University of Pretoria etd – Viljoen, A (2004)

Analysis and Optimum Design of Stiffened Shear Webs in Airframes

by
Awie Viljoen

A dissertation submitted in partial fulfillment
of the requirements for the degree of

Master of Engineering (Mechanical)

in the Faculty of Engineering, Built Environment and Information Technology,
University of Pretoria

March 2004

Supervisor:
Prof. Albert A. Groenwold

University of Pretoria etd – Viljoen, A (2004)

Figure 1: An illustrative example of shear buckling (Reproduced from *Flight International*, 8-14 September, 1999)

University of Pretoria etd – Viljoen, A (2004)

Abstract

Title:	Analysis and Optimum Design of Stiffened Shear Webs in Airframes
Author:	Awie Viljoen
Supervisor:	Prof. A.A. Groenwold
Co-supervisor:	Mr A. G. Visser (CSIR)
Department:	Department of Mechanical Engineering
Degree:	Master of Engineering
Keywords:	Post-buckling analysis, Grisham algorithm, diagonal tension, NACA, non-linear finite element analysis, shear web, aircraft structures, FORTRAN program, structural optimization, genetic algorithm

The analysis and optimum design of stiffened, shear webs in aircraft structures is addressed. The post-buckling behavior of the webs is assessed using the iterative algorithm developed by Grisham. This method requires only linear finite element analyses, while convergence is typically achieved in as few as five iterations. The Grisham algorithm is extensively compared with empirical analysis methods previously used for aircraft structures and also with a refined, non-linear quasi-static finite element analysis.

The Grisham algorithm provides for both compressive buckling in two directions as well as shear buckling, and overcomes some of the conservatism inherent in conventional methods of analysis. In addition, the method is notably less expensive than a complete non-linear finite element analysis, even though global collapse cannot be predicted. While verification of the analysis methodology is the main focus of the study, an initial investigation into optimization is also made. In optimizing stiffened thin walled structures, the Grisham algorithm is combined with a genetic algorithm. Allowable stress constraints are accommodated using a simple penalty formulation.

University of Pretoria etd – Viljoen, A (2004)

Opsomming

Titel:	Analise en Optimale Ontwerp van Verstyfde Skuifpanele in Lugrame
Outeur:	Awie Viljoen
Leier:	Prof. A.A. Groenwold
Mede-Leier:	Mnr A. G. Visser (WNNR)
Departement:	Departement Meganiese Ingenieurswese
Graad:	Meester van Ingenieurswese
Sleutelwoorde:	Post-knik analise, Grisham algoritme, diagonale trekspanning, NACA, nie-lineêre eindige element analise, skuifpaneel, vliegtuig strukture, FORTRAN program, struktuur optimering, genetiese algoritme

Die analise en optimale ontwerp van verstyfde, dunwand skuifpanele in lugvaartstrukture word aangespreek. Die post-knik gedrag van die panele word evalueer met 'n iteratiewe algoritme wat deur Grisham ontwikkel is. Hierdie metode gebruik slegs lineêre eindige element analises en konvergeer gewoonlik binne so min soos vyf iterasies. Grisham se algoritme word vergelyk met empiriese metodes wat voorheen in lugvaartstruktuuranalise gebruik is, asook met 'n nie-lineêre eindige element analise.

Daar word getoon dat die Grisham algoritme voorsiening maak vir samedrukkingsknik in twee rigtings asook knik in skuif, en die algoritme oorkom die konserwatiewe benadering wat in konvensionele ontwerpsmetodes gebruik word. Boonop is die metode ook nie so duur soos nie-lineêre eindige element analises nie, alhoewel dit nie globaal die swigting van 'n struktuur kan voorspel nie. Alhoewel verifikasie van die analiseringsmetodologie die hoof doel van hierdie studie is, word 'n aanvanklike ondersoek in optimering ook gedoen. Om die verstyfde, dunwand strukture te optimeer word die Grisham algoritme gekombineer met 'n genetiese algoritme. Aanvaarbare spanningswaardes word in ag geneem deur 'n eenvoudige boetefunksie te gebruik.

University of Pretoria etd – Viljoen, A (2004)

Acknowledgements

I would like to express my sincere gratitude towards the following persons:

- Prof Albert Groenwold from the University of Pretoria.
- Mr Andries Visser from the CSIR.
- Dr Oliver Damm from the CSIR.

Financial support granted by the CSIR for attending the Fourth World Congress of Structural and Multidisciplinary Optimization (WCSMO-4), held in Dalian, China (4 - 8 June 2001), is gratefully acknowledged and appreciated.

University of Pretoria etd – Viljoen, A (2004)

Contents

University of Pretoria etd – Viljoen, A (2004)

University of Pretoria etd – Viljoen, A (2004)

University of Pretoria etd – Viljoen, A (2004)

List of Figures

University of Pretoria etd – Viljoen, A (2004)

University of Pretoria etd – Viljoen, A (2004)

University of Pretoria etd – Viljoen, A (2004)

University of Pretoria etd – Viljoen, A (2004)

List of Tables

University of Pretoria etd – Viljoen, A (2004)

University of Pretoria etd – Viljoen, A (2004)

List of symbols

A_u	-	Upright area
A_f	-	Flange cross-sectional area
A_{f_u}	-	Upper flange cross-sectional area
A_{f_l}	-	Lower flange cross-sectional area
c	-	Damping factor
d	-	Spacing of uprights
E_f	-	Young's modulus of flange
E_u	-	Young's modulus of upright
E_w	-	Young's modulus of web
E_{w_x}, E_{w_y}	-	Post-buckled Young's modulus of web
G_w	-	Shear modulus of web
G_{pdt}	-	Shear modulus of web associated with pure diagonal tension
G_{IDT}	-	Shear modulus of web associated with incomplete diagonal tension
h	-	Depth of beam
h_c	-	Clear depth of web, either measured between the inside tips of the
	-	two flange cross-sections or between the rivet rows of the upper
	-	and lower flanges
h_e	-	Effective depth of beam measured between the centroids of the two
	-	flanges
k	-	Diagonal tension factor
l	-	Total length of cantilever beam
L_x	-	Length of web along the x-axis
L_y	-	Length of web along the y-axis
N	-	Internal loads of structure
P	-	Applied load
S_b	-	Applied shear load at the onset of buckling
t	-	Thickness of web
t_u	-	Thickness of upright
u_1, u_2, u_3	-	Displacement in global x-, y- and z-directions

University of Pretoria etd – Viljoen, A (2004)

α	-	Diagonal tension angle
α_{prin}	-	Angle of major principal stress
ϵ_x, ϵ_y	-	Web normal strain
ϵ_{x_c}, ϵ_{y_c}	-	Web compressive buckling strain
$\epsilon_{x_{DT}}$, $\epsilon_{y_{DT}}$	-	Web diagonal tension strain
σ_1	-	Maximum principal stress
σ_2	-	Minimum principal stress
σ_{diag}	-	Web diagonal tension stress along the diagonal tension angle
$\sigma_{f_{cpl}}$	-	Crippling stress of flange
σ_{f_l}	-	Lower flange stress
σ_{f_u}	-	Upper flange stress
σ_{mises}	-	Von Mises stress
σ_{norm}	-	Web stress perpendicular to the diagonal tension stress
σ_u	-	Upright stress
$\overline{\sigma_u}$	-	Average upright stress
σ_{ut}	-	Ultimate tensile strength in tension
$\sigma_{u_{max}}$	-	Maximum stress in the upright
σ_x, σ_y	-	Web normal stress
σ_{x_c}, σ_{y_c}	-	Web compressive buckling stress
$\sigma_{x_{cr}}$, $\sigma_{y_{cr}}$	-	Web modified critical buckling normal stress
$\sigma_{x_{cr_0}}$, $\sigma_{y_{cr_0}}$	-	Web critical buckling normal stress based on geometry only
$\sigma_{x_{DT}}$, $\sigma_{y_{DT}}$	-	Web diagonal tension stress
σ_{yt}	-	Yield strength in tension
τ_{cr}	-	Web critical buckling shear stress
τ_{DT}	-	Diagonal tension component of shear stress carried by web
τ_s	-	Pure shear component of shear stress carried by web
τ_{xy}	-	Web shear stress
$\tau_{xy_{cr}}$	-	Web modified critical buckling shear stress
$\tau_{xy_{cr_0}}$	-	Web critical buckling shear stress based on geometry only
γ_{xy}	-	Web shear strain
γ_{xy_c}	-	Web shear strain component due to compressive buckling
$\gamma_{xy_{DT}}$	-	Post-buckled shear strain component of web
μ	-	Poisson's ratio

Subscripts 'x' and 'y' indicate the global axes.

University of Pretoria etd – Viljoen, A (2004)

Chapter 1

Introduction

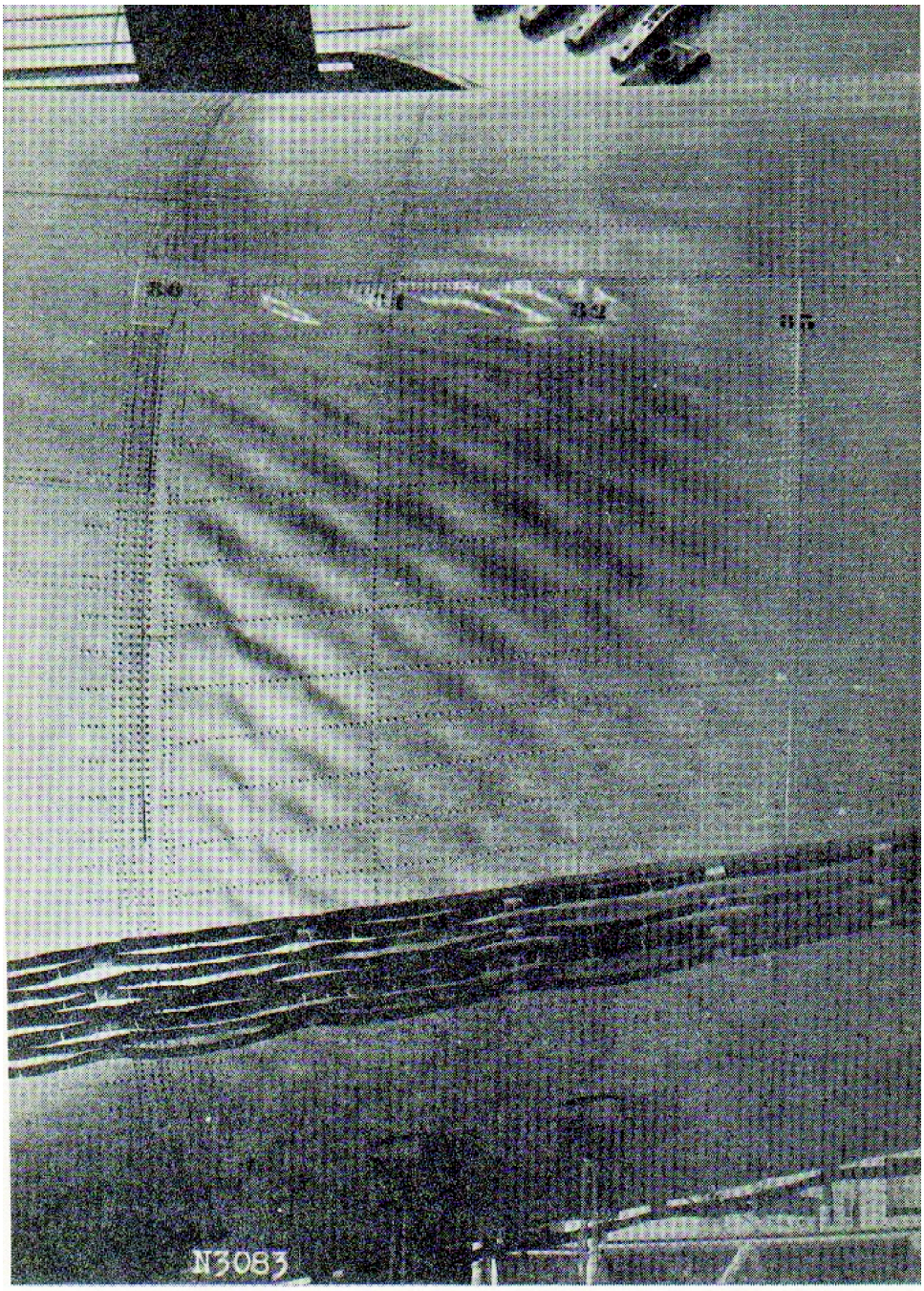
1.1 Overview

In many aircraft structures, thin sheet structural components are designed to buckle under shear load. The buckling can change the internal loads and stresses in neighboring panels and the surrounding structure significantly. Initially, post-buckling effects were taken into account using the theory of pure diagonal tension (PDT) proposed by Wagner [1, 2] in 1929, but this proved to be very conservative in practice. Wagner's approach was gradually modified to eventually cover the general approach of incomplete diagonal tension (IDT). The theory of incomplete diagonal tension was developed by the National Advisory Committee for Aeronautics (NACA) in the 1950's after conducting an extensive testing program to generate empirical relations [3, 4]. This approach, also known as the NACA method, has become the accepted design approach used by most aircraft manufacturers, even though the theory is still considered conservative. One of the factors neglected in the NACA approach is the interaction of stresses in each web element on the element allowables, namely the combination of compression and shear buckling, diagonal tension and post-buckled skin softening in shear.

