	0.70
0.60	0.40	1.00
0.75	0.05	0.90
0.75	0.10	1.10
1.00	0.13	1.00

Table 6-2: Geometries analysed for evaluating PIPETs performance post-training

6.3 Generating and Preparing Data for Training Neural Networks

6.3.1 Introduction

In order to create a robust and competent artificial neural network, one must provide the training algorithms with good data in the first place, the computer programming adage “garbage in, garbage out” is worth bearing in mind when preparing the finite element models for data mining and subsequent use in training neural networks.

Several key stages in the production of an effective system of neural networks for the prediction of stresses as per the mandate were identified:

1. Programming of a suitable algorithm for the rapid creation of finite element models.
2. Validation of FE data as far as possible to ensure accuracy of FE (and, therefore, PIPET) results.
3. Identification of the best route for extracting the relevant FE data and writing a program to do so in an automated manner (or at least minimizing the manual input to the process).

4. Preparation and structuring of FE data in a suitable format for later input into a neural training algorithm.
5. Identification and subsequent programming of a suitable algorithm, for the automated training of neural networks.
6. Generation of new neural network synaptic weights.
7. Identification of those weight vectors generating the lowest training errors for subsequent use in PIPET.
8. Independent evaluation of the new neural network weights using previously unseen (by the new networks) FE data.
9. Cross-validation of new neural network output using available data in open literature.

In order to achieve the above points, a total of four software packages together with additional Fortran and MATLAB programs, Unix shell scripts and proprietary programs written by Dr. Mike Fox at British Energy were used in the FE modelling and analysis, data extraction and neural network training respectively.

The scripts that generated batch input files describing the geometry, mesh, boundary conditions and loads (stage 1) were written using MATLAB 6. After pre-formatting the ABAQUS results file using the Fortran program “filprep” and creating a modified results file known as an “.mdf” file, MATLAB was once more employed to extract the required stress data (stages 2, 3 and 4), concatenating common data into training and validation files.

Programs written and used by British Energy for the original training of the PIPET application was used extensively in the training processes (stages 5 and 6). In particular, the “qckprp” training program used to train the neural networks in conjunction with the Unix shell script “fastprp” written by the current author, which allowed automation and

control of the training process. The best weight files for each of the trained networks were subsequently found using the Unix shell script “bestw” (stage 7).

Testing of the new concatenated weights file implemented in PIPET (stages 8 and 9), was achieved using British Energy’s test programs as described in section 6.5 of this thesis.

FE pre-processing and some graphical post-processing was done using FEMGV 6.1 and the solver used for the analyses was ABAQUS/Standard 5.8-1. A short description of the function of each of the proprietary software packages used is now given.

MATLAB v6

MATLAB is a technical computing tool widely used in the engineering community. This software was used to program a number of scripts, “mgen.m”, to produce ASCII files populated with FEMGV keywords in order to construct a wide range of 3D finite element models. The resultant ASCII files are known as history files and are imported directly into the FEMGV program as a so-called batch files.

After having first been pre-formatted using the Fortran program “filprep”, the MATLAB program “prep.m” was used to gather the requisite stress data from the ABAQUS results file. This program formats and then stores the FE stress data in cell arrays that make up either training or validation files. Once complete, the files holding the cell arrays are used as inputs to the program “concat.m”, which sorts the stress data by profile, load case and stress component (out-of-plane/in-plane) before automated training of the neural networks using the “fastprp” shell script in conjunction with the “qckprp” program.

The MATLAB and Fortran source code, used in achieving each of the stages described above, is given in Appendix H. Appendix I contains all Unix shell script source code.

FEMGV v. 6.1

The pre and post-processing of all FE models was done using the FEMGV package. It processes the batch input files produced by the “mgen.m” program and generates the entire model, including geometry, mesh, loads and boundary conditions in a single step. Once done, the user has the task of making any necessary refinements to the mesh and then generating the ABAQUS input deck (.inp file) for use within the ABAQUS general-purpose FE solver.

ABAQUS v. 5.8-1

The ABAQUS/Standard v.5.8-1 general-purpose FE solver was used to process the FE input files generated by the FEMGV pre-processor. The results from the analysis were taken from the relevant ABAQUS output file (.fil) and subsequently formatted, for training the neural networks, using the “prep.m” MATLAB program.

QCKPRP

The Unix based “qckprp” neural network training program was written by British Energy. It was designed to train feed-forward networks using the back-propagation algorithm (see chapter 5 of this thesis) and was written using the Fortran programming language. It was used extensively to train networks for both the peak and profile stresses.

6.3.2 Finite Element Modelling

As was mentioned earlier in this chapter, MATLAB was used to generate ASCII files containing FEMGV keywords that would, to a large extent, automate the creation of the UFT FE models used to train and evaluate the neural networks. A number of scripts were written to cope with the broad range of geometries involved. The programs “mgens.m”, “mgen.m”, “mgenl.m” and “mgene.m” (see Appendix H for the source code for mgen.m) generated very small, medium, large and equal radius ratio

geometries respectively. This ensured that good quality meshes could be formed for a given geometry, regardless of individual geometric features.

Invocation of one of the model generating programs (see fig. 6-3) prompts the user to specify the number of models they wish to create in that session. One must then enter filenames that contain key geometry details, namely the branch to run radius ratio, the run radius to thickness ratio and the branch to run thickness ratio.

For example, a FEMGV FE model having geometry ratios $r/R = 0.05$, $T/R = 0.25$, $t/T = 1.32$ could be given the following filename: jf-005-025-132.

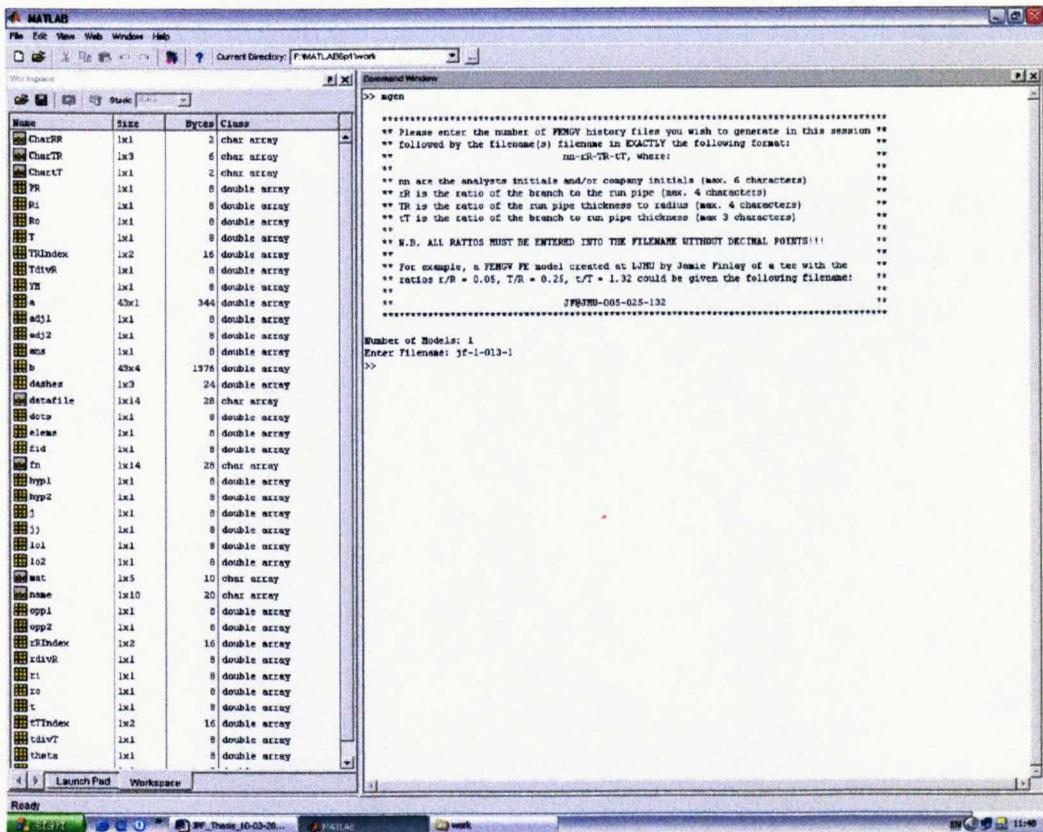


Fig. 6-3: Generating FEMGV input files via the MATLAB command line

The “mgen” programs use the information in the filename to generate a file containing FEMGV keywords, subsequently imported into the pre-processing application. In this

manner, one of the primary objectives of the modelling phase – to automate the modelling process – was achieved.

Since the seven main piping loads experienced by a pipe junction were required to be analysed (see fig. 2-6) the models necessarily had to be in three dimensions. Furthermore, the vast majority of computational effort during a FE analysis is expended decomposing the stiffness matrix for the model and is largely independent of the number of loads in the analysis. Thus, it is more cost effective to produce a single full model including all individual load cases rather than run multiple quarter and half models.

Upon importing the ASCII history file into the FEMGV application, the geometry of each model is built up in the following manner. Firstly, the upper and lower weld profiles and the crotch intersection curve are generated by intersecting two cylinders of the requisite diameters. The remainder of the geometry is then created using straight lines and arcs to produce surfaces for the inner and outer “skins” of a quarter of the geometry – see fig. 6-4(a). Each internal surface is then projected onto its outer surface counterpart creating a solid protrusion, shown in fig. 6-4(b). Completing a quarter model in this fashion, the full model is generated by mirroring first the quarter then the half models.

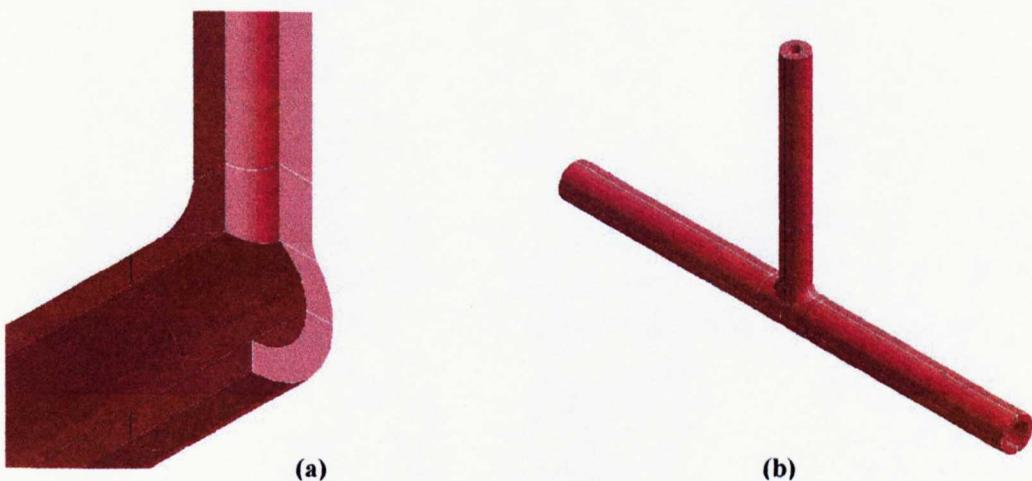


Fig. 6-4: (a) Quarter and (b) Full UFT model geometry

The FE models were each constructed using 20-node 2nd order (quadratic) reduced integration elements. The ABAQUS code for this element is C3D20R. 20-node elements have a proven track record for the linear elastic FE analysis of pressure vessels and pressure components as discussed in chapter 3.

The cantilever model shown in chapter 2, fig. 2-4, represents the boundary condition adopted for analysis of all seven load-cases. With respect to the FE model the cantilever condition fixes all nodes at one end of the run pipe for translation and rotation.

As per the earlier work detailed in chapter 3 of this thesis on reinforced branch outlets, and for the reasons given in that chapter, the location of the nozzle in respect of each end of the run pipe was fixed at a distance of $3.43 \times D_o$ – a factor derived from stress die away studies for 90° branch pipe connections⁴⁰.

Loads consisted of a unit internal pressure load on all internal surfaces and three individual unit moment loads. The moments are applied to a rigid body reference node located at the free ends of the branch and the run pipes on their central axes. Each node on the surface of either the run or branch end is tied to it's rigid body reference node by a rigid beam element. Thus, when a moment is applied to the rigid body reference node, it is smoothly transferred to the surface in question. The assumptions made in modelling the reinforced branch outlet geometries in chapter 3 also stand for the work done herein.

Mesh convergence study

During the course of this study, a substantial amount of FE analysis has been done for piping junctions and a good deal of knowledge has been gained with respect to modelling and meshing them. Therefore, a certain amount of judgement was used when designing the individual meshes.

In addition to doing a small convergence study, and since there is such a broad diversity of geometry parameter ratios, meshing was done on a case-by-case basis using an algorithm to design a mesh that required only a little work on behalf of the user to hone

the final meshed model. Writing a program to cater for all possibilities, producing an optimised mesh in each case via a mesh convergence study, was both beyond the scope of the project and an improbable proposition in any event.

For the aforementioned convergence study, a quarter of the geometry was modelled, a unit pressure load applied to all internal surfaces and symmetry boundary conditions were applied to the two symmetry faces in the XY and ZY planes. One node was constrained in the global Y direction at the inside surface of the run pipe. Equilibrium was maintained by simulating a “closed end” condition at the “free ends” of both the run and the flange.

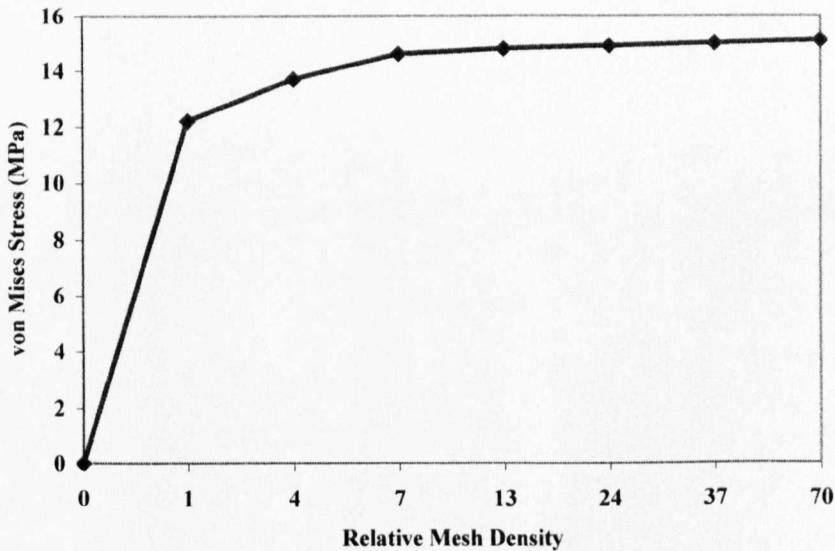


Fig. 6-5: UFT mesh convergence study –mesh density vs. von Mises Stress

A total of seven models were analysed and, since the region in which the branch and the run pipe intersect was the particular site of interest, the mesh convergence study focussed on that area. The von Mises stress was recorded and plotted against relative mesh density in each case (see fig. 6-5). The model with a relative mesh density of 13, with three elements through the thickness as shown in figs. 6-6(a) and (b) was considered to have converged. Relative mesh density is defined here as the density of the mesh in question, divided by the density of the coarsest mesh.

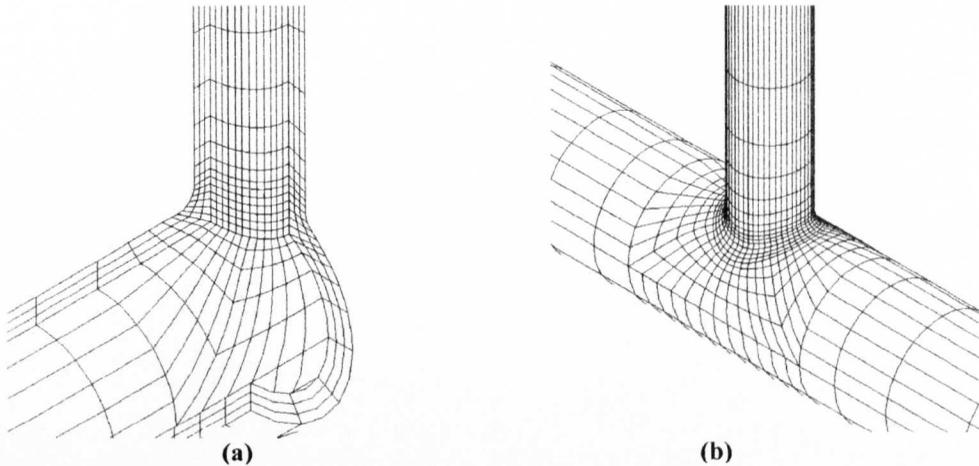


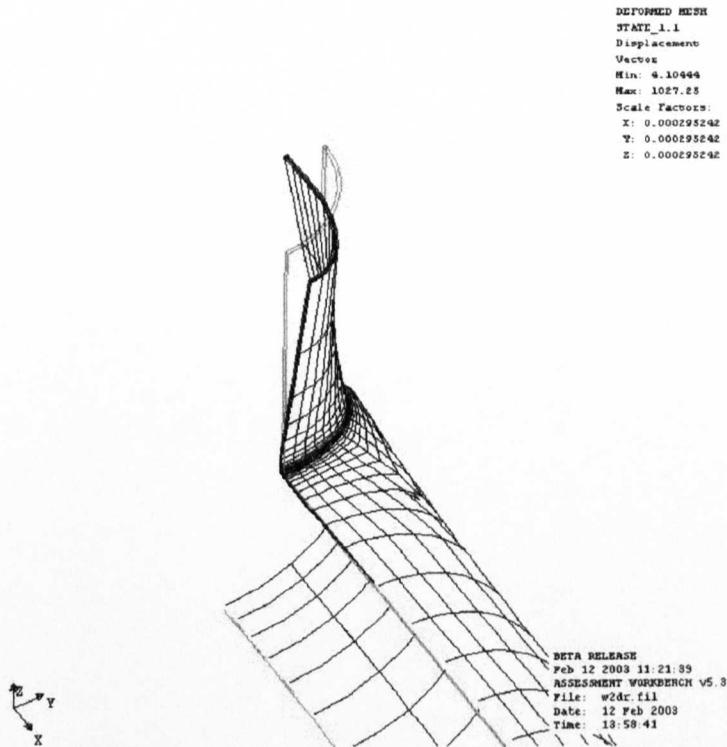
Fig. 6-6: (a,b) UFT converged mesh, inside quarter section. (b) Full model isometric view.

Validation of Finite Element Data

Having completed 97 finite element analyses used for training the neural networks, a comprehensive validation of the FE data was undertaken. During the course of the validation exercise a number of oddities arose.

In the first instance, a number of cases were selected randomly for the validation exercise. For each case the results from the crotch and flank profiles, for pressure and branch bending moment loads, in and out-of-plane stresses, were compared. Some of the results were, at first, inexplicably and significantly divergent.

Upon further exploration the root cause for the divergence was uncovered. It was found that there were previously unknown problems with the original British Energy FE mesh model. It was apparent that, for just a few cases, the branch mesh was distorting in an unexpected manner, in fact the branch was ovalizing under applied internal pressure load. Fig. 6-7 shows the distorted mesh of the model with a radius ratio of 0.7, a run pipe thickness to radius ratio of 0.025 and a branch to run thickness ratio of 1.0.



6-7: Ovalization of the British Energy FE mesh

Of the 97 FE models created for the neural network training phase of this study, 56 were directly comparable with previous FEA undertaken by British Energy. Once the anomaly and its cause had been identified, these 56 sets of FE results were compared with British Energy's original results for in-plane and out-of-plane stresses, pressure, in-plane and out-of-plane moment loads – a total of 448 cases. This was done primarily for the sake of comprehensively validating the results and ensuring confidence in them. In addition, the process served to identify the original cases that were potentially producing erroneous data in the PIPET program due to said errors.

It is not possible to present all 448 graphs and the associated data in this thesis, however, there are four cases (figs. 6-8 – 6-11) presented here that are considered representative of the larger population. In each case, both out-of-plane and in-plane stresses for a single load case (pressure, branch in-plane bending and branch out-of-plane bending) and for a single section (crotch or flank) are shown. The in-plane component used was the larger (in absolute magnitude) of the two in-plane principal

stresses. Thus, for the first example in fig. 6-8, $r/R=0.1$, $T/R=0.09$, $t/T=1.0$ the graphs represent the out-of-plane (hoop) and in-plane stress data in the flank section for unit internal pressure load.

The abscissa is the position parameter, a , and the ordinate represents normalized stress (MPa). Normalization of the stresses is done as part of the process to prepare the data for training in “qckprp”. The “Outside” and “Inside” labels in the legend refer to the position of the data with respect to the surface. Thus, the term “Hoop Stress Outside” in a legend refers to the out-of-plane stress data on the outside surface of the tee junction. The legend tag JF refers to the new FE data; BE refers to the original finite element data produced by British Energy.

The FE validation exercise proved to be very satisfactory, with the overwhelming majority of cases showing good concordance with the original British Energy data as shown in figs. 6-8 – 6-11. This is true, not only of the cases presented below, but also of the other 440 cases that were directly compared in this study. It is felt that the FE data generated for this study is of a higher quality than the original data, as the current models do not display branch ovalizing under applied internal pressure load.

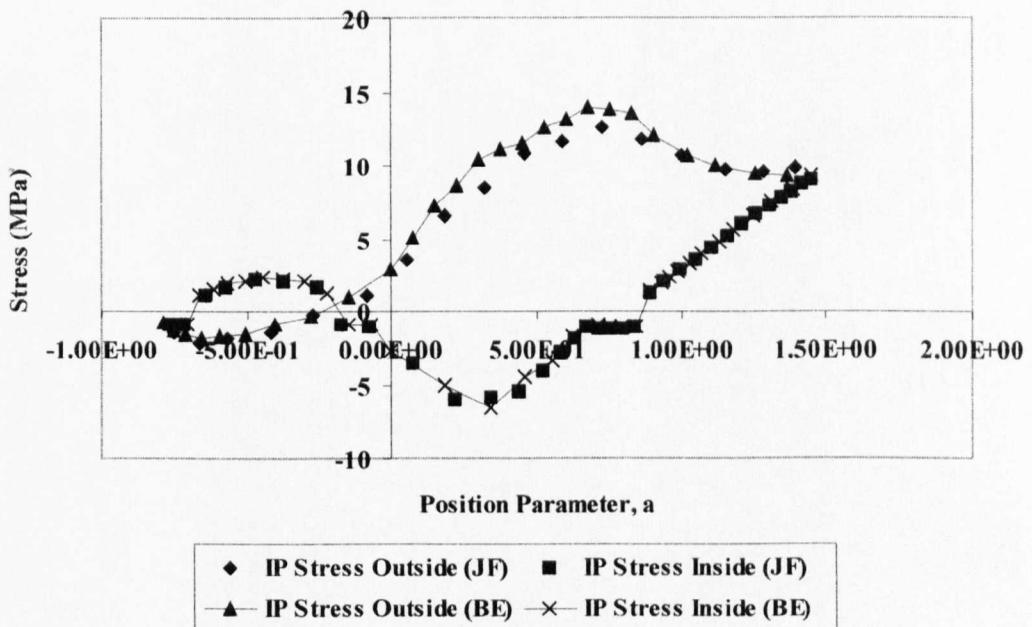
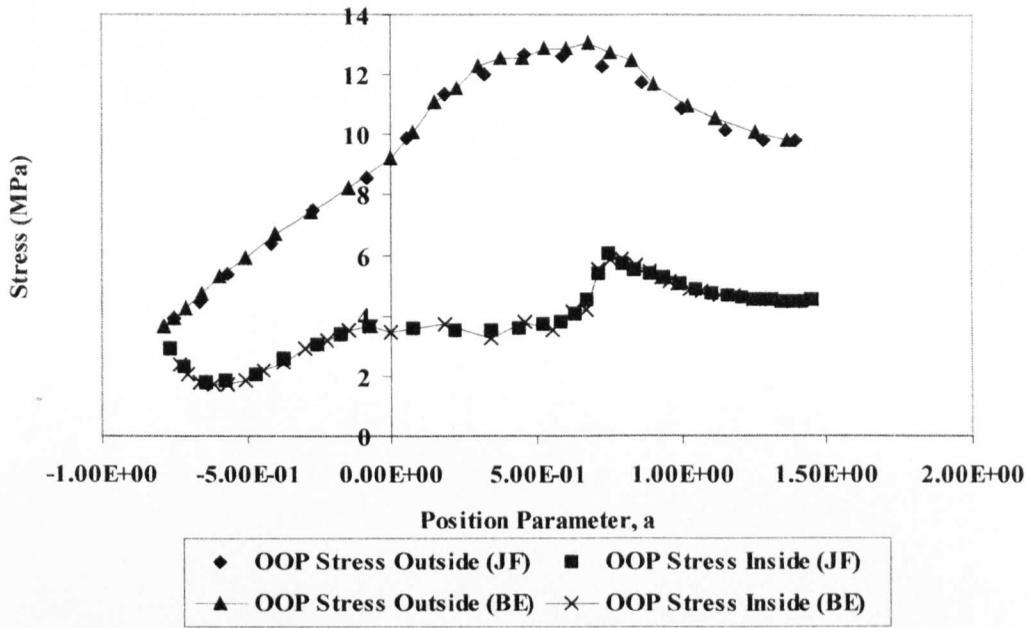


Fig. 6-8: FEA stress data for flank profile internal pressure: $r/R=0.1$, $T/R=0.09$, $t/T=0.36$

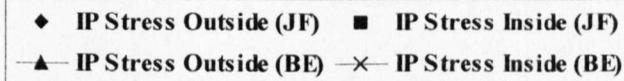
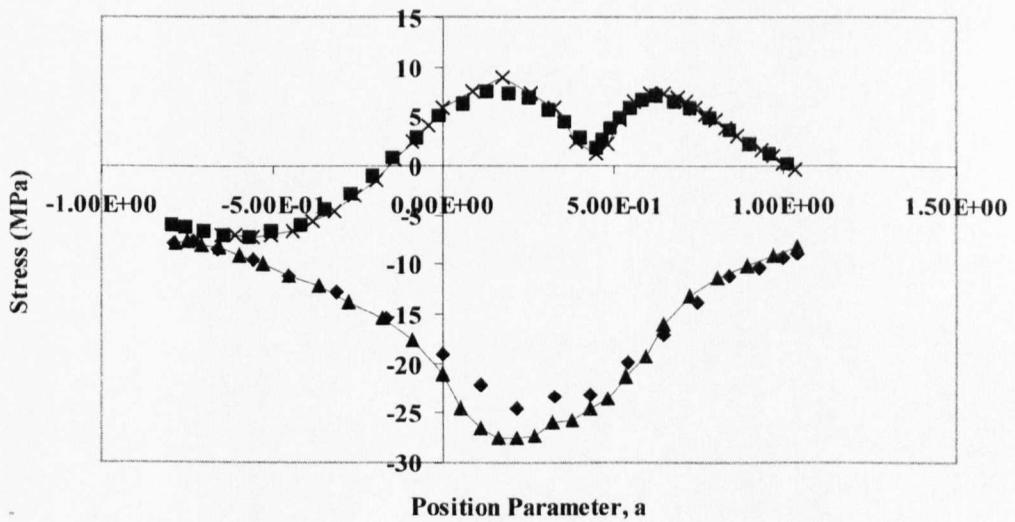
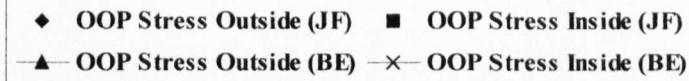
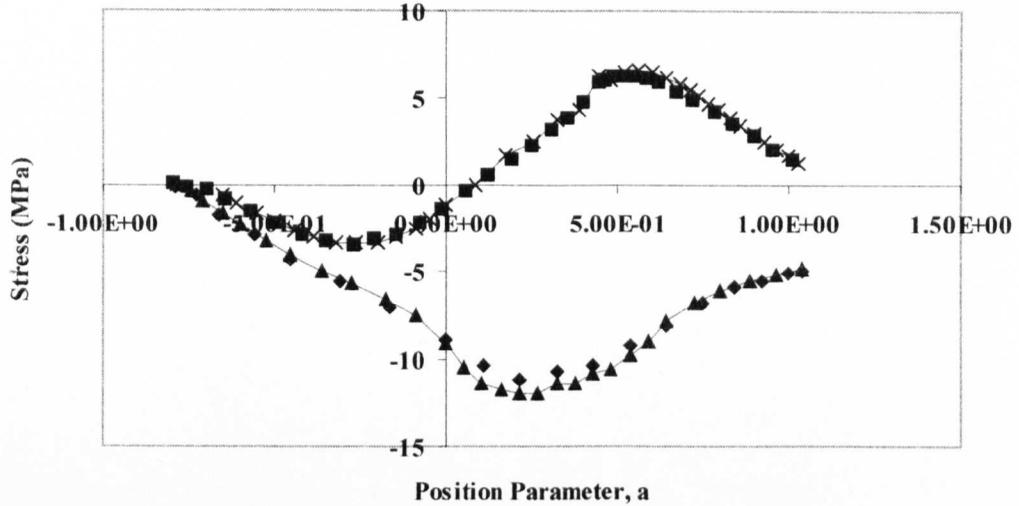


Fig. 6-9: FEA stress data for flank profile branch out-of-plane bending moment: $r/R=0.4$, $T/R=0.25$, $t/T=0.7$

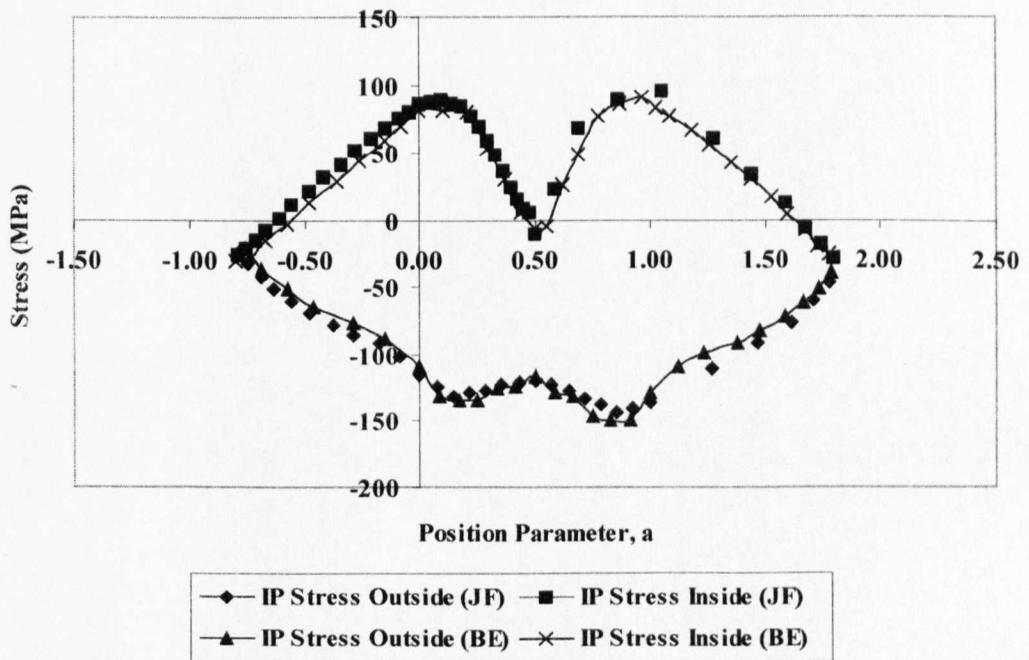
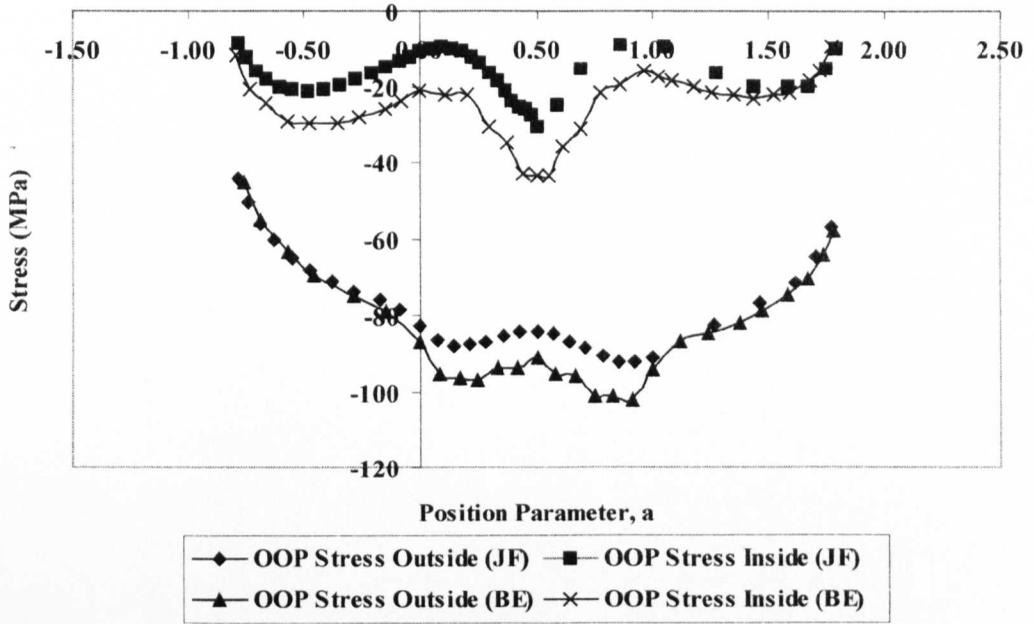


Fig. 6-10: FEA stress data for crotch profile branch in-plane bending moment: $r/R=0.7$, $T/R=0.025$, $t/T=1.0$

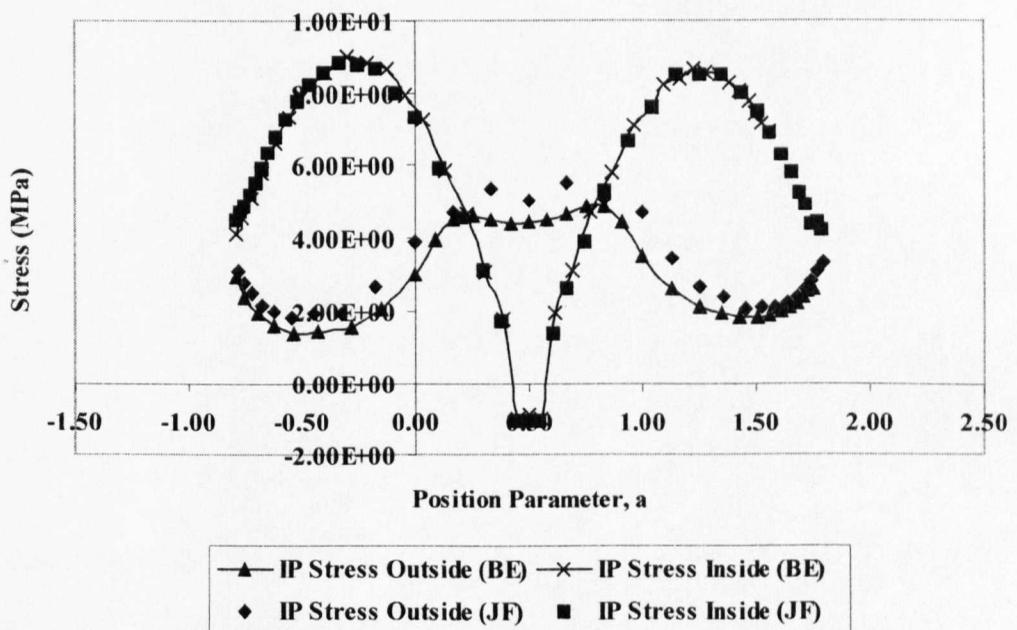
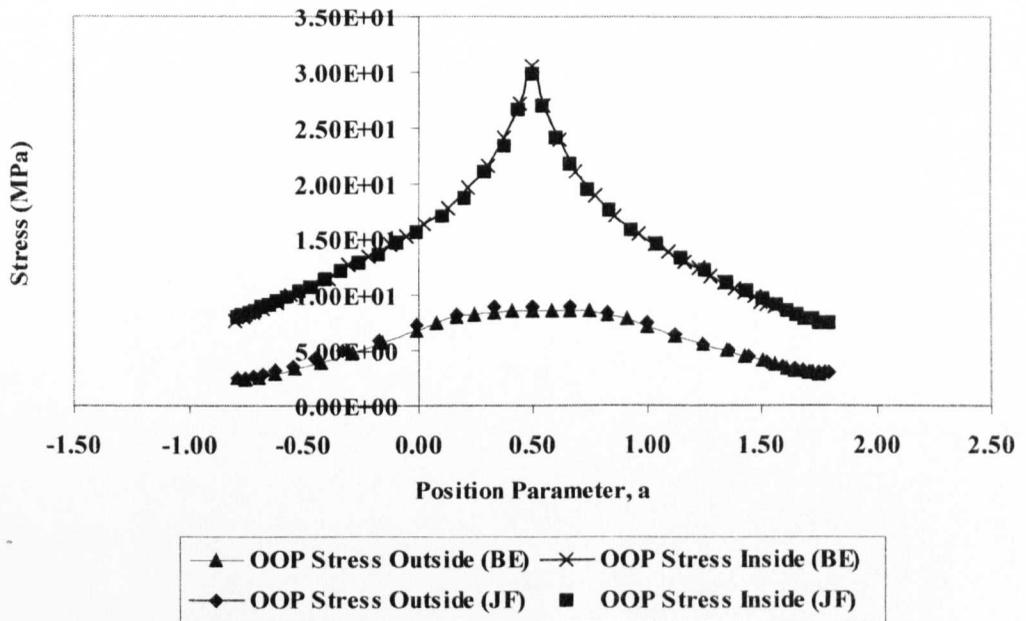


Fig. 6-11: FEA stress data for crotch profile branch in-plane bending moment:
 $r/R=1.0$, $T/R=0.17$, $t/T=1.0$

6.4 Training Neural Networks Using FE Data

Recall that the desired output from the training of neural networks using the new FE data is to provide the PIPET program with a new set of synaptic weights enabling more accurate stress estimates for UFTs. The work that follows describes the steps taken to arrive at, and evaluate the performance of, new weights for use in PIPET.

The process of training an artificial neural network involves a number of stages. The first, manual, process is to decide which data should be put into each of the following sets: training, validation and evaluation. The data must then be collated and formatted in an appropriate manner. To this end, programs written in the MATLAB scripting language enabled the automation of this process. Once the data is correctly formatted, the network must then be designed and trained.

It is desirable to train a neural network on data that includes most, if not all, of the points that lie on the extremities of the defined parameter space. It is logical to select data in such a manner, since it can then be ensured that the network reliably computes solutions for all problems likely to be encountered within the specified bounds or domain. Data used in validating the training data are usually, although not necessarily exclusively, taken from within the bounds of the parameter space.

Data used for evaluating the performance of the network, once trained, may be selected from within the bounds of the parameter space, although points outside the parameter space could also be used to see how the neural network copes with geometries beyond its intended scope. Of course, much depends on the availability and ease of collecting such data.

Generally, neural networks can provide useful solutions even when asked to compute outside their original intended parameters. Evaluation data does not need to be normalised, since the object of the exercise in this case is to compare the networks true output against actual raw FE data.

6.4.1 Extracting and Formatting the FE Data

The default format of the ABAQUS results file is binary and must be transformed in order to ease the manipulation of the stress data within the MATLAB environment. The Fortran program “filprep” interrogates the binary file and outputs only stresses and nodal information. Furthermore, “filprep” puts a break in between each load case and calculates the principal stresses for each nodal stress component in the file. The line breaks in the newly formed “.mdf” ASCII file act as pointers when the file is subsequently pulled into the “prep.m” program for sorting and extraction of the relevant data.

Once all ABAQUS results files have been pre-formatted in this way and placed together in a common directory on the hard disc, the “prep.m” program is invoked and the user is asked to input whether a training or validation data set is to be created. The program then prompts for the number of files (number of training/validation samples) in constructing the set and subsequently prompts the user to select the first results file.

When the “.mdf” file is processed by “prep.m”, a number of things happen. Firstly, the file is stored in a single array and contains all stress data for each node and all seven load cases. The load cases are separated by breaks, which allows the data to be formatted and filtered according to the geometry profile (crotch or flank). Any data that does not lie on either the crotch or flank section profiles is discarded. The MATLAB program “prep.m” then assigns the remaining data to arrays according to profile and each of these arrays are filtered according to the position parameter, a , (see fig. 6-2).