As an alternative to IDT, non-linear finite element codes can be used to assess buckling, even though "design-by-rule" failure criteria are more difficult to assess (Mello *et al.* [5]). In addition, non-linear finite element analyses are computationally very expensive in initial design iterations.

In this thesis, the iterative algorithm developed by Grisham [6], is implemented to assess and evaluate the onset and magnitude of buckling in flat shear webs. Since few verifications of the Grisham algorithm have previously been presented, the method is extensively evaluated in this study. The results obtained using Grisham's algorithm are also compared with a non-linear finite element analysis.

University of Pretoria etd – Viljoen, A (2004)



Flambage local du revêtement d'un fuselage arrière
dans la partie non pressurisée, sous l'effet de l'effort tranchant
et de la flexion de fuselage.

Figure 1.2: Diagonal tension in airframe (Reproduced from [7])

University of Pretoria etd – Viljoen, A (2004)

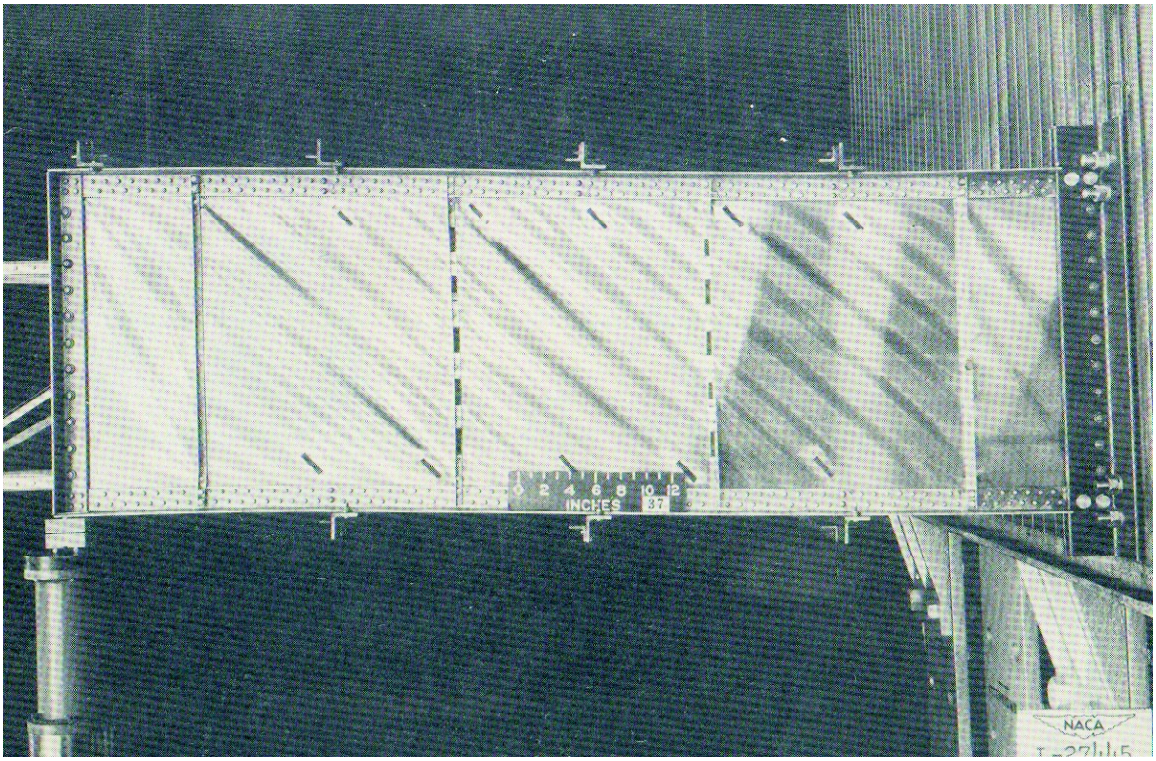


Figure 1.1: Diagonal tension beam (Reproduced from [3])

1.2 Diagonal tension

When a beam consisting of a thin shear web with transverse stiffeners is loaded until the web buckles, it forms diagonal folds at approximately 45 degrees. These diagonal folds carry load in tension. Compression stresses act perpendicular to theses folds while the stiffeners act as compression columns. The beam can carry loads orders of magnitude higher than the initial buckling load. Figure 1.1 shows such a thin web beam under a high test load that is close to failure. The many parallel folds in the web are clearly visible.

If the load is increased further, indefinitely without rupturing the sheet, the compressive stresses become smaller and smaller while the tensile stresses increase more and more, approaching a limiting condition known as pure diagonal tension (PDT). The theory of pure diagonal tension (PDT) was developed by Wagner [1, 2]. Pure diagonal tension can only be approached when the web is very thin, which is impractical. Most webs operate in a state of stress that is intermediate between pure diagonal tension and the state that exists before the web buckles. To cope with this approach, the engineering theory of incomplete diagonal tension (IDT) was developed by NACA [3, 4].

Another example of diagonal tension, this time in a fuselage, is shown in Figure 1.2. Although numerous aircraft skin panels may be in a buckled state during flight, the intensity is very low and is hardly visible to the human eye most of the time.

The incomplete diagonal tension approach divides the nominal web shear into two parts; a

University of Pretoria etd – Viljoen, A (2004)

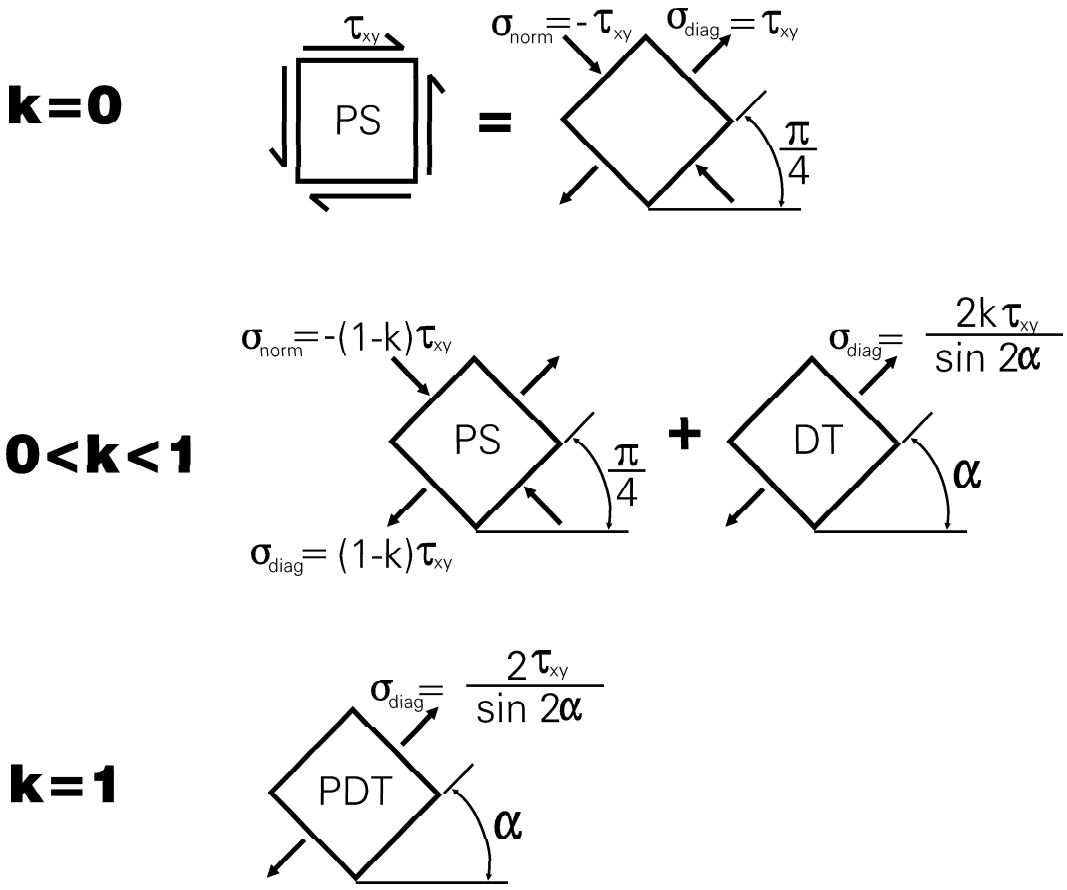


Figure 1.3: Stress systems in web

pure shear component and a diagonal tension component, respectively defined as follows:

$\tau_s = (1-k)\tau_{xy}$ and $\tau_{DT} = k\tau_{xy}$

where k is the diagonal tension factor. Figure 1.3 shows the stress state in a web for the limiting cases when $k = 0$ and $k = 1$ and also for the intermediate case.

1.3 The Grisham algorithm

In 1978 Grisham proposed an iterative algorithm that incorporates web compression-compression and shear buckling (including diagonal tension) in the internal loads solution of a linear finite element analysis [6]. Buckling is incorporated through the use of pre-strain loading of the web finite elements rather than through modifying the stiffness (thickness or elastic moduli) of the webs.

University of Pretoria etd – Viljoen, A (2004)

1.3.1 Outline of the Grisham algorithm

Without presenting the detailed mathematics involved[1] here, the steps in Grisham's iterative algorithm are as follows:

1. Calculate the internal loads N_i of the structure under consideration, using a *linear finite element analysis*. NOTE: the stiffness matrix of the linear finite element analysis is calculated once only and retained for following iterations; small displacement theory applies in this case.
2. Evaluate the onset of buckling using the internal loads obtained in Step 1 above, based on analytical plate buckling criteria and an interaction equation.
3. If buckling does not occur, STOP. Else, calculate the post-buckling relaxation of plates under compressive stress (σ_{x_c} and σ_{y_c}), the post-buckled shear strain $\gamma_{xy_{DT}}$ and the associated diagonal tension ($\sigma_{x_{DT}}$ and $\sigma_{y_{DT}}$).
4. Calculate the post-buckling strains ($\triangle\epsilon_x$ and $\triangle\epsilon_y$) to relieve the compression and shear stress that exceed the plate capability.
5. The post-buckling strains calculated above now become the pre-strains for the next iteration.
6. If the ratio of the final compression stress (σ_{x_c} and σ_{y_c}) in the buckled plate to the modified critical buckling stress ($\sigma_{x_{cr}}$ and $\sigma_{y_{cr}}$) approaches unity, and the diagonal tension stress ($\sigma_{x_{DT}}$ and $\sigma_{y_{DT}}$) converges, STOP. Else, go to Step 1.

1.3.2 Features of the Grisham algorithm

1. Convergence is usually achieved rapidly. Typically, five iterations are required to obtain convergence within 2 % variation between successive values of $\sigma_{x_{DT}}$ and $\sigma_{y_{DT}}$
2. Provision is made for compressive buckling in both the length and width directions of the web, as well as shear buckling. The latter causes the development of diagonal tension, accompanied by associated loading of the surrounding structure and "softening" of the buckled plate in shear.
3. The interaction of compression-compression and shear buckling is accounted for.
4. The stiffness matrix of the finite element model is not altered in any way to include for post-buckling effects.
5. Since the self-equilibrated pre-strains do not modify the stiffness matrix, precisely equilibrated and compatible solutions are obtained.
6. The pre-strains calculated give a direct indication of the degree of "softening" of the structure caused by buckling.

[1] Detailed mathematics is presented in Appendix A.

University of Pretoria etd – Viljoen, A (2004)

7. Multiple load cases (each with a different buckling pattern) may be processed simultaneously in a single finite element analysis.

8. Curved geometries may also be analyzed; structural symmetry can be exploited through the application of symmetric and anti-symmetric pre-strains if interaction equations are available.

9. For curved shells, the membrane loads due to pressurization are incorporated, and the stabilization effects due to hoop/longitudinal loads, are included through an interaction equation.

10. Isotropic, isotropic-stiffened plates, and plates in pure shear may be evaluated using the algorithm.

University of Pretoria etd – Viljoen, A (2004)

Chapter 2

Verification of the Grisham Algorithm

2.1 Overview

No example problems or specific results were published by Grisham [6] to validate the algorithm he presented. In order to verify the results of the current implementation, an example is taken from *Analysis and Design of Flight Vehicle Structures* [8] and the results are compared to the Grisham algorithm prediction.

2.2 Example problem

This example is a shear beam with 6 panels, schematically depicted in Figure 2.1, having the following dimensions:

1. $l = 1524$ mm (60 in)
2. $h = 762$ mm (30 in)
3. $h_e = 748$ mm (29.45 in)
4. $h_c = 725.4$ mm (28.56 in)
5. $d = 254$ mm (10 in)
6. $t = 0.635$ mm (0.025 in)
7. $A_{fl} = 243.9$ mm^2 (0.378 in^2)
8. $A_{fu} = 435.5$ mm^2 (0.675 in^2)
9. $A_u = 151.2$ mm^2 (0.253 in^2)

The web material is 2024-T3 aluminum sheet with the following material properties:

1. $\sigma_{yt} = 289$ MPa (42 ksi)

University of Pretoria etd – Viljoen, A (2004)

2. $\sigma_{ut} = 441$ MPa (64 ksi)
3. $E_w = 72.4$ GPa (10 500 ksi)

The two flanges are fabricated from 7075-T6 aluminium alloy extrusions, with the following material properties:

1. $\sigma_{yt} = 483$ MPa (70 ksi)
2. $\sigma_{ut} = 538$ MPa (78 ksi)
3. $E_f = 71.0$ GPa (10 300 ksi)

The uprights are fabricated from 2014-T6 aluminium alloy extrusions, with the following material properties:

1. $\sigma_{yt} = 366$ MPa (53 ksi)
2. $\sigma_{ut} = 414$ MPa (60 ksi)
3. $E_u = 72.4$ GPa (10 500 ksi)

Both flanges have a T-shaped cross-section. The uprights are angles (25.4 mm $\times$ 25.4 mm $\times$ 3.175 mm or 1 in $\times$ 1 in $\times$ $\frac{1}{8}$ in) and are made from 2014-T6 alloy. The shear load applied to the beam is $P = 60\ 048$ N (13 500 lb).