Once the arrays contain just the stress data for nodes of interest, “prep.m” compares (for all nodes in both crotch and flank sections) the three principal stresses to the out-of-plane direct stress. According to the coordinate system used herein, for nodes in the crotch section, the out-of-plane stresses are in the z-axis direction; the flank section out-of-plane stresses are in the x-axis direction.

The algorithm compares the out-of-plane direct stress against the three principal stresses to determine which one corresponds to the out-of-plane direction. It then takes the

largest of the remaining two principal stresses (in absolute magnitude) as the in-plane stress.

The maximum principal in-plane and out-of-plane stresses are subsequently used to normalize the profile data for that load case. The maximum principal stresses are themselves normalized by taking their natural logs. In this way, the stress range is minimised, equalizing the importance of the output variables and facilitating faster training and better network performance⁶⁸.

The data is then sorted, according to position through the thickness of the profile (represented by the Greek letter tau, τ), load case and stress component (maximum/middle) and stored in a structured cell array, which can be easily indexed when creating the neural network training and validation files.

After creating the training and validation cell arrays, each contained within their respective files, the program “concat.m” is invoked in order to produce training/validation files for specific stress profiles and peaks. Since each FE model has three elements through the thickness of the geometry, one may consider a training or validation profile as being that “string of nodes” on either the outside surface ($\tau = 0$), one-third of the way through the thickness ($\tau = 0.33$), two-thirds of the way through the thickness ($\tau = 0.66$) or on the inside surface ($\tau = 1$). Fig. 6-12 shows the flank section outside surface nodes. Those that lie within the bounds of the position parameter a , as measured from the centre of the weld, are considered to be a training “profile”.

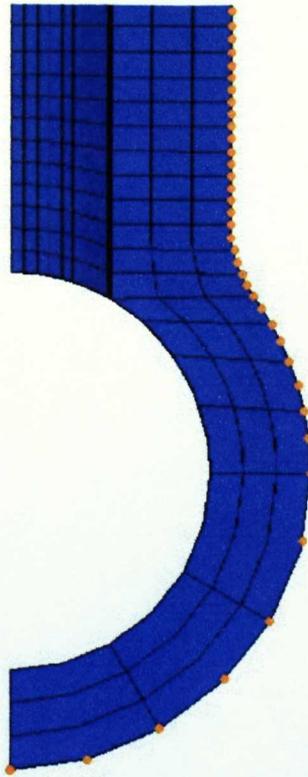


Fig. 6-12: For each section (crotch and flank) nodes are subdivided into training/validation “profiles” according to their through-thickness position

For example, in order to be able to predict the out-of-plane stresses for all positions, a, on the outside surface of a tee under internal pressure load in the crotch section, one must first be able to isolate that data pertaining to that through-thickness position for each of the sample FE cases. One must then create training and validation files from this data and then design a suitable neural network, training it down to a satisfactory level of error using the aforementioned programs.

The “concat.m” program (see Fig. 6-13) allows the user to isolate the required data in this way by indexing each of the training or validation FE samples according to the desired training profile.

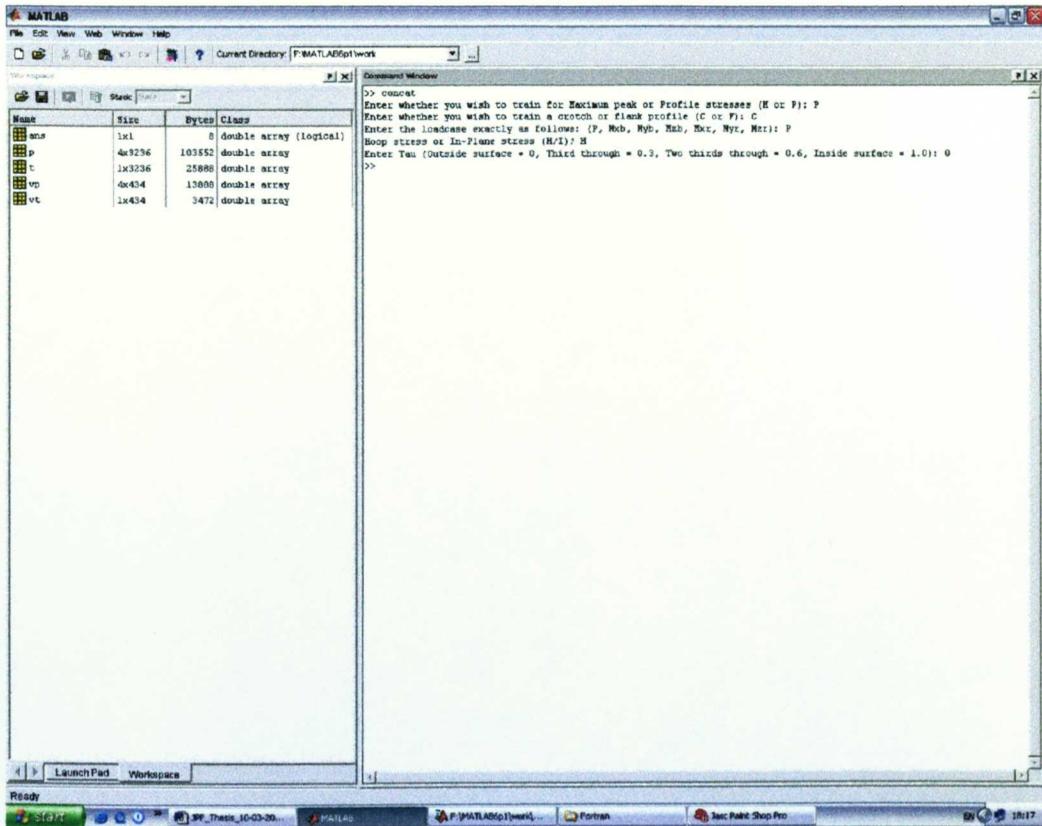


Fig. 6-13: “concat.m” is the MATLAB program used to concatenate all stress profiles and peaks from cell arrays forming new training and validation files

6.4.2 Designing Neural Networks

Initially, the intention was to train neural networks using the MATLAB Neural Network Toolbox and a program was written “qcktrain.m” to enable command line invocation and control of the neural network tools available within the MATLAB Toolbox. The basis for doing so was that the data was already formatted in a form that could be readily used within MATLAB for training neural networks. In addition, the Neural Network Toolbox offers a substantial array of options with respect to which training algorithm is used, the learning function and so on.

However, this approach proved less than successful. Despite literally thousands of hours of computer time, reasonable solutions using the Toolbox were not forthcoming. Compounding the problem was the fact that the root cause could not be found for the

apparent inability to train reasonably well-behaved networks down to satisfactory error levels.

Due to the unsuccessful attempts to train networks using MATLAB, an alternative route to training was initiated: utilising British Energy's in-house "qckprp" software. Although this was not entirely convenient, since the data required further manipulation before being put into files formatted in a way that was acceptable for the bespoke software. Additionally, the software was written for use on Unix workstations and the number of available machines numbered just three, thus the computational resources available for the task at hand were significantly less than would have been the case running MATLAB on the available PCs.

The "qckprp" training program is a multi-layer, feed-forward, back-propagation algorithm with one hidden layer of nodes, as first described in chapter 5 of this work. It uses sigmoid hidden layer activation functions and a linear activation function in the output layer. Since the number of inputs and outputs are fixed for both the peak and profile training scenarios, the only variance in terms of the structure of each network is the number of hidden layer neurons. A schematic representation of such a network is shown in Fig. 6-14.

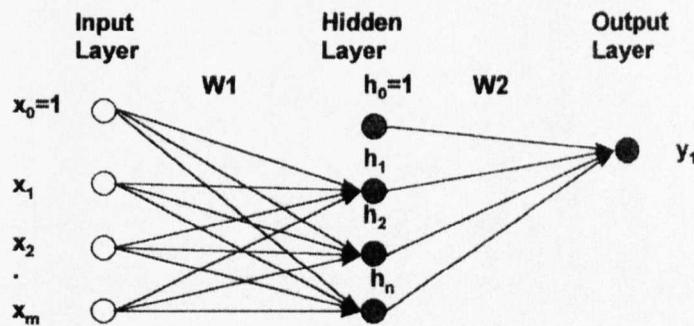


Fig. 6-14: A single hidden layer neural network in "qckprp". Adapted after Hines⁷³

The program trains networks in batch mode, where measurement of a networks performance is done by recording the sum of the squared errors over each epoch. An

epoch is defined here as the interval during which all training data samples are presented to the network. Thus, after presenting all of the training examples and calculating the difference between the networks output and the target vector in each case, the errors are summed and squared.

6.4.3 Training the Networks Using “qckprp”

There are a number of stages in preparing the raw FE data for use with the “qckprp” training program. They include concatenating the training “.tra” and “.val” validation files generated using the “concat.m” MATLAB script, appending a header to the concatenated data (which includes a number of parameters required by “qckprp”) before saving the augmented information in a “.csv” file.

The header in each case contains three records that declare certain parameters that include the number of inputs, outputs and hidden layer neurons. Considering the sheer number of files used in this work, an automated approach to the concatenation of training and validation files, appendage of header information and subsequent training of the networks was considered necessary and subsequently adopted.

4	11	1	95	12			
20000	1000	4	455	1	0	4	
101	102	103	104	0	0	0	0

Table 6-3: Sample neural network training file header

Consider table 6-3. The first record, row 1, contains fields for the number of inputs +1, followed by the number of hidden layer neurons +1, number of output neurons, number of training samples and the number of validation samples.

The second record contains information pertaining to the total number of epochs to train the network for, followed by the frequency of training and validation error output (expressed in epochs). The third field contains the number of columns in the subsequent data records from which the input and training data are to be selected. The fourth field contains a random number seed and this is followed by a weight initialisation indicator.

If this number is 1 then the initial weights are calculated using a random number generator. If it is zero, then the weights from the previous training run are used. The sixth field contains a learning rate (momentum) parameter, which is only relevant if the weights from a previous run are being used. If initial weights are being calculated, a zero is input here and a suitable learning rate value is also calculated by the program. The seventh field contains a Fortran unit number for output of the neural net outputs compared with the target outputs for all the training and validation data, if required.

The third and final record consists of a series of pointers telling “qckprp” where each of the inputs are located. For example the third record in table 6-3 means: “take (from the training file) inputs from columns 1, 2, 3 and the target output from column 4.” The use of 101, 102, 103 and 104 in the third record instead of 1, 2, 3 and 4 indicates that the data are to be used “as is” rather than being scaled to lie in the range 0 to 1. This option is used here, since the outputs were scaled during the data preparation process.

The “fastprp” korn shell script performs the functions described earlier: compiling a comma separated file with a header and concatenated training and validation data sets. It first counts the number of training cases there are in the current working directory. It then matches training and validation file pairs, counts the number of lines in each and then generates a header (as described above) in a comma separated “.csv” file. After this step is complete the relevant training and validation files are appended to the “.csv” file and the new training file is complete.

Since, the optimum number of hidden layer neurons is an unknown, one must approach the process of training in a methodical manner. On this subject, Masters⁶⁸ goes so far as to say:

“It is nearly impossible to specify an effective architecture [of a neural network] given the specifications of the problem. That must be done by experimentation.”

The “fastprp” algorithm allows the user to specify the initial number of neurons in the hidden layer. Thus, following advice laid out by Masters⁶⁸, one is able to start with a

very low number of hidden layer nodes and gradually increase the number in order to arrive at a near optimum number of nodes in the hidden layer.

In training a network one hopes to minimize the error between the target value and the neural networks predicted output. Since each training case has an individual stress profile, it is unrealistic to expect to obtain a uniform level of error across all possible cases. However, in this instance, statistics from British Energy's earlier work pertaining to the training of some, if not all, networks were available.

The "qckprp" program measures a neural networks performance using the following metrics.

1. Sum squared error: calculates the squared error across the entire training and validation set data.
2. Maximum error: the highest errors on the training and validation sets are reported as a percentage of the difference between the target and the output.

After considering British Energy's earlier work, it was decided to train each network down to a maximum error of 20%, for both training and validation sets. In those cases where this level of error was unobtainable, 5% was added to the "acceptable error level" and training restarted with a minimum number of hidden layer neurons. This was done iteratively until convergence was achieved, irrespective of the eventual percentage error on the training and validation sets.

The actual percentage error levels achieved for peak and profile cases successfully trained are presented in tables 6-4 and 6-5 respectively. The latter includes training and validation set error data for both the original training sessions by British Energy and for the current trained networks.

Case	# Neurons	Max. Average Error (%)
CPH	6	7.80
CPI	20	7.80
CMzbH	4	20.00
CMzbl	16	7.90
CMzrH	19	6.40
CMzrl	17	8.00
FPH	27	7.00
FPI	14	7.70
FMxbH	13	14.30
FMxbl	20	15.90
FMybH	13	8.10
FMyrl	9	8.00
FMzrH	11	7.20
FMzrl	12	9.40

Table 6-4: Peak cases successfully trained using “qckprp”

The notation used for the case identifiers in tables 6-4 and 6-5 is as follows:

C = Crotch section

F = Flank section

P = Pressure load case

M = Moment load case

x,y,z = Local axis around which moment load is acting

b = Moment applied to branch pipe

r = Moment applied to run pipe

H, I = Out-of-plane, in-plane stress component

00,03,06,1: Through thickness parameter, τ (profile cases only)

The columns headed T_ErrMax and V_ErrMax in table 6-5 represent the maximum training and validation set errors for each profile case. The numbers, when multiplied by 100, read as maximum percentage errors: 0.19 = 19%, 0.09 = 9%, for example.

It was decided that, in the event that “qckprp” is able to successfully train a given network structure to an “acceptable error level” on six occasions (out of 30 attempts)

starting from different positions in error-weight space each time, the network could be considered to be both stable, in respect of its performance, and successfully trained.

The number of successful runs completed (i.e. those that met the error target) is logged in the column headed “#success”, whilst the column “#neurons” represents the number of neurons required to train that network to the given error levels. Those cases that were not previously attempted by British Energy are marked with a “-”

As one will notice reading table 6-5, the error levels between the two sets of data (British Energy and the current work) show that maximum error levels are comparable for many of the cases. However, due to the differences in approach taken with respect to training, the current study produced slightly higher levels of error probably due to using fewer neurons during training. The method of incrementing neurons gradually whilst training as advocated by Masters⁶⁸ has the benefit of optimising a neural network’s architecture, but may result in higher (but still acceptable) levels of maximum error.

Case	Original Results - Profiles				New Results - Profiles			
	T ErrMax	V ErrMax	# neurons	# success	T ErrMax	V ErrMax	# neurons	# success
CPH00	0.05	0.19	48	4/4	0.18	0.13	15	6/6
CPH03	0.09	0.09	48	4/4	0.17	0.12	12	6/6
CPH06	0.13	0.10	48	4/4	0.19	0.12	12	6/6
CPH1	0.20	0.12	48	4/4	0.23	0.20	21	6/6
CPI00	0.22	0.26	48	4/4	0.24	0.22	39	4/6
CPI03	0.13	0.14	48	4/4	0.18	0.16	18	6/6
CPI06	0.16	0.28	48	4/4	0.27	0.22	30	6/6
CPI1	0.73	0.76	48	4/4	0.48	0.50	69	3/6
CMzbH03	0.10	0.14	48	4/4	0.12	0.08	33	6/6
CMzbH06	0.13	0.21	48	4/4	0.18	0.18	24	6/6
CMzbH00	0.12	0.14	48	4/4	0.24	0.13	33	6/6
CMzbH1	0.31	0.31	48	4/4	0.28	0.23	33	6/6
CMzbi03	0.13	0.15	48	4/4	0.12	0.12	18	6/6
CMzbi06	0.17	0.26	48	4/4	0.20	0.13	12	6/6
CMzbi00	0.16	0.21	48	4/4	0.18	0.13	12	6/6
CMzbi1	0.17	0.15	48	4/4	0.25	0.18	24	6/6
CMzrH00	-	-	-	-	0.18	0.18	24	6/6
CMzrH03	-	-	-	-	0.17	0.18	15	6/6
CMzrH06	-	-	-	-	0.20	0.14	30	6/6
CMzrH1	-	-	-	-	0.44	0.49	48	6/6
CMzrI00	-	-	-	-	0.26	0.25	57	6/6
CMzrI03	-	-	-	-	0.26	0.26	60	6/6
CMzrI06	-	-	-	-	0.25	0.24	66	6/6
CMzrI1	-	-	-	-	0.35	0.30	54	6/6
FPH00	0.11	0.13	48	4/4	0.18	0.15	24	6/6
FPH03	0.07	0.16	48	4/4	0.11	0.13	27	6/6
FPH06	0.13	0.18	48	4/4	0.18	0.15	18	6/6
FPH1	0.27	0.20	48	4/4	0.21	0.21	42	6/6
FPI00	0.12	0.22	48	4/4	0.21	0.22	42	6/6
FPI03	0.17	0.11	48	4/4	0.19	0.16	33	6/6
FPI06	0.18	0.26	48	4/4	0.26	0.21	54	6/6
FPI1	0.32	0.27	48	4/4	0.28	0.24	54	6/6
FMxbH03	0.12	0.19	48	4/4	0.18	0.18	27	6/6
FMxbH06	0.12	0.15	48	4/4	0.16	0.18	24	6/6
FMxbH00	0.12	0.39	48	4/4	0.24	0.14	54	6/6
FMxbH1	0.36	0.23	48	4/4	0.27	0.27	60	6/6
FMxbi03	0.13	0.07	48	4/4	0.23	0.10	12	6/6
FMxbi06	0.16	0.13	48	4/4	0.19	0.10	12	6/6
FMxbi00	0.13	0.14	48	4/4	0.23	0.18	36	6/6
FMxbi1	0.18	0.23	48	4/4	0.28	0.17	84	5/6
FMybH00	-	-	-	-	0.18	0.12	27	6/6
FMybH03	-	-	-	-	0.11	0.13	30	6/6
FMybH06	-	-	-	-	0.12	0.14	30	6/6
FMybH1	-	-	-	-	0.23	0.19	21	6/6
FMyrH00	-	-	-	-	0.15	0.14	18	6/6
FMyrH03	-	-	-	-	0.10	0.12	27	6/6
FMyrH06	-	-	-	-	0.15	0.10	24	6/6
FMyrH1	-	-	-	-	0.22	0.15	18	6/6
FMzrH00	-	-	-	-	0.12	0.11	24	6/6
FMzrH03	-	-	-	-	0.12	0.13	18	6/6
FMzrH06	-	-	-	-	0.12	0.12	27	6/6
FMzrH1	-	-	-	-	0.28	0.25	15	6/6

Table 6-5: Profile cases successfully trained using "qckprp"

The output from a successful training session is shown in table 6-6. For this training run, output was requested every 1,000 epochs and this is represented in the first column. The second and third columns (ERRSQ and ERRMAX) represent the sum squared error and maximum error on the training data respectively. The fourth and fifth columns (ERRSQE and ERRMXE) report the same quantities for the validation set data.

As can be seen in table 6-6, the errors on the training and validation sets are generally driven downwards until the maximum number of epochs is reached, i.e. 20,000 – an arbitrary number, arrived at from experience in training such networks and also taking into account the ever present pressure of time.

The ERRMAX and ERRMXE error metrics, when multiplied by 100, read as maximum percentage errors across the training and validation data sets respectively. In the above case, the network is successfully trained down to 13.1 and 10.8 maximum percentage errors on the training and validation sets after 20,000 epochs.

EPOCH	1000	ERRSQ=0.846	ERRMAX=0.232	ERRSQE=0.127	ERRMXE=0.160
EPOCH	2000	ERRSQ=0.686	ERRMAX=0.223	ERRSQE=0.0817	ERRMXE=0.126
EPOCH	3000	ERRSQ=0.605	ERRMAX=0.222	ERRSQE=0.0607	ERRMXE=0.121
EPOCH	4000	ERRSQ=0.535	ERRMAX=0.226	ERRSQE=0.0531	ERRMXE=0.120
EPOCH	5000	ERRSQ=0.494	ERRMAX=0.227	ERRSQE=0.0592	ERRMXE=0.134
EPOCH	6000	ERRSQ=0.328	ERRMAX=0.140	ERRSQE=0.0548	ERRMXE=0.131
EPOCH	7000	ERRSQ=0.270	ERRMAX=0.149	ERRSQE=0.0522	ERRMXE=0.147
EPOCH	8000	ERRSQ=0.246	ERRMAX=0.151	ERRSQE=0.0518	ERRMXE=0.150
EPOCH	9000	ERRSQ=0.233	ERRMAX=0.153	ERRSQE=0.0497	ERRMXE=0.149
EPOCH	10000	EERRSQ=0.224	ERRMAX=0.151	ERRSQE=0.0467	ERRMXE=0.147
EPOCH	11000	EERRSQ=0.217	ERRMAX=0.150	ERRSQE=0.0441	ERRMXE=0.145
EPOCH	12000	EERRSQ=0.191	ERRMAX=0.147	ERRSQE=0.0255	ERRMXE=0.106
EPOCH	13000	EERRSQ=0.178	ERRMAX=0.144	ERRSQE=0.0279	ERRMXE=0.100
EPOCH	14000	EERRSQ=0.173	ERRMAX=0.142	ERRSQE=0.0312	ERRMXE=0.104
EPOCH	15000	EERRSQ=0.169	ERRMAX=0.140	ERRSQE=0.0340	ERRMXE=0.108
EPOCH	16000	EERRSQ=0.165	ERRMAX=0.138	ERRSQE=0.0356	ERRMXE=0.109
EPOCH	17000	EERRSQ=0.163	ERRMAX=0.136	ERRSQE=0.0368	ERRMXE=0.110
EPOCH	18000	EERRSQ=0.160	ERRMAX=0.134	ERRSQE=0.0377	ERRMXE=0.109
EPOCH	19000	EERRSQ=0.158	ERRMAX=0.132	ERRSQE=0.0385	ERRMXE=0.109
EPOCH	20000	EERRSQ=0.156	ERRMAX=0.131	ERRSQE=0.0391	ERRMXE=0.108

Table 6-6: Sample output from the “qckprp” neural network training program

If all possible load/section/stress component permutations could be solved using the neural method, one would need to train 840 networks. In-fact 390 were eventually completed. Due to the nature of the data derived by the FE solver for certain load case/section combinations, a significant proportion could not be readily trained and early attempts to do so produced data of dubious quality.

For example, one would not expect to obtain useful results from the stress analysis for crotch section branch out-of-plane bending since, theoretically speaking, the stresses will be zero as the crotch section lies on the neutral axis in this case. However, the theoretical stress values of zero are not the true stresses a structure would carry in such circumstances and, inevitably, there will be small (and therefore insignificant in the overall scheme) membrane stresses in the section.

It is time consuming, very difficult (if not impossible), and ultimately not worthwhile trying to train cases such as crotch section branch out-of-plane bending to within 20% of near-zero stresses. To help illustrate the situation, fig. 6-15(a-d) show principal and von Mises stress contour plots (units are MPa) for crotch section unit branch out-of-plane bending. Table 6-7 summarizes the section/stress component/load case combinations successfully trained to “reasonable” error levels in the present work versus the original cases supported by PIPET.

		Crotch		Flank	
Key:		Hoop	In-Plane	Hoop	In-Plane
■ New cases	Pressure	■ X	■ X	■ X	■ X
X Original cases	Mxb			■ X	■ X
	Myb			■	
	Mzb	■ X	■ X		
	Mxr				
	Myr			■	
	Mzr	■	■	■	

Table 6-7: Matrix of successfully trained profile section/loadcase/stress combinations

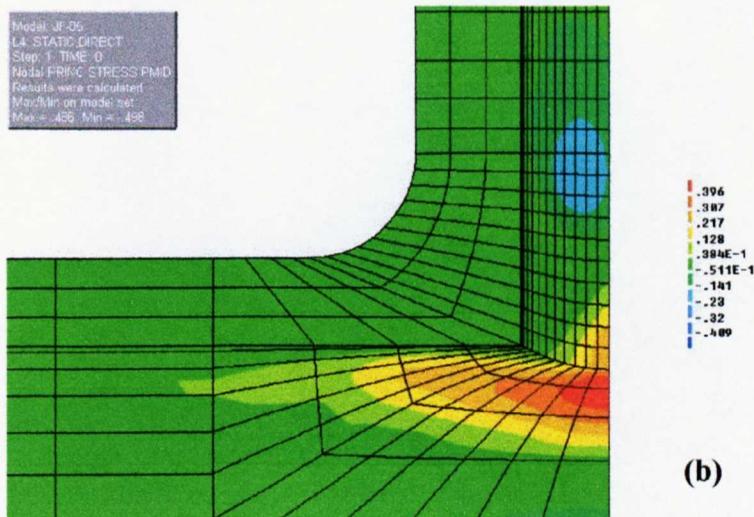
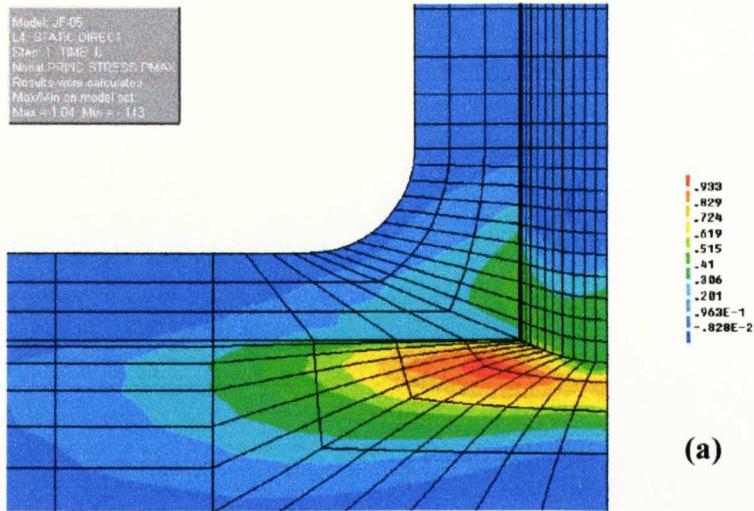


Fig. 6-15: (a) Crotch section maximum principal (b) middle principal stress contour plots

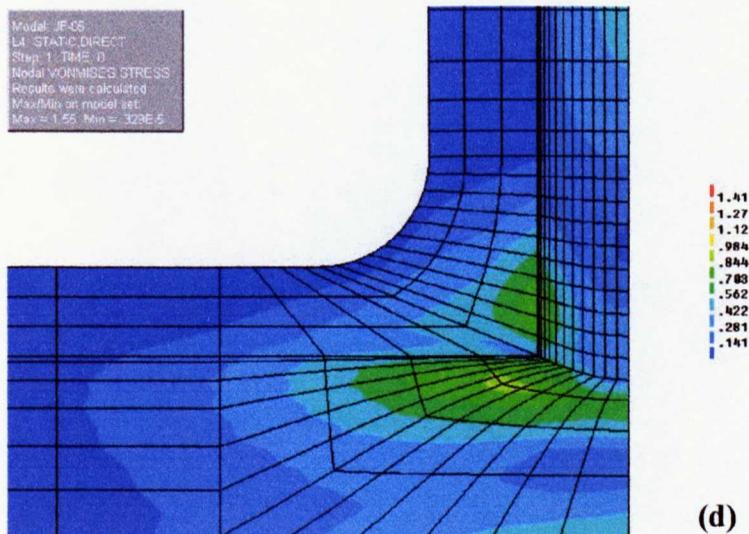
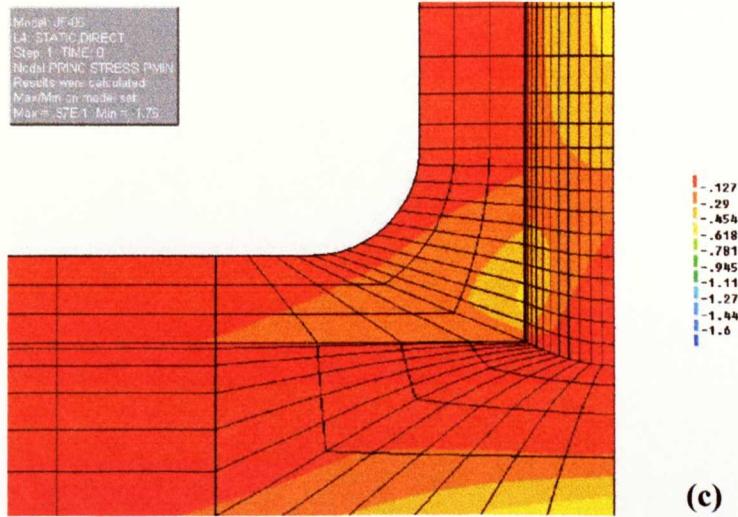


Fig. 6-15: (c) minimum principal and (d) von Mises contour plots for the branch out-of-plane bending loadcase

6.5 Evaluating PIPET's Performance

Evaluating the performance of a newly trained network is a critical step towards producing a fully functional and reliable neural network tool. It was decided that testing and validation of the networks herein required a number of stages to ensure confidence in the final system.

As reported earlier, the FE results obtained for network training were proven to be reliable by comparison with earlier work done at British Energy; two further layers of data testing and validation are now presented to instil confidence in the entire process, from FE analysis through to final operation of the implemented neural network system. In the first instance, the new weights were tested against 11 new finite element cases (see table 6-8), previously unseen by the trained networks and selected at random from within PIPET's geometry parameter range. Since the updated PIPET program now has a larger geometry domain according to British Energy's specifications, it is reasonable to assume that the majority of practical cases would lie within the new bounds, thus the arbitrary choice of geometries for the 11 test cases is justifiable. The second layer of validation was achieved by taking data from a number of sources available in open literature and comparing it against PIPET's neural predictions.

6.5.1 Updating the Weights File

The end-result of training any neural network is a set of weights and biases that can be used in a neural network to produce an output that will, hopefully, be a good estimate of the actual solution.

The weights file used in PIPET consists of an array containing the best three weights obtained (from the six successful training runs) for each component. Thus, for the out-of-plane stress component, pressure load case on the outside surface in the crotch section, there are three sets of weights and biases in the weights file, which are subsequently averaged when the program is run.

Once the best three weights in each case were identified, they were concatenated to form an ASCII weights file. This file was then passed to the program “pipetpw.solaris”, which converts the weights file from ASCII into a binary format and is subsequently put into PIPET’s working directory, where it can be read by the main PIPET program.

6.5.2 Testing PIPET

When PIPET is run with valid geometry and load parameters, for stresses on the surface of the junction, an output is generated that consists of a graph with four curves such as in fig. 6-16. Each curve represents either an upper or lower bound for both the outer surface (open symbols on the graph) and inner surface (closed symbols) stresses. These bounds are offset from the actual neural network result by $\pm 10\%$ - which effectively acts as an error band on the system’s prediction.

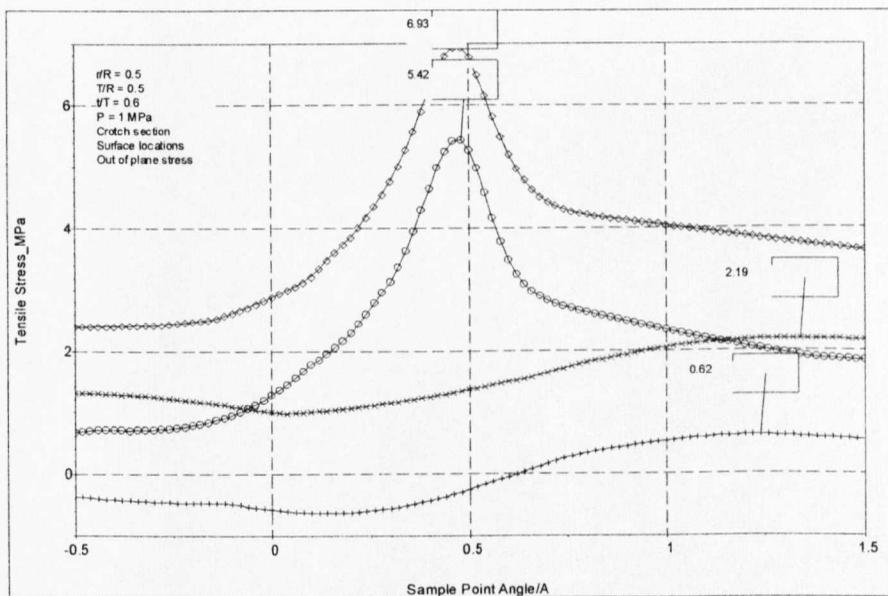


Fig. 6-16: An example of PIPET’s graphical output

Having updated the weights file to include the new data from the training process and completed FE test cases, testing commenced using the “pipet4.solaris” program, which directly compares PIPET’s output against the raw FE data. Any difference between the two is given a non-zero percentage error. A negative number indicates the network overestimates the FE stress value. Conversely, a positive integer indicates that the

system under-predicts value. The results from this test exercise are shown in table 6-8. Note that the OOP and IP abbreviations represent out-of-plane and in-plane respectively.

CASE ID	LOAD	CROTCH				FLANK			
		OOP STRESS OUT.	INN. INN.	IP STRESS OUT.	INN. INN.	OOP STRESS OUT.	INN. INN.	IP STRESS OUT.	INN. INN.
JF-02-04-04	PRESS	0	4	0	-16	-10	0	0	0
	IPLM	0	0	0	0				
	OPLM					0	0	44	29
JF-02-017-1	PRESS	0	0	0	0	0	0	0	0
	IPLM	0	0	1	0				
	OPLM					0	0	16	2
JF-03-005-025	PRESS	0	0	-5	2	0	0	0	2
	IPLM	0	0	0	0				
	OPLM					0	0	0	0
JF-03-009-06	PRESS	0	0	0	0	0	-9	0	2
	IPLM	0	0	0	0				
	OPLM					0	0	0	0
JF-05-03-07	PRESS	0	0	0	0	0	0	0	-3
	IPLM	0	0	-3	0				
	OPLM					0	0	29	2
JF-05-05-12	PRESS	0	0	0	-7	0	0	0	-5
	IPLM	0	0	0	0				
	OPLM					0	0	40	21
JF-06-02-07	PRESS	0	0	0	0	0	0	0	28
	IPLM	0	0	0	0				
	OPLM					0	0	39	0
JF-06-04-1	PRESS	0	0	0	-4	0	0	0	-16
	IPLM	0	0	0	0				
	OPLM					0	0	39	4
JF-075-005-09	PRESS	0	-2	0	0	0	0	0	23
	IPLM	0	0	0	0				
	OPLM					6	0	27	0
JF-075-01-11	PRESS	0	0	0	0	0	0	0	0
	IPLM	0	0	0	0				
	OPLM					0	0	42	0
JF-1-013-1	PRESS	0	0	0	0	0	0	-7	-7
	IPLM	0	0	-4	1				
	OPLM					0	0	3	-35

Table 6-8: Error levels produced by PIPET for 11 test cases

For the pressure load case, most cases have zero error on the FE result and even when there is a deviation from zero, the majority of cases show only a small negative error, thus overestimating the stress for that case. However, there are two cases that under-predict the inside surface in-plane stress levels in the flank section by 28% and 23% respectively – jf-06-02-07 and jf-075-005-09. In both cases, the stresses are negative

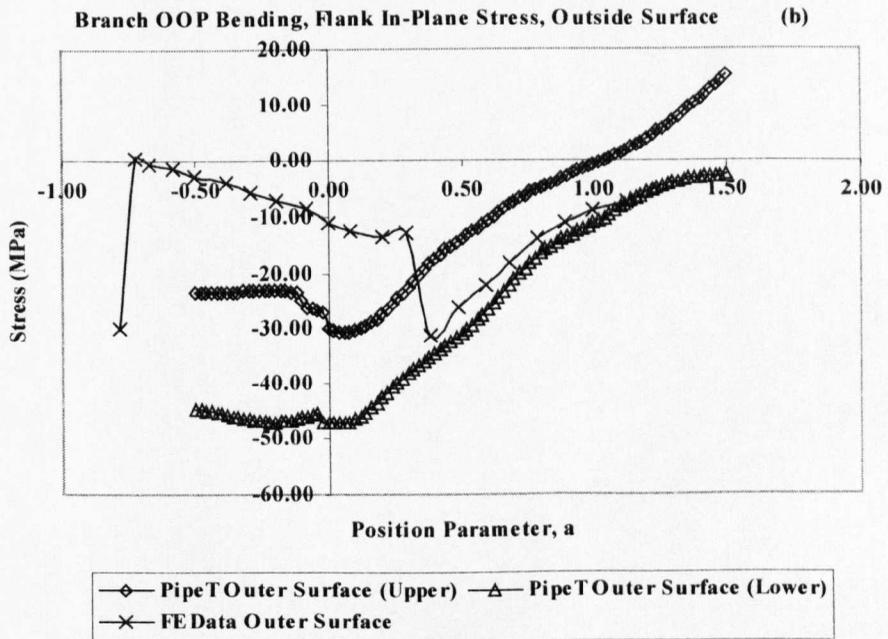
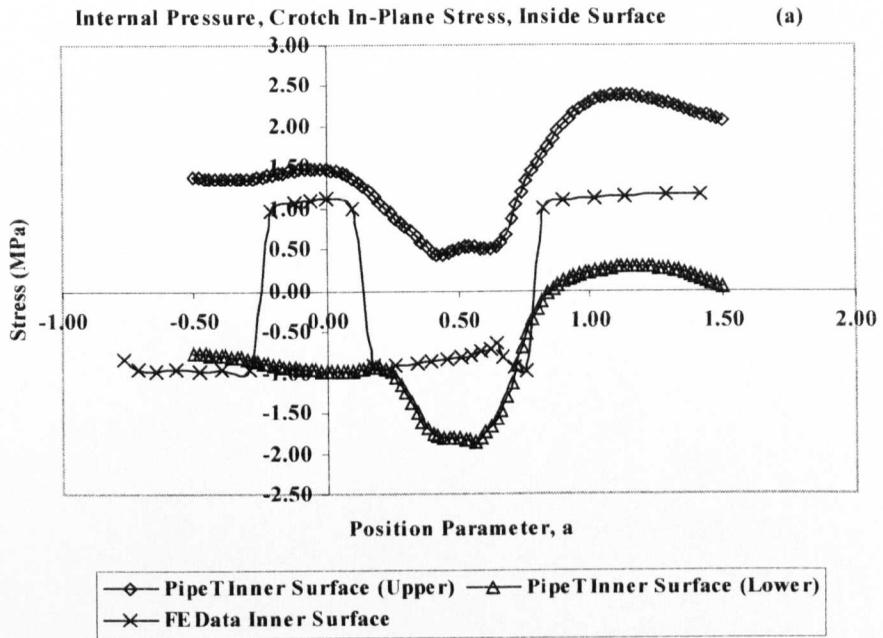
(compressive) and, perhaps, unlikely to be a factor when considering the structural integrity of the geometry under this load case alone. In addition, the case jf-075-005-09 deviates from the FE result at a point in the branch some distance from the junction intersection and would not, therefore be likely to be problematic vis-à-vis structural integrity.

Branch in-plane bending (crotch section only) is the most successful case considered here, with no significant overestimated stresses and just two cases in which the in-plane stresses were underestimated by 1%. Throughout training, validation and evaluation the in-plane bending proved an exceptionally well-behaved case and is certainly a candidate for reducing the “uncertainty level error band” to below the current 10%.

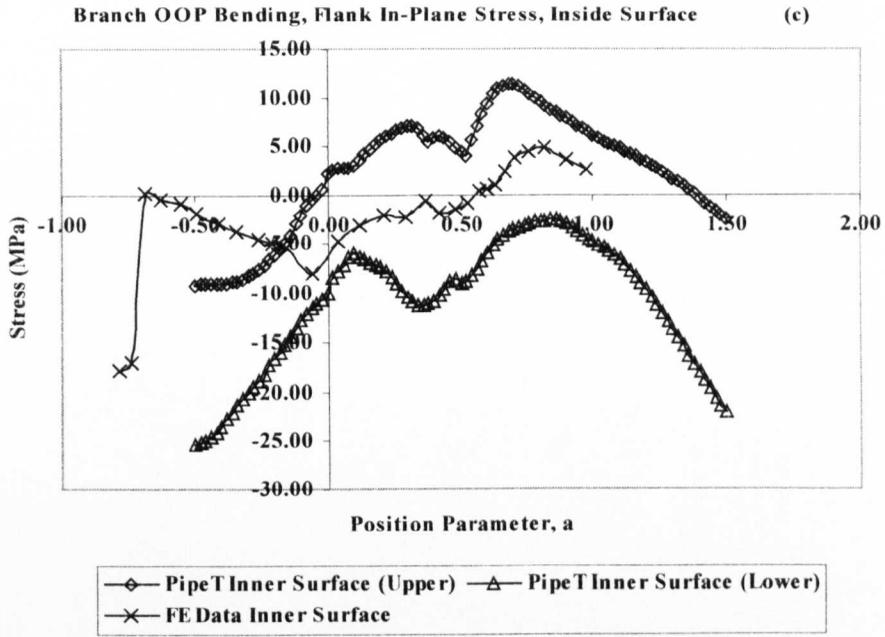
Flank section out-of-plane stresses for both pressure and out-of-plane bending also display excellent potential for reduced stress level error bands, since just three of the 44 cases produced errors of -10%, -9% and 6% for cases jf-02-04-04, jf-03-009-06 and jf-075-005-09 respectively.

Those cases having the highest errors in table 6-8 are the in-plane stresses for flank section out-of-plane bending. This is unsurprising, since branch out-of-plane bending proved a difficult case during the training phase. Both the outer and inner surface profiles for both in-plane and out-of-plane stress required a disproportionately higher number of hidden layer neurons to converge to reasonable levels of error than most other cases. Those cases where the difference between the actual FE result and the PIPET prediction was greater than 10% are presented in figs. 6-17 – 6-25. The legend text “Upper” and “Lower” in the charts below, refer to the upper and lower PIPET stress bounds, which are currently set at $\pm 10\%$ of the actual network prediction.

The vast majority of results for this test exercise have errors (with respect to the FE data) below 10%, suggesting that one of the project’s objectives could potentially be realized – that of reducing the error band within the main PIPET program to allow a more accurate estimate of stress levels. In particular, all load case and stress components in the crotch section demonstrate very low levels of error and are primary candidates for reduced stress level error bands within the PIPET program.



Figs. 6-17(a, b): Stress vs. angle position parameter - $r/R=0.2$, $T/R=0.4$, $t/T=0.4$



Figs. 6-17(c): Stress vs. angle position parameter - $r/R=0.2$, $T/R=0.4$, $t/T=0.4$

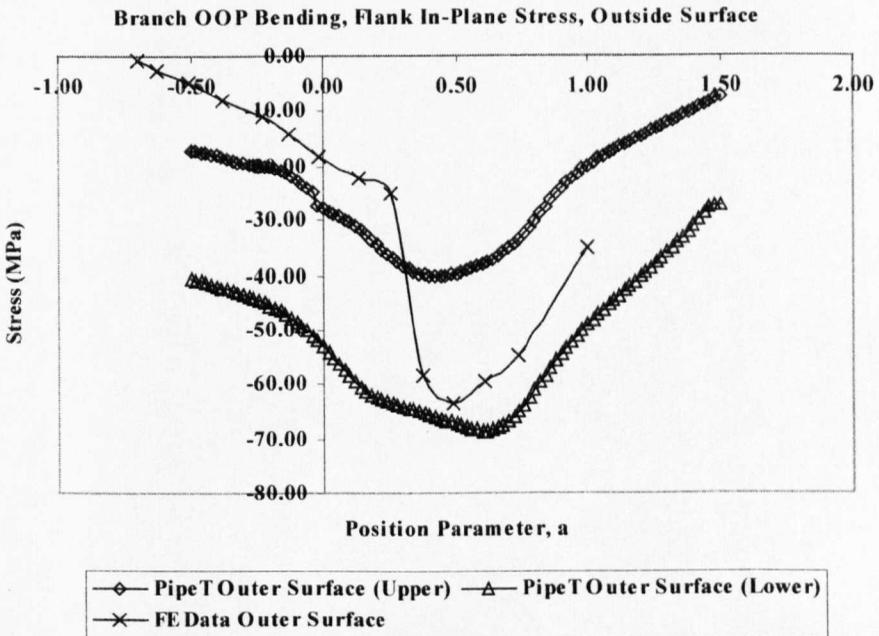


Fig. 6-18: Stress vs. angle position parameter - $r/R=0.2$, $T/R=0.17$, $t/T=1.0$

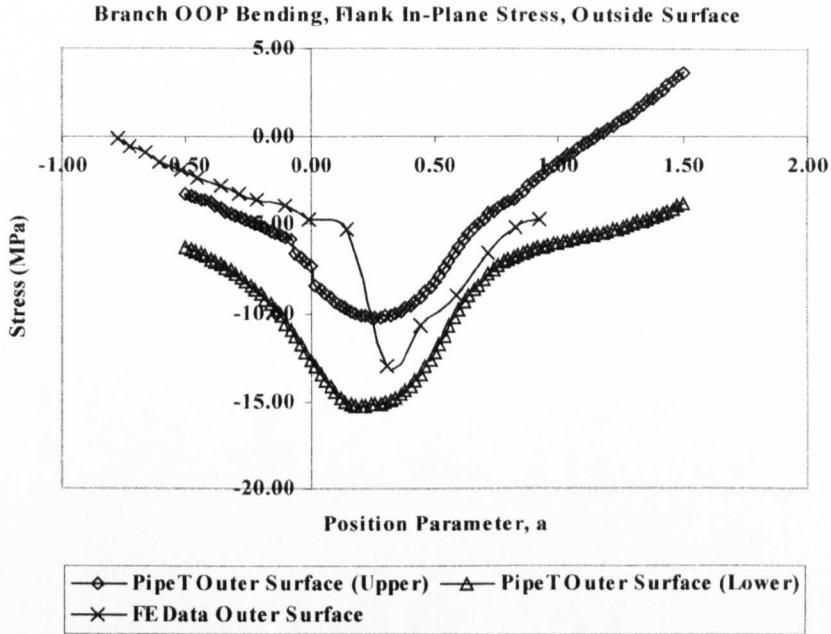


Fig. 6-19: Stress vs. angle position parameter - $r/R=0.5$, $T/R=0.3$, $t/T=0.7$

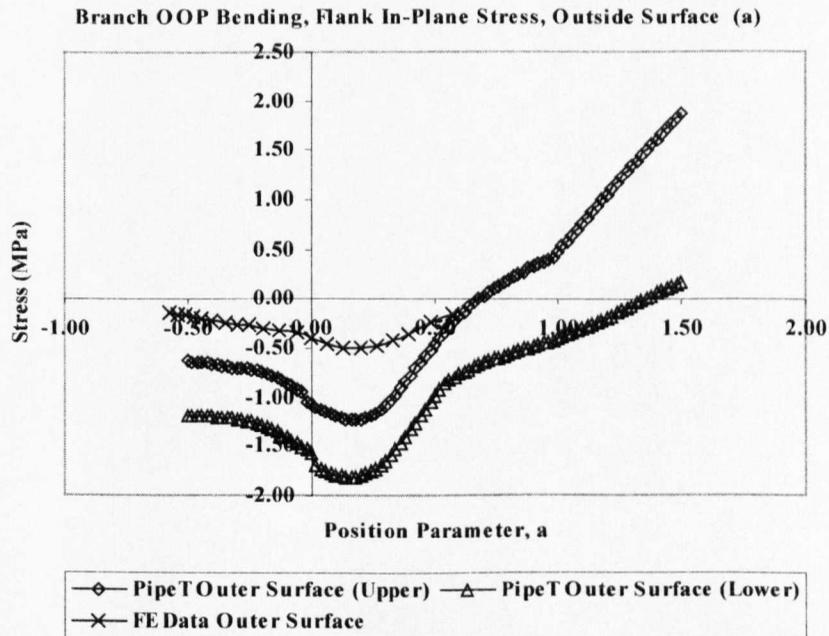


Fig. 6-20(a): Stress vs. angle position parameter - $r/R=0.5$, $T/R=0.5$, $t/T=1.2$

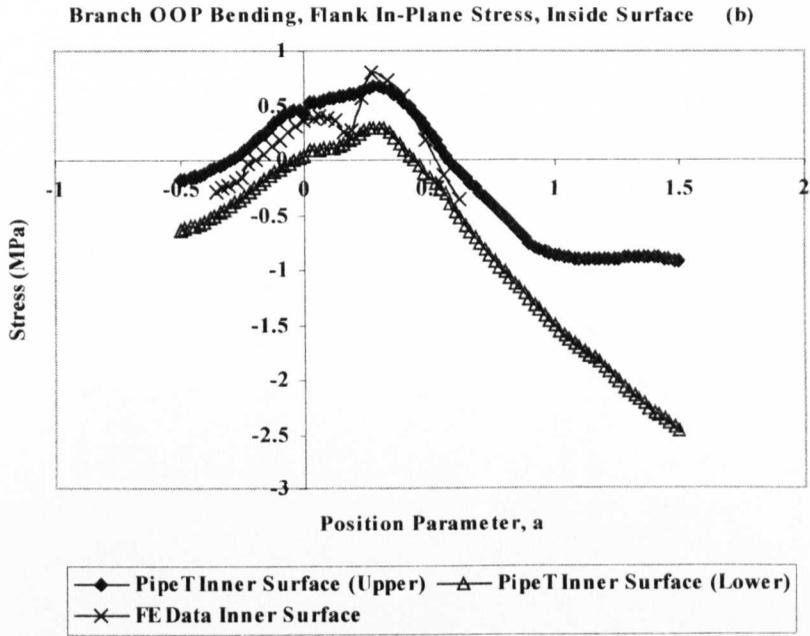


Fig. 6-20(b): Stress vs. angle position parameter - $r/R=0.5$, $T/R=0.5$, $t/T=1.2$

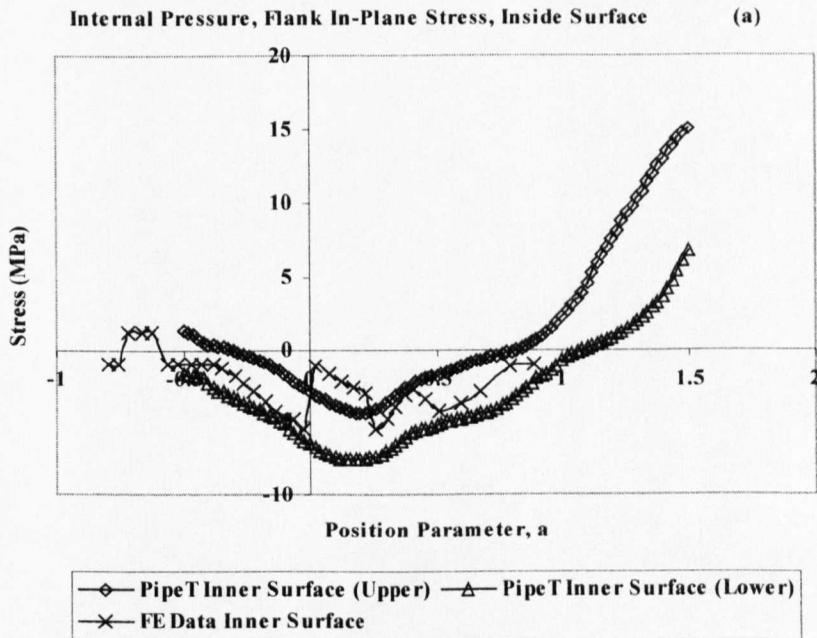


Fig. 6-21(a): Stress vs. angle position parameter - $r/R=0.6$, $T/R=0.2$, $t/T=0.7$

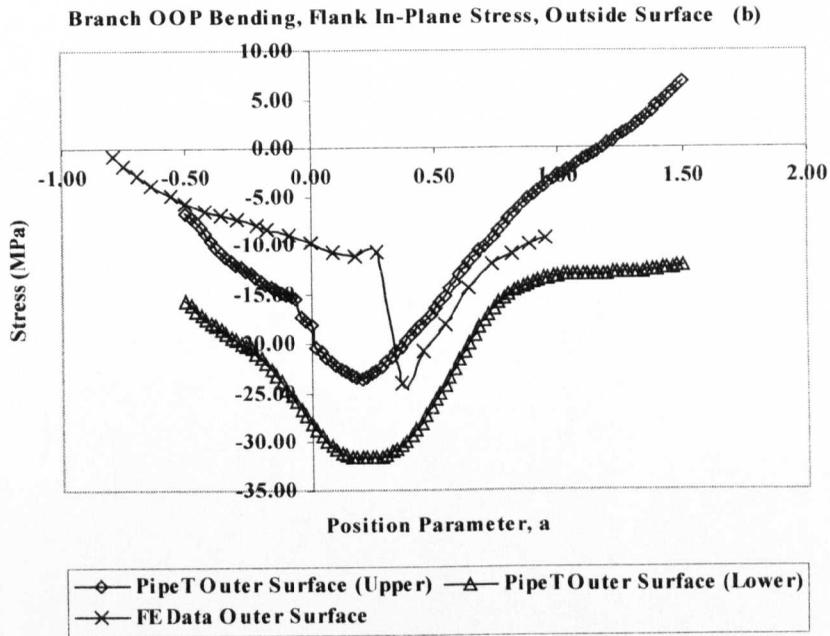


Fig. 6-21(b): Stress vs. angle position parameter - $r/R=0.6$, $T/R=0.2$, $t/T=0.7$

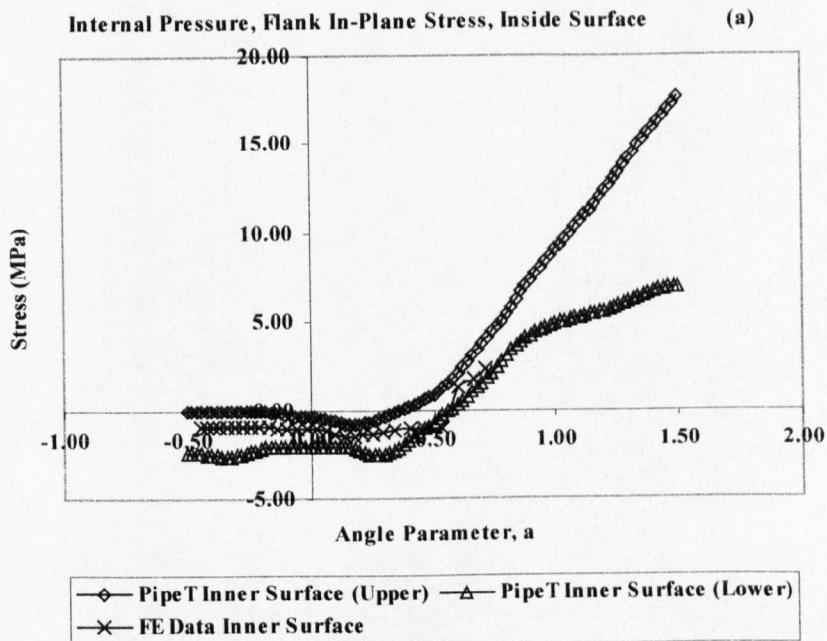


Fig. 6-22(a): Stress vs. angle position parameter - $r/R=0.6$, $T/R=0.4$, $t/T=1.0$

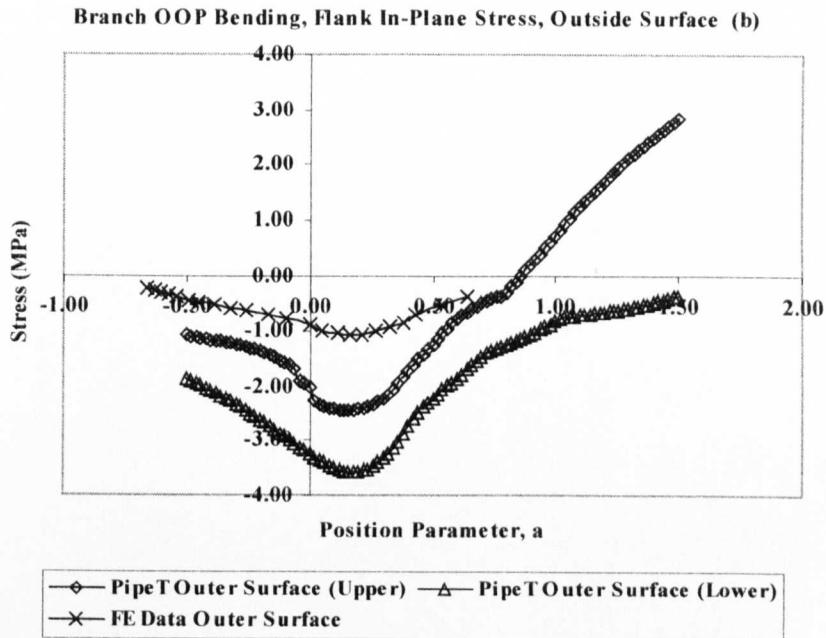


Fig. 6-22(b): Stress vs. angle position parameter - $r/R=0.6$, $T/R=0.4$, $t/T=1.0$

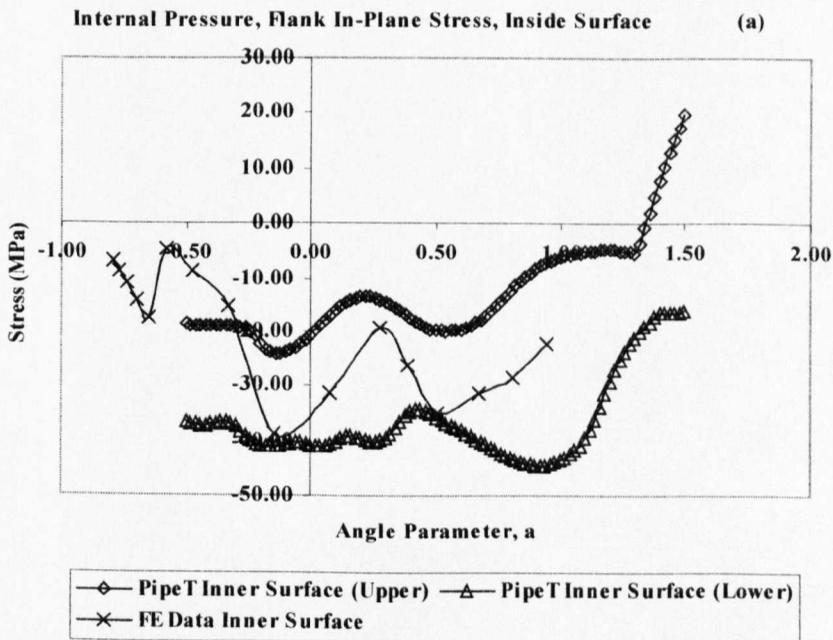


Fig. 6-23(a): Stress vs. angle position parameter - $r/R=0.75$, $T/R=0.05$, $t/T=0.9$

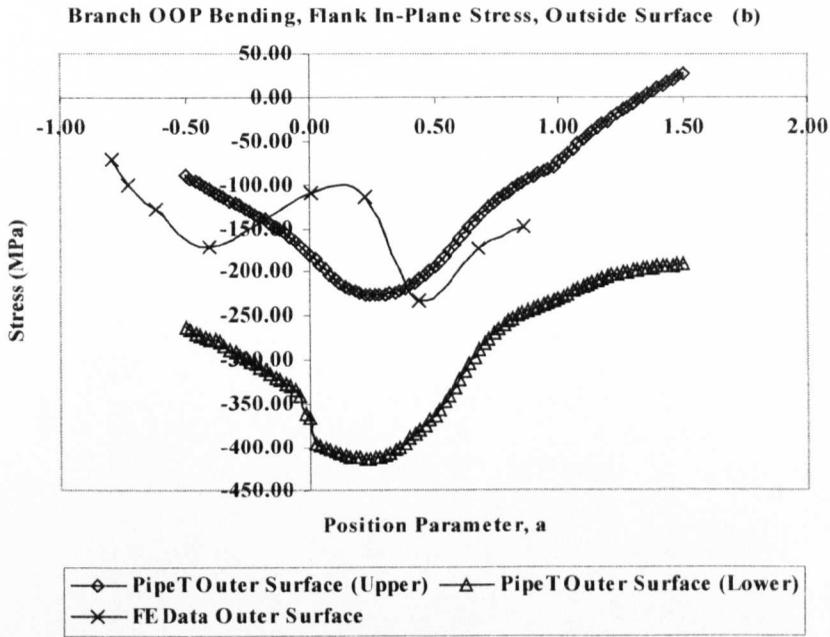


Fig. 6-23(b): Stress vs. angle position parameter - $r/R=0.75$, $T/R=0.05$, $t/T=0.9$

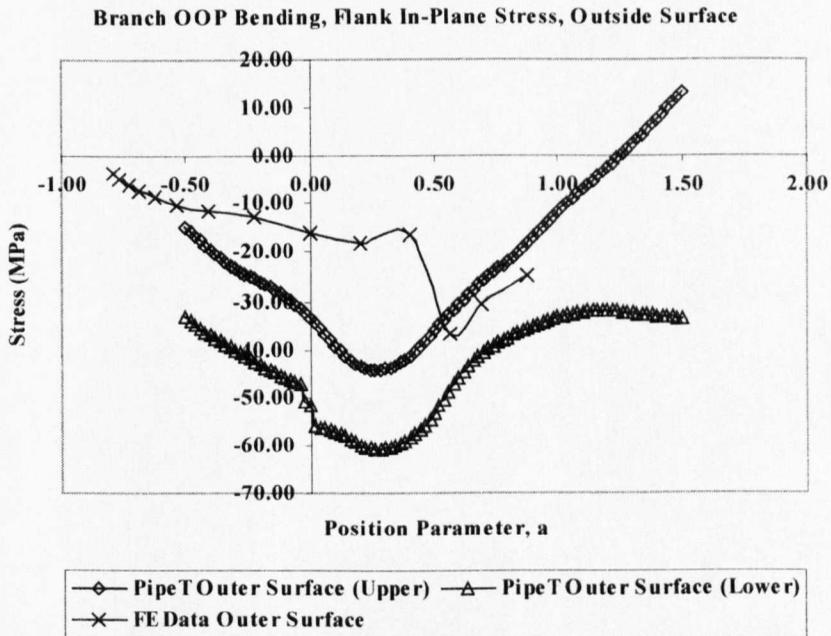


Fig. 6-24: Stress vs. angle position parameter - $r/R=0.75$, $T/R=0.1$, $t/T=1.1$

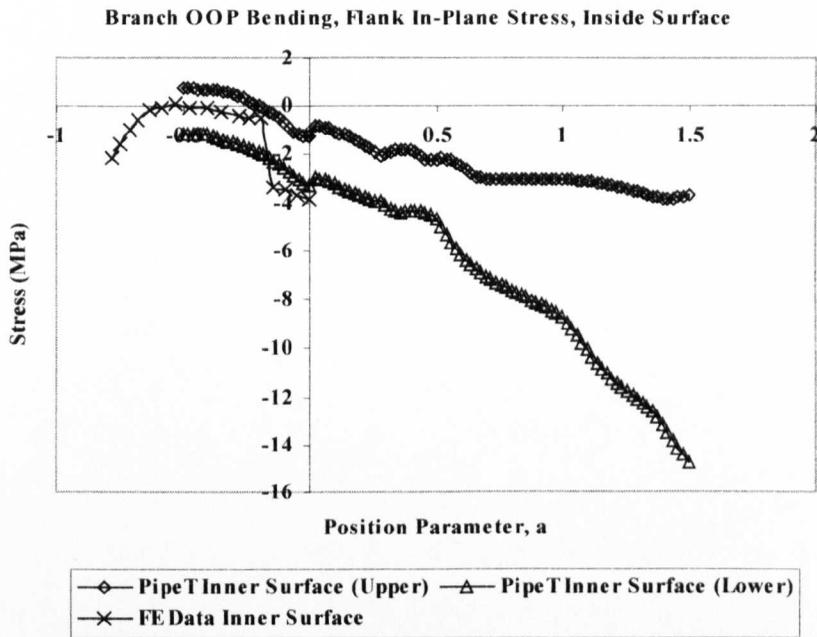


Fig. 6-25: Stress vs. angle position parameter - $r/R=1.0$, $T/R=0.13$ $t/T=1.0$

6.5.3 Evaluation Exercise: Pressure Load Case

Three further evaluation exercises are now presented in addition to the “blind” FE evaluation exercise previously discussed. Firstly though, recall that the stresses derived from the FE analyses earlier are defined as follows: the principal stress closest to the out-of-plane direct stress corresponds to the out-of-plane direction. The largest of the remaining two principal stresses (in absolute magnitude) is the in-plane stress.

It is important to remember this point as results and graphs in the following exercises are presented, since the comparisons drawn in this section are between the current data based on the principal stresses derived as described above and Tresca based effective stress factors. Therefore, and strictly speaking, only crotch section pressure cases are directly comparable, as the out-of-plane principal stresses will be equivalent to the Tresca stress for pressure load conditions. Other load cases in both crotch and flank sections are presented, but should not be interpreted as being directly comparable, since the stresses are not necessarily like-for-like.

The first evaluation exercise is a comparison of stresses done for a number of geometries based on the work of Moffat et al²³, which focussed on producing “Effective Stress Factors” for un-reinforced fabricated tee branch junctions, such as that illustrated in fig. 6-26, from a series of FE analyses.

The results herein build on an earlier comparison initially done by Fox and Mitchell⁷² for the original PIPET system. In their paper, Fox and Mitchell present their out-of-plane stress output from PIPET for crotch section pressure load case against that of Moffat et al. They demonstrated that the overwhelming majority of cases from the latter work fell within the uncertainty levels (upper and lower bound stress curves) of the PIPET system.

Fox and Mitchell found that, for the thicker walled cases, the data from Moffat et al fell within the PIPET uncertainty range. However, for thinner walled junctions, stresses were often overestimated, probably as a result of differences in the weld geometry, since PIPET assumes a weld radius equal to the branch wall thickness, whereas the weld dimensions in²³ were significantly greater than the thickness of the branch.

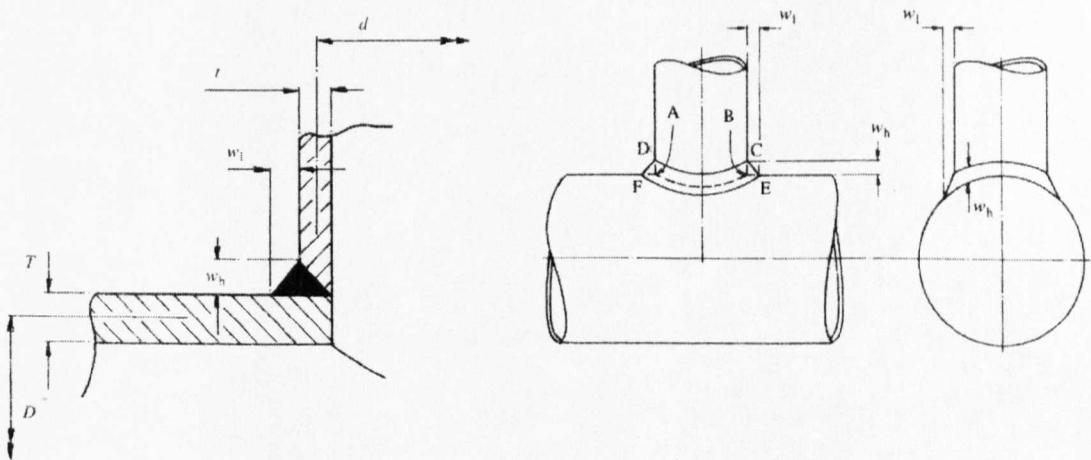


Fig. 6-26: Weld detail and notation used by Moffat²³, Hassan⁷⁴ and Ayob⁷⁵. After Moffat²³

The current work, when compared against the data from the former two, tends to show that whilst the lower limit bound remains almost the same as per the original PIPET data, the upper bound has moved closer to Moffat et al’s data. Thus, the overall

impression is of a narrowing of the uncertainty range for these cases. In a relatively crude approximation, the average difference between the original PIPET stress prediction on the Moffat et al data is -13% and +21% for the lower and upper bounds respectively. The new version of PIPET with updated weights bounds Moffat et al's data by -14% and +14% respectively.

Assumptions herein are as per those in⁷² – that the weld fillet radius, w , is the average of the weld dimensions in²³ and that the peak Tresca stress is equivalent to the maximum principle stress for this load case. Table 6-9 presents results for Moffat et al versus PIPET using the original neural weights versus PIPET using the updated neural weights.

As stated previously, the overall impression from the data shown in table 6-9 is a good one, with the updated PIPET upper bound stresses generally moving closer to Moffat et al's peak stress data.

This exercise showed that PIPET has the capability to safely estimate stresses in pipe junctions, even when the junction under consideration does not have the same weld dimensions as the original FE cases, upon which PIPET was trained. The fact that the three studies (Moffat et al, Fox and the current work) show good concordance is encouraging, however, it would be ill-advised to try and make any further capital from the results above, since all of the data presented is FE based and, therefore, an estimate of an estimate in any event.

Case Identifier	r/R	T/R	t/T	w/T	Moffat ²³	PIPET Stress		PIPET Stress	
						Range (Original)	Range (Updated)	Range (Original)	Range (Updated)
MF1A	0.200	0.222	0.20	0.23	16.1	13.0	19.0	13.3	16.7
MF1C	0.200	0.051	0.20	0.96	75.5	70.0	94.0	65.6	85.2
MF1E	0.350	0.222	0.35	0.35	19.0	16.0	21.0	15.6	20.1
MF1F	0.350	0.105	0.35	0.53	40.0	35.0	47.0	33.2	42.4
MF1G	0.350	0.051	0.35	0.61	86.4	83.0	122.0	75.6	103.0
MF1H	0.350	0.034	0.35	0.68	141.0	149.0	204.0	121.0	185.0
MF1I	0.500	0.222	0.50	0.47	21.5	18.0	23.0	17.7	22.6
MF1J	0.500	0.105	0.50	0.53	44.5	38.0	49.0	37.8	47.7
MF1K	0.500	0.051	0.50	0.78	93.1	86.0	132.0	86.3	114.0
MF1L	0.500	0.034	0.50	0.97	150.0	154.0	229.0	130.0	182.0
MF1M	0.650	0.222	0.65	0.59	23.0	19.0	24.0	19.0	23.9
MF1N	0.650	0.105	0.65	0.56	47.1	40.0	51.0	40.1	50.4
MF1O	0.650	0.051	0.65	1.04	97.1	90.0	136.0	94.3	122.0
MF1P	0.650	0.034	0.65	1.50	155.0	162.0	235.0	143.0	228.0
MF1Q	0.800	0.222	0.80	0.65	23.6	19.0	25.0	19.6	24.6
MF1R	0.800	0.105	0.80	0.74	48.2	40.0	53.0	40.5	51.1
MF1S	0.800	0.051	0.80	1.20	99.4	95.0	143.0	79.4	144.0
MF1T	0.800	0.034	0.80	1.50	158.0	171.0	246.0	145.0	231.0
MF2A	0.111	0.222	1.00	0.98	12.7	10.0	14.0	8.7	11.3
MF2B	0.158	0.105	1.00	1.57	26.4	21.0	31.0	20.9	26.6
MF2C	0.186	0.034	1.00	1.82	98.8	70.0	101.0	82.7	105.0
MF2E	0.278	0.222	1.00	0.99	14.6	12.0	17.0	11.3	14.6
MF2F	0.316	0.105	1.00	1.49	30.4	25.0	35.0	25.8	32.6
MF2G	0.333	0.051	1.00	2.25	67.3	53.0	77.0	64.2	82.2
MF2H	0.339	0.034	1.00	1.88	115.0	87.0	123.0	102.0	127.0
MF2I	0.444	0.222	1.00	0.94	16.9	14.0	19.0	13.5	17.2
MF2J	0.474	0.105	1.00	1.09	35.0	29.0	39.0	29.7	37.6
MF2K	0.487	0.051	1.00	1.71	77.8	63.0	91.0	74.1	95.5
MF2L	0.492	0.034	1.00	1.94	130.0	106.0	150.0	114.0	148.0
MF2M	0.611	0.222	1.00	0.68	19.1	15.0	21.0	15.6	19.6
MF2N	0.632	0.105	1.00	0.86	39.6	33.0	43.0	33.0	41.8
MF2O	0.641	0.051	1.00	1.51	87.6	74.0	107.0	83.1	107.0
MF2P	0.644	0.034	1.00	1.94	144.0	127.0	179.0	125.0	168.0
MF2Q	0.778	0.222	1.00	0.81	21.0	17.0	23.0	17.5	22.0
MF2R	0.789	0.105	1.00	0.93	43.6	36.0	48.0	36.0	45.2
MF2S	0.795	0.051	1.00	1.57	95.3	86.0	124.0	90.7	119.0
MF2T	0.797	0.034	1.00	1.88	154.0	151.0	210.0	149.0	194.0

Table 6-9: Crotch section pressure load case peak stress comparison Moffat²³ vs. PIPET original weights vs. PIPET Updated weights

The second and third verification exercises present ESF data from the PhD theses of Hassan⁷⁴ and Ayob⁷⁵ respectively. Both authors completed a finite element study of an un-reinforced tee junction (of different dimensions) and considered stresses for pressure only in the case of Hassan and for pressure plus six moment loads in the case of Ayob. The geometry presented in each case had a “flat” (as opposed to “circular”) weld profile as per Moffat et al²³ and fig. 6-26 herein.

Hassan considered the effect of elliptical shaping of nozzles in cylindrical pressure vessels and performed both experimental and finite element analysis for a branch junction having the following geometric parameters:

$$d/D = 0.2$$

$$D/T = 7.97$$

$$D_i = 209\text{mm}$$

$$d_i = 42\text{mm}$$

$$T = 30\text{mm}$$

$$t = 6\text{mm}$$

Where d and D are the mean diameters of the nozzle and run pipe and the length of the run and branch pipe limbs were 234.5mm and 210mm respectively.

Hassan modelled a quarter of the geometry and meshed the model using 20-node brick elements, with two elements through the thickness of the run and branch pipe walls and six elements around the circumference of the weld.

For the crotch section out-of-plane stresses, in particular the outside surface stresses, the Hassan data appears well included by the bounds of the PIPET prediction, with the updated (newer) PIPET weights showing “tighter” bounds than the original weights – as noted earlier in this section when comparing the Moffat et al²³ data with PIPETs output. Fig. 6-27 (b) shows a much closer fit of the Hassan data by the PIPET program, in particular the updated weights produce an upper bound curve that is very much like the Hassan curve.

The flank profile results (figs. 6-27c, d) are a little less encouraging. The data for PIPET looks slightly out of phase with that of Hassan for outside surface stresses, the former showing a peak at $a \approx 0.45$, whilst the latter peaks at $a = 0$. In addition, the original weights appear to perform better than the newer weights – the magnitude of the (upper) maximum in the former case is closer to that of Hassan. However, both PIPET upper bound weight plots from $0.3 \leq a \leq 1.5$ appear to show at least some concordance with the Hassan data for this case.

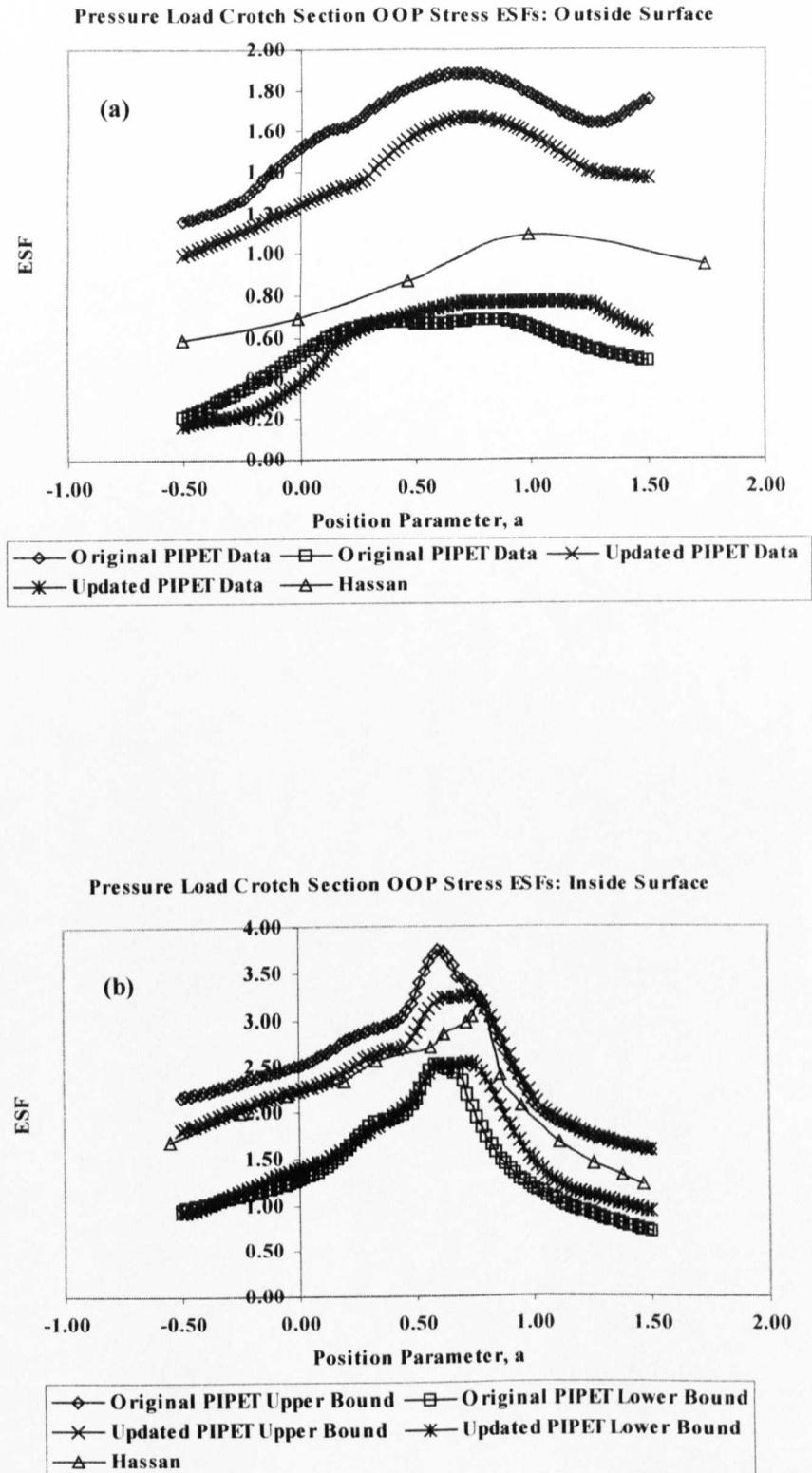


Fig. 6-27(a, b): Crotch section ESF comparison – Hassan vs. PIPET

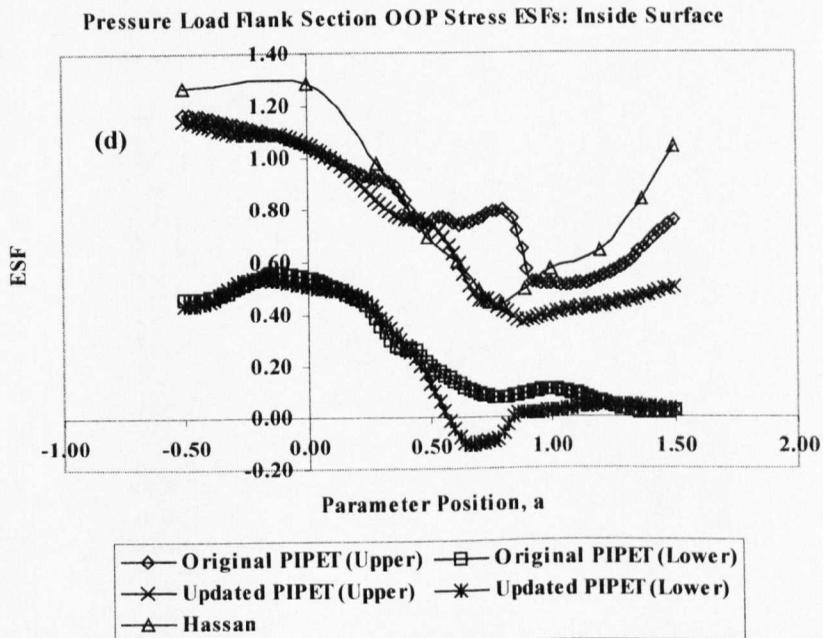
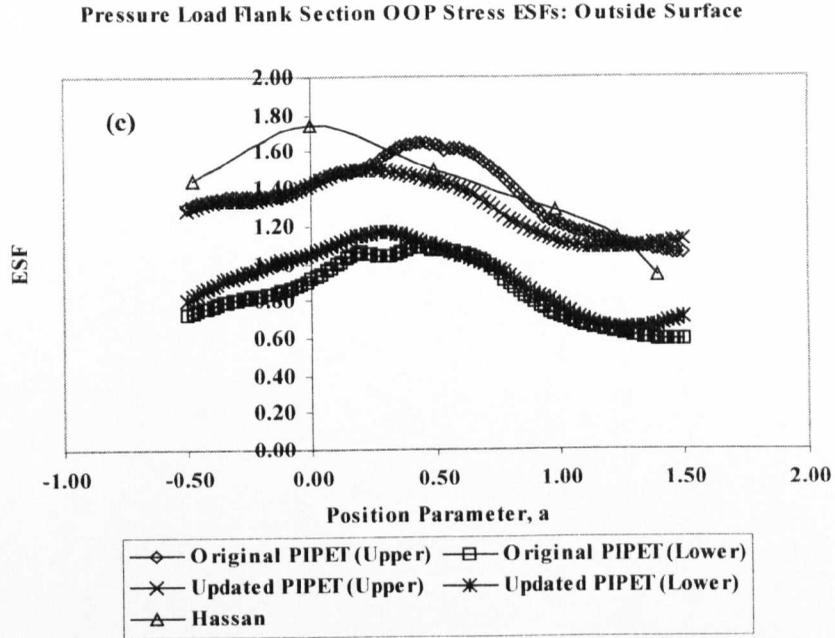


Fig. 6-27(c, d): Flank section ESF comparison – Hassan vs. PIPET

The inside surface stress plots display similar characteristics to that of the outside surface plots in that the peak stress location is once more at $a = 0$ for the Hassan data, whereas PIPET predicts the peak at $a = -0.5$ (both original and updated weights). The forms of the updated weight upper bound curve and the Hassan curve after $a \approx 0.3$ show a similar downward trend to $a \approx 0.75$, at which point both start to increase in stress value. The original PIPET upper bound curve appears to “wobble” a bit more than the others and demonstrates an upward not downward trend when $0.3 \leq a \leq 0.75$, but follows a similar path to the other two upper bound curves from $1.0 \leq a \leq 1.5$.