In this example two different methods are followed to solve the problem, namely the modified Wagner equations based on [1, 2] and the NACA approach [3, 4]. The results of both these approaches are compared to the current implementation of the Grisham algorithm. The various assumptions and limitations of the two methods are discussed below.

The modified Wagner approach incorporates:

1. The shear strength of the beam flanges.
2. The shear carried by the web before the onset of buckling.

The remainder of the shear in the beam after subtracting the components listed in (1) and (2) above is considered to be carried by the web in a buckled state in the form of a diagonal tension field.

The NACA approach is to be used subject to the following restrictions:

1. The uprights on the web stiffeners should not be too thin; the ratio of the stiffener thickness to web thickness should be greater than 0.6; i.e. $\frac{t_u}{t} > 0.6$.
2. The upright or web stiffener spacing shall not be outside the range $0.2 < \frac{d}{h} < 1.0$.

The NACA tests excluded very thin or very thick webs; non-conservative predictions may occur if $\frac{h}{t} > 1500$ or if $\frac{h}{t} < 200$.

Since:

University of Pretoria etd – Viljoen, A (2004)

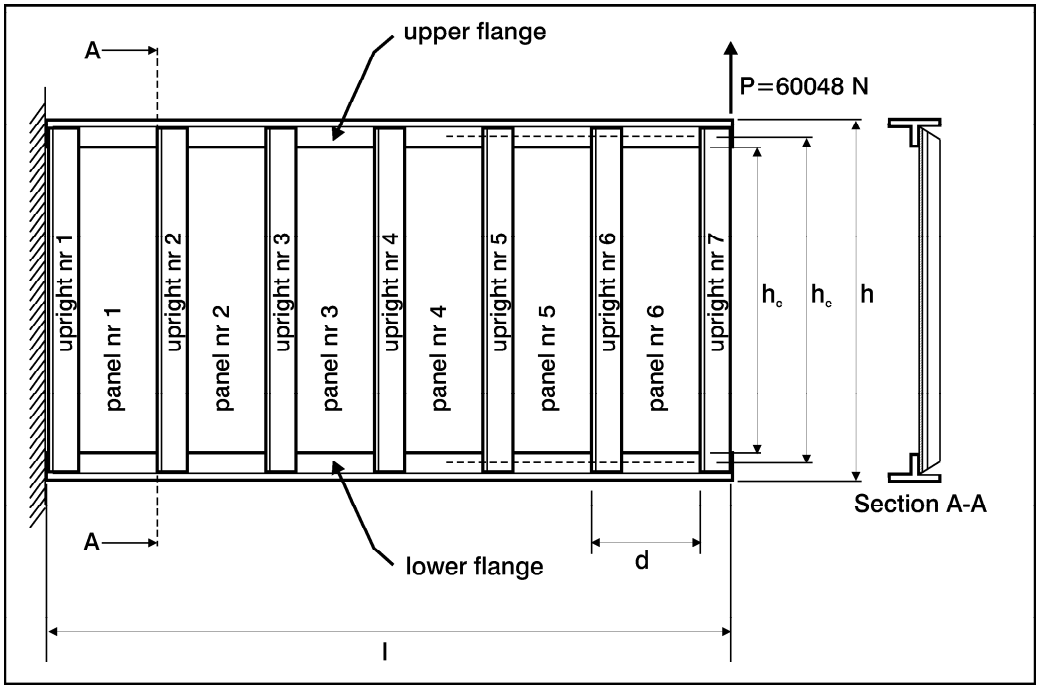


Figure 2.1: Layout of the cantilever beam used in the example

1. $\frac{t_u}{t} = 5.0$
2. $\frac{d}{h} = 0.33$
3. $\frac{h}{t} = 1142$

the use of the NACA equations is acceptable for the example under consideration.

(It should also be noted that, strictly speaking, the empirical relations developed by the NACA only apply to structures fabricated from 7076-T6 and 2024-T3 alloy since these were the only materials used in the NACA's extensive testing program.)

2.2.1 The finite element model

For the linear finite element analysis used in the Grisham algorithm, the flanges are modelled using second order beam elements, having a T-shaped cross-section. The uprights are also modelled with second order beam elements, having L-shaped cross-sections and their eccentricity is accounted for. The thin web is modelled with second order, isoparametric, thin shell elements having eight nodes per element. Small displacement theory is used. The shear load is applied vertically and distributed along all the nodes at the end of the beam. The ABAQUS® commercial finite element package is used for the linear finite element analysis. (It is also used for all subsequent finite element analyses in this thesis.) Plasticity effects are not taken into account and buckling of the uprights is also not considered. All four edges of the web are assumed to be simply supported when calculating the critical shear and

University of Pretoria etd – Viljoen, A (2004)

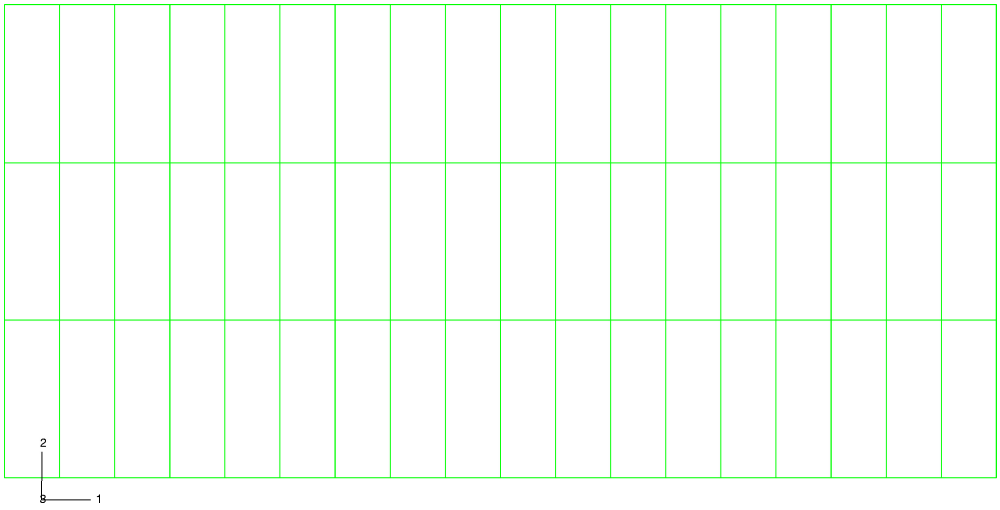


Figure 2.2: Finite element mesh for the verification example

compression buckling coefficients, as is done in the example in Bruhn [8]. Figure 2.2 shows the finite element mesh of the beam used to evaluate the Grisham algorithm. A coarse 3 × 3 mesh is used for the web of each panel.

2.2.2 Web results

In Table 2.1, the results for the panels obtained using the iterative procedure of Grisham are compared with results from the Wagner and NACA approaches. The four variables compared in the table are:

1. k - the diagonal tension factor. It indicates the degree to which the diagonal tension is developed and is defined as $\tau_{DT} = k\tau_{xy}$. When $k = 0$, no diagonal tension is present. When $k = 1$, the web is in pure diagonal tension.

2. α - the diagonal tension angle. In pure diagonal tension (PDT) this is the angle of the major principal stress relative to the horizontal for this example. In incomplete diagonal tension (IDT), it is the angle that the major principal stress would have if the sheet were not carrying part of the load in pure shear.

3. τ_{xy} - the shear stress in the web.

4. τ_{cr} - the critical web shear buckling stress. This is the stress at the onset of shear buckling.

University of Pretoria etd – Viljoen, A (2004)

	Grisham algorithm				Modified	
	Panel 2	Panel 3	Panel 4	Panel 5	Wagner	NACA [3, 4]
k	0.649	0.673	0.673	0.635	—	0.690
α [degrees]	42.1	42.0	42.0	42.2	43.0	41.3
τ_{xy} [MPa]	126.9	127.3	127.2	127.3	130.4	135.1
τ_{cr} [MPa]	2.624	2.624	2.624	2.624	2.358	2.551

Table 2.1: Comparison between Grisham algorithm, modified Wagner equations, and NACA approach

The results of panels 1 and 6 are ignored since they include edge effects that are automatically included via the finite element analysis and which are not accounted for in the NACA or Wagner approaches. Only five iterations are required for the Grisham results; this means that the system equations are solved only five times, while the stiffness matrix remains constant throughout (since the analyses are linear, simple superposition principles can be applied).

The results for the four panels obtained with the Grisham algorithm are very similar and compare closely to those of the other two methods. The value of the diagonal tension factor k from the Grisham algorithm is slightly lower than that of the NACA method. The diagonal tension factor k indicates the percentage of the total load that is carried in diagonal tension by the web. The value of $k = 0.69$ using the NACA methodology indicates that 69 % of the total load carried by the web is carried in diagonal tension and the rest in normal shear. The Grisham algorithm takes compression-compression buckling in both directions into account through an interaction equation which has an effect on the magnitude of the diagonal tension. In the NACA method this is not taken into account, resulting in conservative predictions. *When neglecting this effect of compression-compression buckling in Grisham's method, the NACA result of* $k = 0.69$ *is obtained exactly.*

The critical shear stress τ_{cr} (shear stress at the onset of buckling) for the Grisham algorithm and Wagner approach are based on analytical relations that depend on geometry and material properties. The NACA critical shear stress τ_{cr} includes empirical data in its calculation, relating to the stiffness of the flanges and uprights. The large difference between the critical shear stress value τ_{cr} and the total web shear stress shows that the final stress carried by the web is almost 50× higher than the initial buckling shear stress. The low value of τ_{cr} is a result of the relatively thin plate ($t = 0.635$ mm).

Table 2.2 gives the diagonal tension stress values as well as the compression buckling stresses of the web for each panel. The Modified Wagner and NACA methods do not produce values that can be compared to these and are therefore not included in Table 2.2. The final compressive stress values are very low. This indicates that the compressive-compressive buckling effects are not significant in this example. Figures 2.3, 2.4 and 2.5 show the stress results in the web from the Grisham algorithm's finite element analysis. The shear stress plot compares well to the results in Table 2.1. Figures 2.6 and 2.7 show vector plots of the maximum and minimum principal stresses in the web. The magnitudes of the stress values are very high for the web. In the worked example the intricacies of the finite element

University of Pretoria etd – Viljoen, A (2004)

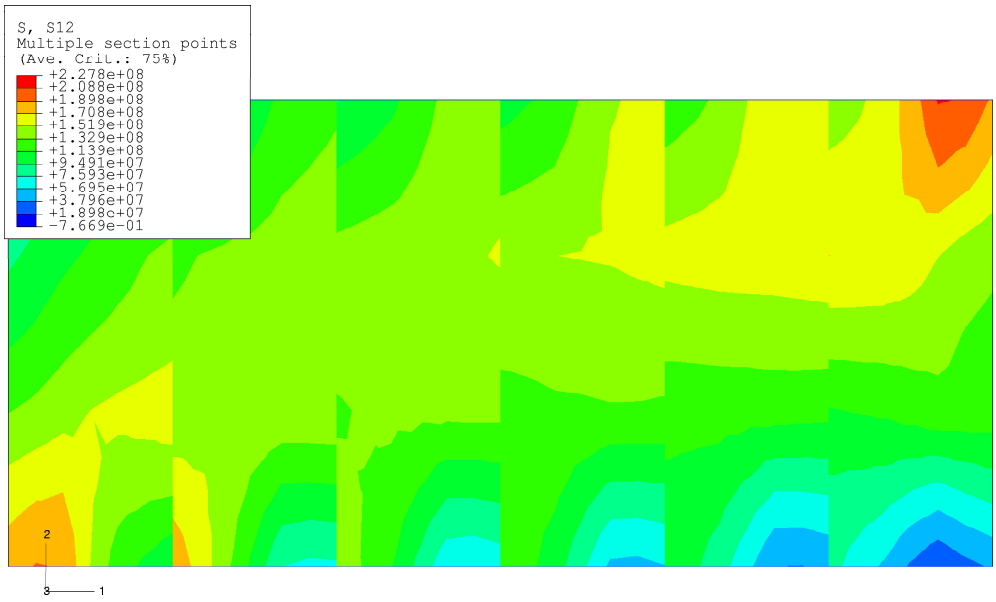


Figure 2.3: Unsmoothed shear stress distribution (τ_{xy}) in the web after the final iteration

method are not taken into account. Boundary conditions, having a localized effect are also not considered.