The above graphs show that some reliability is to be found for UFT junctions having a “flat” rather than “round” weld profile – particularly for the crotch section out-of-plane stresses for this load case. Whilst the flank section stress profiles do show some similarities much more FE validation would be required for UFTs with both “flat” and “circular” weld profiles for any meaningful conclusions to be drawn for the UFTs having a weld profile as per Moffat et al²³, Hassan⁷⁴ and Ayob⁷⁵.

Ayob⁷⁵ considered the interaction of pressure and bending loads on a UFT using the finite element method. The branch junction having the following geometric parameters:

$$d/D = 0.8$$

$$D/T = 20$$

$$D_i = 200\text{mm}$$

$$d_i = 160\text{mm}$$

$$T = 10\text{mm}$$

$$t = 10\text{mm}$$

Where d and D are the mean diameters of the nozzle and run pipe and the length of the run and branch pipe limbs were 800mm and 640mm respectively.

Ayob modelled the full geometry and meshed the model using 20-node brick elements, with just one element through the thickness of the run and branch pipe walls and 48 elements around the circumference of the weld.

For the pressure load case, both out-of-plane and in-plane stresses were compared for the crotch section profiles only. Comparison of the pressure load out-of-plane stresses was also done for the flank section profile.

The crotch out-of-plane and in-plane stress results presented in figs. 6-29(b, d) for the inside surface show excellent correlation with the Ayob data, with practically no difference between the original PIPET weight data and the updated version. The inside surface data is perhaps more significant than the outside surface data for this case, since one would expect the peak stress for the crotch section to occur somewhere on that surface, close to the crotch corner.

The crotch outside surface out-of-plane and in-plane stresses in figs. 6-29(a, d) do not compare as well as those for the inside surface. It is felt that this is partly because there was less data for the outside surface and also that the stresses produced by PIPET would be higher in any case, since the strengthening effect of the weld would not be as pronounced as it would be in Ayob's model. The weld dimensions for the latter would exceed that of its circular counterpart used for the FE models herein, where a weld radius equal to the branch wall thickness is assumed. Therefore, one would expect that, for a thin-walled junction such as this, stresses would be overestimated by PIPET due to differences in the weld geometry.

The flank results in figs. 6-30(a, b) for pressure, show that the Ayob results fall largely within the bounds of both sets of PIPET weights, although the Ayob curves do not closely resemble PIPET's. It is interesting to observe that, once more, stresses for this differing geometry are included to a significant extent within PIPET's bounds.

It is noted that, when the position parameter exceeds unity, PIPET's prediction and Ayob's data significantly diverge. This is almost certainly a result of PIPET's limitations for large diameter ratio junctions, such as in this case, where $d/D = 0.8$. Due to the nature of the indexing algorithm and the profile of the flank section for such diameter ratios, data is unavailable for alpha positions around and in excess of unity. In order to graph across the full alpha position parameter range, PIPET is forced to extrapolate beyond this bound.

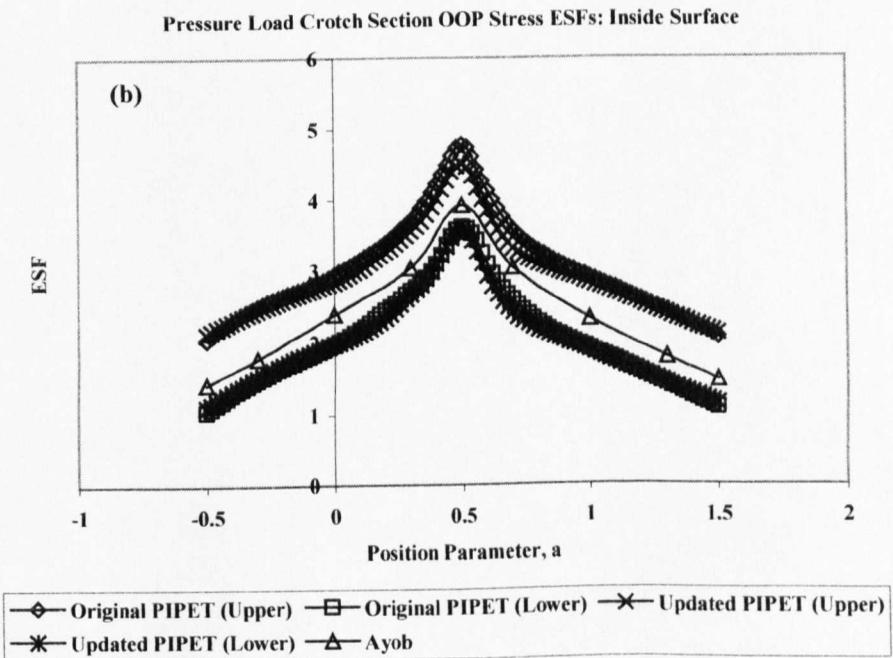
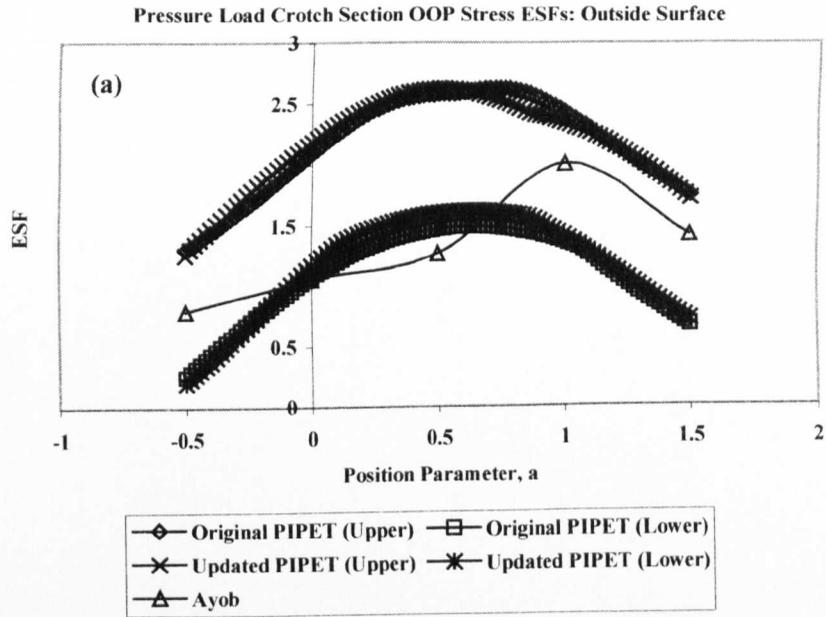


Fig. 6-29(a, b): Crotch section out-of-plane stress ESF comparisons Ayob vs. PIPET

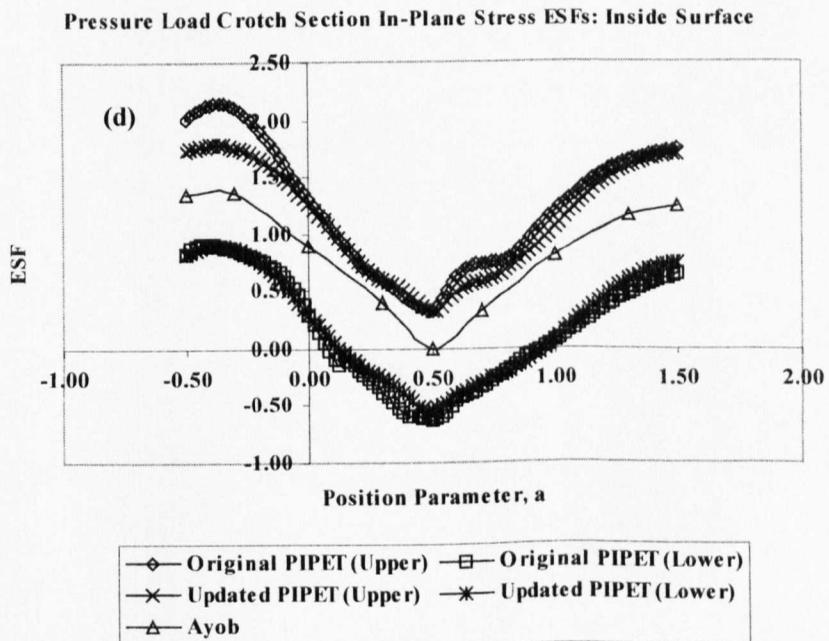
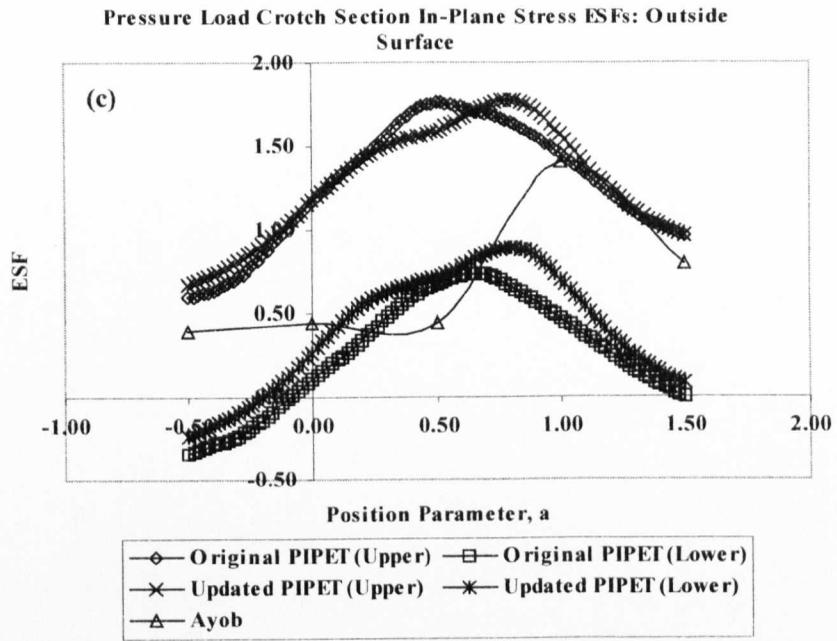


Fig. 6-29(c, d): Crotch section in-plane stress ESF comparisons Ayob vs. PIPET

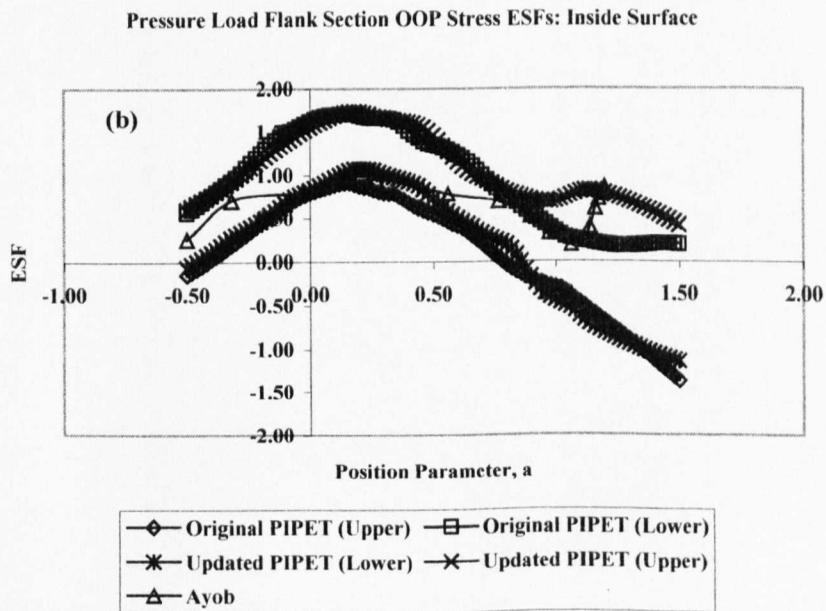
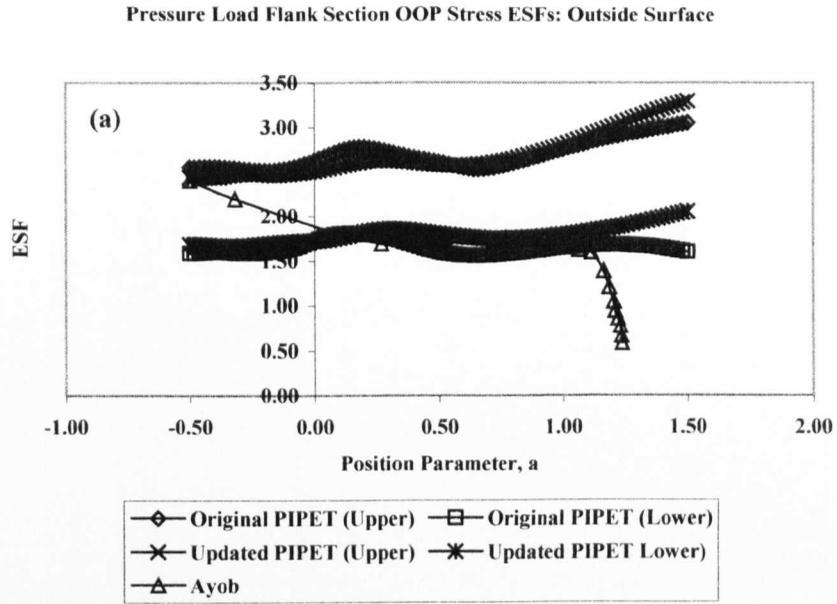


Fig. 6-30(a, b): Flank section out-of-plane stress ESF comparisons Ayob vs. PIPET

6.5.4 Evaluation Exercise: Bending Moment Load Cases

Finally, some data for branch moment load cases is presented for PIPET vs Ayob. The two load cases under consideration here are branch in-plane bending (crotch section only) and branch out-of-plane bending (flank section only). Unlike the pressure load case in the crotch section, where the out-of-plane stress is considered equivalent to the Tresca stress, this will not be the case for moment loads. In attempting to redress this imbalance, the assumed out-of-plane and in-plane PIPET data was converted from principal to von Mises stress using the following equation:

$$\text{von Mises} = \sqrt{\frac{\sigma_1^2 + \sigma_2^2 + (\sigma_1 - \sigma_2)^2}{2}} \quad (6-1)$$

Where: σ_1 = OOP principal stress, σ_2 = IP principal stress

Fig. 6-32(a, b) are plots of crotch section branch in-plane bending, for the outside and inside surfaces respectively. Whilst the outside surface curves for the original PIPET and Ayob data (fig. 6-32a) show similar profiles, the ESF values are considerably divergent and are comparable only in the range $-0.5 \leq a \leq 0$.

The inside surface ESF curves in Fig. 6-32b demonstrate much greater levels of concordance than the outside surface plots, mentioned above. However, there still remains a significant difference in ESF levels between PIPET and Ayob, with the latter data largely falling below the lower bound curves predicted by PIPET.

The flank section branch out-of-plane bending cases in figs. 6-33 a and b show even less correlation with their PIPET counterparts than the branch in-plane bending cases. For this load case, it would be unwise to use PIPET in predicting stresses for the “flat” UFT geometry profile. Both the inside and outside surface profile curves for the Ayob data are much flatter than the PIPET equivalents, this is perhaps partly a function of the weld detail, with more reinforcement available in Ayob’s model.

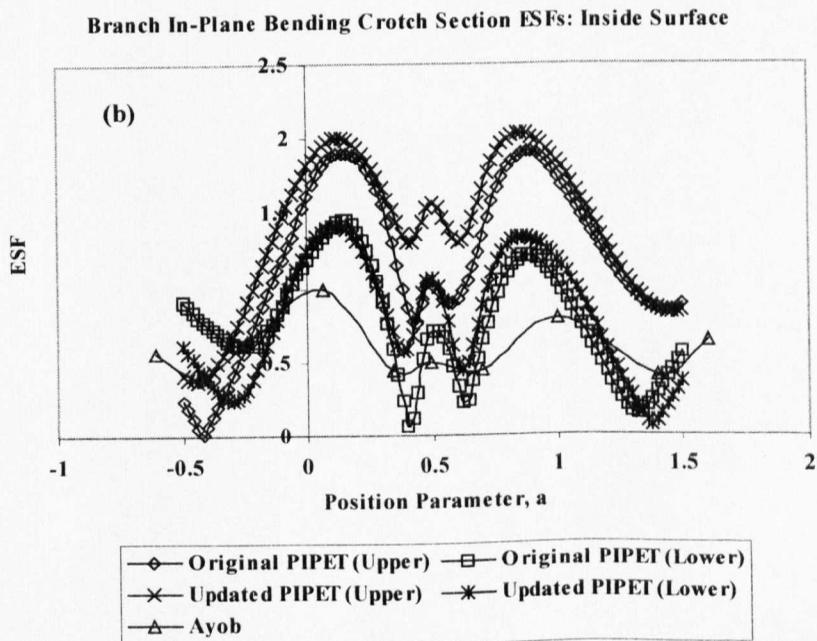
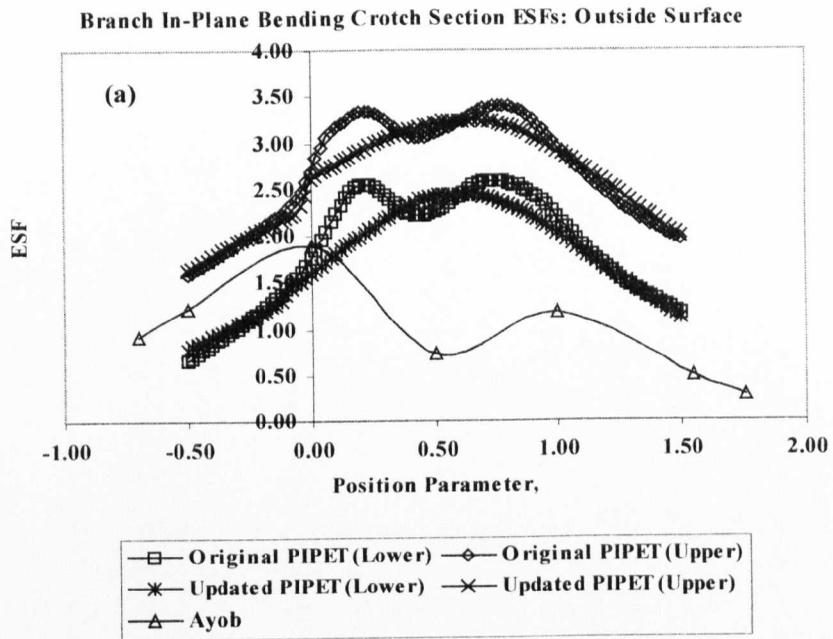


Fig. 6-32(a, b): Crotch section branch in-plane bending ESFs comparisons Ayob vs. PIPET

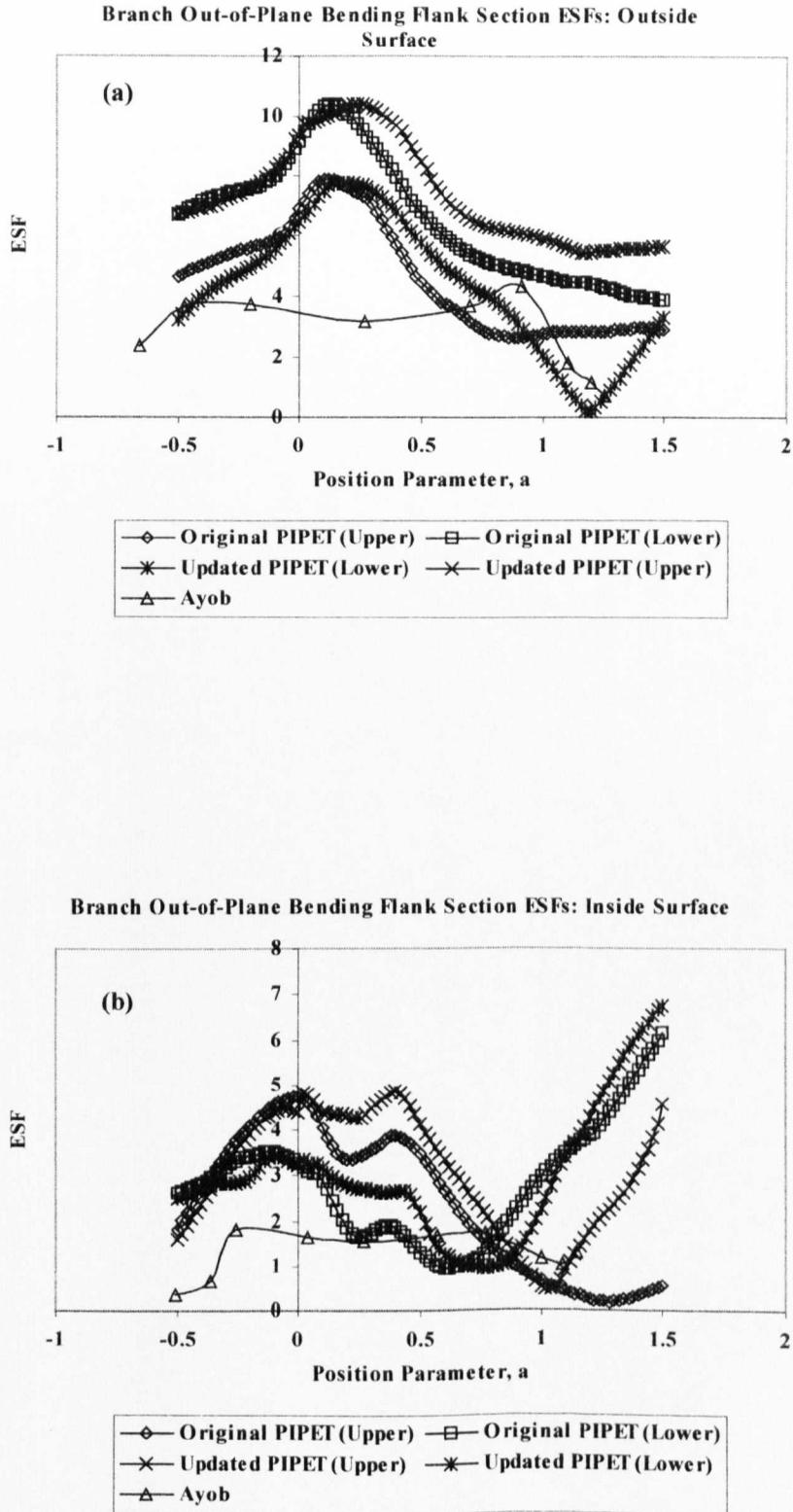


Fig. 6-33(a, b): Flank section branch out-of-plane bending ESF comparisons Ayob vs. PIPET

In addition to the obvious inequality between the stresses being compared, another possible contributing factor for the widely divergent results is that the meshes used herein are somewhat different. The current FE meshes had three 20-node elements through the wall thickness of both branch and run pipes to enable estimation of the through-thickness stresses within PIPET (see fig. 6-31), whereas the Ayob model had just a single 20-node element through the thickness. In addition, the current author's mesh convergence study suggested at least two elements would be necessary in order to effectively assess the stresses for the range of geometries presented here.

Ayob's justification for using a single 20-node element through the thickness was based on his review of papers by Moffat et al²³, Mwenifumbo⁴⁰ and Natarajan et al⁷⁶. However, the authors⁷⁶ noted that poor results were obtained for out-of-plane stresses until the mesh in the area of interest was sufficiently refined. The current author suspects that, due to having just a single element through the thickness, it is possible that the stresses presented by Ayob are not as well defined as those predicted by PIPET.

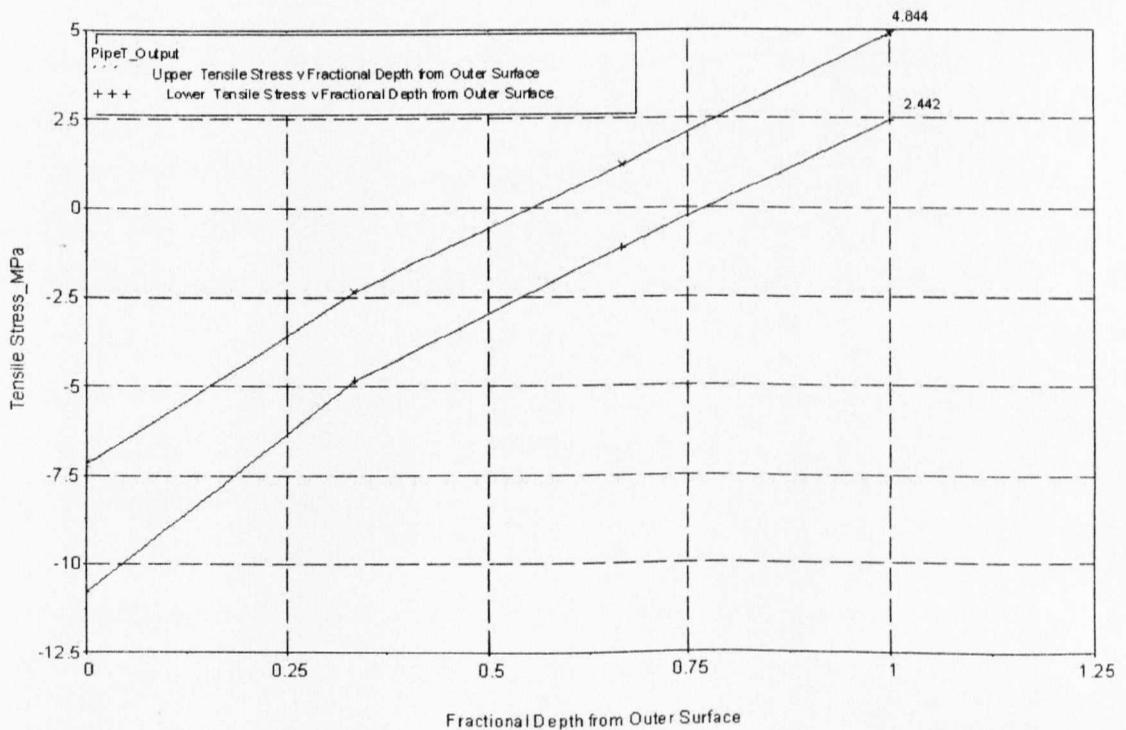


Fig. 6-31: PIPET flank in-plane through-thickness ESFs at $a=0$, for $r/R=0.8$, $R/T=20$, $t/T=1.0$

One is finally drawn to the conclusion that, there is no single reason for the disparity between the Ayob and PIPET results and the differences are probably due to a combination of the stresses not being totally comparable, different geometry profiles and number of through-thickness elements.

Recent research appears to advocate the use of more than a single element through the thickness for pipe junction analysis in order to properly develop through-thickness stresses. In his Ph.D. thesis on cracked pipe junctions, Lynch⁷⁷ commented:

“The typical density required for internal pressure loading has previously been established at Liverpool by Yahiaoui et al.⁷⁸ for forged tee junctions. In this previous study, four elements through the thickness were used in order to be consistent with the cracked work, although it was found that one element through the thickness gave ‘adequate accuracy’. In addition, a draft report⁷⁹, providing guidelines for FE modelling of intersections, recommends that a minimum of two elements through the thickness are used near the intersection area.”

Lynch performed his own mesh convergence study for one of the branch junctions with the parameters $d/D=0.5$, $D/T=10$ and $t/T=1.0$ by varying the number of elements around the circumference of the weld between 48 and 80. He also altered the number of elements through the thickness between one and five and used four to ten elements through the height of the weld. After the study was completed, he noted:

“Despite these large differences in the mesh density, and hence the total number of elements used, the maximum variation (measured from the limit pressure for the finest mesh) was just 1.1%.”

Eventually, Lynch decided that the number of through-thickness elements for $D/T = 5$, $D/T = 10$, 20, 30 and $D/T > 30$ should be five, three and two respectively.

It would appear that, perhaps due to the advent of ever-greater computing power, the number of through-thickness elements are no longer being limited to just one and an acceptance of “adequate accuracy”. Instead multiple elements though the thickness for

pressure vessel finite element analysis is more likely. In any event, the number of through-thickness elements used herein was necessarily three, since this is the number of elements required by PIPET in order to be fully functional.

6.6 Closure

This chapter described work undertaken with British Energy to extend their current capability of stress prediction and estimation in un-reinforced fabricated tee junctions using a neural network approach.

The current extent of British Energy's neural network program PIPET was first presented before examining the scope of the project in more detail. The programming and analysis tools used during this work were discussed, outlining the role of each in the project.

After presenting a sample of the FE data validation exercise, an explanation of the methods used to utilize the data in training neural networks, and their subsequent implementation into British Energy's neural network stress prediction program, PIPET, was given.

The newly updated PIPET program, was then evaluated by testing the system against new, previously unseen, FE data. The overall impression was very satisfactory, with the system producing very low levels of error for the vast majority of load case/stress component combinations. The flank branch out-of-plane bending in-plane stress results produced reasonably high percentage errors in some cases – a trend seen during the training phase. However, the author feels that PIPET's error band levels could be reduced for certain load cases, such as branch in-plane bending.

Evaluation and validation of the neural network program was extended by comparing the output from PIPET with FE data obtained from the reviewed literature, namely Moffat et al²³, Hassan⁷⁴ and Ayob⁷⁵. Data from the newly updated version of PIPET alongside the output from the original incarnation of PIPET were presented and discussed with respect to that obtained from the aforementioned works.

The evaluation exercises showed that, for pressure load case, a significant proportion of the FE data from^{23,74,75} was within the PIPET predicted stress ranges, with the updated

weights version of the program tending to have slightly lower stress levels than its predecessor. However, whilst the pressure load case comparisons proved useful, the branch bending cases showed less concordance with PIPET's predicted stress ranges and the system could not, therefore, be used with any degree of assurance for the "flat" UFT weld profile under such conditions.

7 CONCLUSIONS

The primary objective of this research was, to further extend the bounds of knowledge in the field of pressure vessel and piping stress analysis by presenting new data to the wider research community pertaining to the stress analysis of 90° welded pipe junctions with and without reinforcement.

The work undertaken on reinforced branch outlets (RBOs) was done during the period 1999 – 2001 for Spromak Ltd., when an extensive finite element analysis of 92 reinforced butt-welded pipe junctions was carried out. After comparing the resulting ESF data with the only other readily available source of data of a similar scale – that of Moffat et al²³ – it was concluded that, for the majority of loads, reinforced branch outlets appear better able to contain stresses than their un-reinforced counterparts, particularly for diameter ratios exceeding 0.5.

For the pressure load case, reinforcement of the thinner junctions for the vast majority of d/D ratios proved more than adequate and it can be seen from the results detailed in Appendix D that the current design ensures that ESF values for all diameter ratios and all values of D/T analysed herein remain between 1.8 and 3.5. In addition, when this design was compared with the Money⁴¹, Moffat et al²³ and Decock⁴² data for comparable D/T ratios it consistently outperformed UFT. Indeed, it may be that for heavier weight junctions, the reinforcement provided under the ASME B31.3 design code is more than adequate and, perhaps, more than necessary.

The study showed some trends that might be unwelcome if one is designing a piping structure using UFT: with the exception of pressure and run pipe out-of-plane bending moment, all load case comparisons reveal that ESF increases dramatically with increasing diameter ratio beyond $d/D = 0.5$ – this is particularly noticeable for thin-walled junctions.

For branch out-of-plane bending moment loading, the extent of shell bending in the run pipe increases with decreasing wall thickness. For very thin walled junctions ($D/T < 20$) the extent of shell bending in the run pipe is so large that the stresses are greatly magnified at approximately $d/D \approx 0.7$. This phenomenon, first investigated by Wordsworth⁴³ and subsequently Schneider⁴⁴, Woods and Rodabaugh⁴⁵, was also discovered in the experimental data of Chen and Wu⁴⁹. This differs considerably to the behaviour of UFT as reported by Moffat et al²³ for this load, where no such peak at $d/D \approx 0.7$ exists, rather, the ESF curve increases steadily with diameter ratio.

For the remaining branch bending moment loads, Myb and Mzb, this research has shown that in nearly all cases the reinforced fitting performs well under such conditions. This performance is enhanced when viewed in the light of ESFs for the un-reinforced fabricated tee design when diameter ratios exceed 0.5 and, in particular, when pipe walls are comparatively thin.

The results from all run moment load cases further enhanced the reinforced design's ability to deal with the stresses induced by these loads. Run pipe moment loads do not receive as much attention in open literature as pressure and branch out-of-plane bending, perhaps justifiably. However, the effects of these loads on piping systems design cannot be overlooked if overall structural integrity in any given situation is of concern.

The study of three reinforced branch junctions in chapter 4 set out to establish the potential use of inelastic FE analysis as a tool simulating proof pressure burst tests. Experimental material tests were described and an account was given with respect to inclusion of such data into a finite element model for inelastic analysis.

The differences between small and large displacement analysis were discussed before presenting results based on both types, alongside results from traditional hand calculations for estimating the bursting pressure. The different FE results obtained from the two sets of independently obtained material data tends to concur with Boyle and

Spence's²⁷ hypothesis that inelastic analysis can be expensive and is only as good as the material model.

Despite the conservatism of the results obtained from the large displacement FE method, the difference between corresponding FE and hand calculation results were not sufficient to warrant the cost of a full non-linear finite element analysis. Especially since the results obtained are in no way surprising or offer further insight for either a designer or stress analyst in respect of pressure vessel behaviour under such conditions.

The research regarding the stress analysis of un-reinforced fabricated tee using neural networks was a collaborative effort involving British Energy Ltd. over three years from 2001 – 2004. The basic principles of neural networks, their origins and some of the theory behind them was introduced in chapter 5 and particular attention was given to the multi-layer, feed-forward, back-propagation algorithm, which was later used to solve the problem posed in chapter 6.

Chapter 6 described work undertaken with British Energy Ltd. to extend their current capability of stress prediction and estimation in un-reinforced fabricated tee (UFT) branch junctions using a neural network approach.

It was discovered, during the FE validation exercise, that some of the original analysis done by British Energy was erroneous and may have led to non-conservative results from the PIPET program. Once updated with the new weights derived from the current study, PIPET was then evaluated by testing the system against new, previously unseen, FE data. The overall impression was very satisfactory, with the system producing very low levels of error (with respect to the actual FE result in each case) for the vast majority of load case/stress component combinations.

One exception was the flank branch out-of-plane bending in-plane stress results, which produced comparatively high percentage errors in some cases – a trend seen during the training phase. However, the author feels that one of the projects objectives – lowering the current 10% error band on PIPET's neural predictions for certain load cases, such as branch in-plane bending – could be achieved with the new weights.

Evaluation and validation of the neural network program was extended by comparing the output from PIPET with data from similar studies, namely those of Moffat et al²³, Hassan⁷⁴ and Ayob⁷⁵. Data from the newly updated version of PIPET alongside the output from the original incarnation of PIPET were presented and discussed with respect to data obtained from the aforementioned works.

The exercise showed that, for pressure load case, a significant proportion of the FE data from^{23,74,75} was within the PIPET predicted stress ranges, with the updated weights version of the program tending to have slightly lower stress levels than its predecessor. However, whilst the pressure load case comparisons proved useful, the branch bending cases showed less concordance with PIPET's predicted stress ranges and the system could not, therefore, be used with any degree of assurance for the "flat" UFT geometry profile under such conditions.

The use of neural networks for this sort of application is still in its infancy, yet it has proved a useful addition in the engineer's arsenal for rapidly evaluating stresses in unreinforced pipe junctions at British Energy Ltd. It is hoped that the program will be used with more frequency now that the range of the system has been increased (in terms of its geometry and load case parameters) as a direct result of this work.

7.1 Recommendations for Future Work

One possible avenue for further investigation into the stress analysis of reinforced pipe junctions is the potential use of neural networks for the evaluation of effective stress factors (ESFs). The work conducted in chapter 6 showed that neural networks can be readily trained to evaluate peak stresses with low levels of error. The scale of such research need not be overwhelming, despite the number of possible permutations of branch/run pipe combinations. Thoughtful selection of FE cases across the diameter range for a limited range of T/R ratios could be done as was for the UFT study detailed in chapter 6. If peak stresses only were required (for calculation of ESFs) the number of neural networks to be trained is significantly reduced from those that would be necessary for detailed stress predictions as was done in this work for UFT.

With respect to the work undertaken on UFT and presented in chapter 6, the author feels that the next logical step in the development of the PIPET system, would be to incorporate the evaluation of the stress distributions around the circumference of the weld. This would allow engineers to assess stresses in pipe junctions without being confined to the flank and crotch plane sections. This can be achieved using the current FE data that has been archived on DVD and could be made available for further research. In addition the source code for all programs written for this project is included in the appendices, thus successive researchers should be able to seamlessly continue this line of investigation with a view to further extending PIPET's potential.