	Grisham algorithm			
	Panel 2	Panel 3	Panel 4	Panel 5
$\sigma_{x_{DT}}$ [MPa]	91.19	95.17	95.20	89.37
$\sigma_{y_{DT}}$ [MPa]	74.47	77.17	77.18	73.28
σ_{x_c} [MPa]	-0.473	-0.250	0.047	0.518
σ_{y_c} [MPa]	0.149	0.382	0.364	0.466

Table 2.2: Diagonal tension and compressive stress values calculated by means of the Grisham algorithm

2.2.3 Upright results

The results for the upright stresses are plotted in Figures 2.9 to 2.11. Values are plotted along the length of the upright at three different section points on the cross-section of the upright. Uprights number 1 and 7 are not considered because of boundary conditions. Figure 2.8 shows the position of these section points on the cross-section of the upright.

From the data in Figures 2.9 to 2.11 it can be seen that for all the uprights, the values at section points 5 and 9 are negative and therefore in compression while the values at

University of Pretoria etd – Viljoen, A (2004)

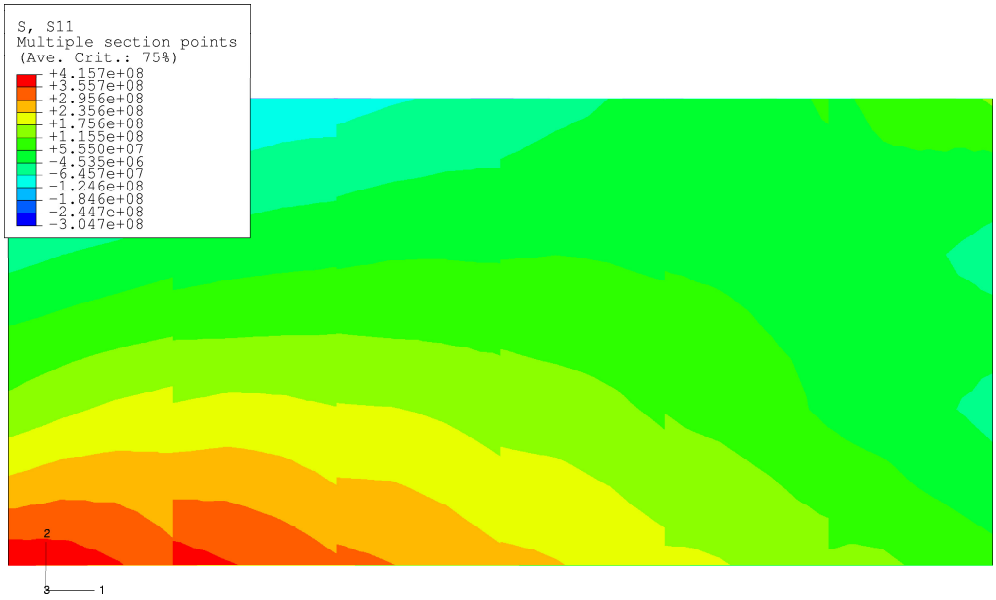


Figure 2.4: Unsmoothed normal stress (σ_x) in the web after the final iteration

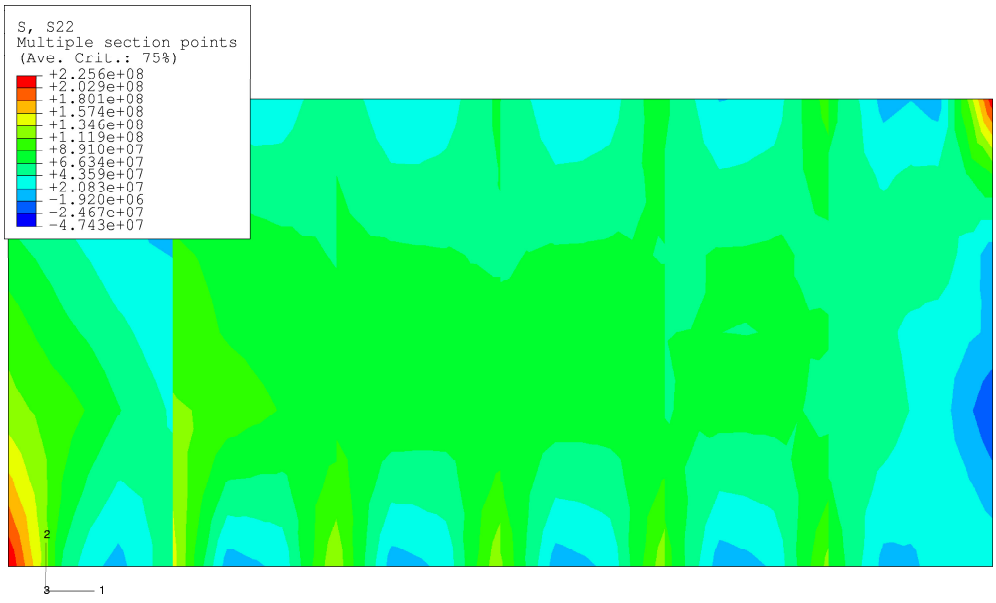


Figure 2.5: Unsmoothed normal stress (σ_y) in the web after the final iteration

University of Pretoria etd – Viljoen, A (2004)

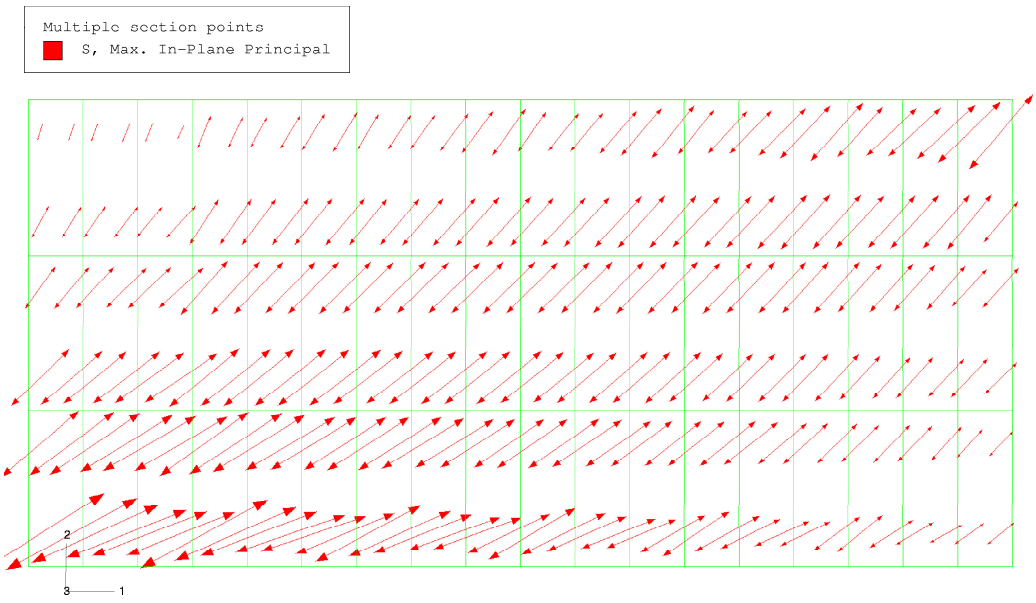


Figure 2.6: Maximum principal web stress vector plot after the final iteration

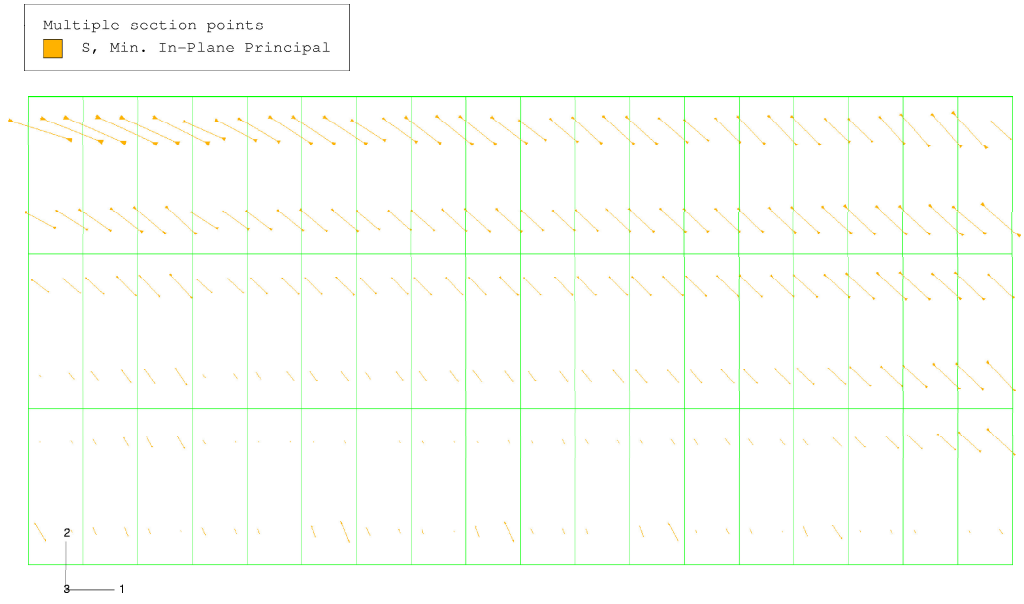


Figure 2.7: Minimum principal web stress vector plot after the final iteration

University of Pretoria etd – Viljoen, A (2004)

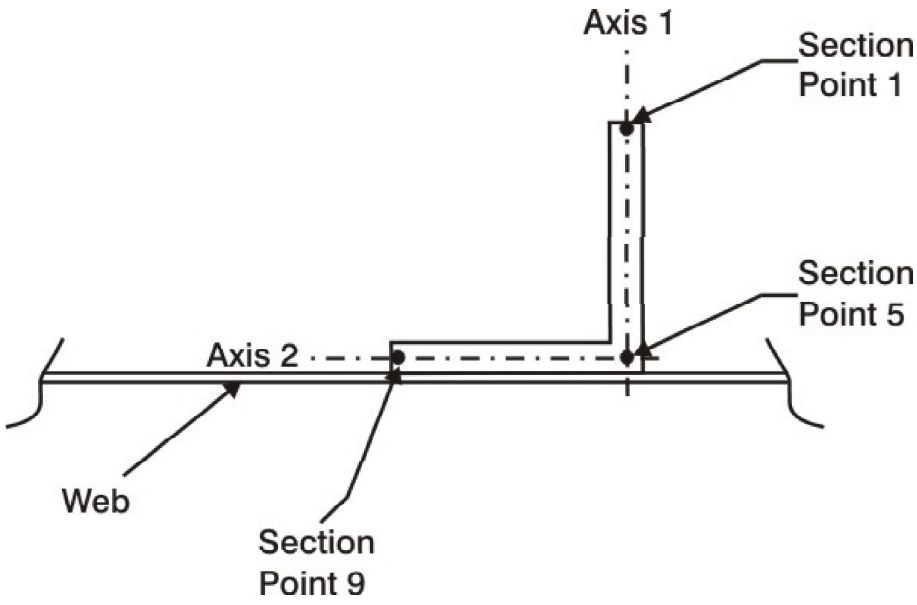


Figure 2.8: Cross-section of the upright showing the location of the section points

section point 1 are positive and therefore in tension. The upright behaves as a column loaded eccentrically along the median plane of the web by diagonal tension effects in the web. Although this tends to pull the two flanges together, creating a compressive load in the upright, it also causes bending due to the eccentricity and therefore the compressive stress tends to decrease in the upright when moving away from the web plane (towards section point 1). If the upright were symmetrical about the median plane of the web, the stresses would be compressive at all section points. The NACA equations predict that the maximum stress occurs in the upright at a point that coincides with the neutral axis of the beam. The upright stress then reduces in magnitude (parabolically) until the minimum upright stress is found at the two ends that are attached to the two flanges (known as the "gusset effect"). The neutral axis of the beam in this example is slightly above the centroid of the upright. At very high loading ratios ($\frac{\tau_{xy}}{\tau_{cr}} \gg 1$), such as in this example, the parabolic profile flattens out and all the stresses along the length of the upright are similar in magnitude.

Table 2.3 compares the results from the NACA method and that of the Wagner approach with the Grisham results. The average stress values are calculated along the length of the upright, using all the integration point values at section points 5 and 9. To compensate for eccentric loading the NACA method uses an effective area and only calculates average stresses at the median plane of the web. Section points 5 and 9 are closest to the median plane of the web. Wagner assumes that the uprights are like columns with elastic supports; that the web tension prevents the uprights from buckling out of the plane and that the total cross-section of the upright is in compression.

The average upright stress values of the Grisham algorithm are lower than that predicted by the NACA and Wagner approach. The stress magnitude in the upright is a direct indication of the magnitude of diagonal tension in the web. Since the diagonal tension factor for the Grisham algorithm is slightly lower than that predicted by the above mentioned methods,

University of Pretoria etd – Viljoen, A (2004)

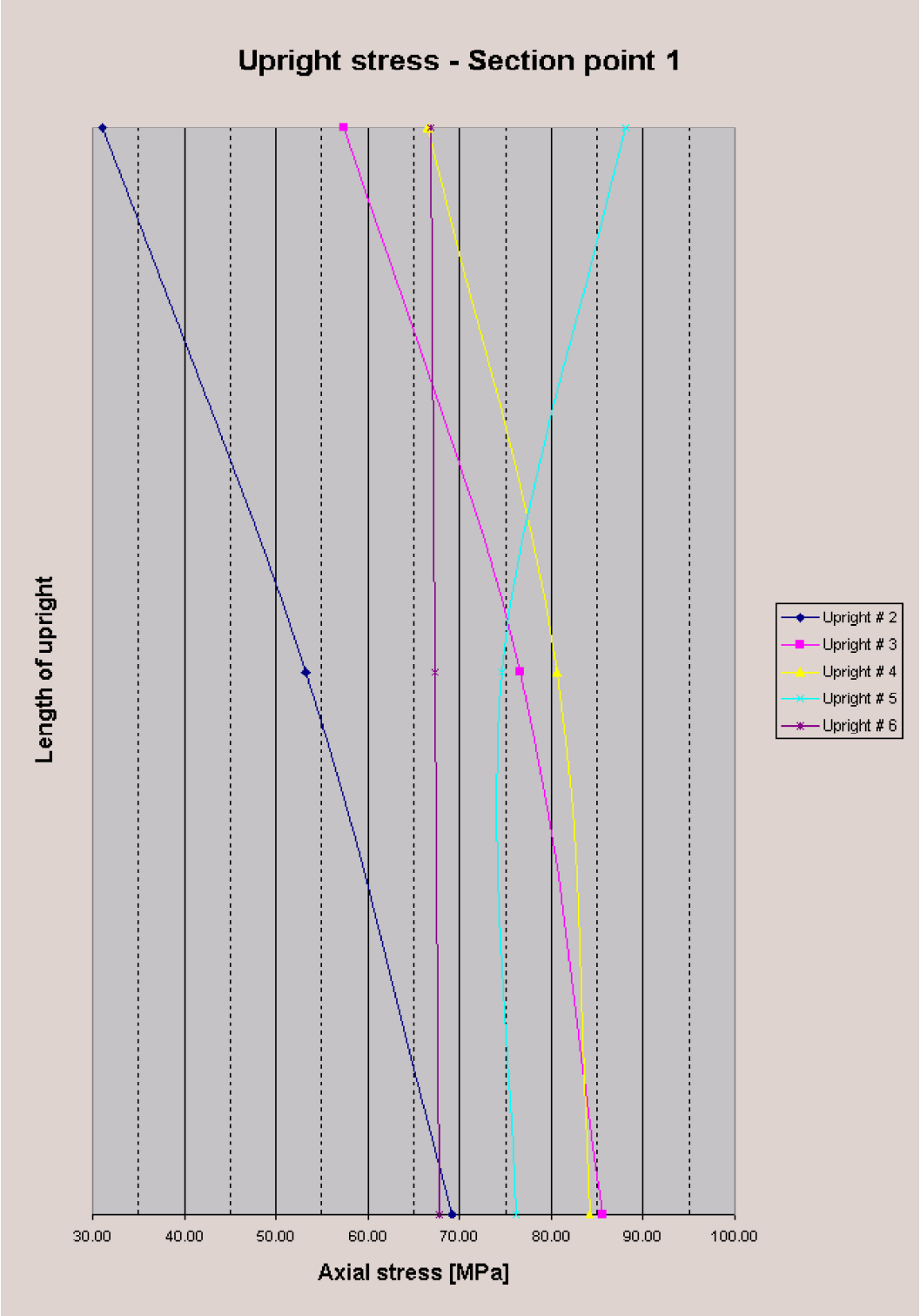


Figure 2.9: Axial stress along the length of the upright (σ_u) at section point 1

University of Pretoria etd – Viljoen, A (2004)

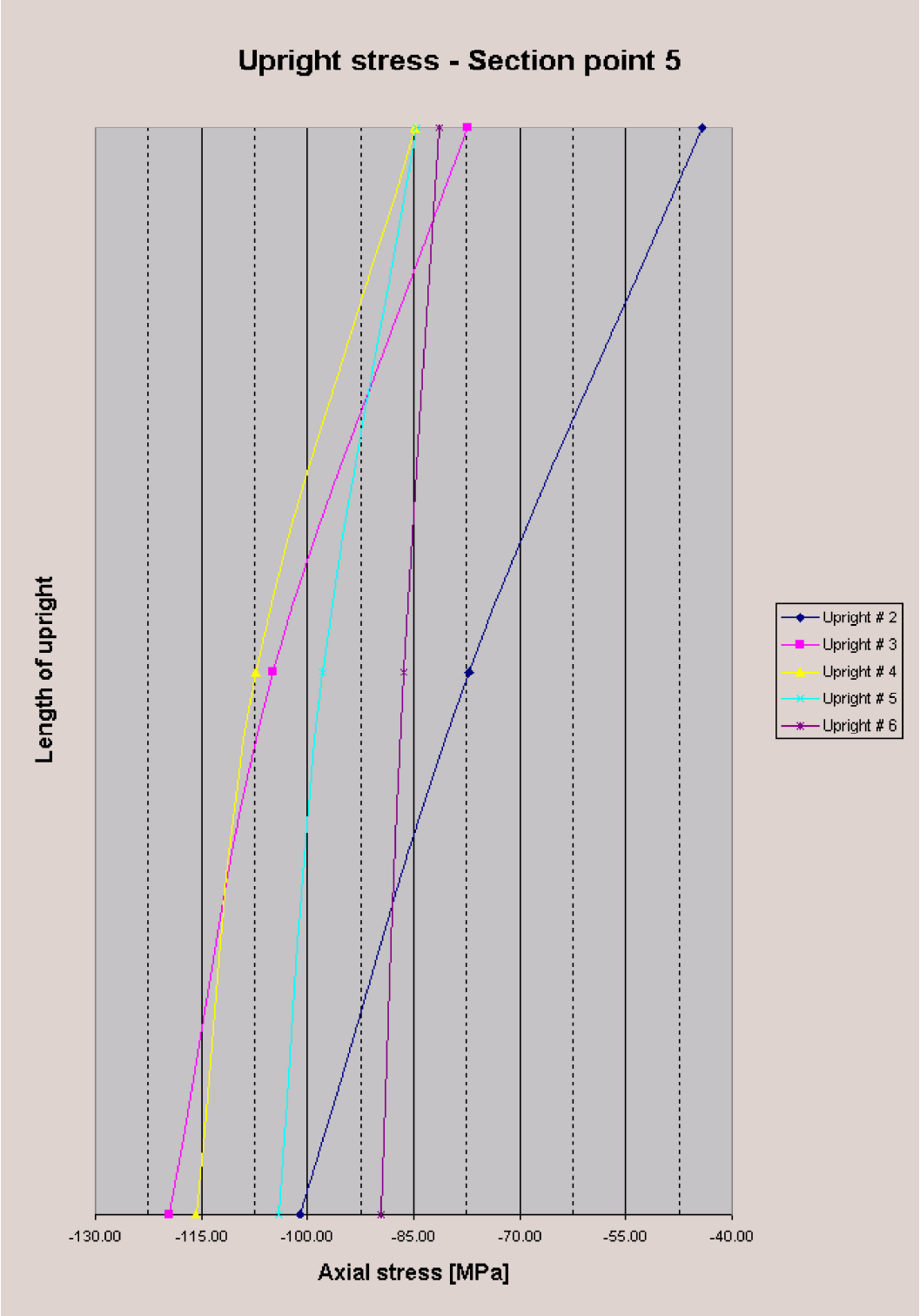


Figure 2.10: Axial stress along the length of the upright (σ_u) at section point 5

University of Pretoria etd – Viljoen, A (2004)

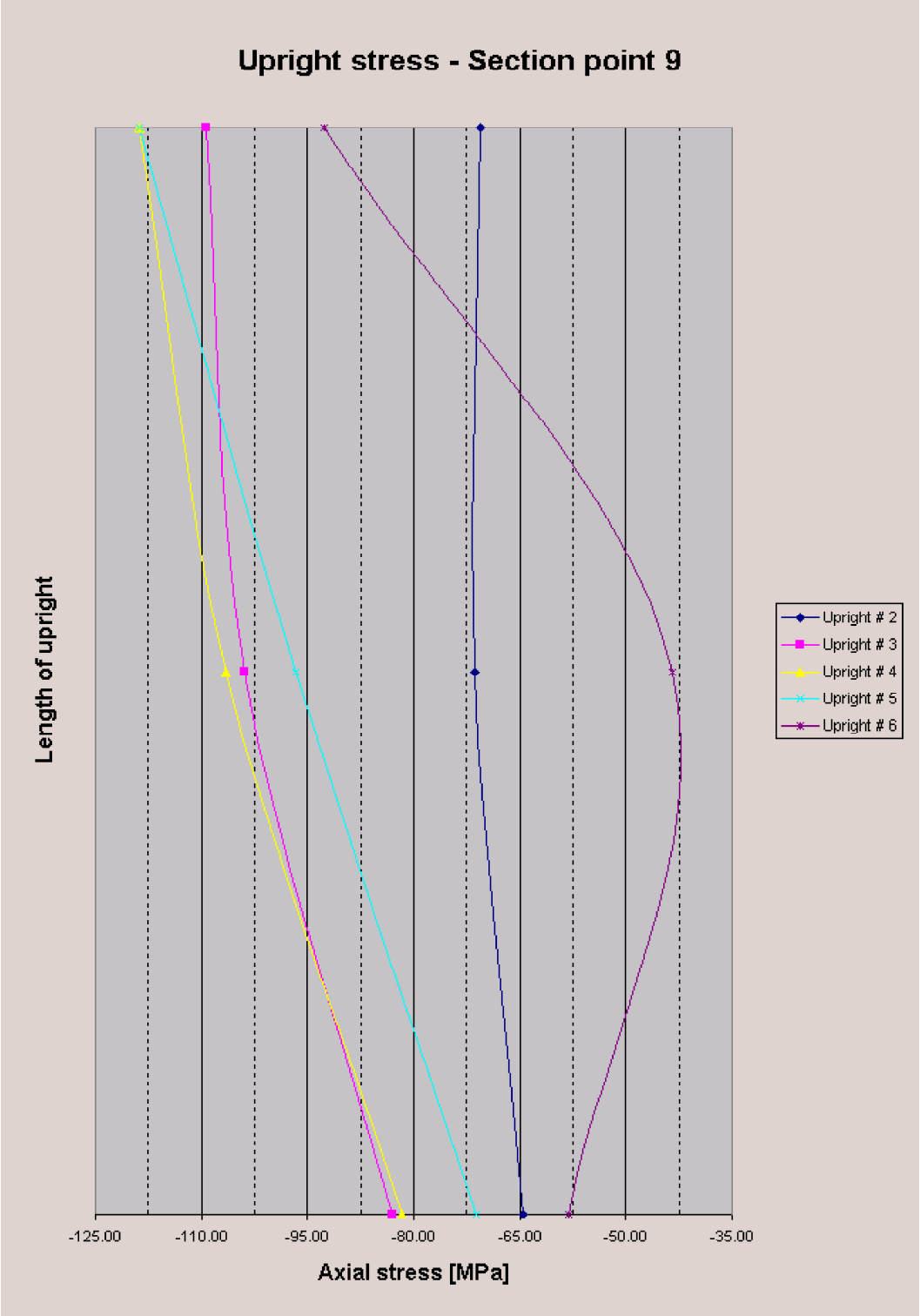


Figure 2.11: Axial stress along the length of the upright (σ_u) at section point 9

University of Pretoria etd – Viljoen, A (2004)

	Grisham algorithm					Modified	
	Upright 2	Upright 3	Upright 4	Upright 5	Upright 6	Wagner	NACA
$\overline{\sigma}_u$[MPa]	-71.4	-99.7	-102.4	-95.5	-75.3	-123.6	-116.5
$\sigma_{u_{max}}$[MPa]	-101.2	-119.7	-118.7	-118.7	-92.8	-123.6	-136.5

Table 2.3: Upright stress results

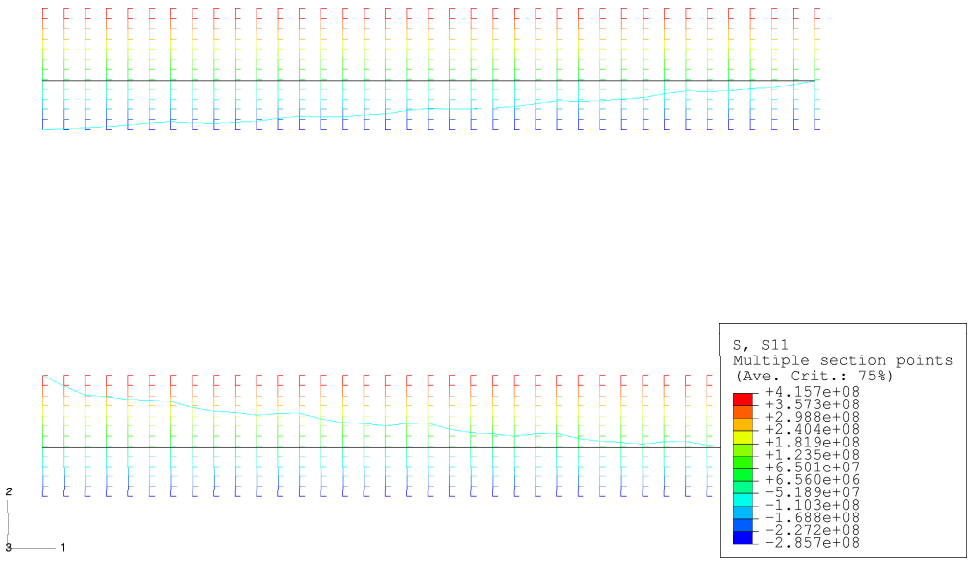


Figure 2.12: Axial stress distribution in the flanges after the final iteration of the Grisham algorithm

the upright stresses for the Grisham algorithm would also be lower.