References

1. Gill, S. S., The stress analysis of pressure vessels and pressure vessel components. Pergamon, London, UK, 1970.
2. Flugge, W., Stresses in shells. Springer Verlag, Berlin, Germany, 1973.
3. Kraus, H., Thin elastic shells. J. Wiley, New York, USA, 1967.
4. Spence J., Tooth A. S., Pressure vessel design: concepts and principles. E & FN Spon, Oxford, 1994.
5. BS EN-13445 Unfired pressure vessels. British Standards Institute, London, UK, 2002.
6. BS PD-5500 Specification for unfired fusion welded pressure vessels. British Standards Institute 1997.
7. ASME B 31.1 ASME B31.1 Power piping. American Society of Mechanical Engineers 1995 Edition – 1996 Addenda.
8. ASME B 31.3 Process Piping. American Society of Mechanical Engineers 1996 Edition – 2000 Addenda.
9. ASME BPVC Section III Boiler and pressure vessel code Division 1 Subsection NB. American Society of Mechanical Engineers 1989.
10. ASME BPVC Section VIII Boiler and pressure vessel code Division 2. American Society of Mechanical Engineers 1998.
11. Leckie F. A., Penny R. K. Stress concentration factors for the stresses at nozzle intersections in pressure vessels. Welding Research Council Bulletin No. 90 1963.
12. Lock J. D., Carmichael G. D. T., Moffat D. G., Finite element stress analysis of an equal diameter branch pipe intersection subjected to internal pressure and in-plane moment loadings. Pipe-work Design and Operation Institution of Mechanical Engineers, 1985; C11/85: 59-70.
13. Ando Y. Yagawa G. Kikuchi F. Stress distributions in thin-walled intersecting cylindrical shells subjected to pressure and in-plane force. Proceedings Structural Mechanics in Reactor Technology, Berlin, Paper G2/2, 1971.
14. Ando Y. Yagawa G. Kikuchi F. Stress distributions in thin-walled intersecting cylindrical shells subjected to pressure and in-plane force. Proceedings Structural Mechanics in Reactor Technology, Berlin, Paper G2/2, 1971.
15. Baldur R., Sabourin M. Finite element analysis of cylindrical intersections loaded by moments. ASME Paper 80-C2/Pressure Vessels and Piping 3, 1980.
16. Gwaltney R. C. Corum, J. M. Bolt, S. E. Bryson, J. W. Experimental stress analysis of cylinder-to-cylinder shell models and comparisons with theoretical predictions. J Pressure Vessel Technol Trans ASME, v 98 Ser J, n 4, Nov 1976, p 283-290.
17. Bryson J. W. Johnson W. G. Bass B. R. Stresses in reinforced nozzle-cylinder attachments under internal pressure loading analysed by the finite element method – a parameter study. American Society of Mechanical Engineers, Pressure Vessels and Piping Division (Publication) PVP, v 50 pp 51-65, 1981.
18. Steele C. R., Steele M. L. Stress analysis of nozzles in cylindrical vessels with external load. ASME Paper no.82, Pressure Vessels and Piping (14), 1982.
19. Mershon J. L. Mokhtarian K. Ranjan G. V. Rodabaugh E. C. Local stresses in cylindrical shells due to external loadings on nozzles – supplement to WRC bulletin No. 107. Welding Research Council Bulletin, September 1984.
20. Khan A. S., Woods G. A comparative study of the stress field around reinforced and unreinforced normally intersecting cylindrical shells. ASME Paper no. 82, Pressure Vessels and Piping (9), 1982.
21. Berger B. R. Woods G. Finite element determination of stress indices. ASME Paper no. 82, Pressure Vessels and Piping (12), 1982.
22. Moffat D. G. Experimental stress analysis of four fabricated equal diameter branch intersections subjected to moment loadings and the implications on branch junction design. Proceedings of the Institution of Mechanical Engineers, Part A: Power and Process Engineering 1985; 199(A4): 261-284.

23. Moffat D. G. Mwenifumbo J. A. M. Xu S. H. Mistry J. Effective stress factors for piping branch junctions due to internal pressure and external moment loads. *IMechE Journal of Strain Analysis*, 1991; 26(2): 85-101.
24. Markl A. R. C. Fatigue tests of piping components. *Trans ASME*, 1952; 74(3): 287-303.
25. Gilroy J. E. Clark J. S. Tosh P. A. Finite element stress analysis of extruded outlet tee junctions. *IMechE Conference Publications (Institution of Mechanical Engineers)* 1985; C15/85: 83-95.
26. Pastor T. P. Hechmer J. L. ASME Task Group report on primary stress. *ASME PVP*, 1994; 277. Recertification and stress classification issues. Ed. Petrinec Jr. J. N.
27. Boyle J. T. Spence J. Developments in methods of inelastic piping analysis. *Proc Inst Mech Engrs*, 1987; 201(A2): 105-126.
28. Mackerle J. Finite elements in the analysis of pressure vessels and piping, an addendum: a bibliography (1998–2001). *Int. J. Pres. Ves. Piping* 2002; 79(1): 1-26.
29. Nadarajah C. Mackenzie D. Boyle J. T. A method of estimating limit loads by iterative elastic analysis. II – Nozzle sphere intersections with internal pressure and radial load. *Int. J. Pres. Ves. Piping* 1993; 53: 97-119.
30. Mackenzie D. Shi J. Boyle J. T. Finite element modelling for limit analysis by the elastic compensation method. *Computers & Structures*. 1994; 51(4): 403-410.
31. Carino C. Carli F. Cinquini C. Pipe self-reinforcing outlets: non-linear analysis towards rational design. American Society of Mechanical Engineers, Design Engineering Division (Publication) DE, v 82, n 1, 21st Annual Design Automation Conference, Advances in Design Automation, 1995, 915-922.
32. Sanal Z. Nonlinear analysis of pressure vessels: some examples. *Int. J. Pres. Ves. Piping* 2000; 77(12): 705-709.
33. Gerdeen J. C. A critical evaluation of plastic behaviour data and a unified definition of plastic loads for pressure components. *PVRC Welding Research Council Bulletin* 254, ASME, 1979.
34. Moffat D. G. Hsieh M. F. Lynch M. An assessment of ASME III and CEN TC54 methods of determining plastic and limit loads for pressure system components. *Journal of Strain Analysis for Engineering Design*. 2001; 36(3): 301-312.
35. Hibbit Karlsson Sorensen. ABAQUS/Standard users manual volume II. HKS 1998; 14.1.1-1–14.1.1-3.
36. Williams D. K. Development of stress intensity factors for fabricated tee branch connections. *ASME Pressure Vessel Piping* 1995; 301: 55-66.
37. Moffat D. G. Kirkwood M. G. Flexibility factors for fabricated equal diameter branch pipe intersections. *Institution of Mechanical Engineers* 1985 C5/85: 49-58.
38. Moffat D. G. Mistry J. Moore S. E. Effective stress correlation equations for piping branch junctions under internal pressure loading. *ASME Journal of Pressure Vessel Technology* 1999; 121: 121-126.
39. Weis E. Joost H. Local and global flexibility of nozzle-to-vessel-intersections under local loads as boundary conditions for piping system design. *International Journal of Pressure Vessels & Piping* 1997; 73: 241-247.
40. Mwenifumbo J. A. M. Stress and flexibility analysis of branch pipe junctions by the finite element method. Thesis MEng, Liverpool University, 1988.
41. Money H. A. Designing flush cylinder-to-cylinder intersections to withstand pressure. *Proceedings ASME* 1968.
42. Decock J. Reinforcement method of openings in cylindrical pressure vessels subjected to internal pressure. *Centre de Recherches Scientifiques et Techniques de L'Industrie des Fabrications Metalliques* 1975; MT104.
43. Wordsworth A. C. Smedley G. P. Stress concentrations of unstiffened tubular joints. *European Offshore Research Seminar*, Cambridge 1978.
44. Schneider R. W. Early warnings of a problem regarding code stress intensity factors for reducing outlet branch connections. *ASME Pressure Vessels and Piping Division* 1988; 136: 73-74.
45. Woods G. E. Rodabaugh E. C. WFI/PVRC moment fatigue tests on 4x3 ANSI B16.9 tees. *Welding Research Council Bulletin* 1989; 346: 1-8.
46. Rodabaugh E. C. Accuracy of stress intensification factors for branch connections. *Welding Research Council Bulletin*, pp 1-44. Dec. 1987.
47. Woods, G. E. Stress intensification factor equations for fabrication branch connections. *ASME Pressure Vessels & Piping* 1987; 120, *Design and Analysis of Pressure Vessel Components*, American Society of Mechanical Engineers, New York.
48. Markl A. R. C. Piping flexibility analysis. *Trans ASME*, 1955; 77: 127-149.

49. Chen J. C. Wu Y. C. Low cycle fatigue strength of integrally reinforced branch connections. *Pressure Vessel and Piping Codes and Standards ASME* 1997; 353: 315-323.
50. Plummer, M. FEA Determination of ESF data and burst test simulation for a range of branch outlet fittings. Thesis B.Eng. (Hons), Liverpool John Moores University, 2000.
51. Gurney, K. An introduction to neural networks. UCL Press, 1997.
52. Haykin, S. *Neural Networks: A comprehensive foundation*. Prentice Hall, 1999.
53. Tsoukalas, L. H. Uhrig, R. E. *Fuzzy and neural approaches in engineering*. John Wiley & Sons, 1997.
54. McCulloch, W. Pitts, W. A logical calculus of the ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics*, 1943.
55. Hebb, D. O. *The organization of behaviour: A neuropsychological theory*, Wiley, 1949.
56. Rochester, N. Holland, J. H. Haibt, L. H. Duda, W. L. Tests on a cell assembly theory of the action of the brain, using a digital computer. *IRE Transactions on Information Theory*. IT-2: 80-93.
57. Uttley, A. M. A theory of the mechanism of learning based on the computation of conditional probabilities. *Procs. of the First International Conference on Cybernetics*, Namur, Gauthier-Villars, Paris, 1956.
58. Rosenblatt, F. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, 1958; 65: 386-408.
59. Widrow, B. Hoff, Jr. M. E. Adaptive switching circuits. *IRE WESCON Convention Record*, 1960: 96-104.
60. Rummelhart, D. E. Hinton, G. E., Williams, R. J. Learning representations of back-propagation Errors. *Nature*, 1986; 323: 533-536.
61. Lippmann, R. P. An introduction to computing with neural nets. *IEEE ASSP Magazine*, 1987: 4-22.
62. Hush, D. R. Horne, B. G. Progress in supervised neural networks: what's new since Lippmann? *IEEE Signal Processing Magazine*, 1993: 8-38.
63. Irie, B. Miyake, S. Capabilities of three-layered perceptrons. *Proc. IEEE International Conference on Neural Networks*, 1988: 1-641.
64. Hornik, K. Stinchcombe, M. White, H. Multi-layer feed-forward networks are universal approximators. *Neural Networks*, 1989; 2(5): 359-366.
65. Marquardt, D. An algorithm for least squares estimation of non-linear parameters. *J. Soc. Ind. Appl. Math.* 1963: 431-441.
66. Hagan, M. T. Menhaj, M. Training feed-forward networks with the Marquardt algorithm. *IEEE Transactions on Neural Networks*. 1994; 5(6): 989-993.
67. Demuth, H. Beale, M. *Matlab neural network toolbox user guide version 4*. The MathWorks, Inc. 2002.
68. Masters, T. *Practical neural network recipes in C++*. Academic Press, 1993.
69. Werbos, P. J. Beyond regression: new tools for prediction and analysis in the behavioural sciences. Ph. D. Thesis, Harvard University, Cambridge, MA. 1974.
70. LeCun, Generalization and network design strategies. Technical Report CRG-TR-89-4, Dept. of Computer Science, University of Toronto, Canada, 1989.
71. LeCun, Efficient Learning and second-order methods: a tutorial at NIPS 93, Denver, 1993.
72. Fox M. J. H. Mitchell I. PIPET: Neural network stress prediction, 1996; I. Nuclear Electric Internal Report EPD/GEN/REP/0148/96.
73. Hines, J. W. *Matlab supplement to fuzzy and neural approaches in engineering*. John Wiley & Sons, Inc., 1997.
74. Hassan, A.Y. B. Manufacture and stress analysis of elliptically shaped pressure vessel nozzles. Thesis Ph.D., Liverpool University, 1990.
75. Ayob, A. B. Development of load interaction design rules for pressurised components subjected to combined loading, Thesis Ph.D., Liverpool University, 1994.
76. Natarajan, R. Widera, G. E. O. Afsari, P. A finite element model to analyse cylinder-cylinder intersections. *Journal of Pressure Vessel Technology*, 1985; 109: 411-420.
77. Lynch, M. A. Branch junctions with cracks. Thesis Ph.D., Liverpool University, 2001.
78. Yahiaoui, K. Moffat, D. G. Moreton, D. N. The collapse behaviour of a forged piping branch junction under internal pressure. 9th Int. Conf. Pressure Vessel Technology, Sydney, 2000.
79. Wang, Y. Kuo, A. Development of guidelines for FEA modelling of cylinder-to-cylinder intersections (Draft Report). Optimal Corporation (for PVRC), 1999.

Bibliography

1. Calladine, C. R. *Plasticity for Engineers*. Ellis Horwood Limited, 1985.
2. Chapman, S. J. *Matlab Programming for Engineers*. Brooks/Cole 2000.
3. Demuth, H. Beale, M. *Matlab Neural Network Toolbox User Guide Version 4*. The MathWorks, Inc. 2002.
4. Gurney, K. *An Introduction to Neural Networks*. UCL Press, 1997.
5. Haykin, S. *Neural Networks: A Comprehensive Foundation*. Prentice Hall, 1999.
6. Hibbit Karlsson Sorensen. *Getting Started with ABAQUS/Standard*. HKS 1998.
7. Hibbit Karlsson Sorensen. *ABAQUS/Standard users manual volume II*. HKS 1998.
8. Hines, J. W. *Matlab Supplement to Fuzzy and Neural Approaches in Engineering*. John Wiley & Sons, Inc., 1997.
9. Spence, J. Tooth, A. S. *Pressure Vessel Design: Concepts and Principles*. E & FN Spon, 1994.
10. Hearn EJ. *Mechanics of Materials Vol II, 3rd Ed*. Oxford: Butterworth-Heinemann. 1996.
11. Masters, T. *Practical Neural Network Recipes in C++*. Academic Press, 1993.
12. Tsoukalas, L. H. Uhrig, R. E. *Fuzzy and Neural Approaches in Engineering*. John Wiley & Sons, 1997.

Appendix A: Butt-weld Outlet Bores and ASA Dimensions of Seamless Pipe to BS1600, ANSI B36.10 and B3619

BUTT WELD OUTLET BORES and ASA SCHEDULES (dimensions in inches)

Nominal pipe size	1/2	3/4	1	1 1/4	1 1/2	2	2 1/2	3	3 1/2	4	5	6	8	10	12	14	16	18	20	22	24	26
Outside diameter	0.840	1.050	1.315	1.660	1.900	2.375	2.875	3.500	4.000	4.500	5.563	6.625	8.625	10.750	12.750	14.000	16.000	18.000	20.000	22.000	24.000	26.000
Schedule 5S	0.710	0.920	1.185	1.530	1.770	2.245	2.709	3.334	3.834	4.334	5.345	6.407	8.407	10.482	12.438	13.688	15.670	17.670	19.624	21.624	23.564	-
Schedule 10S	0.674	0.884	1.097	1.442	1.682	2.157	2.635	3.260	3.760	4.260	5.295	6.357	8.329	10.420	12.390	13.624	15.624	17.624	19.564	21.564	23.500	-
Schedule 10	0.674	0.884	1.097	1.442	1.682	2.157	2.635	3.260	3.760	4.260	5.295	6.357	8.329	10.420	12.390	13.500	15.500	17.500	19.500	21.500	23.500	-
Schedule 20	-	-	-	-	-	-	-	-	-	-	-	-	8.125	10.250	12.376	13.376	15.376	17.376	19.250	21.250	23.250	-
Schedule 30	-	-	-	-	-	-	-	-	-	-	-	-	8.071	10.137	12.090	13.250	15.250	17.124	19.000	21.000	22.876	-
Schedule 40S	0.622	0.824	1.049	1.380	1.610	2.067	2.469	3.068	3.548	4.026	5.047	6.065	7.981	10.020	12.000	-	-	-	-	-	-	-
Standard Wall	0.622	0.824	1.049	1.380	1.610	2.067	2.469	3.068	3.548	4.026	5.047	6.065	7.981	10.020	12.000	13.250	15.250	17.250	19.250	21.250	23.250	25.250
Schedule 40	0.622	0.824	1.049	1.380	1.610	2.067	2.469	3.068	3.548	4.026	5.047	6.065	7.981	10.020	11.938	13.124	15.000	16.876	18.814	20.750	22.626	-
Schedule 60	-	-	-	-	-	-	-	-	-	-	-	-	7.813	9.750	11.626	12.814	14.688	16.500	18.376	20.250	22.064	-
Schedule 80S	0.546	0.742	0.957	1.278	1.500	1.939	2.323	2.900	3.364	3.826	4.813	5.761	7.625	9.750	11.750	13.000	15.000	17.000	19.000	21.000	23.000	25.000
Extra Strong	0.546	0.742	0.957	1.278	1.500	1.939	2.323	2.900	3.364	3.826	4.813	5.761	7.625	9.750	11.376	12.500	14.314	16.126	17.938	19.750	21.564	-
Schedule 80	0.546	0.742	0.957	1.278	1.500	1.939	2.323	2.900	3.364	3.826	4.813	5.761	7.625	9.564	11.064	12.126	13.938	15.688	17.438	19.250	20.938	-
Schedule 100	-	-	-	-	-	-	-	-	-	-	-	-	7.439	9.314	11.064	12.126	13.938	15.688	17.438	19.250	20.938	-
Schedule 120	-	-	-	-	-	-	-	-	-	-	-	-	7.189	9.064	10.750	11.814	13.564	15.250	17.000	18.750	20.376	-
Schedule 140	-	-	-	-	-	-	-	-	-	-	-	-	7.001	8.750	10.500	11.500	13.124	14.876	16.500	18.250	19.876	-
Schedule 160	0.466	0.614	0.815	1.160	1.338	1.689	2.125	2.624	-	-	-	-	6.813	8.500	10.126	11.168	12.814	14.438	16.064	17.750	19.314	-
Double Extra Strong	0.252	0.434	0.599	0.896	1.100	1.503	1.771	2.300	2.728	3.152	4.063	4.897	6.875	8.750	10.750	-	-	-	-	-	-	-

DIMENSIONS OF SELECTED SIZES OF SEAMLESS L PIPE TO BS.1600 and ANSI B.36.10 and B3619 Wall Thickness (All dimensions in inches)

Nominal pipe size	1/2	3/4	1	1 1/4	1 1/2	2	2 1/2	3	3 1/2	4	5	6	8	10	12	14	16	18	20	22	24	26	30	36
Outside diameter	0.840	1.050	1.315	1.660	1.900	2.375	2.875	3.500	4.000	4.500	5.563	6.625	8.625	10.750	12.750	14.000	16.000	18.000	20.000	22.000	24.000	26.000	30.000	36.000
Schedule 5S	0.065	0.065	0.065	0.065	0.065	0.083	0.083	0.083	0.083	0.083	0.109	0.109	0.109	0.134	0.156	0.156	0.165	0.165	0.188	0.188	0.218	0.250	-	-
Schedule 10S	0.083	0.083	0.109	0.109	0.109	0.120	0.120	0.120	0.120	0.134	0.134	0.134	0.148	0.165	0.180	0.188	0.188	0.188	0.218	0.218	0.250	-	-	-
Schedule 10	0.083	0.083	0.109	0.109	0.109	0.120	0.120	0.120	0.120	0.134	0.134	0.134	0.148	0.165	0.180	0.230	0.250	0.250	0.250	0.250	0.312	0.312	0.312	0.312
Schedule 20	-	-	-	-	-	-	-	-	-	-	-	-	0.250	0.250	0.250	0.312	0.312	0.312	0.375	0.375	0.375	0.500	0.500	0.500
Schedule 30	-	-	-	-	-	-	-	-	-	-	-	-	0.277	0.307	0.330	0.375	0.375	0.438	0.500	0.500	0.562	-	-	-
Schedule 40S	0.109	0.113	0.133	0.140	0.145	0.154	0.203	0.216	0.226	0.237	0.258	0.280	0.322	0.365	0.375	-	-	-	-	-	-	-	-	-
Standard Wall	0.109	0.113	0.133	0.140	0.145	0.154	0.203	0.216	0.226	0.237	0.258	0.280	0.322	0.365	0.375	0.375	0.375	0.375	0.375	0.375	0.375	0.375	0.375	0.375
Schedule 40	0.109	0.113	0.133	0.140	0.145	0.154	0.203	0.216	0.226	0.237	0.258	0.280	0.322	0.365	0.406	0.438	0.500	0.562	0.562	0.687	-	-	-	0.750
Schedule 60	-	-	-	-	-	-	-	-	-	-	-	-	0.406	0.500	0.562	0.593	0.656	0.750	0.812	0.875	0.968	-	-	-
Schedule 80S	0.147	0.154	0.179	0.191	0.200	0.218	0.276	0.300	0.318	0.337	0.375	0.432	0.500	0.500	0.500	0.500	0.500	0.500	0.500	0.500	0.500	0.500	0.500	0.500
Extra Strong	0.147	0.154	0.179	0.191	0.200	0.218	0.276	0.300	0.318	0.337	0.375	0.432	0.500	0.500	0.500	0.500	0.500	0.500	0.500	0.500	0.500	0.500	0.500	0.500
Schedule 80	0.147	0.154	0.179	0.191	0.200	0.218	0.276	0.300	0.318	0.337	0.375	0.432	0.500	0.500	0.500	0.500	0.500	0.500	0.500	0.500	0.500	0.500	0.500	0.500
Schedule 100	-	-	-	-	-	-	-	-	-	-	-	-	0.438	0.500	0.562	0.644	1.000	1.093	1.218	1.375	1.531	-	-	-
Schedule 120	-	-	-	-	-	-	-	-	-	-	-	-	0.438	0.500	0.562	0.718	0.843	0.937	1.031	1.156	1.281	1.375	1.531	-
Schedule 140	-	-	-	-	-	-	-	-	-	-	-	-	0.812	1.000	1.125	1.250	1.438	1.562	1.750	1.875	2.062	-	-	-
Schedule 160	0.187	0.218	0.250	0.250	0.281	0.343	0.375	0.438	-	-	-	-	0.531	0.625	0.718	0.906	1.125	1.312	1.406	1.593	1.781	1.968	2.125	2.343
Double Extra Strong	0.294	0.308	0.358	0.382	0.400	0.436	0.552	0.600	0.636	0.674	0.750	0.864	0.875	1.000	1.000	-	-	-	-	-	-	-	-	-

Appendix B: Matlab Source Code: ESF Stress Analysis of RBO

```

% =====
% MAIN PROGRAM FCRMAIN.M FOR COUNTER BORED FLANGED BUTTWELDED FITTINGS
% =====
%
% PRELIMINARY CALCULATIONS FROM USER INPUTS
%
R=Do/2;
rH=R-wt;
fr=fr/2;
ri=ri/2;
rcb=rcb/2;
rsh=rsh/2;
hs=hs90-bh;
nl=(hb-fl-hs-sw);
ddivD=nombranch/nomhead;
shl=hs+R+rg;
shh=shl+sw;
rblo=rcb+bl;
lol=-(P*(ri^2))/(fr^2-ri^2);          % FLANGE STRESS CALC.
lo2=-(P*(rH^2))/(R^2-rH^2);          % HEADER STRESS CALC AT RIGHT HAND END.
% SET UP MATRIX "A" TO STORE ALL CO-ORDINATE POINTS GENERATED
a=zeros(91,36);
% GENERATE POINT NUMBERS FOR USE IN FEMGV
j=1:91;
k=92:182;
l=183:273;
m=274:364;
n=365:455;
o=456:546;
p=547:637;
q=638:728;
s=729:819;
% ANGLES FROM 0° TO 90° CONVERTED INTO RADIANs
angle=0:90;
alpha=pi*angle/180;
x=cos(alpha);
z=sin(alpha);
% CALCULATE THE X, Y AND Z VALUES FOR THE TWO SHOULDER CURVES
xshl=rsh*x;
zshl=rsh*z;
yshl=ones(91,1)*shl;
% TOP SHOULDER CURVE HAS SAME X AND Y CO-ORDS AS THE LOWER SHOULDER.
% THEREFORE ONLY NEED NEW Y VALUES - YSHH
yshh=ones(91,1)*shh;
% CALCULATE THE INTERSECTION BETWEEN THE INSIDE BASE LANDING AND THE HEADER
xibl=x*rcb;
zibl=z*rcb;
zibl2=zibl.^2;
yibl=sqrt(((R+rg)^2)-zibl2);
% CALCULATE THE INTERSECTION BETWEEN THE OUTSIDE BASE LANDING AND THE HEADER
xobl=x*rblo;
zobl=z*rblo;
zobl2=zobl.*zobl;
yobl=sqrt(((R+rg)^2)-zobl2);
% CALCULATE THE INTERSECTION BETWEEN THE COUNTER BORE OF THE BRANCH WITH THE
% INSIDE OF THE HEADER
xinter=x*rcb;
zinter=z*rcb;
zinter2=zinter.^2;
yinter=sqrt((rH^2)-zinter2);
% CALCULATE THE INTERSECTION BETWEEN THE COUNTER BORE OF THE BRANCH WITH THE
% INSIDE OF THE HEADER
xintsec=x*rcb;
zintsec=z*rcb;
zintsec2=zintsec.^2;
yintsec=sqrt((R^2)-zintsec2);
% FILL MATIX "A" WITH X, Y AND Z CO-ORDINATES FOR THE TWO SHOULDER CURVES
% AND THE BASE LANDING INTERSECTIONS
a(:,5)=k";
a(:,6)=xshl";
a(:,7)=yshl;
a(:,8)=zshl";
a(:,9)=1";

```

```

a(:,10)=xshl";
a(:,11)=yshh";
a(:,12)=zshl";
a(:,13)=m";
a(:,14)=xibl";
a(:,15)=yibl";
a(:,16)=zibl";
a(:,17)=n";
a(:,18)=xobl";
a(:,19)=yobl";
a(:,20)=zobl";
a(:,21)=o";
a(:,22)=xinter";
a(:,23)=yinter";
a(:,24)=zinter";
% CALL SUBPROGRAM TO CALCULATE THE CO-ORDS FOR THE TOP OF THE WELD
TOP;
% CALL SUBPROGRAM TO CALCULATE THE CO-ORDS FOR THE TOE OF THE WELD
TOE;
a(:,33)=s";
a(:,34)=xintsec";
a(:,35)=yintsec";
a(:,36)=zintsec";
% CALL SUBPROGRAM TO STORE ALL GEOMETRY CO-ORDS OTHER THAN THOSE IN THE WELD.
CBRpoints;

% =====
% Subprogram top.m: calculate the cords for the top of the weld
% =====
degrad=(pi/180);
dy=hs+sw; % Get the y value for the top of the weld at 0°
dr=rsh-rcb-bl; % Get the x value for the top of the weld at 0°
DR=rtop-bl; % Calculate the initial radius (DR) at the top
% of the weld from the outer base landing (90°)
DY=DR*tan(Y1*pi/180); % Calculate the initial height (DY) at the top
% of the weld from the outer base landing (90°)
Adj=(hs90+sw)-DY; % Calculate the difference in height between the top of
% the shoulder at 90° and the top of the weld
Opp=(rsh-rcb-rtop); % The radial distance between the shoulder and the top
% of the weld at 90°
if nombranch<=2;
    Y0=(atan(dy/dr))*180/pi;
    Y3=atan(Opp/Adj)*180/pi;
    else
        Y0=Y0;
        Y3=Y3;
end;
A=zeros(91,6);
angle=0:90;
alpha=pi*angle/180;
x=cos(alpha)";
z=sin(alpha)";
Yi=Y0-((Y0-Y1)/90)*angle;
Yi=Yi*pi/180;
m1=tan(Yi);
intang=(90-Y3)*degrad; % Calculate the angle where "line 2" intersects the y axis
m2=tan(intang); % The fixed gradient of "line 2"
% Track the change in height of the base landing
% with respect to it's original height (zero) at 0°
L4inter=((hs90-hs)/90)*angle";
% c = y-mr, where c is the y axis intercept
Yint90=(hs90+sw)-m2*(rsh-rcb-bl);
SHINT90=(-Yint90)+hs90;

% Calculate the y intercept value from the shoulder point in the transverse
% plane for "line 2"
SHOBL=L4inter+hs; % Calculate the height of the base landing from the shoulder
% starting at 0°
cintL2=SHOBL-SHINT90; % These are the values for the C intercept for all angles from
% 0° - 90°.
A(:,1)=angle";
A(:,2)=Yi";
A(:,3)=m1";

```

```

A(:,4)=cintL2;
for n=1:91; % Start a loop,n, with the number of rows 1 TO 91
    slope=A(n,3); % ON EACH PASS, N, READ MATRIX A COLUMN 3 INTO THE VECTOR "SLOPE"
    CL2=A(n,4); % ON EACH PASS, N, READ MATRIX A COLUMN 4 INTO THE VECTOR "CL2"
    lhs=[1 -m2;1 -slope]; % PUT THE NEW VALUE FOR "SLOPE" INTO THE MATRIX REPRESENTING
        % THE LEFT HAND SIDE OF Y-MR=C
    rhs=[CL2;0]; % STORE THE "C" VALUES FROM THE RIGHT HAND SIDE OF THE
        % EQUATION Y-MR=C IN THE MATRIX "RHS"
    sol=lhs\rhs; % SOLVE THE EQUATIONS SIMULTANEOUSLY OBTAINING Y AND
        % R VALUES, THEN STORE IN "SOL"
    yr(n,1)=sol(1,1); % FROM "SOL" ROW 1 COLUMN 1, STORE EACH NEW Y VALUE IN
        % MATRIX "YR" COLUMN 1
    yr(n,2)=sol(2,1); % FROM SOL ROW 2 COLUMN 1, STORE EACH NEW R VALUE IN
        % MATRIX "YR" COLUMN 2
end;
r=yr(:,2); % USING Y AND R VALUES CALCULATE THE LOCAL COORDINATES IN
xtow=r.*x; % X,Y, AND Z RELATIVE TO THE OUTER BASE LANDING FOR EACH
ztow=r.*z; % DEGREE, BETA, FROM THE TRANSVERSE TO THE LONGITUDINAL PLANE
ytow=yr(:,1);
A(:,5)=xtow;
A(:,6)=ytow;
A(:,7)=ztow;
% GET X,Y AND Z VALUES FROM MATRIX "A" COLUMNS 18-20, AND ADD TO THE CO-ORDS
% FOR THOSE IN MATRIX "A" TO GET THE GLOBAL CO-ORDS.
% PUT INTO TEMPORARY MATRIX "DTOW" AND EXTRACT THESE TO MATRIX "A" COLUMNS 26-28.
dtow=a(1:91,18:20)+A(1:91,5:7);
a(:,25)=p";
a(:,26:28)=dtow;
% STORE GLOBAL X,Y AND Z CO-ORDS FOR TOP OF THE WELD IN SEPARATE VECTORS TO
% ALLOW SIMPLE PLOTTING PROCEDURES
gtwx=a(:,26);
gtwy=a(:,27);
gtwz=a(:,28);

% =====
% SUBPROGRAM TOE.M TO CALCULATE THE COORDS FOR THE INTERSECTION
% BETWEEN THE TOE OF THE WELD AND THE RUN PIPE
% =====
% Store all values for calculating the coords of the weld toe
BMAT=zeros(91,9);
angle=0:90;
angle=angle+(angle==0)*eps;
BETA=pi*angle/180;
x=cos(BETA)";
z=sin(BETA)";
Zi=ones(1,91)";
Zi=Z3-((Z3-Z8)/80)*angle;
Zi=(Zi*pi/180)";
Zi(82:91)=Zi(81); % SET LAST 10 VALUES OF THE INTERMEDIATE Z3 ANGLE TO Z8°
THETA=(90*pi/180)-Zi;
GRAD=tan(THETA);
GRAD2=GRAD.^2;
YINT=(r+rblo).*GRAD;
C=YINT+gtwy;
C2=C.^2;
% CALCULATE THE "A" AND "B" VALUES FOR ELLIPSES 1 THROUGH TO 90 DEGREES
ELLb=ones(91,1)*R; % B IS ALWAYS EQUAL TO THE HEADER OD
ELLa=ELLb./sin(BETA)";
ELLa2=ELLa.^2; % SQUARE OF "A" AND "B" VALUES
ELLb2=ELLb.^2;
% PUT ALL CALCULATED VALUES INTO BMAT
BMAT(:,1)=angle";
BMAT(:,2)=Zi;
BMAT(:,3)=YINT;
BMAT(:,4)=GRAD;
BMAT(:,5)=GRAD2;
BMAT(:,6)=C;
BMAT(:,7)=C2;
BMAT(:,8)=ELLb2;
BMAT(:,9)=ELLa2;
% DEFINE EQUATIONS TO DETERMINE CO-EFFICIENTS OF X^2, X AND K
X2TERM=ELLb2+ELLa2.*GRAD2;
XTERM=2.*ELLa2.*(-GRAD).*C; % GRAD CHANGED TO NEG FOR REDUCING WHERE Z8 IS +VE
K=(ELLa2.*C2)-(ELLa2.*ELLb2);

```

```

CMAT=zeros(91,3);
CMAT(:,1)=X2TERM;
CMAT(:,2)=XTERM;
CMAT(:,3)=K;
% SET UP A LOOP TO FIND ALL ROOTS OF THE POLYNOMIALS IN CMAT. THESE ROOTS ARE THE
% "r" VALUES OF THE TOE OF THE WELD.
% ONCE OBTAINED, THE HIGHEST "r" VALUE IN EACH ROW WILL BE TAKEN AND THEN BROKEN
% DOWN INTO X AND Z CO-ORDS FOR THE TOE OF THE WELD
rootstr=zeros(91:4);
for n=1:91;
    POLVAL=CMAT(n,1:3);
    rootstr=roots(POLVAL);
    RVALS(n,1)=rootstr(1,1)";
    RVALS(n,2)=rootstr(2,1)";
end;
ytoe1=C-(GRAD.*RVALS(:,1));
ytoe2=C-(GRAD.*RVALS(:,2));
xtoe=x.*RVALS(:,2);
ztoe=z.*RVALS(:,2);
a(:,29)=q";
a(:,30)=xtoe;
a(:,31)=ytoe2;
a(:,32)=ztoe;

% =====
% SUBPROGRAM CBRPOINTS.M FOR REDUCING COUNTER BORED BUTTWELDED FITTINGS
% HAVING A BURN HEIGHT LESS THAN 10MM
% =====
%
% FILL COLUMNS 1-4 OF MATRIX "A" WITH X, Y AND Z CO-ORDINATES ALL POINTS
% OTHER THAN THE WELD PROFILES SHOULDER AND INTERSECTION POINTS
%
if (typeval==2)|(typeval==4);
    a(:,1)=j"; % ALL OF THESE POINTS ARE CENTRES FOR ARCS
    a(2,2)=R*1.5; % THAT MAKE UP THE GEOMETRY OF THE HEADER
    a(3,2)=(R*2)*2*3.43; % AND THE BRANCH. THEY ALL LIE ON EITHER
    a(4,2)=-(R*2)*2*3.43; % THE BRANCH AXIS (I.E. X=0) OR THE HEADER
    a(5,3)=R+rg; % AXIS (I.E. Y=0).
    a(6,3)=((rcb-ri)*tan(Y2))+a(5,3);
    a(7,3)=R+rg+hs+sw+(nl/2);
    a(8,3)=R+rg+hb-fl;
    a(9,3)=R+rg+hb;
else;
    a(:,1)=j"; % ALL OF THESE POINTS ARE CENTRES FOR ARCS
    a(2,2)=R*1.5; % THAT MAKE UP THE GEOMETRY OF THE HEADER
    a(3,2)=(R*2)*2*3.43; % AND THE BRANCH. THEY ALL LIE ON EITHER
    a(4,2)=-(R*2)*2*3.43; % THE BRANCH AXIS (I.E. X=0) OR THE HEADER
    a(5,3)=R+rg; % AXIS (I.E. Y=0).
    a(6,3)=((rcb-ri)*tan(Y2))+a(5,3);
    a(7,3)=(rsh-ro)/tan(Z1)+sw+hs+rg+R;
    a(8,3)=R+rg+hb+blength;
end;
% POINTS TO CREATE THE INTERSECTION BETWEEN THE C"BORE AND THE INSIDE OF
% THE HEADER
a(10,2)=a(1,22); % POINTS FOR THE START POINT OF THE INTERSECTION CURVE
a(10,3)=a(1,23); % BETWEEN C"BORE AND INSIDE SURFACE OF HEADER
a(11,3)=a(91,23); % POINTS FOR THE END POINT OF THE INTERSECTION CURVE
a(11,4)=a(91,24); % BETWEEN C"BORE AND INSIDE SURFACE OF HEADER
a(12,2)=a(46,22); % THESE ARE THE CO-ORDS FOR THE MIDDLE POINT OF THE
a(12,3)=a(46,23); % INTERSECTION CURVE BETWEEN THE C"BORE AND THE INSIDE
a(12,4)=a(46,24); % SURFACE OF THE HEADER
a(13,4)=rH; % THESE POINTS (14, 17 & 18)
a(16,2)=a(2,2); % FORM THE BOUNDARY OF SURFACE NUMBER S4,
a(16,4)=rH; % ALONG WITH POINTS 11,12 AND 13.
a(17,2)=a(2,2);
a(17,3)=rH;
a(14,3)=-rH; % POINTS 14 & 15 ALONG WITH 13 & 16 FOR THE BOUNDARY
a(15,2)=a(2,2); % OF SURFACE NUMBER S3.
a(15,3)=-rH;
a(18,2)=a(3,2); % POINTS 18 & 19 WITH 16 & 17 FORM THE BOUNDARY
a(18,3)=rH; % OF SURFACE NUMBER S1. THIS IS ON THE INNER SURFACE
a(19,2)=a(3,2); % OF THE HEADER.
a(19,4)=rH;
a(20,2)=a(3,2); % POINT 20 ALONG WITH 15,16 & 19 FORM THE BOUNDARY

```

```

a(20,3)=-rH;          % TO SURFACE NUMBER S2.
a(21,4)=R;           % THESE POINTS FORM SURFACE NUMBER S11.
a(22,3)=-R;
a(23,2)=a(2,2);
a(23,3)=-R;
a(24,2)=a(2,2);
a(24,4)=R;
a(25,2)=a(2,2);     % THIS POINT FORMS SURFACE NUMBER S12 ALONG WITH POINTS
a(25,3)=R;           % 21 AND 24 AND THE START AND END POINTS OF THE WELD TOE.
a(26,2)=a(3,2);     % POINTS 26 & 27 ALONG WITH 24 & 25 FORM OUTER HEADER
a(26,3)=R;           % SURFACE S9.
a(27,2)=a(3,2);
a(27,4)=R;
a(28,2)=a(3,2);     % THIS POINT FORMS THE BOUNDARY FOR SURFACE NUMBER S10 WITH
a(28,3)=-R;          % POINTS 23, 24 & 27.
a(29,2)=rcb;         % THE POINTS 29 & 30 ARE THE START AND END POINTS FOR THE
a(29,3)=a(5,3);     % ARC AT THE BOTTOM OF THE C"BORE.
a(30,3)=a(5,3);
a(30,4)=rcb;         % POINTS 29,30,31,32 FORM INTERNAL SURFACE NUMBER S6.
a(31,2)=ri;          % THE POINTS 31,32 ARE THE START AND END POINTS OF THE ARC
a(31,3)=a(6,3);     % AT THE TOP OF THE C"BORE.
a(32,3)=a(6,3);
a(32,4)=ri;         % POINTS 31,32,35,36 FORM THE INTERNAL SURFACE NUMBER S7.
if (typeval==2)|(typeval==4);
    a(33,3)=a(1,27);
    a(33,4)=rsh;
    a(34,2)=ri;
    a(34,3)=a(7,3);
    a(35,3)=a(7,3);
    a(35,4)=ri;
    a(36,2)=(hb-fl-hs-sw)/2*tan(Z1)+rsh;
    a(36,3)=a(7,3);
    a(37,3)=a(7,3);
    a(37,4)=(hb-fl-hs-sw)/2*tan(Z1)+rsh;
    a(38,2)=ri;
    a(38,3)=a(8,3);
    a(39,3)=a(8,3);
    a(39,4)=ri;
    a(40,2)=(hb-fl-hs-sw)*tan(Z1)+rsh;
    a(40,3)=a(8,3);
    a(41,4)=(hb-fl-hs-sw)*tan(Z1)+rsh;
    a(41,3)=a(8,3);
    a(42,2)=fr;
    a(42,3)=a(8,3);
    a(43,3)=a(8,3);
    a(43,4)=fr;
    a(44,2)=ri;
    a(44,3)=a(9,3);
    a(45,3)=a(9,3);
    a(45,4)=ri;
    a(46,2)=(hb-fl-hs-sw)*tan(Z1)+rsh;
    a(46,3)=a(9,3);
    a(47,3)=a(9,3);
    a(47,4)=(hb-fl-hs-sw)*tan(Z1)+rsh;
    a(48,2)=fr;
    a(48,3)=a(9,3);
    a(49,3)=a(9,3);
    a(49,4)=fr;
    a(50,4)=-R;
else;
    a(33,2)=ro;
    a(33,3)=a(7,3);
    a(34,3)=R+rg+hs+sw;
    a(34,4)=rsh;
    a(35,3)=a(7,3);
    a(35,4)=ro;
    a(36,2)=ri;
    a(36,3)=a(8,3);
    a(37,3)=a(8,3);
    a(37,4)=ri;
    a(38,2)=ro;
    a(38,3)=a(8,3);
    a(39,4)=ro;
    a(39,3)=a(8,3);
    a(40,4)=-R;

```

```

end;
if ((LL1==1)|(BT1==1))&(typeval==10);
    BCRwriteburst;
elseif ((ES1==1)&(typeval==10));
    BCRwriter;
elseif ((LL1==1)|(BT1==1))&((typeval==2)|(typeval==4));
    FCRwriteburst;
else ((ES1==1)&(typeval==2)|(typeval==4));
    FCRwriter;
end;
Y2=Y2*180/pi;
Z1=Z1*180/pi;

% =====
% SUB-PROGRAM FCRWRITER TO CREATE THE FEMGV BATCH FILE FROM INPUT DATA
% =====
%
% FILE IS ALREADY OPEN
%
fprintf(fn,"FEMGEN "); % First line containing filename data
fprintf(fn,name);
fprintf(fn,"\n");
% SET TOLERANCE FOR MODEL TO 1E-3
fprintf(fn,"CON SPA TOL ABS 1e-3\n");
% PRINT ALL POINTS FOR CONSTRUCTION OF THE REST OF THE GEOMETRY MINUS THE WELD
fprintf(fn,"G P C P%-4.0f %8.4f %8.4f %8.4f\n",a(1:50,1:4));
fprintf(fn,"EYE FR ALL\n"); % Set frame size
% CONSTRUCT A SHAPE CALLED MINN REPRESENTING THE INSIDE SURFACE OF THE HEADER.
% THIS IS NEEDED TO CREATE THE INTERSECTION LINE (BRANCH/HEADER) AND ALSO TO
% PRESS THE MESH ONTO THE INNER SURFACE.
fprintf(fn,"CON SH CYL MINN P%-4.0f",a(4,1));
fprintf(fn,"P%-4.0f",a(3,1));
fprintf(fn,"%8.4f",rH);
fprintf(fn," TRUNC\n");
% CONSTRUCT A SHAPE CALLED CBORE REPRESENTING THE INSIDE SURFACE OF THE
% COUNTER-BORE. THIS IS NEEDED TO CREATE THE INTERSECTION LINE (BRANCH/HEADER).
fprintf(fn,"CON SH CYL CBORE P%-4.0f",a(1,1));
fprintf(fn,"P%-4.0f",a(9,1));
fprintf(fn,"%8.4f",rcb);
fprintf(fn," TRUNC\n");
% CONSTRUCT A SHAPE CALLED MOUT REPRESENTING THE OUTER SURFACE OF THE HEADER.
% THIS IS NEEDED TO PRESS THE MESH ONTO THE OUTER SURFACE.
fprintf(fn,"CON SH CYL MOUT P%-4.0f",a(4,1));
fprintf(fn,"P%-4.0f",a(3,1));
fprintf(fn,"%8.4f",R);
fprintf(fn," TRUNC\n");
% CREATE GEOMETRY LINES ...
...
% CREATE SURFACES - INNER
fprintf(fn,"GE SURF 4SIDES S1 L11 L3 L12 L4\n");
fprintf(fn,"GE SURF 4SIDES S2 L12 L6 L13 L7\n");
fprintf(fn,"GE SURF 4SIDES S3 L9 L5 L10 L6\n");
fprintf(fn,"GE SURF 4SIDES S4 L8 L1 L2 INNER\n");
fprintf(fn,"GE SURF 4SIDES S5 L15 INECK1 L1 INECK2\n");
fprintf(fn,"GE SURF 4SIDES S6 L16 L24 L15 L25\n");
fprintf(fn,"GE SURF 4SIDES S7 L17 L26 L16 L27\n");
% CREATE SURFACES - OUTER
fprintf(fn,"GE SURF 4SIDES S8 L39 L31 L40 L32\n");
fprintf(fn,"GE SURF 4SIDES S9 L40 L34 L41 L35\n");
fprintf(fn,"GE SURF 4SIDES S10 L37 L33 L38 L34\n");
fprintf(fn,"GE SURF 4SIDES S11 L36 L29 L30 OUTER\n");
fprintf(fn,"GE SURF 4SIDES S12 L42 ONECK1 L29 ONECK2\n");
fprintf(fn,"GE SURF 4SIDES S13 L43 L52 L42 L53\n");
fprintf(fn,"GE SURF 4SIDES S14 L45 L54 L43 L55\n");
fprintf(fn,"GE SURF 4SIDES S15 L46 L56 L44 L57\n");
fprintf(fn,"GE BOD 2SURF S1 S8\n");
fprintf(fn,"GE BOD 2SURF S2 S9\n");
fprintf(fn,"GE BOD 2SURF S3 S10\n");
fprintf(fn,"GE BOD 2SURF S4 S11\n");
fprintf(fn,"GE BOD 2SURF S5 S12\n");
fprintf(fn,"GE BOD 2SURF S6 S13\n");
fprintf(fn,"GE BOD 2SURF S7 S14\n");
fprintf(fn,"GE BOD 2SURF S14 S15\n");

```

```

% CLEAR MESH TYPES AND THEN CHANGE TO C3D20RS. SET GLOBAL MESH DIVISIONS TO 6.
fprintf(fn,"ME TYPES ALL NONE\n");
fprintf(fn,"ME TYPES ALL HE20 C3D20R\n");
fprintf(fn,"ME DIV DEF 6\n");
fprintf(fn,"ME DIV LIN ALL\n");
% SET SPECIFIC MESH GRADATIONS
% B1 - HEADER
fprintf(fn,"ME DIV DEF 924\n");
fprintf(fn,"ME DIV LIN L11\n");
fprintf(fn,"ME DIV LIN L12\n");
fprintf(fn,"ME DIV LIN L39\n");
fprintf(fn,"ME DIV LIN L40\n");
fprintf(fn,"ME DIV DEF 8\n");
fprintf(fn,"ME DIV LIN L3\n");
fprintf(fn,"ME DIV LIN L4\n");
fprintf(fn,"ME DIV LIN L31\n");
fprintf(fn,"ME DIV LIN L32\n");
% CONTINUE TO SET MESH GRADATIONS FOR THE REST OF THE MODEL ...
...
...
fprintf(fn,"ME GEN\n"); % GENERATE MESH.
fprintf(fn,"ME SH S1 MINN\n");
fprintf(fn,"ME SH S2 MINN\n");
fprintf(fn,"ME SH S3 MINN\n");
fprintf(fn,"ME SH S4 MINN\n");
fprintf(fn,"ME SH S8 MOUT\n"); % PRESS MESH ONTO THE HEADER SHAPES.
fprintf(fn,"ME SH S9 MOUT\n");
fprintf(fn,"ME SH S10 MOUT\n");
fprintf(fn,"ME SH S11 MOUT\n");
fprintf(fn,"ME GEN\n");
fprintf(fn,"ME MER ALL\n"); % MERGE MESH.
fprintf(fn,"VIEW HIDD SHA\n"); % VIEW MESH IN 3D RENDERED AND SHADED.
fprintf(fn,"VIEW MESH\n");
fprintf(fn,"E D -1\n");
fprintf(fn,"EYE FR ALL\n");
fprintf(fn,"@#3\n");
fprintf(fn,"E D 0\n");
fprintf(fn,"EYE FR ALL\n");
fprintf(fn,"CON SE OP QUARTER\n"); % STORE QUARTER OF GEOMETRY IN A SET.
fprintf(fn,"CON SE APP ALL\n");
fprintf(fn,"CON SE CL\n");
fprintf(fn,"GE COP QUARTER HALF MIR P3 P4\n");
fprintf(fn,"CON SE OP HALF\n"); % STORE HALF OF GEOMETRY IN A SET.
fprintf(fn,"CON SE APP ALL\n");
fprintf(fn,"CON SE CL\n");
fprintf(fn,"GE COP HALF FULL MIR P21 P50\n");
fprintf(fn,"CON SE OP FULL\n"); % STORE FULL OF GEOMETRY IN A SET.
fprintf(fn,"CON SE APP ALL\n");
fprintf(fn,"CON SE CL\n");
fprintf(fn,"CON SE OP INTPRES\n");
fprintf(fn,"CON SE APP SUR S1 S2 S3 S4 S5 S6 S7\n");
fprintf(fn,"CON SE APP SURF S43 S44 S45 S46 S47 S48 S49\n");
fprintf(fn,"CON SE APP SURF S79 S80 S81 S82 S83 S84 S85\n");
fprintf(fn,"CON SE APP SURF S113 S114 S115 S116 S117 S118 S119\n");
fprintf(fn,"CON SE CL\n");
fprintf(fn,"E D 1 1 1\n");
fprintf(fn,"EYE FR ALL\n");
fprintf(fn,"V G INTPRES\n");
fprintf(fn,"@#3\n");
fprintf(fn,"V G ALL\n");
fprintf(fn,"EYE FR ALL\n");
fprintf(fn,"@#1\n");
% CREATE SETS FOR RIGHT & LEFT HAND ENDS OF THE HEADER PIPE
% AND FOR THE TOP OF THE FLANGE
fprintf(fn,"CON SE OP LHEND\n"); % APPEND NEW SURFACES TO LHEND.
fprintf(fn,"CON SE APP SURF S61 S64 S96 S98\n");
fprintf(fn,"CON SE CL\n");
fprintf(fn,"CON SE OP RHEND\n"); % APPEND NEW SURFACES TO RHEND.
fprintf(fn,"CON SE APP SURF S19 S22 S130 S132\n");
fprintf(fn,"CON SE CL\n");
fprintf(fn,"CON SE OP TOPFLG\n"); % APPEND NEW SURFACES TO TOPFLG.
fprintf(fn,"CON SE APP SURF S36 S39 S74 S76 S108 S110 S138 S139\n");
fprintf(fn,"CON SE CL\n");
% APPLY BOUNDARY CONDITIONS TO LHEND
fprintf(fn,"PROP BOUN CON C01 LHEND ENCASTRE\n");

```

```

% APPLY BOUNDARY CONDITIONS TO THE TOP OF THE FLANGE - SETNAME TOPFLG
fprintf(fn,"MESH TYPES P9 P-EL NODE-GEN\n"); % Mesh point P9 as a RB ref. node.
fprintf(fn,"PROP BOUN MPC RBEAM CO2 TOPFLG P9 ALL\n");
% APPLY MPC BOUNDARY CONDITION TO THE FREE END OF THE HEADER PIPE - SETNAME RHEND.
fprintf(fn,"MESH TYPES P3 P-EL NODE-GEN\n"); % Mesh point P3 as a RB ref. node.
fprintf(fn,"PROP BOUN MPC RBEAM CO3 RHEND P3 ALL\n");
fprintf(fn,"ME GEN\n"); % RE-GENERATE MESH.
fprintf(fn,"ME MER ALL\n"); % MERGE MESH.
fprintf(fn,"DR DIS\n");
fprintf(fn,"VIEW MESH\n");
fprintf(fn,"@#3\n");
fprintf(fn,"VIEW HIDD OFF\n");
fprintf(fn,"LAB MESH CON\n");
fprintf(fn,"@#3\n");
if (ES1==1)|(ES1==ES3);
fprintf(fn,"PROP MATERIAL MAT "); % WRITE THE MATERIAL NAME...
fprintf(fn,mat);
fprintf(fn," %8.4f",YM); % YOUNG'S MODULUS...
fprintf(fn," %8.4f",PR); % AND POISSON'S RATIO.
fprintf(fn,"\n");
fprintf(fn,"PROP ATT ALL MATERIAL "); % THEN ATTACH THESE VALUES TO THE ENTIRE MODEL
fprintf(fn,mat);
fprintf(fn,"\n");
% WRITE THE LOAD CASE HISTORY FOR AN ESF ANALYSIS
fprintf(fn,"PROP LOADS PRES LO1 1 TOPFLG %8.4f",lo1");
fprintf(fn," N\n");
fprintf(fn,"PROP LOADS PRES LO2 2 RHEND %8.4f",lo2");
fprintf(fn," N\n");
fprintf(fn,"PROP LOADS PRES LO3 3 INTPRES 1 N\n");
fprintf(fn,"PROP LOADS PRES LO4 3 TOPFLG %8.4f",lo1");
fprintf(fn," N\n");
fprintf(fn,"PROP LOADS PRES LO5 3 RHEND %8.4f",lo2");
fprintf(fn," N\n");
fprintf(fn,"PROP LOADS FORCE LO6 4 P9 1 RX\n");
fprintf(fn,"PROP LOADS FORCE LO7 5 P9 1 RY\n");
fprintf(fn,"PROP LOADS FORCE LO8 6 P9 1 RZ\n");
fprintf(fn,"PROP LOADS FORCE LO9 7 P3 1 RX\n");
fprintf(fn,"PROP LOADS FORCE LO10 8 P3 1 RY\n");
fprintf(fn,"PROP LOADS FORCE LO11 9 P3 1 RZ\n");
fprintf(fn,"VIEW HIDD FILL\n");
fclose("all");
else;
fprintf(fn,"VIEW HIDD FILL\n");
fclose("all");
end;

```

Appendix C: Reinforced Branch Outlet Range of Geometries Analysed

Model	d/D	D/T	t/T	Schedule
FB1686	0.06	9.86	0.34	XXS
FB1627	0.19	9.86	0.46	XXS
FB2090	0.25	9.86	0.50	XXS
FB1687	0.38	9.86	0.69	XXS
FB1631	0.50	9.86	0.77	XXS
FB1648	0.63	9.86	0.86	XXS
FB8001	0.75	9.86	0.99	XXS
FB8004	1.00	9.86	1.00	XXS
FB1679	0.06	17.25	0.29	S80S
FB0026	0.19	17.25	0.40	S80S
FB2087	0.25	17.25	0.44	S80S
FB1680	0.38	17.25	0.60	S80S
FB1630	0.50	17.25	0.67	S80S
FB1639	0.63	17.25	0.75	S80S
FB1681	0.75	17.25	0.86	S80S
FB8003	1.00	17.25	1.00	S80S
FB9998	0.06	26.87	0.34	S40S
FB0764	0.19	26.87	0.45	S40S
FB2084	0.25	26.87	0.48	S40S
FB9997	0.38	26.87	0.67	S40S
FB1629	0.50	26.87	0.74	S40S
FB1638	0.63	26.87	0.80	S40S
FB9996	0.75	26.87	0.87	S40S
FB9995	1.00	26.87	1.00	S40S
FB1677	0.06	58.27	0.56	S10S
FB1626	0.19	58.27	0.74	S10S
FB2081	0.25	58.27	0.74	S10S
FB1367	0.38	58.27	0.81	S10S
FB1628	0.50	58.27	0.81	S10S
FB1637	0.63	58.27	0.90	S10S
FB1678	0.75	58.27	0.90	S10S
FB8002	1.00	58.27	1.00	S10S

Table C-1: 8" run nominal bore range of reinforced branch outlet models for four different schedules

Model	d/D	D/T	t/T	Schedule
FB1658	0.05	10.75	0.29	XXS
FB1659	0.10	10.75	0.36	XXS
FB1660	0.15	10.75	0.40	XXS
FB1661	0.30	10.75	0.60	XXS
FB1662	0.50	10.75	0.75	XXS
FB1001	0.60	10.75	0.86	XXS
FB1005	0.80	10.75	0.88	XXS
FB1006	1.00	10.75	1.00	XXS
FB1657	0.05	21.50	0.29	S80S
FB1656	0.10	21.50	0.36	S80S
FB0129	0.15	21.50	0.40	S80S
FB1655	0.30	21.50	0.60	S80S
FB1654	0.50	21.50	0.75	S80S
FB1690	0.60	21.50	0.86	S80S
FB1693	0.80	21.50	1.00	S80S
FB1004	1.00	21.50	1.00	S80S
FB1103	0.05	29.46	0.30	S40S
FB1650	0.10	29.46	0.36	S40S
FB1649	0.15	29.46	0.40	S40S
FB1647	0.30	29.46	0.59	S40S
FB1643	0.50	29.46	0.71	S40S
FB1689	0.60	29.46	0.77	S40S
FB1692	0.80	29.46	0.88	S40S
FB1003	1.00	29.46	1.00	S40S
FB1652	0.05	65.17	0.50	S10S
FB1651	0.10	65.17	0.66	S10S
FB0094	0.15	65.17	0.66	S10S
FB1646	0.30	65.17	0.73	S10S
FB1642	0.50	65.17	0.81	S10S
FB1688	0.60	65.17	0.81	S10S
FB1691	0.80	65.17	0.90	S10S
FB1002	1.00	65.17	1.00	S10S

Table C-2: 10" run nominal bore range of reinforced branch outlet models for four different schedules

Model	d/D	D/T	t/T	Schedule
FB1599	0.08	12.75	0.36	XXS
FB2039	0.17	12.75	0.44	XXS
FB1560	0.33	12.75	0.68	XXS
FB1564	0.50	12.75	0.86	XXS
FB1596	0.67	12.75	0.88	XXS
FB1597	0.83	12.75	1.00	XXS
FB1598	1.00	12.75	1.00	XXS
FB1574	0.08	25.50	0.36	S80S
FB0995	0.17	25.50	0.44	S80S
FB1575	0.33	25.50	0.67	S80S
FB1576	0.50	25.50	0.86	S80S
FB1577	0.67	25.50	1.00	S80S
FB1601	0.83	25.50	1.00	S80S
FB0007	1.00	25.50	1.00	S80S
FB1578	0.08	33.98	0.35	S40S
FB2029	0.17	33.98	0.41	S40S
FB1579	0.33	33.98	0.63	S40S
FB1580	0.50	33.98	0.75	S40S
FB1581	0.67	33.98	0.86	S40S
FB1600	0.83	33.98	0.97	S40S
FB0006	1.00	33.98	1.00	S40S
FB1582	0.08	70.86	0.61	S10S
FB1153	0.17	70.86	0.61	S10S
FB1583	0.33	70.86	0.67	S10S
FB1584	0.50	70.86	0.75	S10S
FB1585	0.67	70.86	0.82	S10S
FB1599	0.83	70.86	0.92	S10S
FB0005	1.00	70.86	1.00	S10S

Table C-3: 12" run nominal bore range of reinforced branch outlet models for four different schedules

Appendix D: ESF Charts for Reinforced Branch Outlets

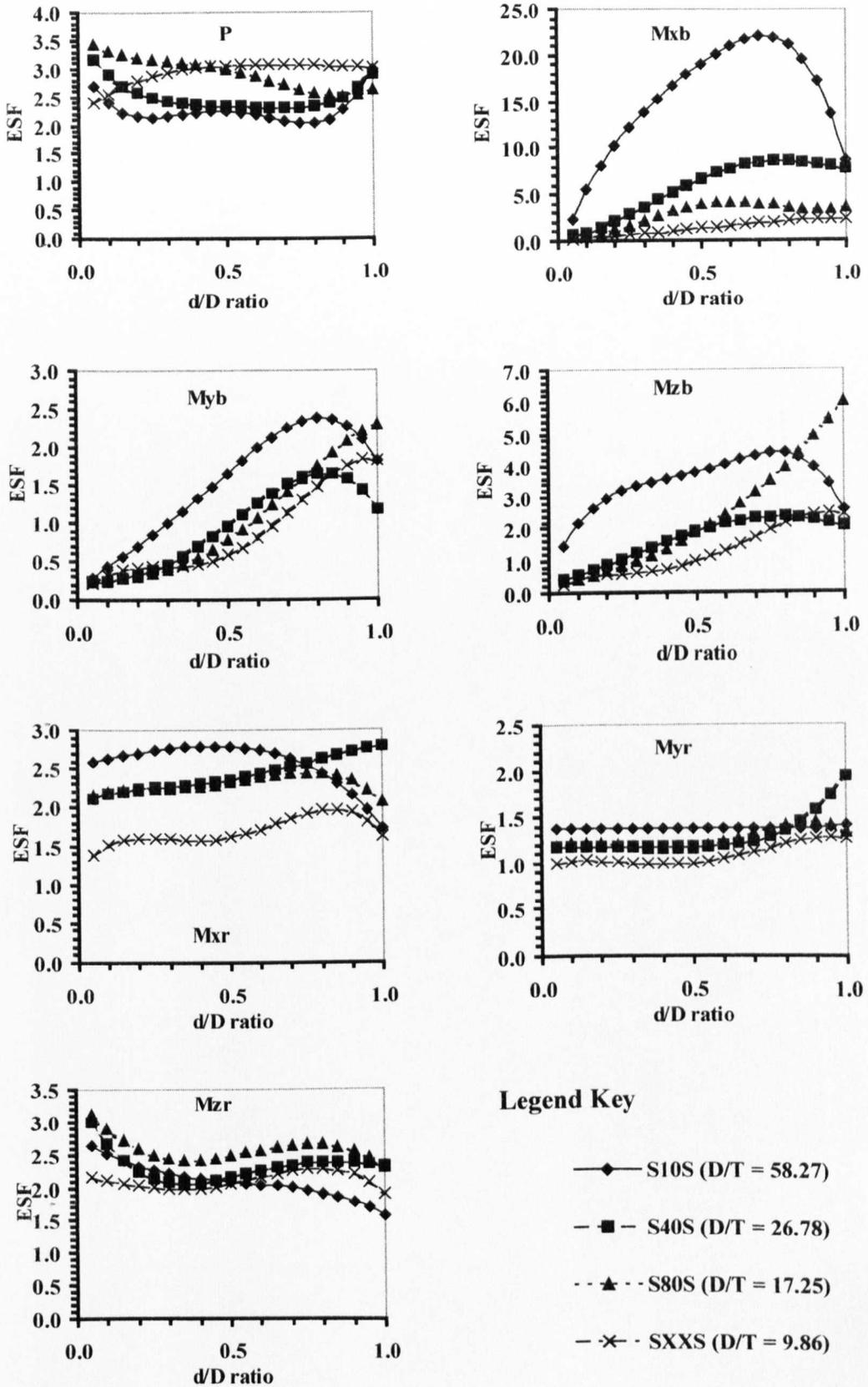


Fig. D1: 8" Run Pipe ESF Charts

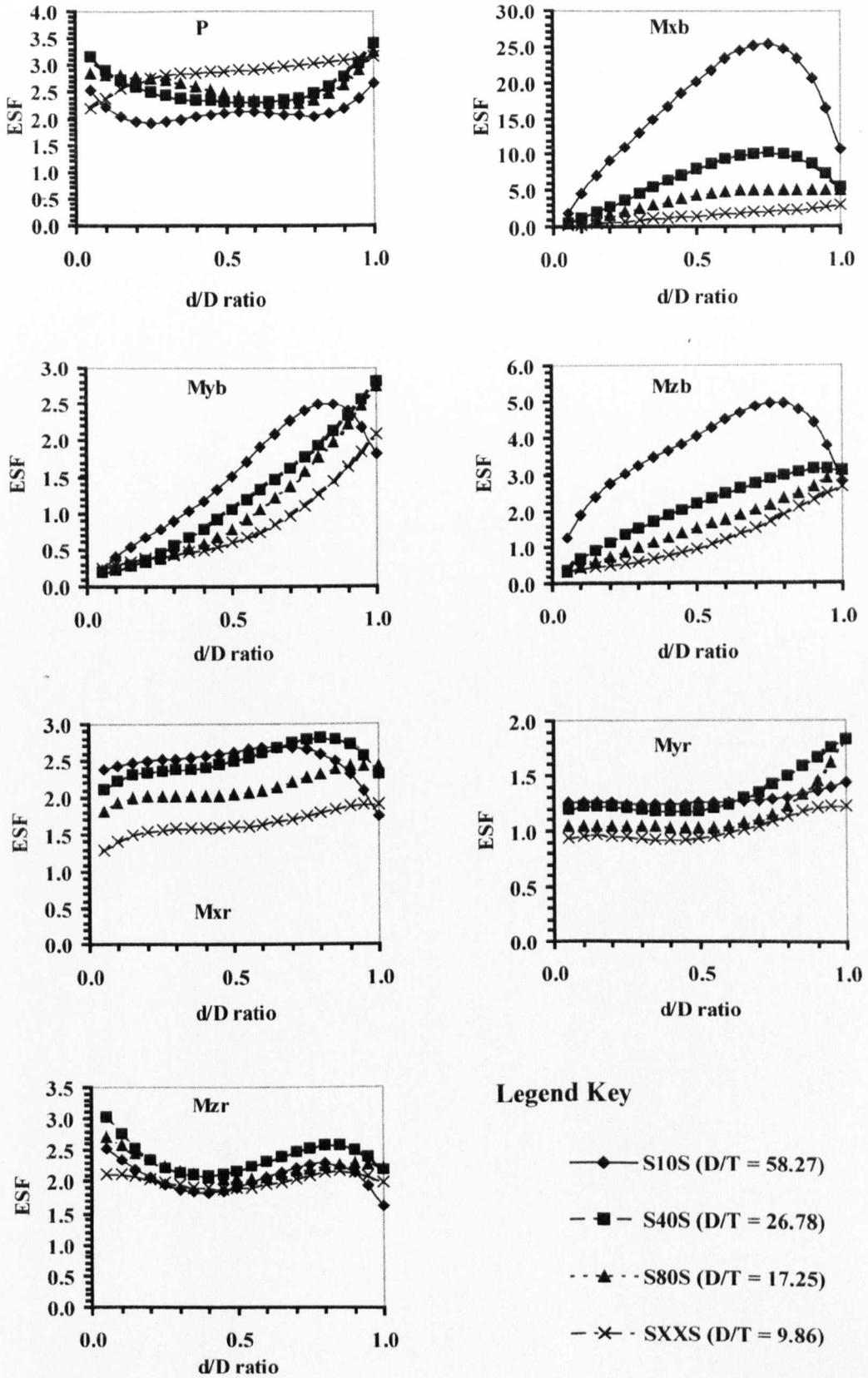


Fig. D2: 10" Run Pipe ESF Charts

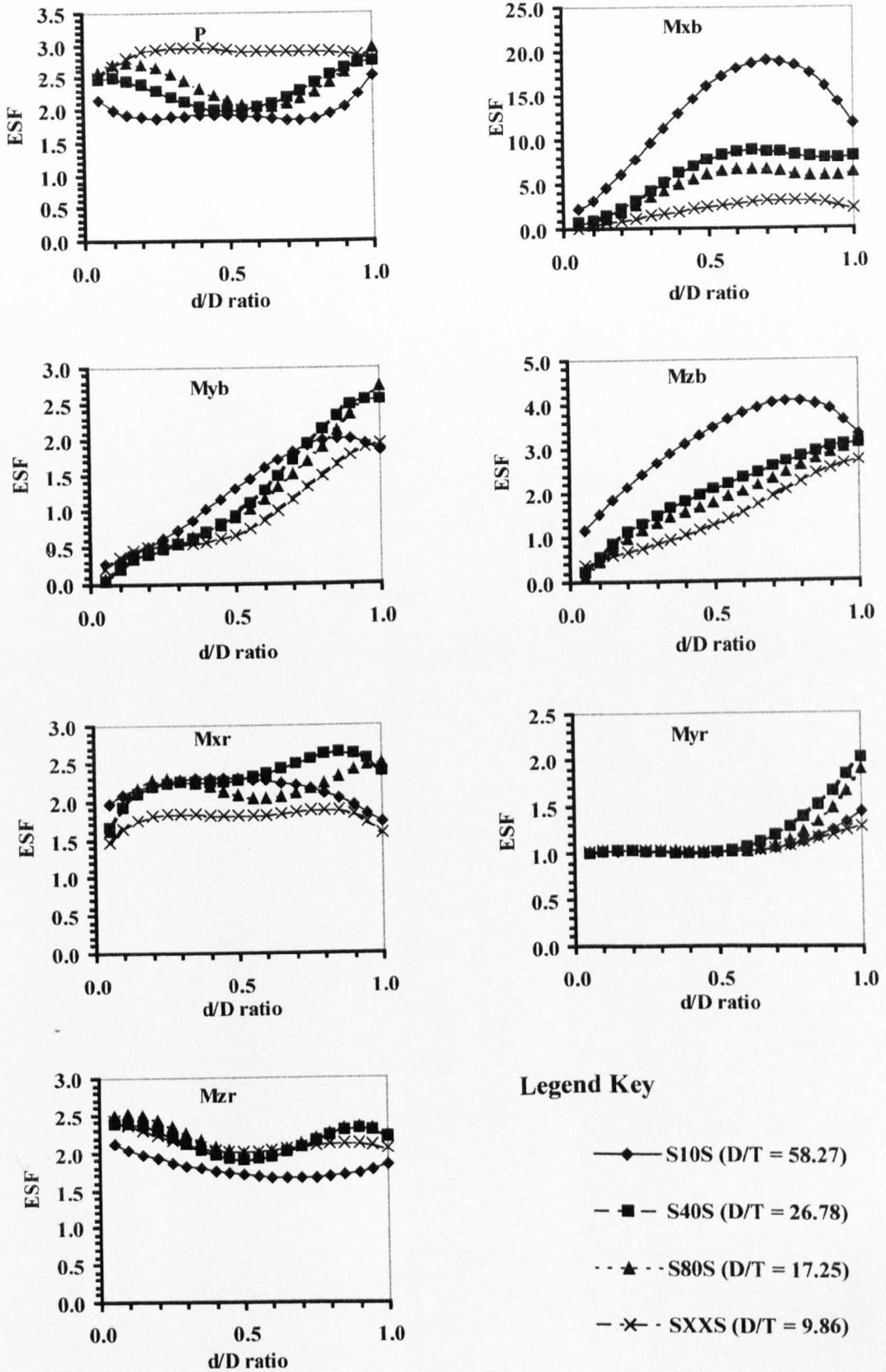
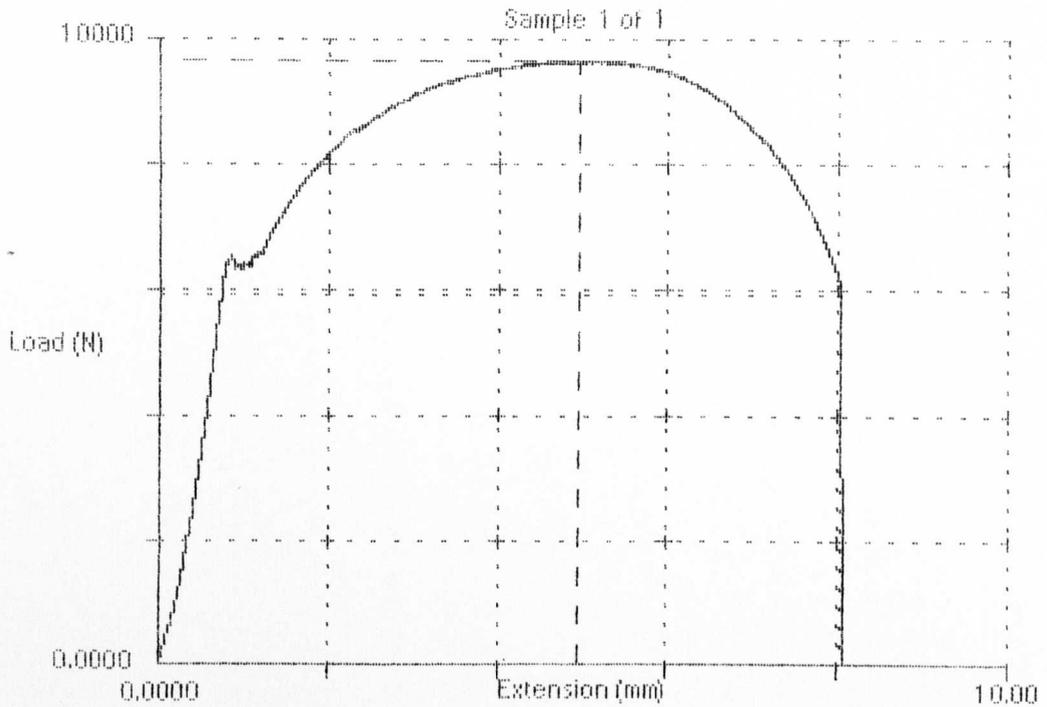


Fig. D3: 12" Run Pipe ESF Charts

**PAGES
NOT SCANNED
AT THE REQUEST OF
THE UNIVERSITY**

**SEE ORIGINAL COPY
OF THE THESIS FOR
THIS MATERIAL**

Appendix G: Material Tensile Test Data

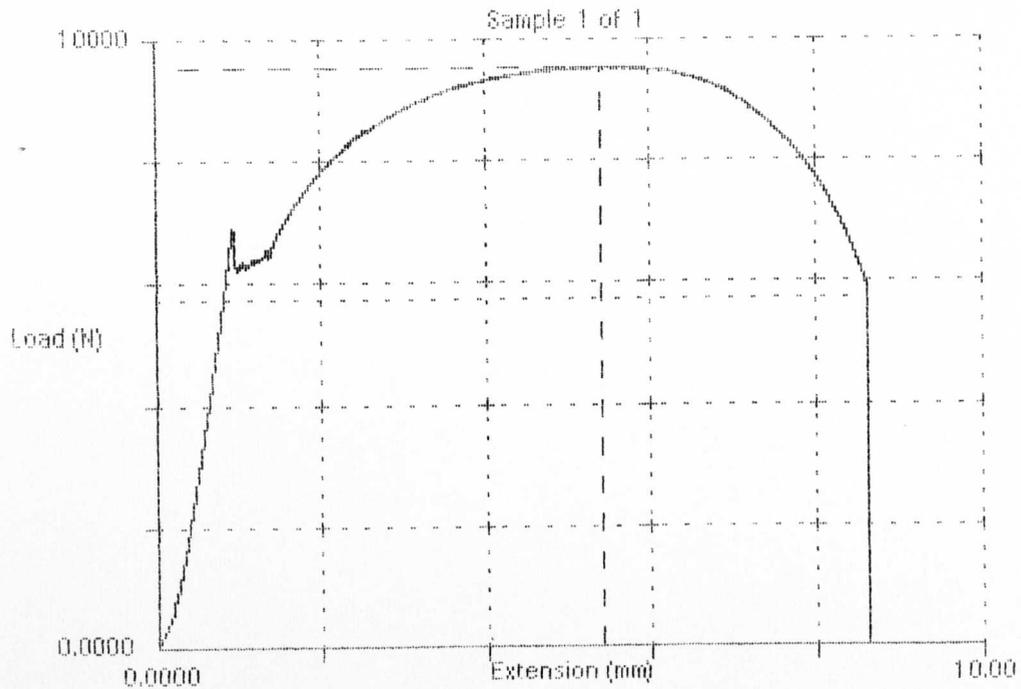


Maximum Load (N)	Ext @ Max Load (mm)	Load @ Break (N)	Ext @ Break (mm)	Work Done (N-mm)	Modulus of Elast (N/mm ²)	Sample Diameter (mm)
9679	4.928	5917	8.033	65130	10040	5.05000

Person : Jamie Finlay
 Description : 2761 Circumferential
 Form : DOG BONE
 Size : 20
 Date : Wed 4 Oct 2000

Maximum Load (N)	Ext @ Max Load (mm)	Load @ Break (N)	Ext @ Break (mm)	Work Done (N-mm)	Modulus of Elast (N/mm ²)	Sample Diameter (mm)
9679	4.928	5917	8.033	65130	10040	5.05000

d 1 : 1.000 mm/min
 gth 1 : 27.000 mm
 : 30000.00 N
 s : 0.5
 Extensometer
 d as: C:\LLOYD\DAPMAT\DATA\2761_C.FSD

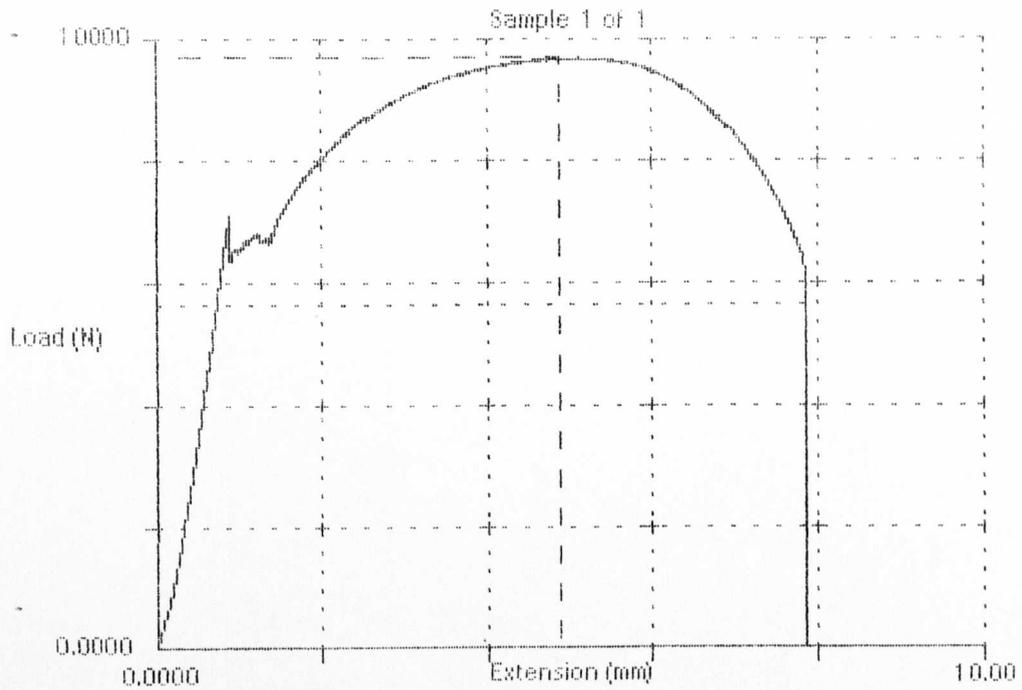


Maximum Load (N)	Ext @ Max Load (mm)	Load @ Break (N)	Ext @ Break (mm)	Work Done (N-mm)	Modulus of Elast (N/mm ²)	Sample Diameter (mm)
9558	5.396	5750	8.613	68770	9241	5.05000

Person : Jamie Finlay
 Description : 2761 Longitudinal
 Form : D06 BONE
 Age : 20
 Date : Wed 4 Oct 2000

Maximum Load (N)	Ext @ Max Load (mm)	Load @ Break (N)	Ext @ Break (mm)	Work Done (N-mm)	Modulus of Elast (N/mm ²)	Sample Diameter (mm)
9558	5.396	5750	8.613	68770	9241	5.05000

Speed : 1.000 mm/min
 Length : 27.000 mm
 Force : 30000.00 N
 Scale : 0.5
 Extensometer
 Data as: C:\LLOYD\DAPMAT\DATA\2761_L.FSD
 Output file : C:\LLOYD\DAPMAT\DATA\2761_L.WKS

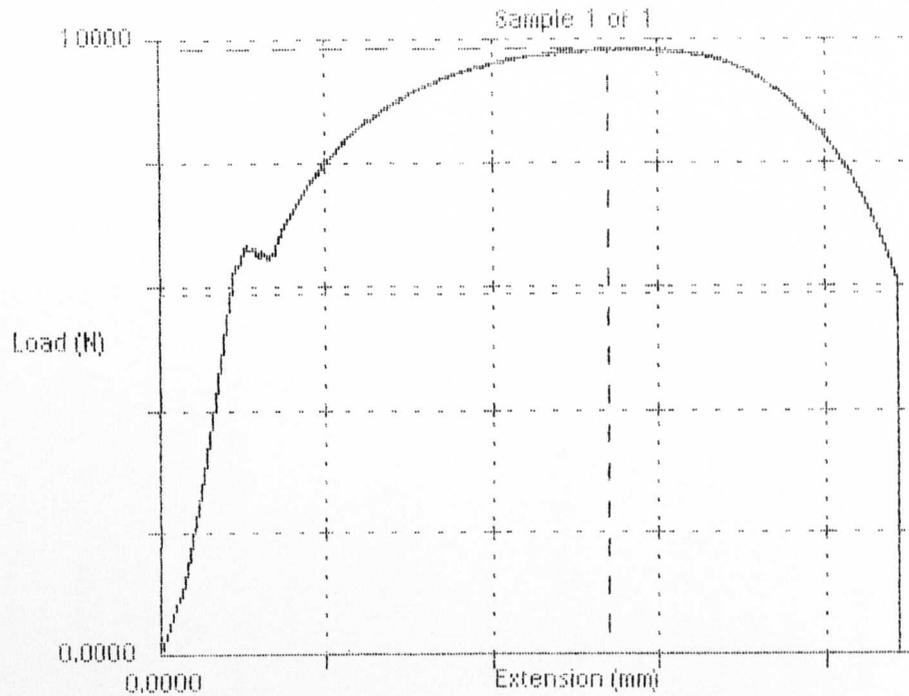


Maximum Load (N)	Ext @ Max Load (mm)	Load @ Break (N)	Ext @ Break (mm)	Work Done (N-mm)	Modulus of Elast (N/mm ²)	Sample Diameter (mm)
9695	4.876	5663	7.849	63430	9935	5.05000

person : Jamie Finlay
 description : 546 Circumferential
 form : DOG BONE
 size : 20
 date : Wed 4 Oct 2000