2.2.4 Flange results

A contour plot of the axial stress distribution in the upper and lower flanges after the final iteration of the Grisham algorithm is shown in Figure 2.12. The flange stress values as calculated at section A-A in Figure 2.1 are given in Table 2.4. In the upper flange (compression flange), the high values of σ_{f_u} = -545.0 MPa and σ_{f_u} = -467.2 MPa are due to the secondary bending effects caused by diagonal tension in the web. Likewise in the lower flange (tension flange), the compressive values of σ_{f_l} = -427.0 MPa and σ_{f_l} = -288.4 MPa are also due to secondary bending effects caused by diagonal tension in the web. In a real life design situation both flanges would have to be redesigned because these values either exceed the yield stress of the material or are too close to the crippling stress [8] of $\sigma_{f_{cpl}}$ = -434.4 MPa.

University of Pretoria etd – Viljoen, A (2004)

The stress values for the Modified Wagner method as well as the NACA method can be split into three components: (i) the bending stress due to the applied shear load, (ii) the axial stress due to diagonal tension, and (iii) secondary bending stress due to diagonal tension. According to Kuhn [9] the calculation of the secondary bending stresses is conservative and can usually be neglected in practice. The flange stresses without the secondary bending stresses are given in Table 2.5. All lower flange stresses are now in tension and the values are also in much better agreement with those of the Grisham algorithm.

	Grisham Algorithm	Modified Wagner	NACA
σ_{f_u} [MPa] (upper fibre)	-216.9	-242.0	-237.2
σ_{f_u} [MPa] (lower fibre)	-255.0	-545.0	-467.2
σ_{f_l} [MPa] (upper fibre)	162.9	-427.0	-288.4
σ_{f_l} [MPa] (lower fibre)	275.2	460.0	408.4

Table 2.4: Flange stress results

	Grisham Algorithm	Modified Wagner	NACA
σ_{f_u} [MPa] (upper fibre)	-216.9	-312.3	-292.7
σ_{f_u} [MPa] (lower fibre)	-255.0	-304.5	-276.2
σ_{f_l} [MPa] (upper fibre)	162.9	292.4	281.0
σ_{f_l} [MPa] (lower fibre)	275.2	294.5	276.7

Table 2.5: Flange stress results excluding secondary bending effects

According to Peery [10] the stresses in the uprights and flanges may be five times less than that predicted by the theory of pure diagonal tension. The results predicted by Grisham are certainly less conservative, making it a more attractive method to use.

2.2.5 Deflection results

For the sake of completeness the displaced shape of the beam at the end of the iterative analysis is shown in Figure 2.13.

University of Pretoria etd – Viljoen, A (2004)

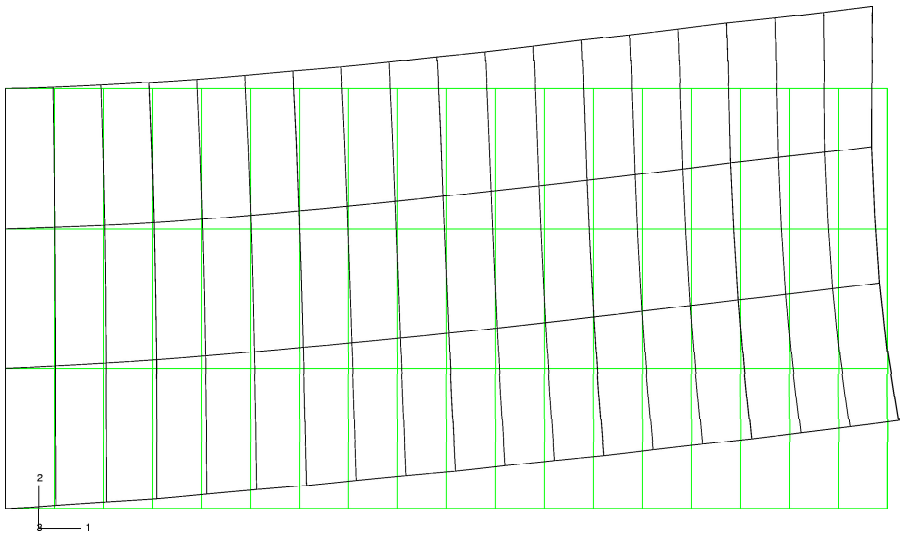


Figure 2.13: Displaced shape of the finite element model after the final iteration (undeformed shape in green)

University of Pretoria etd – Viljoen, A (2004)

Chapter 3

Non-linear finite element analysis

In this chapter, the Grisham algorithm is further verified. The Grisham algorithm results for the *example problem in the previous chapter* are now compared with the results of a complete non-linear finite element analysis.

3.1 Overview

Global load control methods (e.g. the arc-length control method proposed by Riks and Wempner [11, 12]) are suitable for global buckling and post-buckling analysis. However, they are not ideal when the buckling is localized (when there is a local transfer of strain energy from one part of the model to neighbouring parts). Alternatives are to analyze the problem dynamically, or to introduce an artificial damping factor. In the dynamic case, the strain energy released during local buckling is transformed into kinetic energy; in the damping case the strain energy is dissipated. To solve a quasi-static problem dynamically is expensive, however.

Hence, the modified Riks algorithm, combined with damping, is selected as an analysis option in the ABAQUS® environment. The modified Riks method is used where the load magnitudes are controlled by a single scalar parameter. The load is considered unknown and the algorithm solves for loads and displacements simultaneously. An arc length along the equilibrium path is used as an indication of the progress of the solution and can be used to specify the termination of the analysis. To account for instability in the non-linear static problem, volume-proportional damping is added to the model. This provides viscous forces that are large enough to prevent instantaneous collapse, but small enough not to affect the results while the problem is stable and is a computationally efficient approach for the analysis of localized buckling.

The damping factor is determined in such a way that the extrapolated dissipated energy is a small fraction of the extrapolated strain energy. The damping factor is dependent on the mesh size and material behavior.

University of Pretoria etd – Viljoen, A (2004)

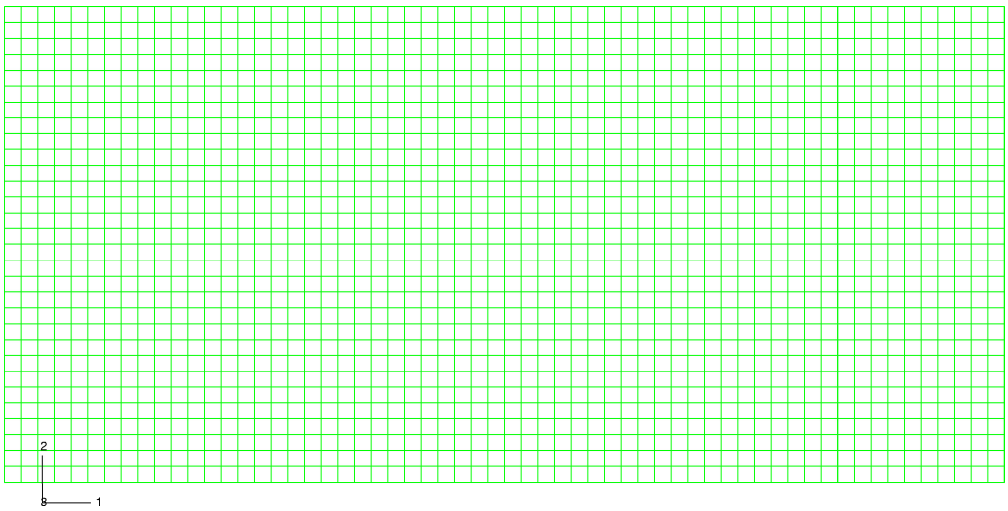


Figure 3.1: Non-linear finite element mesh

3.2 The finite element model

Again, the flanges and uprights are modelled using second order beam elements with the appropriate cross-section. The eccentricity of the single uprights (angles on one side of the web only) is taken into account. The web is modelled using isoparametric, second order, thin shell elements. Figure 3.1 shows the mesh used in the non-linear analysis, which is much more dense than the mesh used in the linear analysis for Grisham's algorithm. In the non-linear analysis 10×30 elements were used for the web of each panel as opposed to the 3×3 in the linear analysis. A distributed, vertical shear load is applied at the one end. The flanges as well as upright number 7 are constrained not to deform out of the plane.

3.3 Web results

The onset of shear buckling occurs after 13 load increments at an applied shear load of 1 552 N, resulting in a nominal web shear stress of $\tau_{cr} = 3.370$ MPa, which is slightly higher than the values predicted by the Grisham algorithm ($\tau_{cr} = 2.624$ MPa), the Wagner method ($\tau_{cr} = 2.358$ MPa) and the NACA approach ($\tau_{cr} = 2.551$ MPa). The three above methods, however, assume all four edges to be simply supported when calculating the buckling coefficients. This is conservative since the flanges and uprights are not completely flexible in torsion. The non-linear finite element method takes the torsional stiffness of the flanges and uprights into account and therefore gives a more accurate, higher critical stress value. A less conservative analytical estimate by Fehrenbach [7] gives a value of $\tau_{cr} = 3.195$ MPa for this problem. The

University of Pretoria etd – Viljoen, A (2004)

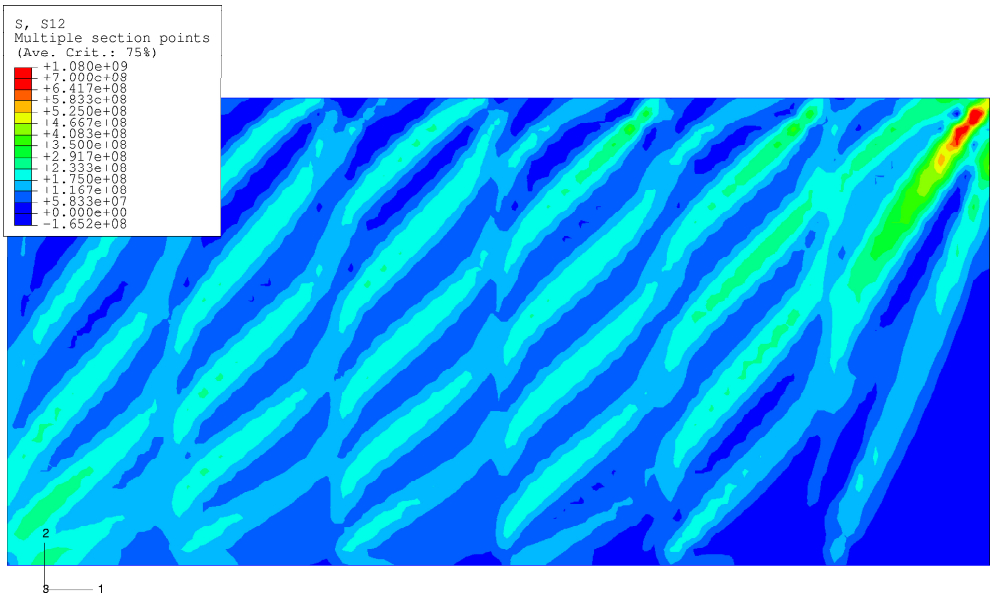


Figure 3.2: Shear stress (τ_{xy}) at the end of the analysis

analysis takes 102 load increments to reach the fully applied load of 60 048 N. The applied load of 60 048 N is almost 39 times more than the initial buckling load of 1 552 N, showing that the structure can hold a load orders of magnitude larger than the initial buckling load.

The stress results are shown in Figures 3.2 to 3.4. Figure 3.2 shows the shear stresses while Figures 3.3 and 3.4 shows the maximum and minimum principal stresses respectively. The effects of the folds in the web are clear as are the angles of the folds relative to the beam axis.

3.4 Upright results

Table 3.1 shows the non-linear finite element results in the uprights compared with the Grisham algorithm results (see Figure 2.8 for the location of the section points). The average stress values compare well for section points 5 and 9. At section point 1 the average stress results of the non-linear finite element analysis are slightly lower than the Grisham algorithm results, varying by up to 50 %. This shows that the out of plane bending effects due to the upright eccentricity is less evident in the non-linear finite element analysis. Except for upright number 2, the maximum stress values of the non-linear finite element analysis are higher than the Grisham algorithm results at all section points for all other uprights. The values vary from 0.8 % to 34 %.

The non-linear finite element upright stress result plots are shown in Figures 3.5 to 3.7. The upright stresses are negative at section points 5 and 9, indicating that the upright is in

University of Pretoria etd – Viljoen, A (2004)

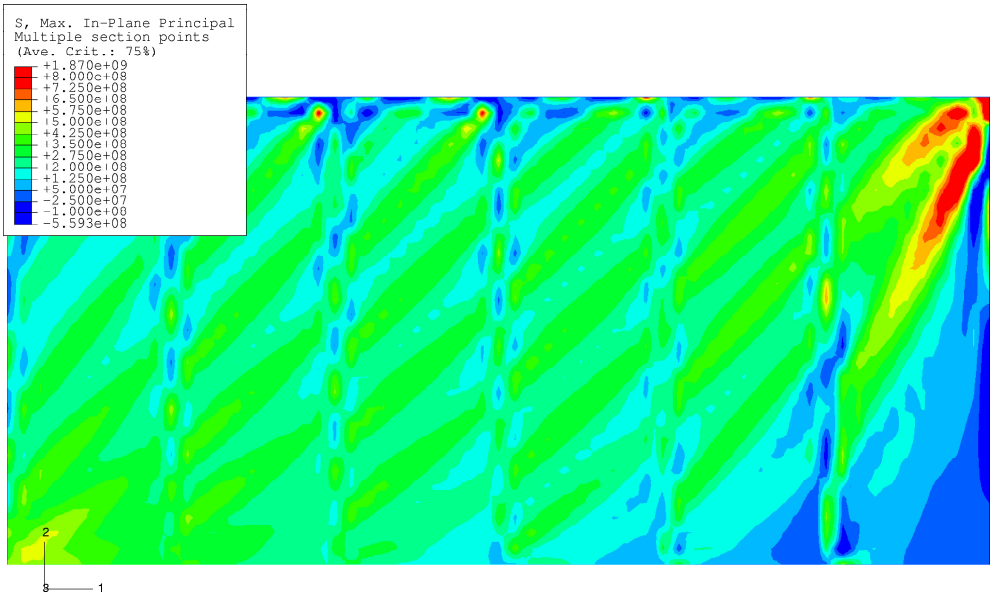


Figure 3.3: Maximum principal stress (σ_1) at the end of the analysis

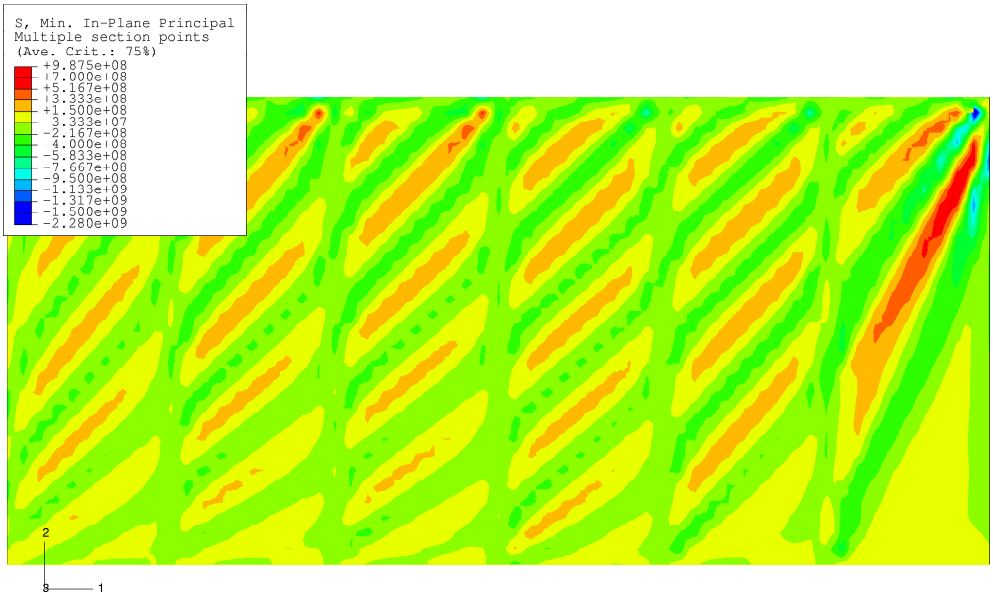


Figure 3.4: Minimum principal stress (σ_2) at the end of the analysis

University of Pretoria etd – Viljoen, A (2004)

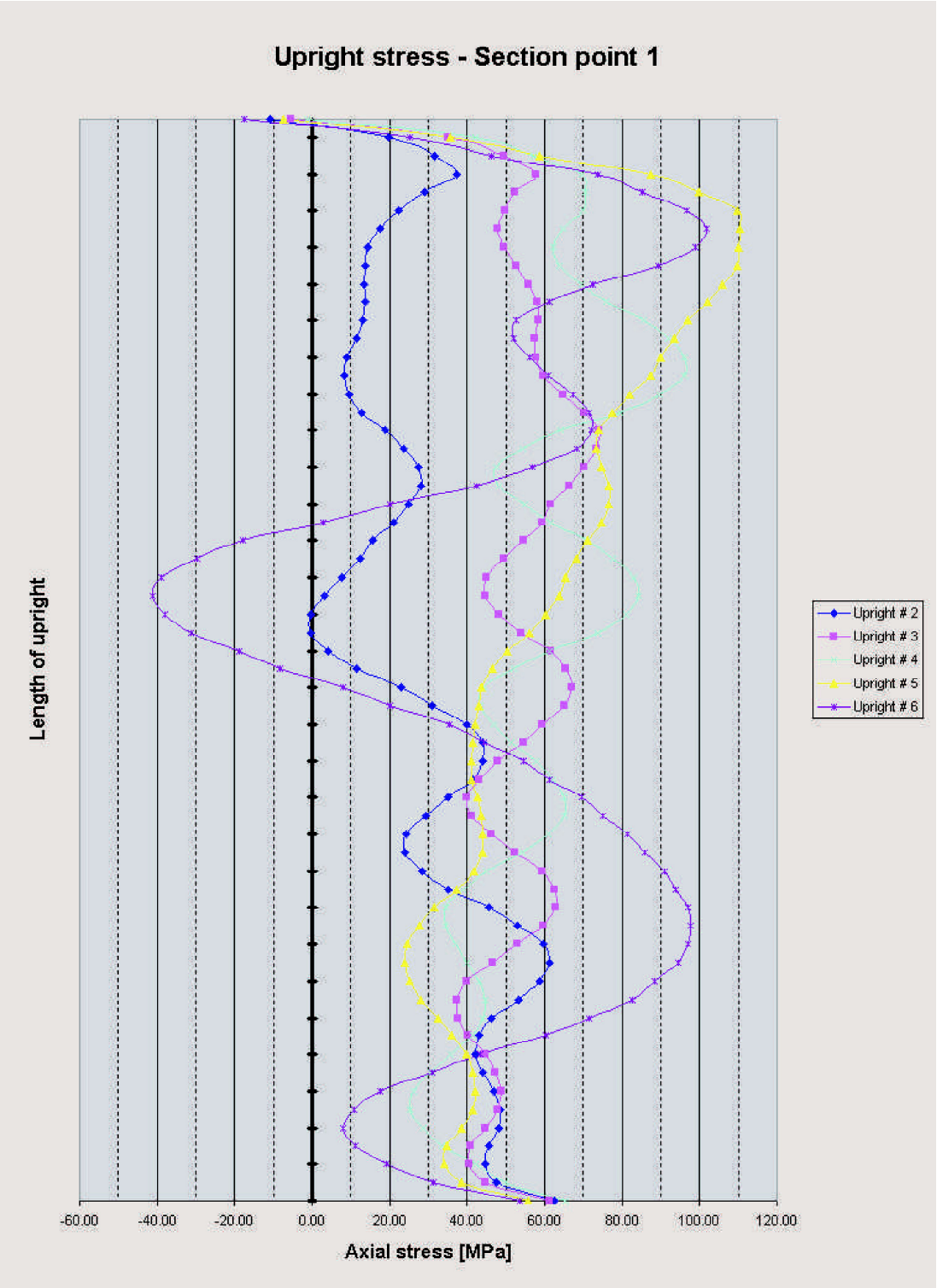


Figure 3.5: Stresses along the uprights at section point 1

University of Pretoria etd – Viljoen, A (2004)

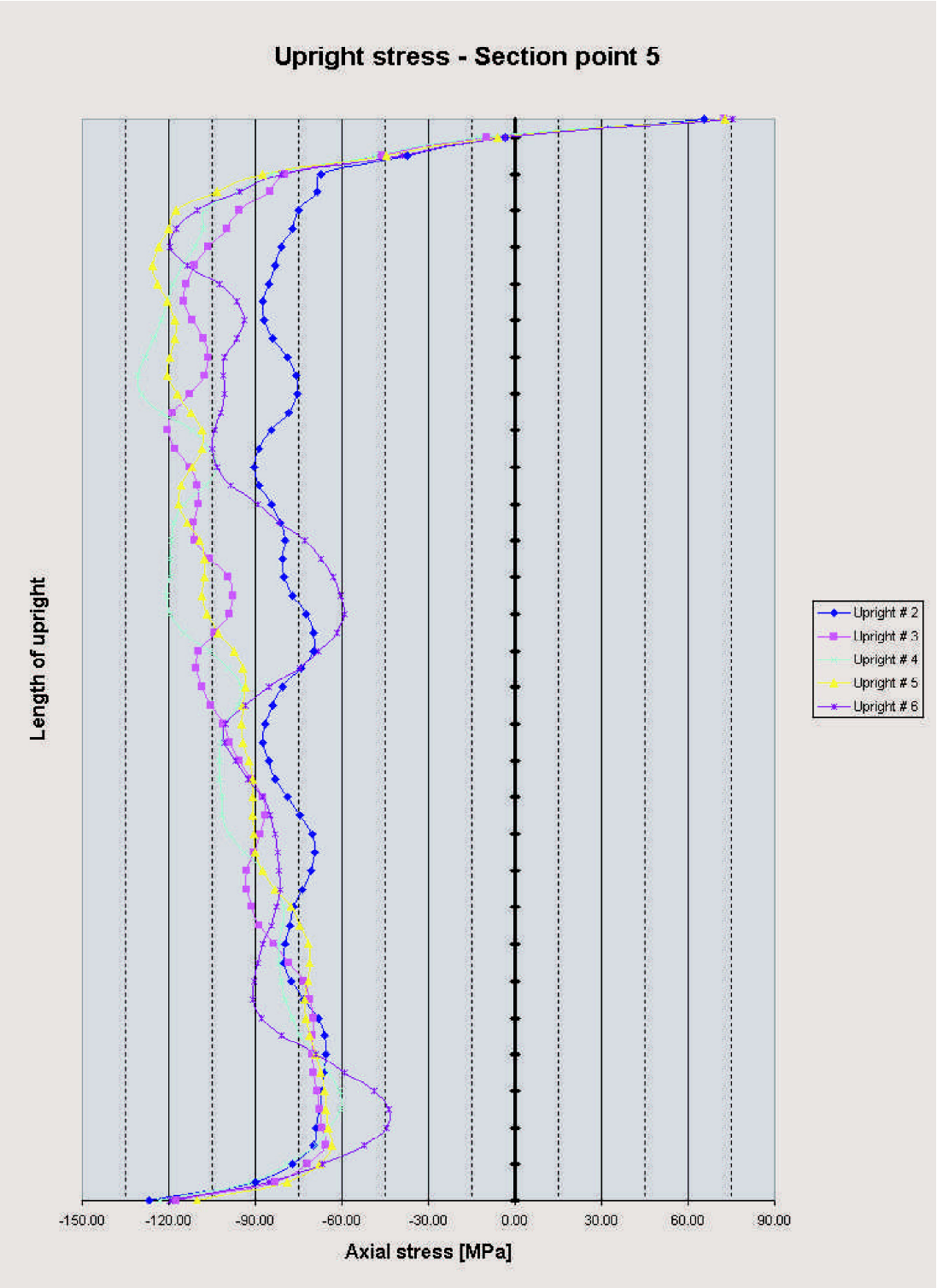


Figure 3.6: Stresses along the uprights at section point 5

University of Pretoria etd – Viljoen, A (2004)