Maximum Load (N)	Ext @ Max Load (mm)	Load @ Break (N)	Ext @ Break (mm)	Work Done (N-mm)	Modulus of Elast (N/mm ²)	Sample Diameter (mm)
9695	4.876	5663	7.849	63430	9935	5.05000

speed 1 : 1.000 mm/min
 length 1 : 27.000 mm
 force : 30000.00 N
 size : 0.5
 Extensometer
 data file : C:\LLOYD\DAPMAT\DATA\546_C.FSD
 output file : C:\LLOYD\DAPMAT\DATA\546_C.WKS

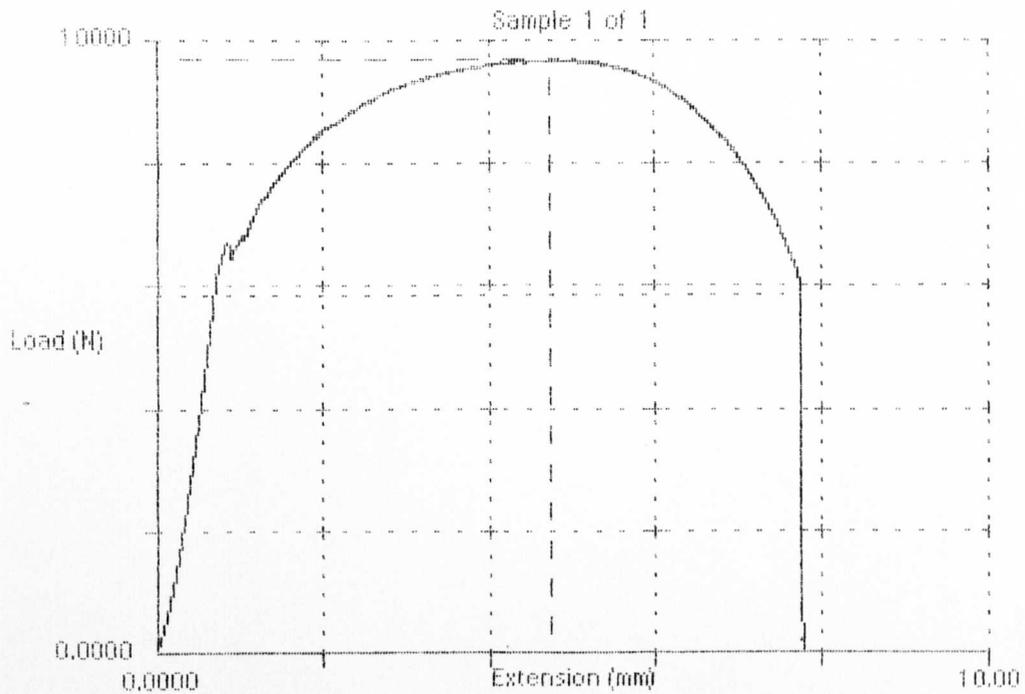


Maximum Load (N)	Ext @ Max Load (mm)	Load @ Break (N)	Ext @ Break (mm)	Work Done (N-mm)	Modulus of Elast Di (N/mm ²)
9865	5.392	5896	8.864	72970	8500

person : Jamie Finlay
 description : 54G Longitudinal
 arm : DOG BONE
 size : 20
 date : Wed 4 Oct 2000

Maximum Load (N)	Ext @ Max Load (mm)	Load @ Break (N)	Ext @ Break (mm)	Work Done (N-mm)	Modulus of Elast (N/mm ²)	Sample Diameter (mm)
9865	5.392	5896	8.864	72970	8500	5.05000

speed : 1.000 mm/min
 length : 27.000 mm
 force : 30000.00 N
 size : 0.5
 Extensometer
 saved as: C:\LLOYD\DAPMAT\DATA\54G_L.FSD
 output file : C:\LLOYD\DAPMAT\DATA\54G_L.WKS



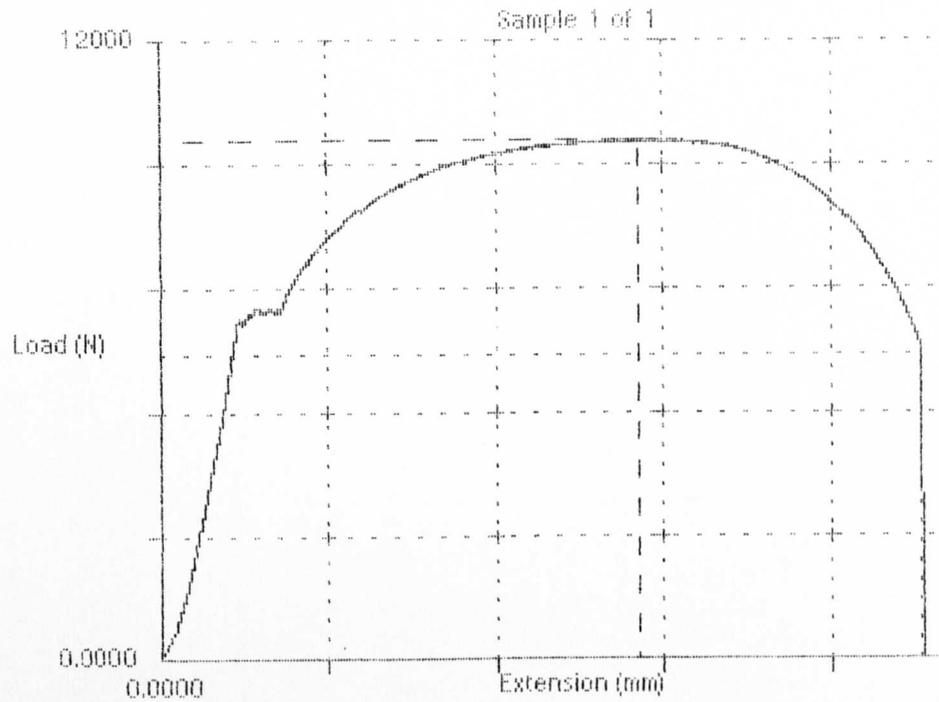
Maximum Load (N)	Ext @ Max Load (mm)	Load @ Break (N)	Ext @ Break (mm)	Work Done (N-mm)	Modulus of Elast (N/mm ²)	Sample Diameter (mm)
9689	4.736	5848	7.754	63520	10460	5.05000

Person : Jamie Finlay
 Description : B3F Circumferential
 Firm : DGG BONE
 Age : 20
 Date : Wed 4 Oct 2000

Maximum Load (N)	Ext @ Max Load (mm)	Load @ Break (N)	Ext @ Break (mm)	Work Done (N-mm)	Modulus of Elast (N/mm ²)	Sample Diameter (mm)
9689	4.736	5848	7.754	63520	10460	5.05000

Speed 1 : 1.000 mm/min
 Length 1 : 27.000 mm
 Force : 30000.00 N
 Displacement : 0.5
 Extensometer
 Path as: C:\LLOYD\DAFMAT\DATA\B3F_C.FSD
 Output file : C:\LLOYD\DAFMAT\DATA\B3F_C.WKS

83F_L

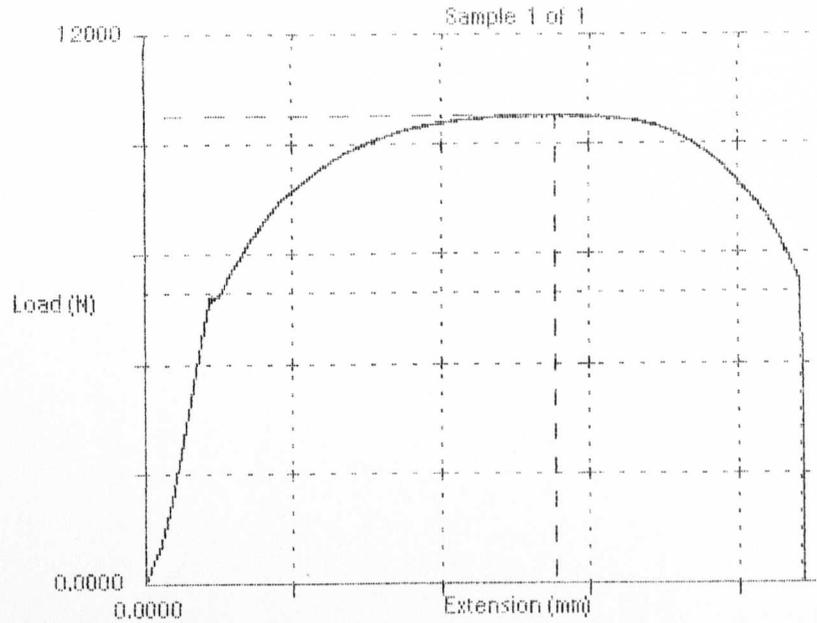


Maximum Load (N)	Ext @ Max Load (mm)	Load @ Break (N)	Ext @ Break (mm)	Work Done (N-mm)	Modulus of Elast (N/mm ²)	Sample Diameter (mm)
10080	5.680	5951	9.078	76500	8515	5.05000

Name of person : Jamie Finlay
 Sample description : 83F Longitudinal
 Sample form : DOG BONE
 Temperature : 20
 Date : Fri 6 Oct 2000

sample	Maximum Load (N)	Ext @ Max Load (mm)	Load @ Break (N)	Ext @ Break (mm)	Work Done (N-mm)	Modulus of Elast (N/mm ²)	Sample Diameter (mm)
1	10080	5.680	5951	9.078	76500	8515	5.05000

Test speed 1 : 1.000 mm/min
 Gauge length 1 : 27.000 mm
 Load cell : 30000.00 N
 Cell class : 0.5
 Internal Extensometer
 Data saved as: C:\LLOYD\DAFNAT\DATA\83F_L.FSD

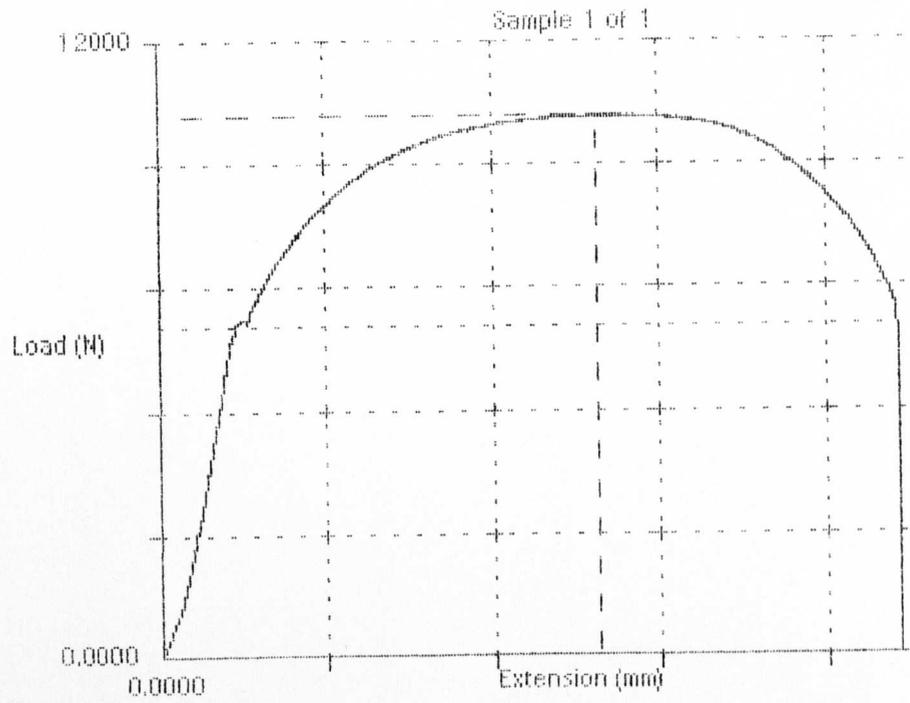


Maximum Load (N)	Ext @ Max Load (mm)	Load @ Break (N)	Ext @ Break (mm)	Work Done (N-mm)	Modulus of Elast (N/mm ²)	Sample Diameter (mm)
10250	5.520	6365	8.837	76310	8957	5.05000

Name of person : Jamie Finlay
 Sample description : A106 PIPE CIRCUMFERENTIAL
 Sample form : DOG BONE
 Temperature : 20
 Date : Fri 6 Oct 2000

sample	Maximum Load (N)	Ext @ Max Load (mm)	Load @ Break (N)	Ext @ Break (mm)	Work Done (N-mm)	Modulus of Elast (N/mm ²)	Sample Diameter (mm)
1	10250	5.520	6365	8.837	76310	8957	5.05000

Test speed 1 : 1.000 mm/min
 Gauge length 1 : 27.000 mm
 Load cell : 30000.00 N
 Cell class : 0.5
 Internal Extensometer
 Data saved as: C:\LLOYD\DAFMAT\DATA\A106_C.F80
 LOTUS 123 output file : C:\LLOYD\DAFMAT\DATA\A106_C.WKS



Maximum Load (N)	Ext @ Max Load (mm)	Load @ Break (N)	Ext @ Break (mm)	Work Done (N-mm)	Modulus of Elast (N/mm ²)	Sample Diameter (mm)
10550	5.252	6473	8.837	78460	8934	5.05000

Name of person : Jamie Finlay
 Sample description : A106 PIPE LONGITUDINAL
 Sample form : DOG BONE
 Temperature : 20
 Date : Fri 6 Oct 2000

sample	Maximum Load (N)	Ext @ Max Load (mm)	Load @ Break (N)	Ext @ Break (mm)	Work Done (N-mm)	Modulus of Elast (N/mm ²)	Sample Diameter (mm)
1	10550	5.252	6473	8.837	78460	8934	5.05000

Test speed 1 : 1.000 mm/min
 Gauge length 1 : 27.000 mm
 Load cell : 30000.00 N
 Cell class : 0.5
 Internal Extensometer
 Data saved as: C:\LLOYD\DAFMAT\DATA\A106_L.FSD
 LOTUS 123 output file : C:\LLOYD\DAFMAT\DATA\A106_L.WKS

Appendix H: Fortran and Matlab Source Code: Stress Analysis of UFT Using Neural Networks

```

INCLUDE "aba_param.inc"
DIMENSION ARRAY(513), JRRAY(NPRECD,513), LRUNIT(2,1)
EQUIVALENCE (ARRAY(1), JRRAY(1,1))

DIMENSION S(6), E(6), PS(3), PE(3), ANPS(3,3), ANPE(3,3)
CHARACTER FNAME*80

WRITE(6,*) "*** Enter the name of the input file (w/o ext.) ***"
READ(5,"(A)") FNAME

C=====
C   OPEN THE OUTPUT FILE
C=====
OPEN(UNIT=9,FILE=FNAME,STATUS="NEW")
NRU=1
LOUTF=0
LRUNIT(1,1)=8
LRUNIT(2,1)=2
CALL INITPF(FNAME,NRU,LRUNIT,LOUTF)
JUNIT=8
CALL DBRNU(JUNIT)

C=====
C   READ RECORDS FROM THE RESULTS (.fil) FILE AND PROCESS THE DATA
C   COVER A MAXIMUM OF 10 MILLION RECORDS IN THE FILE
C=====
DO 1000 K100 = 1, 100
DO 1000 K1 = 1, 99999

CALL DBFILE(0,ARRAY,JRCD)
IF (JRCD .NE. 0) GO TO 1001
KEY = JRRAY(1,2)

C=====
C   GET THE NODE NUMBER AND COORDINATE DATA
C=====
IF (KEY.EQ.1901) THEN
WRITE(9,1200) JRRAY(1,3), (ARRAY(IXX),IXX=4,6)
1200   FORMAT(2X,I5, 8X,E12.5, 3X,E12.5, 3X,E12.5,
1      3X,"0.00000E+00", 3X,"0.00000E+00")

C=====
C   INSERT A ROW OF ZEROS AT THE START OF EVERY INCREMENT
C=====
ELSE IF (KEY.EQ.2000) THEN
WRITE(9,1201)
1201   FORMAT(2X,"0.00000E+00", 3X,"0.00000E+00", 4X,"0.00000E+00",
1      4X,"0.00000E+00", 3X,"0.00000E+00", 3X,"0.00000E+00")

C =====
C   Get the element and integration point numbers, JELNUM and INTPN, and the
C   location of INTPN (0--at Integration Point, 1--at centroid, 4--Nodal
C   Average) and the number of direct and shear components in the analysis
C =====
ELSE IF (KEY .EQ. 1) THEN
JELNUM = JRRAY(1,3)
INTPN = JRRAY(1,4)
LOCATE = JRRAY(1,6)
NDI = JRRAY(1,8)
NSHR = JRRAY(1,9)
NDIP1 = NDI + 1

C =====
C   GET THE STRESS TENSOR
C =====
ELSE IF (KEY .EQ. 11) THEN
DO 10 IXX = 1, NDI
10   S(IXX) = ARRAY(IXX+2)
CONTINUE
DO 20 IYY = NDI + 1, NSHR + NDI
20   S(IYY) = ARRAY(IYY+2)
CONTINUE

C =====
C   CALCULATE THE PRINCIPAL STRESSES BY CALLING SUB SPRIND

```

```
C =====
      LSTR = 1
      CALL SPRIND(S, PS, ANPS, LSTR, NDI, NSHR)

C =====
C      WRITE ALL NODAL AVERAGED STRESSES AND PRINCIPAL STRESSES TO FILE
C =====
      WRITE(9,1204) (S(IZZ), IZZ = 1, NDI),
1          PS(1), PS(3), PS(2)
1204      FORMAT(1X, E12.5, 2X, E12.5, 3X, E12.5,
1          3X, E12.5, 2X, E12.5, 2X, E12.5)

      END IF

1000 CONTINUE
1001 CONTINUE

C =====
C      WRITE AN ADDITIONAL ROW OF ZEROS AT THE END OF THE FILE FOR USE AS A
C      POINTER IN THE MATLAB SORTING ROUTINE
C =====
      WRITE(9,1205)
1205      FORMAT(2X, "0.00000E+00", 3X, "0.00000E+00", 4X, "0.00000E+00",
1          4X, "0.00000E+00", 3X, "0.00000E+00", 3X, "0.00000E+00")

      CLOSE(UNIT=9)

      STOP
      END
```

```

% Script file: mgen.m
%
% PURPOSE OF PROGRAM:
%       To generate FEMGV history files for UFT models that can be read into
%       FEMGV automatically creating
%       the user's specified UFT pipe junction
%
% RECORD OF REVISIONS:
%       Date           Programmer           Description of Change
%       =====
%       16/10/2002     J. Finlay           Version 1.0 - Original code
%       21/10/2002     J. Finlay           Included a loop to allow users to input the
%                               number of models they want to process in
%                               one session
%
% DEFINITION OF VARIABLES:
% j       - User input number of models
% jj      - Loop counter
% fn      - Model Filename
% name    - Model Filename, w/o .his extension for use in first line of FGV history file
% dashes  - Vector containing locations of dashes in the filename
% dots    - Scalar value having the position of the period extension character
% CharRR  - Character Array stores the value of the branch to run radius ratio
% CharTR  - Character Array stores the value of the run thickness to radius ratio
% ChartT  - Character Array stores the value of the branch to run thickness ratio
% rdivR   - Array to store the branch to run radius ratio as a real number
% TdivR   - Array to store the run pipe thickness to radius ratio as a real number
% tdivT   - Array to store the branch to run thickness ratio as a real number
% rRIndex - Array containing the size of CharRR
% TRIndex - Array containing the size of CharTR
% tTIndex - Array containing the size of ChartT
% elems   - Scalar: the number of elems. around the circum. of the weld (12)
% datafile - Text file that stores the UFT's basic geometric parameters for
%           checking/referencing
% Ri      - Run pipe inside radius
% ri      - Branch inside radius
% T       - Run pipe wall thickness
% Ro      - Run pipe outside radius
% t       - Branch wall thickness
% wr      - Weld radius
% ro      - Branch outside radius
% finty   - Flank plane branch/run pipe intersection point
% mat     - Material name
% YM      - Young's Modulus
% PR      - Poisson's Ratio
% lol , lo2 - Branch and run pipe pressure equilibrium loads respectively
% a       - Vector containing the number of points to construct model geometry
% b       - Array to store all model point numbers and x,y,z coordinates for the
%           geometry

disp(" ");
disp("*****");
disp("*** Enter the # of FEMGV history files to generate in this session ** ");
disp("*** followed by the filename(s) in EXACTLY the following format: ** ");
disp("*** nn-rR-TR-tT, where: ** ");
disp("*** nn=analysts initials and/or company initials (max. 6 characters) ** ");
disp("*** rR=ratio of the branch to the run pipe (max. 4 characters) ** ");
disp("*** TR=ratio of the run pipe thickness to radius (max. 4 characters) ** ");
disp("*** tT=ratio of the branch to run pipe thickness (max 3 characters) ** ");
disp("*** N.B. ALL RATIOS MUST BE ENTERED INTO THE F/N W/O DECIMAL POINTS!!! ** ");
disp("*** EG, a FEMGV FE model created at LJMU by Jamie Finlay of a tee with the ** ");
disp("*** ratios r/R = 0.05, T/R = 0.25, t/T = 1.32 has the following filename:** ");
disp("*** JF@JMU-005-025-132 ** ");
disp("*****");
disp(" ");
% GET THE NUMBER OF MODELS THE USER WISHES TO CREATE IN ON SESSION
j=input("Number of Models: ");
for jj = 1:j;
    % GET THE MODEL FILENAME AND OPEN A FEMGV HISTORY FILE
    fn = input("Enter Filename: ','s");
    name = [fn]; % STORE THE FILENAME W/O EXTENSION - THIS WILL BE THE FEMGV FILENAME
    fn = [fn, ".his"]; % Add on the extension .his to the input filename
    % OBTAIN RADIUS, RUN THICKNESS TO RADIUS AND BRANCH TO RUN THICKNESS RATIOS
    dashes = find(fn == "-");
    dots = find(fn == ".");

```

```

CharRR = fn(dashes(1)+1:dashes(2)-1);
CharTR = fn(dashes(2)+1:dashes(3)-1);
ChartT = fn(dashes(3)+1:dots(1)-1);
% CONVERT STRINGS TO REAL NUMBERS SO THAT THEIR VALUES CAN BE USED IN CALCULATIONS
rdivR = str2num(CharRR);
  TdivR = str2num(CharTR);
  tdivT = str2num(ChartT);
  % DETERMINE HOW MANY DIGITS ARE IN EACH OF THE RATIO PARAMETERS
  rRIndex = size(CharRR);
  TRIndex = size(CharTR);
  tTIndex = size(ChartT);
  % PUT THE DECIMAL POINT IN FOR THE RADIUS RATIO...
  if rRIndex(2) == 4;
    rdivR = rdivR/1000;
  elseif rRIndex(2) == 3;
    rdivR = rdivR/100;
  elseif rRIndex(2) == 2;
    rdivR = rdivR/10;
  else;
    end;
  % ...AND FOR THE RUNPIPE THICKNESS TO RADIUS RATIO...
  if TRIndex(2) == 4;
    TdivR = TdivR/1000;
  elseif TRIndex(2) == 3;
    TdivR = TdivR/100;
  elseif TRIndex(2) == 2;
    TdivR = TdivR/10;
  else;
    end;
  % ...AND FOR THE BRANCH TO RUN THICKNESS RATIO
  if tTIndex(2) == 4;
    tdivT = tdivT/1000;
  elseif tTIndex(2) == 3;
    tdivT = tdivT/100;
  elseif tTIndex(2) == 2;
    tdivT = tdivT/10;
  else;
    end;
  % SET THE DEFAULT # OF ELEMENTS AROUND THE CIRCUMFERENCE OF THE WELD TO BE 12
  elems = 12;
  elems = elems*2; % TAKE INTO CONSIDERATION ELEMENT TYPE WITH MID-SIDE NODES
  % OPEN A DATA FILE COMPRISING THE MODEL NAME (FN) AND THE EXTENSION .TXT
  % THIS FILE IS A WAY OF VERIFYING ALL GEOMETRIC PARAMETERS THAT WERE USED IN
  % CONSTRUCTING A PARTICULAR MODEL WERE CORRECT AND CAN BE VERIFIED AGAINST
  % THE .HIS FILE AND THE FEMGV MODEL DATABASE
  datafile = [name];
  datafile = [datafile,".txt"];
  fid = fopen(datafile,"wt");
  % CALCULATE PIPE GEOMETRY PARAMETERS
  Ri = 1;
  ri = Ri*rdivR;
  T = Ri*TdivR;
  Ro = Ri+T;
  t = T*tdivT;wr=t;
  ro = ri+t;
  % WRITE THE PIPE GEOMETRY PARAMETERS TO THE DATA FILE
  fprintf(fid,"Table of geometry parameters for the model ");
  fprintf(fid,name);
  fprintf(fid,"\n");
  fprintf(fid,"\n");
  fprintf(fid,"r/R = %-8.4e",rdivR);
  fprintf(fid,"\n");
  fprintf(fid,"T/R = %-8.4e",TdivR);
  fprintf(fid,"\n");
  fprintf(fid,"t/T = %-8.4e",tdivT);
  % CLOSE THE DATAFILE
  fclose(fid);
  % CALCULATE THE LOWER WELD TOE POINT (FLANK SECTION). FIND THE ANGLE THETA
  % FIRST...
  hyp1 = Ri + T + wr;
  adj1 = ri + t + wr;
  theta = acos(adj1/hyp1);
  % ...NOW FIND THE Y AND Z COORDINATES OF THE POINT
  hyp2 = Ri + T;
  adj2 = hyp2 * cos(theta);

```

```

opp2 = sqrt(hyp2^2 - adj2^2);
% CALCULATE THE UPPER WELD TOE POINT (FLANK SECTION).
opp1 = sqrt(hyp1^2 - adj1^2);
% SET MATERIAL PROPERTIES
mat='steel';
YM=2.04e11;
PR=0.3;
% EQUILIBRIUM STRESSES FOR THE TOP OF THE BRANCH AND THE RIGHT HAND END OF THE
% HEADER PIPE (PRESSURE LOADCASE)
lo1=- (ri^2)/(ro^2-ri^2); % Branch end equilibrium stress
lo2=- (Ri^2)/(Ro^2-Ri^2); % Run end equilibrium stress
a=(1:43)';
b=zeros(43,4);
% POINTS FOR THE GEOMETRY GO HERE ...
...
% OPEN THE FILE FN.HIS AND WRITE THE HISTORY FILE
fid=fopen(fn,'wt');
fprintf(fid,'FEMGEN '); % FIRST LINE CONTAINING FILENAME DATA
fprintf(fid,name);
fprintf(fid,'\n');
% SET TOLERANCE FOR MODEL TO 1E-3
fprintf(fid,'CON SPA TOL ABS 1e-3\n');
% FILL COLUMNS 1-4 OF MATRIX "P" WITH X, Y AND Z CO-ORDINATES
% PRINT POINTS 1 TO 7 OF GEOMETRY
fprintf(fid,'G P C P%-4.0f %8.4f %8.4f %8.4f\n',b(1:7,1:4)');
fprintf(fid,'EYE FR ALL\n'); % Set frame size
fprintf(fid,'CON SH PL PLX0 X\n'); % CONSTRUCT PLANE SHAPE IN X=0
fprintf(fid,'CON SH PL PLY0 Y\n'); % CONSTRUCT PLANE SHAPE IN Y=0
fprintf(fid,'CON SH PL PLZ0 Z\n'); % CONSTRUCT PLANE SHAPE IN Z=0
% CREATE SHAPES TO CONSTRUCT INTERSECTION CURVES. MOUT & MINN WILL ALSO
% ALLOW THE MESH TO BE PRESSED THE RUN SURFACES.
fprintf(fid,'CON SH CYL MINN P%-4.0f',b(3,1)');
fprintf(fid,'P%-4.0f',b(4,1)');
fprintf(fid,'%8.4f',Ri);
fprintf(fid,' TRUNC\n');
fprintf(fid,'CON SH CYL MOUT P%-4.0f',b(3,1)');
fprintf(fid,'P%-4.0f',b(4,1)');
fprintf(fid,'%8.4f',Ro);
fprintf(fid,' TRUNC\n');
fprintf(fid,'CON SH CYL MFILL P%-4.0f',b(3,1)');
fprintf(fid,'P%-4.0f',b(4,1)');
fprintf(fid,'%8.4f',Ro+wr);
fprintf(fid,' TRUNC\n');
fprintf(fid,'CON SH CYL BINN P%-4.0f',b(1,1)');
fprintf(fid,'P%-4.0f',b(6,1)');
fprintf(fid,'%8.4f',ri);
fprintf(fid,' TRUNC\n');
fprintf(fid,'CON SH CYL BOUT P%-4.0f',b(1,1)');
fprintf(fid,'P%-4.0f',b(6,1)');
fprintf(fid,'%8.4f',ro);
fprintf(fid,' TRUNC\n');
fprintf(fid,'CON SH CYL BFILL P%-4.0f',b(1,1)');
fprintf(fid,'P%-4.0f',b(6,1)');
fprintf(fid,'%8.4f',ro+wr);
fprintf(fid,' TRUNC\n');
% NOW CREATE THE GEOMETRY LINES...
...
% CREATE SURFACES - INNER
fprintf(fid,'GE SURF 4SIDES S1 L17 L9 L15 L11\n');
fprintf(fid,'GE SURF 4SIDES S2 L15 L8 L13 L10\n');
fprintf(fid,'GE SURF 4SIDES S3 INNER L16 L5 L7\n');
fprintf(fid,'GE SURF 4SIDES S4 L14 L6 L12 L8\n');
fprintf(fid,'GE SURF 4SIDES S5 L4 L20 L5 L21\n');
fprintf(fid,'GE SURF 4SIDES S6 L18 L22 L4 L23\n');
fprintf(fid,'GE SURF 4SIDES S7 L19 L24 L18 L25\n');
% CREATE SURFACES - OUTER
fprintf(fid,'GE SURF 4SIDES S8 L38 L30 L36 L32\n');
fprintf(fid,'GE SURF 4SIDES S9 L36 L29 L34 L31\n');
fprintf(fid,'GE SURF 4SIDES S10 OUTER L37 L3 L27\n');
fprintf(fid,'GE SURF 4SIDES S11 L35 L26 L33 L29\n');
fprintf(fid,'GE SURF 4SIDES S12 L1 FILLET L3 L46\n');
fprintf(fid,'GE SURF 4SIDES S13 L39 L42 L1 L43\n');
fprintf(fid,'GE SURF 4SIDES S14 L40 L44 L39 L45\n');

```

```

% CREATE GEOMETRY BODIES
fprintf(fid,"GE BOD 2SURF S1 S8\n");
fprintf(fid,"GE BOD 2SURF S2 S9\n");
fprintf(fid,"GE BOD 2SURF S3 S10\n");
fprintf(fid,"GE BOD 2SURF S4 S11\n");
fprintf(fid,"GE BOD 2SURF S5 S12\n");
fprintf(fid,"GE BOD 2SURF S6 S13\n");
fprintf(fid,"GE BOD 2SURF S7 S14\n");
% CLEAR MESH TYPES AND THEN CHANGE TO C3D20RS
% SET NUMBER OF ELEMENTS THROUGH THICKNESS TO 3
fprintf(fid,"ME TYPES ALL NONE\n");
fprintf(fid,"ME TYPES ALL HE20 C3D20R\n");
fprintf(fid,"ME DIV DEF 6\n");
fprintf(fid,"ME DIV LIN ALL\n");
% SET SPECIFIC MESH GRADATIONS FOR EACH BODY ...
...
% STORE GEOMETRY IN SET CALLED QUARTER
fprintf(fid,"CON SE OP QUARTER\n");
fprintf(fid,"CON SE APP ALL\n");
fprintf(fid,"CON SE CL\n");
% MIRROR QUARTER GEOMETRY TO CREATE HALF
fprintf(fid,"GE COP QUARTER HALF MIR P3 P4\n");
fprintf(fid,"CON SE OP HALF\n");
fprintf(fid,"CON SE APP ALL\n");
fprintf(fid,"CON SE CL\n");
% MIRROR HALF GEOMETRY TO CREATE FULL
fprintf(fid,"GE COP HALF FULL MIR P27 P42\n");
fprintf(fid,"CON SE OP FULL\n");
fprintf(fid,"CON SE APP ALL\n");
fprintf(fid,"CON SE CL\n");
% CREATE INTERNAL PRESSURE SURFACES
fprintf(fid,"CON SE OP INTPRES\n");
fprintf(fid,"CON SE APP SUR S1 S2 S3 S4 S5 S6 S7\n");
fprintf(fid,"CON SE APP SUR S38 S39 S40 S41 S42 S43 S44\n");
fprintf(fid,"CON SE APP SUR S70 S71 S72 S73 S74 S75 S76\n");
fprintf(fid,"CON SE APP SUR S100 S101 S102 S103 S104 S105 S106\n");
fprintf(fid,"CON SE CL\n");
% CREATE LHEND BOUNDARY CONDITION SURFACE SET
fprintf(fid,"CON SE OP LHEND\n");
fprintf(fid,"CON SE APP SUR S55 S58 S86 S88\n");
fprintf(fid,"CON SE CL\n");
% CREATE RHEND BOUNDARY CONDITION SURFACE SET
fprintf(fid,"CON SE OP RHEND\n");
fprintf(fid,"CON SE APP SUR S18 S21 S116 S118\n");
fprintf(fid,"CON SE CL\n");
% CREATE BREND BOUNDARY CONDITION SURFACE SET
fprintf(fid,"CON SE OP BREND\n");
fprintf(fid,"CON SE APP SUR S35 S68 S98 S124\n");
fprintf(fid,"CON SE CL\n");
% APPLY BOUNDARY CONDITIONS TO LHEND
fprintf(fid,"PROP BOUN CON CO1 LHEND ENCASTRE\n");
% APPLY BC'S TO THE TOP OF THE BRANCH. MESH POINT P6 AS A RB REF. NODE.
fprintf(fid,"MESH TYPES P6 P-EL NODE-GEN\n");
fprintf(fid,"PROP BOUN MPC RBEAM CO2 BREND P6 ALL\n");
% APPLY MPC TO THE FREE END OF THE HEADER PIPE. MESH POINT P3 AS A RB REF. NODE.
fprintf(fid,"MESH TYPES P3 P-EL NODE-GEN\n");
fprintf(fid,"PROP BOUN MPC RBEAM CO3 RHEND P3 ALL\n");
% WRITE THE MATERIAL PROPERTIES AND LOAD CASE HISTORY FOR LINEAR STATIC ANALYSIS
% APPLY MATERIAL PROPERTIES
fprintf(fid,"PROP MATERIAL MAT ");
fprintf(fid,mat);
fprintf(fid," %8.4e",YM); % Young's Modulus
fprintf(fid," %8.4e",PR); % Poisson's Ratio
fprintf(fid,"n");
fprintf(fid,"PROP ATT ALL MATERIAL ");
fprintf(fid,mat);
fprintf(fid,"n");
% APPLY LOADS
fprintf(fid,"PROP LOADS PRES LO1 1 BREND %8.4f",lo1);
fprintf(fid," N\n");
fprintf(fid,"PROP LOADS PRES LO2 2 RHEND %8.4f",lo2);
fprintf(fid," N\n");
fprintf(fid,"PROP LOADS PRES LO3 3 BREND %8.4f",lo1);
fprintf(fid," N\n");

```

```
fprintf(fid,"PROP LOADS PRES LO4 3 RHEND %8.4f",lo2");
fprintf(fid," N\n");
fprintf(fid,"PROP LOADS PRES LO5 3 INTPRES 1 N\n");
fprintf(fid,"PROP LOADS FORCE LO6 4 P6 1 RX\n");
fprintf(fid,"PROP LOADS FORCE LO7 5 P6 1 RY\n");
fprintf(fid,"PROP LOADS FORCE LO8 6 P6 1 RZ\n");
fprintf(fid,"PROP LOADS FORCE LO9 7 P3 1 RX\n");
fprintf(fid,"PROP LOADS FORCE LO10 8 P3 1 RY\n");
fprintf(fid,"PROP LOADS FORCE LO11 9 P3 1 RZ\n");
% CLOSE FILE
fclose(fid);
if jj ~= j;
jj = jj + 1;
else;
jj = j;
end;
end;
```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%
%%% Script file: prep.m
%%%
%%% PURPOSE OF PROGRAM:
%%%     To process ASCII files containing nodal coordinate and stress data
%%%     into MATLAB arrays for manipulation and subsequent output as PIPET
%%%     training or validation files.
%%%
%%% RECORD OF REVISIONS:
%%%     Date           Programmer           Description of Change
%%%     =====
%%%     16/03/2004    J. Finlay           Version 0.1 - Original Code
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% TRAINING OR VALIDATION FILE? NO. OF MODELS TO PROCESS IN THIS SESSION
dataSet = input("Create a [T]raining or a [V]alidation data set?","s");
k = input("Number of files to process in this session? ");

% DECLARE INPUT AND TARGET CELL ARRAYS NOW
inputs = cell(1,k);
targets = cell(1,k);
clc;

% Set a loop to process all user specified files
for kk = 1:k;
    clear global;

    % DECLARATION AND DEFINITION OF VARIABLES:
    global FN;           % FILENAME OF ASCII FILE CONTAINING NODAL AND STRESS DATA
    global PNAME;       % DIRECTORY INFORMATION
    global DATA;       % ALL NODAL AND STRESS DATA
    global MAXX;        % MAX. X COORDINATE VALUE (INDEX 1)
    global IX;          % MAX. X COORDINATE VALUE (INDEX 2)
    global MAXY;        % MAX. Y COORDINATE VALUE (INDEX 1)
    global IY;          % MAX. Y COORDINATE VALUE (INDEX 2)
    global SND;         % SIZE OF THE ARRAY "NUDATA" - USED AS A LOOP INDEX
    global MAXX_DATA;   % ALL Y-POSITIVE, ZERO Z VALUED NODES WITH MAX. X COORD
    global Ri;          % RUN PIPE INSIDE RADIUS (MM)
    global Ro;          % RUN PIPE OUTSIDE RADIUS (MM)
    global T;           % RUN PIPE THICKNESS (MM)
    global ri;          % BRANCH PIPE INSIDE RADIUS
    global ro;          % BRANCH PIPE OUTSIDE RADIUS
    global t;           % BRANCH PIPE THICKNESS
    global rdivR;       % BRANCH TO RUN PIPE RADIUS RATIO
    global TdivR;       % RUN PIPE THICKNESS TO RADIUS RATIO
    global tdivT;       % BRANCH TO RUN PIPE THICKNESS RATIO
    global ROWVAL;      % LOCATIONS OF ALL ROWS OF ZEROS "DATA" MATRIX
    global YZDATA_ROWS; % LOCATIONS OF ALL FLANK DATA (X=0) FOR THE WHOLE MODEL
    global XYDATA_ROWS; % LOCATIONS OF ALL CROTCH DATA (Z=0) FOR THE WHOLE MODEL
    global FLANK;       % ALL DATA FOR THE FLANK PROFILE (QUARTER MODEL)
    global CROTCH;      % ALL DATA FOR THE CROTCH PROFILE (QUARTER MODEL)
    global NUFLANK;
    global NUCROTCH;
    global AA;
    global BB;
    global XX;
    global A;
    global B;
    global ANGL;
    global J;           % LOOP COUNTER
    global JJ;         % LOOP COUNTER
    global ANGLE;
    global B_RBREF;
    global BRANCH_MAX;
    global CS;
    global D;
    global DELETE_CROWS;
    global DELETE_FROWS;
    global FILECHECK;
    global FS;
    global HOOP;
    global HOOPMX;
    global LC_ID;       % CELL ARRAY (CONTAINING STRINGS) TO IDENTIFY A LOADCASE
    global MAXY_DATA;

```

```

global NCS;
global NFS;
global NUDATA;
global PHI;
global PLAN2X;
global PLANE;
global PLANE2;
global PLANMX;
global R_RBREF;
global RADIUS;
global RB_REFS;
global RB_SORT;
global RUN_MAX;
global S1;
global S2;
global SVAL1;
global SVAL2;
global SVALS;
global THETA;
global WR;
global X;
global XCORN;
global YCORN;
global ZCORN;
global ANGMAX;

% LET THE USER SPECIFY THE DATA FILE TO OPEN FROM THE POP-UP DIALOG
[FN PNAME] = uigetfile("*.mdf","Open");
clc;
status = ["Processing *** " FN " *** - file " num2str(kk) " out of " ...
         num2str(k) " files..."];
disp(status);

% LOAD IN THE NODAL AND STRESS DATA
DATA = load(FN);

% FIND ROWS IN "DATA" WITH ZERO VALUES (THESE ARE LOADSTEP DATA BREAKS)
ROWVAL = find(DATA(:,1) == 0.0 & DATA(:,2) == 0.0 & DATA(:,3) == 0.0 & ...
             DATA(:,4) == 0.0 & DATA(:,5) == 0.0 & DATA(:,6) == 0.0 );

% STORE NODAL DATA IN ARRAY "NUDATA"
NUDATA(1:ROWVAL(1)-1,1:4) = DATA(1:ROWVAL(1)-1,1:4);

% DETERMINE THE MAXIMUM Y AND X COORDINATE VALUES AND LOCATIONS
[MAXX IX] = max(NUDATA(:,2));
[MAXY IY] = max(NUDATA(:,3));

% REMOVE THE NODAL DATA FOR THE RIGID BODY REFERENCE NODES
R_RBREF = find(NUDATA(:,3) == 0.0 & NUDATA(:,4) == 0.0);
B_RBREF = find(NUDATA(:,2) == 0.0 & NUDATA(:,3) == MAXY & ...
             NUDATA(:,4) == 0.0);
RB_REFS = [R_RBREF B_RBREF]';

% SORT INTO ASCENDING ORDER SO THAT THE HIGHEST NODE NUMBER VALUE IS DELETED
% FROM THE ARRAY "NUDATA" FIRST
RB_SORT = sort(RB_REFS);
NUDATA(RB_SORT(2),:) = [];
NUDATA(RB_SORT(1),:) = [];
SND = size(NUDATA); % Get the size of the array "NUDATA"

% FIND ALL NODES AT THE END OF THE RUN PIPE (+VE Y-COORD, 0 Z-COORD)
RUN_MAX = find(NUDATA(:,2) == MAXX & NUDATA(:,3) >= 0.0 & ...
             NUDATA(:,4) == 0.0);
MAXX_DATA = NUDATA(RUN_MAX,:);
MAXX_DATA = sortrows(MAXX_DATA,[3]);

% GET THE GEOMETRIC PARAMETERS FOR THE RUN PIPE
Ri = MAXX_DATA(1,3);
Ro = max(MAXX_DATA(:,3));
T = Ro - Ri;

% FIND ALL NODES AT THE END OF THE BRANCH PIPE (+VE X-COORD, 0 Z-COORD)
BRANCH_MAX = find(NUDATA(:,2) >= 0.0 & NUDATA(:,3) == MAXY & ...
             NUDATA(:,4) == 0.0);
MAXY_DATA = NUDATA(BRANCH_MAX,:);

```

```

MAXY_DATA = sortrows(MAXY_DATA, [2]);

% GET THE GEOMETRIC PARAMETERS FOR THE RUN PIPE
ri = MAXY_DATA(1,2);
[ro] = max(MAXY_DATA(:,2));
t = ro - ri; WR = t;
tdivT = t/T;
rdivR = ri/Ri;
TdivR = T/Ri;

% NOW APPEND THE STRESS DATA FROM THE 7 LOAD-CASES
NUDATA(1:SND(1),5:10) = DATA(ROWVAL(3)+1:ROWVAL(4)-1,1:6);
NUDATA(1:SND(1),11:16) = DATA(ROWVAL(4)+1:ROWVAL(5)-1,1:6);
NUDATA(1:SND(1),17:22) = DATA(ROWVAL(5)+1:ROWVAL(6)-1,1:6);
NUDATA(1:SND(1),23:28) = DATA(ROWVAL(6)+1:ROWVAL(7)-1,1:6);
NUDATA(1:SND(1),29:34) = DATA(ROWVAL(7)+1:ROWVAL(8)-1,1:6);
NUDATA(1:SND(1),35:40) = DATA(ROWVAL(8)+1:ROWVAL(9)-1,1:6);
NUDATA(1:SND(1),41:46) = DATA(ROWVAL(9)+1:ROWVAL(10)-1,1:6);

% FIND NODES ON X = 0 PLANE WITH +VE Y AND Z COORDS. STORE IN "FLANK"
YZDATA_ROWS = find(NUDATA(:,2) == 0.0 & NUDATA(:,3) >= 0.0 & ...
    NUDATA(:,4) > 0.0);
FLANK = NUDATA(YZDATA_ROWS,:);

% FIND NODES ON Z = 0 PLANE WITH -VE X AND Y COORDS. STORE IN "CROTCH"
% -VE NODES ARE TAKEN FOR THE CROTCH SECTION TO ENSURE THAT THE MAXIMUM
% STRESS VALUES ARE USED THEREFORE ENSURING CONSERVATISM
XYDATA_ROWS = find(NUDATA(:,2) < 0.0 & NUDATA(:,3) > 0.0 & ...
    NUDATA(:,4) == 0.0);
CROTCH = NUDATA(XYDATA_ROWS,:);

% CONVERT X COORDS TO +VE VALUES FOR FILTERING BY THE ANGLE PARAMETER
CROTCH(:,2) = CROTCH(:,2).*-1;

% GET SIZES OF ARRAYS "CROTCH" AND "FLANK"
CS = size(CROTCH); FS = size(FLANK);

% FILTER ALL CROTCH DATA BY ANGULAR PARAMETER "ANGLE"
YCORN = 1.0+T+WR;
XCORN = ri+t+WR;
ANG1 = 2*atan2(T+WR,t+WR)/pi;
J = 0; JJ = 0;
for J = 1:1:CS(1);
    D = sqrt((YCORN-CROTCH(J,3))^2 + (XCORN-CROTCH(J,2))^2);
    ANGLE = atan2((YCORN-CROTCH(J,3)), (XCORN-CROTCH(J,2)));
    ANGLE = ANGLE*2/pi;

% APPEND THE ANGLE PARAMETER TO "CROTCH"
CROTCH(J,CS(2)+1) = ANGLE;
if (ANGLE > 1.8) | (ANGLE < -0.8);
    JJ = JJ + 1;
    DELETE_CROWS(1, JJ) = J;
    S1 = 0; S2 = 0; D = 0;
elseif (ANGLE < 0.0);
    S1 = WR/cos(0.5*pi*ANGLE);
elseif (ANGLE < 1.0);
    S1 = WR;
else;
    S1 = WR/sin(0.5*pi*ANGLE);
end;
if (ANGLE < ANG1);
    S2 = (WR+t)/cos(0.5*pi*ANGLE);
else;
    S2 = (WR+T)/sin(0.5*pi*ANGLE);
end;
RADIUS = (D - S1)/(S2 - S1);

% REMOVE ROWS FROM "CROTCH" OUTSIDE THROUGH-THICKNESS RANGE OF INTEREST
if (RADIUS > 1.02) | (RADIUS > 0.68 & RADIUS < 0.98) ...
    | (RADIUS > 0.35 & RADIUS < 0.64) | (RADIUS > 0.02 & ...
    RADIUS < 0.31);
    JJ = JJ + 1;
    DELETE_CROWS(1, JJ) = J;
else;
end;
end;

```

```

% APPEND THE RADIUS PARAMETER TO THE ARRAY "CROTCH"
CROTCH(J,CS(2)+2) = RADIUS;
end;

% REMOVE ALL ROWS FROM THE ARRAY "CROTCH" OUTSIDE -0.8 < ANGLE < 1.8
CROTCH(DELETE_CROWS,:) = [];
CS = size(CROTCH);

% REARRANGE THE CROTCH DATA IN "NUCROTCH" TO ENABLE EASY PROCESSING OF HOOP
% AND IN-PLANE STRESSES. INFILL THE GEOMETRY PARAMETERS FIRST...

NUCROTCH(1:CS(1)*7,1:12) = zeros;
NUCROTCH(1:CS(1)*7,1:2) = repmat(CROTCH(1:CS(1),CS(2)-1:CS(2)),7,1);
NUCROTCH(1:CS(1)*7,3) = repmat(ri,CS(1)*7,1);
NUCROTCH(1:CS(1)*7,4) = repmat(T,CS(1)*7,1);
NUCROTCH(1:CS(1)*7,5) = repmat(tdivT,CS(1)*7,1);
NUCROTCH(1:CS(1)*7,6) = repmat(WR,CS(1)*7,1);

% INFILL THE REMAINING CELLS OF THE ARRAY WITH THE STRESSES...
NUCROTCH(1:CS(1),7:12) = repmat(CROTCH(1:CS(1),5:10),1,1);
NUCROTCH(CS(1)+1:CS(1)*2,7:12) = repmat(CROTCH(1:CS(1),11:16),1,1);
NUCROTCH(CS(1)*2+1:CS(1)*3,7:12) = repmat(CROTCH(1:CS(1),17:22),1,1);
NUCROTCH(CS(1)*3+1:CS(1)*4,7:12) = repmat(CROTCH(1:CS(1),23:28),1,1);
NUCROTCH(CS(1)*4+1:CS(1)*5,7:12) = repmat(CROTCH(1:CS(1),29:34),1,1);
NUCROTCH(CS(1)*5+1:CS(1)*6,7:12) = repmat(CROTCH(1:CS(1),35:40),1,1);
NUCROTCH(CS(1)*6+1:CS(1)*7,7:12) = repmat(CROTCH(1:CS(1),41:46),1,1);
NCS = size(NUCROTCH); % Get the size of the array "NUCROTCH"

% IDENTIFY WHICH OF THE PRINCIPAL STRESSES IS THE LARGER INPLANE STRESS
% STORE THE MAXIMUM HOOP, LARGER INPLANE AND SMALLER INPLANE STRESSES
HOOPMX(1:7) = zeros;
PLANMX(1:7) = zeros;

% DEFINE LOADCASE IDS
LC_ID = {"P" "Mxb" "Myb" "Mzb" "Mxr" "Myr" "Mzr"};

% SET COUNTERS AND INITIATE SORT LOOP
J = 1; JJ = 0;
for JJ = 1:NCS(1);
    HOOP = NUCROTCH(JJ,9);

% FIRST CONSIDER THAT THE HOOP STRESS IS THE MAXIMUM PRINCIPAL
% (NUCROTCH(:,10)) - THEN SELECT THE LARGER INPLANE STRESS TO BE EITHER THE
% MIDDLE PRINCIPAL (NUCROTCH(:,11)) OR THE MINIMUM PRINCIPAL
% (NUCROTCH(:,12)), WHICHEVER HAS THE HIGHEST ABSOLUTE VALUE.
    if (abs(NUCROTCH(JJ,10) - NUCROTCH(JJ,9)) <= ...
        abs(NUCROTCH(JJ,11) - NUCROTCH(JJ,9)) & ...
        abs(NUCROTCH(JJ,10) - NUCROTCH(JJ,9)) <= ...
        abs(NUCROTCH(JJ,12) - NUCROTCH(JJ,9)));
        if abs(NUCROTCH(JJ,11)) > abs(NUCROTCH(JJ,12));
            PLANE = NUCROTCH(JJ,11);
            PLANE2 = NUCROTCH(JJ,12);
        else;
            PLANE = NUCROTCH(JJ,12);
            PLANE2 = NUCROTCH(JJ,11);
        end;
    elseif (abs(NUCROTCH(JJ,11) - NUCROTCH(JJ,9)) <= ...
        abs(NUCROTCH(JJ,10) - NUCROTCH(JJ,9)) & ...
        abs(NUCROTCH(JJ,11) - NUCROTCH(JJ,9)) <= ...
        abs(NUCROTCH(JJ,12) - NUCROTCH(JJ,9)));
        if abs(NUCROTCH(JJ,10)) > abs(NUCROTCH(JJ,12));
            PLANE = NUCROTCH(JJ,10);
            PLANE2 = NUCROTCH(JJ,12);
        else;
            PLANE = NUCROTCH(JJ,12);
            PLANE2 = NUCROTCH(JJ,10);
        end;
    % FINALLY CONSIDER THAT THE HOOP STRESS IS THE MINIMUM PRINCIPAL
    % (NUCROTCH(:,12)) - THEN SELECT THE LARGER INPLANE STRESS TO BE EITHER THE

```

```

% MAXIMUM PRINCIPAL (NUCROTCH(:,10)) OR THE MIDDLE PRINCIPAL
% (NUCROTCH(:,11)), WHICHEVER HAS THE HIGHEST ABSOLUTE VALUE.
    elseif abs(NUCROTCH(JJ,10)) > abs(NUCROTCH(JJ,11));
        PLANE = NUCROTCH(JJ,10);
        PLANE2 = NUCROTCH(JJ,11);
    else;
        PLANE = NUCROTCH(JJ,11);
        PLANE2 = NUCROTCH(JJ,10);
    end;

% FIND THE MAXIMUM CROTCH HOOP STRESS FOR EACH LOADCASE
    if abs(HOOP) > HOOPMX(J);
        HOOPMX(J) = abs(HOOP);
    else;
    end;

% FIND THE MAXIMUM CROTCH IN-PLANE STRESS FOR EACH LOADCASE
    if abs(PLANE) > PLANMX(J);
        PLANMX(J) = abs(PLANE);
    else;
    end;

% STORE THE VALUES OF THE HOOP AND LARGEST IN-PLANE STRESSES
NUCROTCH(JJ,7) = HOOP;
NUCROTCH(JJ,8) = PLANE;
if JJ == CS(1)*J;
    J = J + 1;
else;
end;
end;

% NORMALIZE CROTCH HOOP STRESSES AGAINST THE MAX. STRESS FOR EACH LOAD CASE
NUCROTCH(1:CS(1),9) = NUCROTCH(1:CS(1),7) ./HOOPMX(1);
NUCROTCH(CS(1)+1:CS(1)*2,9) = NUCROTCH(CS(1)+1:CS(1)*2,7) ./HOOPMX(2);
NUCROTCH(CS(1)*2+1:CS(1)*3,9) = NUCROTCH(CS(1)*2+1:CS(1)*3,7) ./HOOPMX(3);
NUCROTCH(CS(1)*3+1:CS(1)*4,9) = NUCROTCH(CS(1)*3+1:CS(1)*4,7) ./HOOPMX(4);
NUCROTCH(CS(1)*4+1:CS(1)*5,9) = NUCROTCH(CS(1)*4+1:CS(1)*5,7) ./HOOPMX(5);
NUCROTCH(CS(1)*5+1:CS(1)*6,9) = NUCROTCH(CS(1)*5+1:CS(1)*6,7) ./HOOPMX(6);
NUCROTCH(CS(1)*6+1:CS(1)*7,9) = NUCROTCH(CS(1)*6+1:CS(1)*7,7) ./HOOPMX(7);

% ... DO LIKEWISE FOR THE CROTCH IN-PLANE STRESSES
NUCROTCH(1:CS(1),10) = NUCROTCH(1:CS(1),8) ./PLANMX(1);
NUCROTCH(CS(1)+1:CS(1)*2,10) = NUCROTCH(CS(1)+1:CS(1)*2,8) ./PLANMX(2);
NUCROTCH(CS(1)*2+1:CS(1)*3,10) = NUCROTCH(CS(1)*2+1:CS(1)*3,8) ./PLANMX(3);
NUCROTCH(CS(1)*3+1:CS(1)*4,10) = NUCROTCH(CS(1)*3+1:CS(1)*4,8) ./PLANMX(4);
NUCROTCH(CS(1)*4+1:CS(1)*5,10) = NUCROTCH(CS(1)*4+1:CS(1)*5,8) ./PLANMX(5);
NUCROTCH(CS(1)*5+1:CS(1)*6,10) = NUCROTCH(CS(1)*5+1:CS(1)*6,8) ./PLANMX(6);
NUCROTCH(CS(1)*6+1:CS(1)*7,10) = NUCROTCH(CS(1)*6+1:CS(1)*7,8) ./PLANMX(7);

% STORE ALL CROTCH INPUT AND TARGET DATA INTO THE CELL ARRAYS "INPUTS" AND
% "TARGETS" RESPECTIVELY. ZERO ALL COUNTERS
a = 0; b = 0; c = 0; d = 0; JJ = 0; J = 1;
for JJ = 1:NCS(1);

% GET TAU = 0 CROTCH POSITIONS
if NUCROTCH(JJ,2) <= 0.02;
    a = a + 1;
    inputs{1,kk}{1,1}{1,J}{1,a}(1) = NUCROTCH(JJ,1);
    inputs{1,kk}{1,1}{1,J}{1,a}(2:4,:) = [rdivR TdivR tdivT];
    targets{1,kk}{1,1}{1,J}{1,a}(1,1) = NUCROTCH(JJ,9);
    targets{1,kk}{1,1}{1,J}{1,a}(2,1) = NUCROTCH(JJ,10);

% GET TAU = 0.333 CROTCH POSITIONS
elseif (NUCROTCH(JJ,2) >= 0.2) & (NUCROTCH(JJ,2) < 0.5);
    b = b + 1;
    inputs{1,kk}{1,1}{2,J}{1,b}(1) = NUCROTCH(JJ,1);
    inputs{1,kk}{1,1}{2,J}{1,b}(2:4,:) = [rdivR TdivR tdivT];
    targets{1,kk}{1,1}{2,J}{1,b}(1,1) = NUCROTCH(JJ,9);
    targets{1,kk}{1,1}{2,J}{1,b}(2,1) = NUCROTCH(JJ,10);

% GET TAU = 0.667 CROTCH POSITIONS
elseif (NUCROTCH(JJ,2) > 0.5) & (NUCROTCH(JJ,2) < 0.7);
    c = c + 1;
    inputs{1,kk}{1,1}{3,J}{1,c}(1) = NUCROTCH(JJ,1);
    inputs{1,kk}{1,1}{3,J}{1,c}(2:4,:) = [rdivR TdivR tdivT];

```

```

targets{1,kk}{1,1}{3,J}{1,c}(1,1) = NUCROTCH(JJ,9);
targets{1,kk}{1,1}{3,J}{1,c}(2,1) = NUCROTCH(JJ,10);

% GET TAU = 1.0 CROTCH POSITIONS
else NUCROTCH(JJ,2) >= 0.98;
d = d + 1;
inputs{1,kk}{1,1}{4,J}{1,d}(1) = NUCROTCH(JJ,1);
inputs{1,kk}{1,1}{4,J}{1,d}(2:4,:) = [rdivR TdivR tdivT];
targets{1,kk}{1,1}{4,J}{1,d}(1,1) = NUCROTCH(JJ,9);
targets{1,kk}{1,1}{4,J}{1,d}(2,1) = NUCROTCH(JJ,10);
end;

% GET THE MAXIMUM HOOP AND PLANE STRESSES FOR EACH LOADCASE
targets{1,kk}{1,1}{5,J}(1,1) = log(HOOPMX(J));
targets{1,kk}{1,1}{5,J}(2,1) = log(PLANMX(J));

% CHECK TO SEE IF THE LOOP HAS COME TO THE END OF THE LOADCASE. IF NOT
% CONTINUE TO FIND THE INPUT AND TARGET VALUES FOR THAT LOADCASE OTHERWISE
% START A NEW LOADCASE
if JJ == CS(1)*J;
J = J + 1;
a = 0; b = 0; c = 0; d = 0;
else;
end;
end;

% FILTER ALL FLANK DATA BY ANGULAR PARAMETER "ANGLE"
ZCORN = ri+t+WR;

% RE-DEFINE Y-COORDINATE VALUE FOR THE CENTRE OF FILLET RADIUS
YCORN = sqrt((1.0+T+WR)^2 - ZCORN^2);
THETA = acos((ri+t+WR)/(1.0+T+WR));
THETA = THETA*2/pi;
ANGMAX = 2*acos(sqrt((WR+T)^2+2*(WR+T))/(WR+T+1.0)^2)/pi;
% RE-SET COUNTER AND ANGLE PARAMETERS "ANGLE" AND "D"
J = 0; JJ = 0; D = 0; ANGLE = 0;
for J = 1:1:FS(1);
D = sqrt((YCORN-FLANK(J,3))^2 + (ZCORN-FLANK(J,4))^2);
ANGLE = atan2((YCORN-FLANK(J,3)), (ZCORN-FLANK(J,4)));
ANGLE = ANGLE*2/pi;
PHI = 0.5*pi*(ANGLE-THETA);
A = WR;
B = 1.0 + T;
X = (A+B)*cos(PHI);
AA = WR + T;
BB = 1.0;
XX = (AA+BB)*cos(PHI);
FLANK(J,FS(2)+1) = ANGLE; % Append the ANGLE parameter to "FLANK"
if (ANGLE >= THETA + 0.8*ANGMAX) | (ANGLE < -0.8);
JJ = JJ + 1;
DELETE_FROWS(1, JJ) = J;
S1 = 0; S2 = 0; D = 0;
elseif (ANGLE < 0.0);
S1 = WR/cos(0.5*pi*ANGLE);
elseif ANGLE < THETA;
S1 = WR;
else;
S1 = abs(X-sqrt(X*X-A*A-2*A*B));
end;
if ANGLE >= 1.0;
S2 = abs(XX-sqrt(XX*XX-AA*AA-2*AA*BB));
elseif ((THETA - ANGLE) < ANGMAX);
SVAL1 = (WR + t)/cos(0.5*pi*ANGLE);
SVALS(1) = SVAL1;
SVAL2 = abs(XX-sqrt(XX*XX-AA*AA-2*AA*BB));
SVALS(2) = SVAL2;
S2 = min(abs(SVALS));
else;
S2 = (WR+t)/cos(0.5*pi*ANGLE);
end;
RADIUS = (D - S1)/(S2 - S1);

% REMOVE IN "FLANK" OUTSIDE THE THROUGH-THICKNESS RANGE OF INTEREST
if (RADIUS > 1.02) | (RADIUS > 0.68 & RADIUS < 0.98) ...
| (RADIUS > 0.35 & RADIUS < 0.64) | (RADIUS > 0.02 & RADIUS < 0.31);

```

```

      JJ = JJ + 1;
      DELETE_FROWS(1, JJ) = J;
    else;
    end;
    % Append the RADIUS parameter to "FLANK"
    FLANK(J, FS(2)+2) = RADIUS;
  end;
  % Remove rows from "FLANK" outside -0.8 < ANGLE < 1.8
  FLANK(DELETE_FROWS,:) = [];
  FS = size(FLANK);

  % REARRANGE "FLANK" DATA IN "NUFLANK" TO ENABLE EASY PROCESSING OF HOOP
  % AND IN-PLANE STRESSES. INFILL THE GEOMETRY PARAMETERS FIRST...
  NUFLANK(1:FS(1)*7,1:12) = zeros;
  NUFLANK(1:FS(1)*7,1:2) = repmat(FLANK(1:FS(1), FS(2)-1:FS(2)), 7, 1);
  NUFLANK(1:FS(1)*7,3) = repmat(ri, FS(1)*7, 1);
  NUFLANK(1:FS(1)*7,4) = repmat(T, FS(1)*7, 1);
  NUFLANK(1:FS(1)*7,5) = repmat(tdivT, FS(1)*7, 1);
  NUFLANK(1:FS(1)*7,6) = repmat(WR, FS(1)*7, 1);

  % INFILL THE REMAINING CELLS OF THE ARRAY WITH THE STRESSES...
  NUFLANK(1:FS(1), 7:12) = repmat(FLANK(1:FS(1), 5:10), 1, 1);
  NUFLANK(FS(1)+1:FS(1)*2, 7:12) = repmat(FLANK(1:FS(1), 11:16), 1, 1);
  NUFLANK(FS(1)*2+1:FS(1)*3, 7:12) = repmat(FLANK(1:FS(1), 17:22), 1, 1);
  NUFLANK(FS(1)*3+1:FS(1)*4, 7:12) = repmat(FLANK(1:FS(1), 23:28), 1, 1);
  NUFLANK(FS(1)*4+1:FS(1)*5, 7:12) = repmat(FLANK(1:FS(1), 29:34), 1, 1);
  NUFLANK(FS(1)*5+1:FS(1)*6, 7:12) = repmat(FLANK(1:FS(1), 35:40), 1, 1);
  NUFLANK(FS(1)*6+1:FS(1)*7, 7:12) = repmat(FLANK(1:FS(1), 41:46), 1, 1);
  NFS = size(NUFLANK); % Get the size of the array "NUFLANK"

  % ZERO VECTORS FOR MAXIMUM HOOP AND IN-PLANE STRESSES. ZERO COUNTERS J AND JJ
  HOOPMX(1:7) = zeros;
  PLANMX(1:7) = zeros;
  J = 1; JJ = 0;
  for JJ = 1:NFS(1);
    HOOP = NUFLANK(JJ, 7);

    % FIRST CONSIDER THAT THE HOOP STRESS IS THE MAXIMUM PRINCIPAL
    % (NUFLANK(:, 10)) - THEN SELECT THE LARGER INPLANE STRESS TO BE EITHER THE
    % MIDDLE PRINCIPAL (NUFLANK(:, 11)) OR THE MINIMUM PRINCIPAL
    % (NUFLANK(:, 12)), WHICHEVER HAS THE HIGHEST ABSOLUTE VALUE.
    if (abs(NUFLANK(JJ, 10) - NUFLANK(JJ, 7)) <= abs(NUFLANK(JJ, 11) ...
        - NUFLANK(JJ, 7)) & abs(NUFLANK(JJ, 10) - NUFLANK(JJ, 7)) <= ...
        abs(NUFLANK(JJ, 12) - NUFLANK(JJ, 7)));
      if abs(NUFLANK(JJ, 11)) > abs(NUFLANK(JJ, 12));
        PLANE = NUFLANK(JJ, 11);
        PLANE2 = NUFLANK(JJ, 12);
      else;
        PLANE = NUFLANK(JJ, 12);
        PLANE2 = NUFLANK(JJ, 11);
      end;

    % NEXT CONSIDER THAT THE HOOP STRESS IS THE MIDDLE PRINCIPAL
    % (NUFLANK(:, 11)) - THEN SELECT THE LARGER INPLANE STRESS TO BE EITHER THE
    % MAXIMUM PRINCIPAL (NUFLANK(:, 10)) OR THE MINIMUM PRINCIPAL
    % (NUFLANK(:, 12)), WHICHEVER HAS THE HIGHEST ABSOLUTE VALUE.
    elseif (abs(NUFLANK(JJ, 11) - NUFLANK(JJ, 7)) <= ...
        abs(NUFLANK(JJ, 10) - NUFLANK(JJ, 7)) & ...
        abs(NUFLANK(JJ, 11) - NUFLANK(JJ, 7)) <= ...
        abs(NUFLANK(JJ, 12) - NUFLANK(JJ, 7)));
      if abs(NUFLANK(JJ, 10)) > abs(NUFLANK(JJ, 12));
        PLANE = NUFLANK(JJ, 10);
        PLANE2 = NUFLANK(JJ, 12);
      else;
        PLANE = NUFLANK(JJ, 12);
        PLANE2 = NUFLANK(JJ, 10);
      end;

    % FINALLY CONSIDER THAT THE HOOP STRESS IS THE MINIMUM PRINCIPAL
    % (NUFLANK(:, 12)) - THEN SELECT THE LARGER INPLANE STRESS TO BE EITHER THE
    % MAXIMUM PRINCIPAL (NUFLANK(:, 10)) OR THE MIDDLE PRINCIPAL
    % (NUFLANK(:, 11)), WHICHEVER HAS THE HIGHEST ABSOLUTE VALUE.
    elseif abs(NUFLANK(JJ, 10)) > abs(NUFLANK(JJ, 11));
      PLANE = NUFLANK(JJ, 10);
  end;

```

```

        PLANE2 = NUFLANK(JJ,11);
    else;
        PLANE = NUFLANK(JJ,11);
        PLANE2 = NUFLANK(JJ,10);
    end;

% FIND THE MAXIMUM FLANK HOOP STRESS FOR EACH LOADCASE
    if abs(HOOP) > HOOPMX(J);
        HOOPMX(J) = abs(HOOP);
    else;
    end;

% FIND THE MAXIMUM FLANK IN-PLANE STRESS FOR EACH LOADCASE
    if abs(PLANE) > PLANMX(J);
        PLANMX(J) = abs(PLANE);
    else;
    end;

% STORE THE VALUE OF THE LARGEST IN-PLANE STRESS
NUFLANK(JJ,8) = PLANE;

% CHECK TO SEE IF THE LOOP HAS COME TO THE END OF THE LOADCASE. IF NOT
% CONTINUE TO FIND THE MAXIMUM STRESS VALUES FOR THAT LOADCASE OTHERWISE
% START A NEW LOADCASE
    if JJ == FS(1)*J;
        J = J + 1;
    else;
    end;
end;

% NORMALIZE FLANK HOOP STRESSES AGAINST THE MAX. STRESS FOR EACH LOAD CASE
NUFLANK(1:FS(1),9) = NUFLANK(1:FS(1),7)/HOOPMX(1);
NUFLANK(FS(1)+1:FS(1)*2,9) = NUFLANK(FS(1)+1:FS(1)*2,7)/HOOPMX(2);
NUFLANK(FS(1)*2+1:FS(1)*3,9) = NUFLANK(FS(1)*2+1:FS(1)*3,7)/HOOPMX(3);
NUFLANK(FS(1)*3+1:FS(1)*4,9) = NUFLANK(FS(1)*3+1:FS(1)*4,7)/HOOPMX(4);
NUFLANK(FS(1)*4+1:FS(1)*5,9) = NUFLANK(FS(1)*4+1:FS(1)*5,7)/HOOPMX(5);
NUFLANK(FS(1)*5+1:FS(1)*6,9) = NUFLANK(FS(1)*5+1:FS(1)*6,7)/HOOPMX(6);
NUFLANK(FS(1)*6+1:FS(1)*7,9) = NUFLANK(FS(1)*6+1:FS(1)*7,7)/HOOPMX(7);

% ... DO LIKEWISE FOR THE FLANK IN-PLANE STRESSES
NUFLANK(1:FS(1),10) = NUFLANK(1:FS(1),8)/PLANMX(1);
NUFLANK(FS(1)+1:FS(1)*2,10) = NUFLANK(FS(1)+1:FS(1)*2,8)/PLANMX(2);
NUFLANK(FS(1)*2+1:FS(1)*3,10) = NUFLANK(FS(1)*2+1:FS(1)*3,8)/PLANMX(3);
NUFLANK(FS(1)*3+1:FS(1)*4,10) = NUFLANK(FS(1)*3+1:FS(1)*4,8)/PLANMX(4);
NUFLANK(FS(1)*4+1:FS(1)*5,10) = NUFLANK(FS(1)*4+1:FS(1)*5,8)/PLANMX(5);
NUFLANK(FS(1)*5+1:FS(1)*6,10) = NUFLANK(FS(1)*5+1:FS(1)*6,8)/PLANMX(6);
NUFLANK(FS(1)*6+1:FS(1)*7,10) = NUFLANK(FS(1)*6+1:FS(1)*7,8)/PLANMX(7);

% STORE ALL FLANK INPUT AND TARGET DATA INTO THE CELL ARRAYS "INPUTS" AND
% "TARGETS" RESPECTIVELY. SET COUNTERS
a = 0; b = 0; c = 0; d = 0; JJ = 0; J = 1;
for JJ = 1:NFS(1);

% GET TAU = 0 FLANK POSITIONS
if NUFLANK(JJ,2) <= 0.02;
    a = a + 1;
    inputs{1,kk}{1,2}{1,J}{1,a}(1) = NUFLANK(JJ,1);
    inputs{1,kk}{1,2}{1,J}{1,a}(2:4,:) = [rdivR TdivR tdivT];
    targets{1,kk}{1,2}{1,J}{1,a}(1,1) = NUFLANK(JJ,9);
    targets{1,kk}{1,2}{1,J}{1,a}(2,1) = NUFLANK(JJ,10);

% GET TAU = 0.333 FLANK POSITIONS
elseif (NUFLANK(JJ,2) >= 0.2) & (NUFLANK(JJ,2) < 0.5);
    b = b + 1;
    inputs{1,kk}{1,2}{2,J}{1,b}(1) = NUFLANK(JJ,1);
    inputs{1,kk}{1,2}{2,J}{1,b}(2:4,:) = [rdivR TdivR tdivT];
    targets{1,kk}{1,2}{2,J}{1,b}(1,1) = NUFLANK(JJ,9);
    targets{1,kk}{1,2}{2,J}{1,b}(2,1) = NUFLANK(JJ,10);

% GET TAU = 0.667 FLANK POSITIONS
elseif (NUFLANK(JJ,2) > 0.5) & (NUFLANK(JJ,2) < 0.7);
    c = c + 1;
    inputs{1,kk}{1,2}{3,J}{1,c}(1) = NUFLANK(JJ,1);
    inputs{1,kk}{1,2}{3,J}{1,c}(2:4,:) = [rdivR TdivR tdivT];
    targets{1,kk}{1,2}{3,J}{1,c}(1,1) = NUFLANK(JJ,9);

```

```

    targets{1, kk}{1, 2}{3, J}{1, c}(2, 1) = NUFLANK(JJ, 10);

    % GET TAU = 1.0 FLANK POSITIONS
    else NUFLANK(JJ, 2) >= 0.98;
        d = d + 1;
        inputs{1, kk}{1, 2}{4, J}{1, d}(1) = NUFLANK(JJ, 1);
        inputs{1, kk}{1, 2}{4, J}{1, d}(2:4, :) = [rdivR TdivR tdivT];
        targets{1, kk}{1, 2}{4, J}{1, d}(1, 1) = NUFLANK(JJ, 9);
        targets{1, kk}{1, 2}{4, J}{1, d}(2, 1) = NUFLANK(JJ, 10);
    end;

    % GET THE MAX. HOOP AND PLANE STRESSES FOR EACH LOADCASE, FLANK PROFILE
    targets{1, kk}{1, 2}{5, J}(1, 1) = log(HOOPMX(J));
    targets{1, kk}{1, 2}{5, J}(2, 1) = log(PLANMX(J));

    % CHECK TO SEE IF THE LOOP HAS COME TO THE END OF THE LOADCASE. IF NOT
    % CONTINUE TO FIND THE INPUT AND TARGET VALUES FOR THAT LOADCASE OTHERWISE
    % START A NEW LOADCASE
    if JJ == FS(1)*J;
        J = J + 1;
        a = 0; b = 0; c = 0; d = 0;
    else;
        end;
    end;

    % CHECK TO SEE IF THE "NO. OF FILES TO PROCESS" TARGET HAS BEEN MET.
    if kk ~= k;
        kk = kk + 1;
    else;
        kk = k;
    end;

    % SAVE ALL DATA FROM THE "INPUTS" AND "TARGETS" ARRAYS INTO FILENAMES
    % BASED ON WHETHER THE DATA CONTAINED IN THE ARRAYS IS FOR TRAINING OR
    % VALIDATION PURPOSES
    inpFile = strcat(dataSet, "eval_inp.mat");
    tarFile = strcat(dataSet, "eval_tar.mat");
    save(inpFile, "inputs");
    save(tarFile, "targets");
end;

```

Appendix I: Unix Shell Script Source Code

```

#!/bin/ksh

# Version 0.1.2: Changes made include driving the error down to less than 30%
# for a given loadcase and forcing conversion for a job having the number of
# neurons that first successfully broke the 30% barrier

# To enter de-debugging mode and print to screen all commands and values as the
# program runs, uncomment the next line
#set -x

# Find out how many training and validation files there are in the current
# working directory to be processed. If the number of training files does not
# equal the number of validation files report the error and exit.

tfcount=$(ls *.tra | wc -l)
vfcount=$(ls *.val | wc -l)

if (( $tfcount != $vfcount )); then
    echo "Error: The number of training and validation files to be processed does not
match."
    exit 1;
fi

# List the training and validation files to be processed. Within the loop determine
# determine the name of the current validation file ($validfile) that matches
# the current (truncated) training filename ($truncfile). Concatenate the current
# training file and validation file to the .csv file.

tfiles=$(ls *.tra)
vfiles=$(ls *.val)

for trainfile in $tfiles;do
    truncfile=${trainfile%.tra}
    validfile=$(ls $truncfile.val)
    j=1
    n=12
    maxValErr=300
    success=0

# Count the number of lines in the training file and validation files, store the
# values in tlines and vlines respectively.

    tlines=$(wc -l $trainfile | awk {"print $1"})
    vlines=$(wc -l $validfile | awk {"print $1"})

# Start the loop to run qckprp for each input file until the validation error
# $valErr has been met 3 times

    while [ $success -lt 7 ];do

# Store the first of three header lines in the appropriate .csv file

        echo 5,$n,1,$tlines,$vlines > ${truncfile}-${n}-${j}.csv

# Create a random number seed using a pseudorandom generated number.

        seed="$RANDOM"
        echo 20000,1000,5,$seed,1,0.0,4 >> ${truncfile}-${n}-${j}.csv
        echo 101,102,103,104,105,0,0,0 >> ${truncfile}-${n}-${j}.csv

# Append the training and validation file data in ${truncfile}-${n}-${j}.csv.

        cat $trainfile >> ${truncfile}-${n}-${j}.csv
        cat $validfile >> ${truncfile}-${n}-${j}.csv

# Run the qckprp.solaris training program for the .csv input file, 6 times. After
# each run concatenate the output text file and rename all of the associated fort
# files to match the filename of the completed job.

        sDate=`date +%d/%m/%y`
        sTime=`date +%H:%M:%S`
        echo ""
        echo "**** Started job no. ${j} on ${sDate} at ${sTime} ****"
        echo "**** Processing file ${truncfile}-${n}-${j}.csv with ${n} HL neurons
****"

```

```

./qckprp.solaris < ${truncfile}-${n}-${j}.csv > ${truncfile}-${n}-${j}-
output.txt & wait
cat ${truncfile}-${n}-${j}-output.txt >> ${truncfile}_cat.txt

mv fort.3 ${truncfile}-${n}-${j}-fort.3
mv fort.4 ${truncfile}-${n}-${j}-fort.4
mv fort.8 ${truncfile}-${n}-${j}-fort.8
fDate=`date +%d/%m/%y`
fTime=`date +%H:%M:%S`
echo "--- Job number ${j} for ${truncfile}-${n}-${j}.csv completed on
${fDate} at ${fTime} ---"
echo ""

# Get the validation percentage error, expressed as an integer, then evaluate it against
# the target vaidaiion error of $maxValErr%

valErr=$(grep 20000 ${truncfile}-${n}-${j}-output.txt | awk {"print $10"}
| sed 's/0[^\.]*.//');
if [ $valErr -gt $maxValErr ] && [ $success -eq 0 ];then
n=$((n + 3))
elif [ $valErr -le $maxValErr ]; then
success=$((success + 1))
maxValErr=$valErr
else [ $valErr -gt $maxValErr ] && [ $success -gt 0 ]
success=$success
valErr=$valErr
fi

j=$((j + 1))
done

# Move all files associated with the current job to a directory named according to
# the name of the current job

mkdir ${truncfile}-${fDate}
mv $truncfile*. * ${truncfile}-${fDate}
done
exit 0;

```

```

#!/bin/ksh

#####
###
### This script goes through each directory and copies the concatenated files into a
### single directory ready for moving to another system
###
#####

# List the directories to be accessed.
dirs=$(ls -a | grep "C")
for curDir in $dirs;do
    cd $curDir
    rm -rf *.tmp *.txt *.csv *.val *.tra *.xls

# Find the number of fort.8 files in current working dir

nfiles=$(ls *.8)
for file in $nfiles;do
    t=$(grep 20000 ${file} | awk {"print $7"} | sed 's/0[^.]*/"/)
    v=$(grep 20000 ${file} | awk {"print $9"} | sed 's/0[^.]*/"/)
    let ave=$((t/2+v/2))
    echo $file $ave >> ${curDir}.tmp
done
cat ${curDir}.tmp | sort +1 >> ${curDir}.txt
rm -rf 3 *.tmp
cd ../
done

```