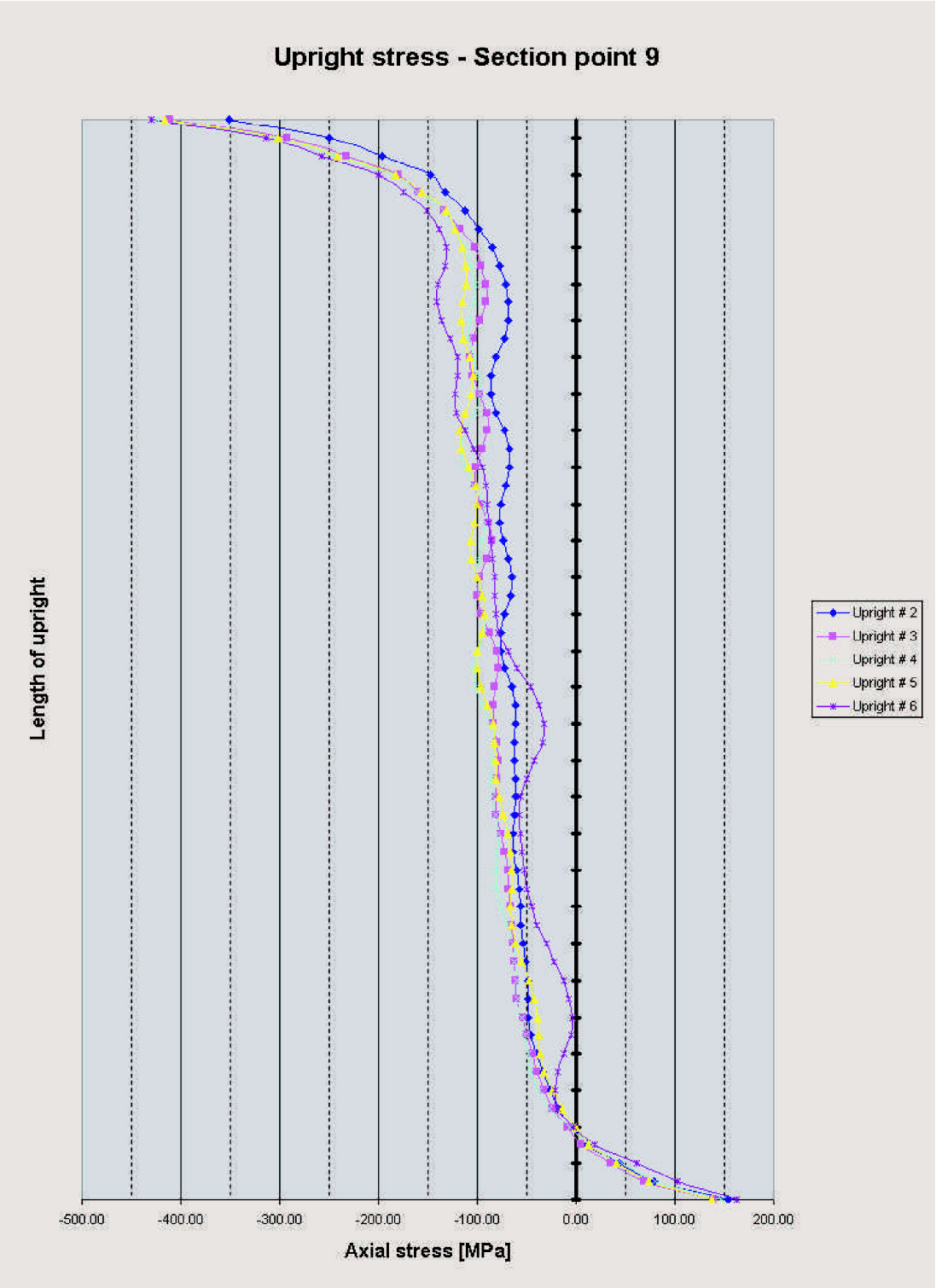


Figure 3.7: Stresses along the uprights at section point 9

University of Pretoria etd – Viljoen, A (2004)

	Six panel cantilever beam example - Bruhn [8]				
	Upright 2	Upright 3	Upright 4	Upright 5	Upright 6
$\overline{\sigma}_u$ (sp 1) - Grisham	51.2	73.2	77.1	72.9	67.3
$\overline{\sigma}_u$ (sp 1) - NL FEM	26.83	54.54	59.07	59.54	48.42
$\overline{\sigma}_u$ (sp 5) - Grisham	-74.2	-100.6	-102.6	-95.5	-85.8
$\overline{\sigma}_u$ (sp 5) - NL FEM	-78.29	-98.01	-102.25	-98.01	-87.39
$\overline{\sigma}_u$ (sp 9) - Grisham	-68.6	-98.8	-102.2	-95.5	-64.7
$\overline{\sigma}_u$ (sp 9) - NL FEM	-65.31	-81.77	-88.03	-85.74	-71.32
$\sigma_{u_{max}}$ (sp 1) - Grisham	69.3	85.7	84.2	88.1	67.7
$\sigma_{u_{max}}$ (sp 1) - NL FEM	61.35	73.98	96.09	110.36	101.65
$\sigma_{u_{max}}$ (sp 5) - Grisham	-101.2	-119.6	-115.7	-104.0	-89.6
$\sigma_{u_{max}}$ (sp 5) - NL FEM	-90.44	-120.64	-130.83	-125.69	-119.48
$\sigma_{u_{max}}$ (sp 9) - Grisham	-71.4	-109.4	-118.7	-118.7	-92.8
$\sigma_{u_{max}}$ (sp 9) - NL FEM	-99.21	-117.81	-122.24	-122.97	-141.72

Table 3.1: Upright stress results: non-linear finite element analysis

compression in the area attached to the web while the stresses at section point 1 are positive. This is due to bending of the upright out of the plane, as a result of the eccentric loading. Very high stresses occur at the two ends of the uprights at their attachment points to the flanges. This is especially evident from the graphs of section points 5 and 9. These peak stresses were ignored when calculating the average stress values and also when selecting the maximum stress value in each upright. The top and bottom three elements in each upright are ignored. These very high stresses are not realistic. In the finite element model, the attachment point of the uprights to the flanges are at a single node, resulting in very rigid joints with zero relative rotation of the joined elements. In practice, the uprights are riveted to the vertical leg of the flange by two or more rivets that introduce a measure of flexibility, thereby reducing the stresses at the extremities of the upright.

At high loading ratios ($\frac{\tau_{xy}}{\tau_{cr}} \gg 1$) the parabolic distribution in the upright flattens out according to the NACA. This is also evident from the results of the non-linear finite element analysis. The sinusoidal shape of the plots are due to the wrinkling effects of the web on the upright. To compare the stress distribution in the uprights for the two approaches, plots of each upright are made for the non-linear finite element model and the Grisham algorithm, at each section point. The comparative results for the five uprights, at each section point, are shown in Figures 3.8 to 3.22. The results of the two plots for section points 5 and 9 (adjacent to the web), compare very well. The results for section point 1 (the upright leg perpendicular to the web and loaded eccentrically) are more erratic although the average values still compare well.

University of Pretoria etd – Viljoen, A (2004)

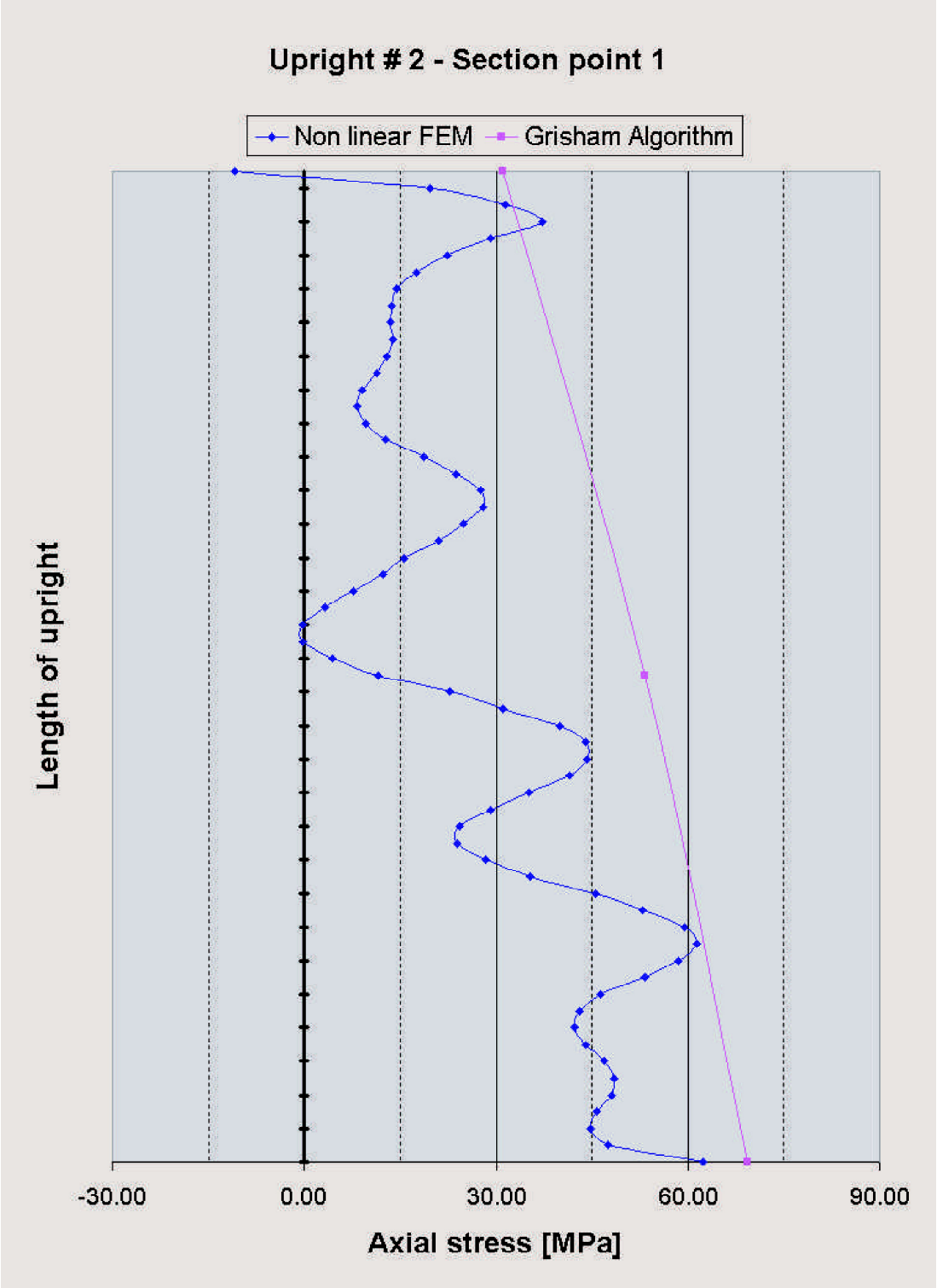


Figure 3.8: Comparative stress values in upright 2 at section point 1

University of Pretoria etd – Viljoen, A (2004)

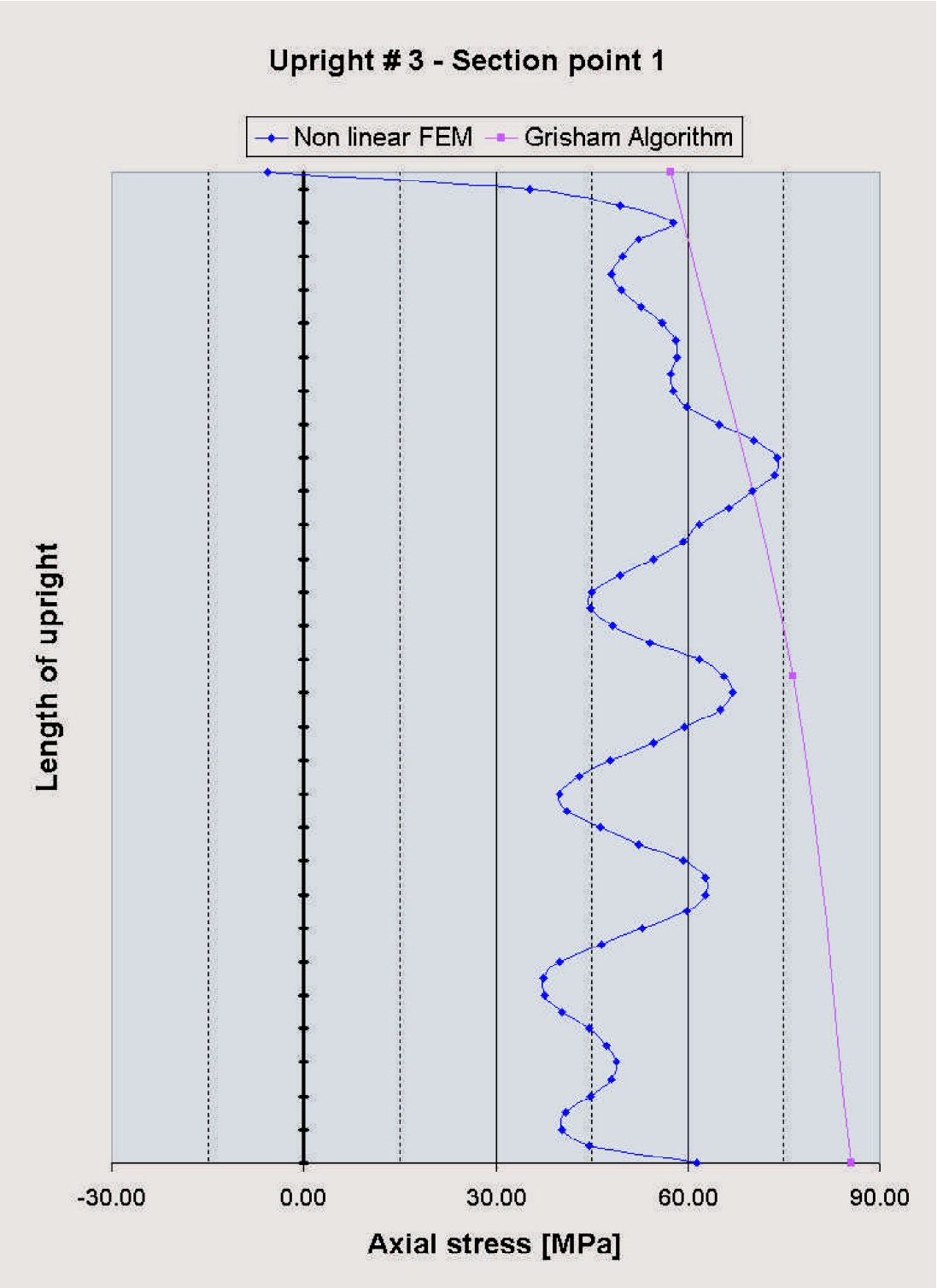


Figure 3.9: Comparative stress values in upright 3 at section point 1

University of Pretoria etd – Viljoen, A (2004)

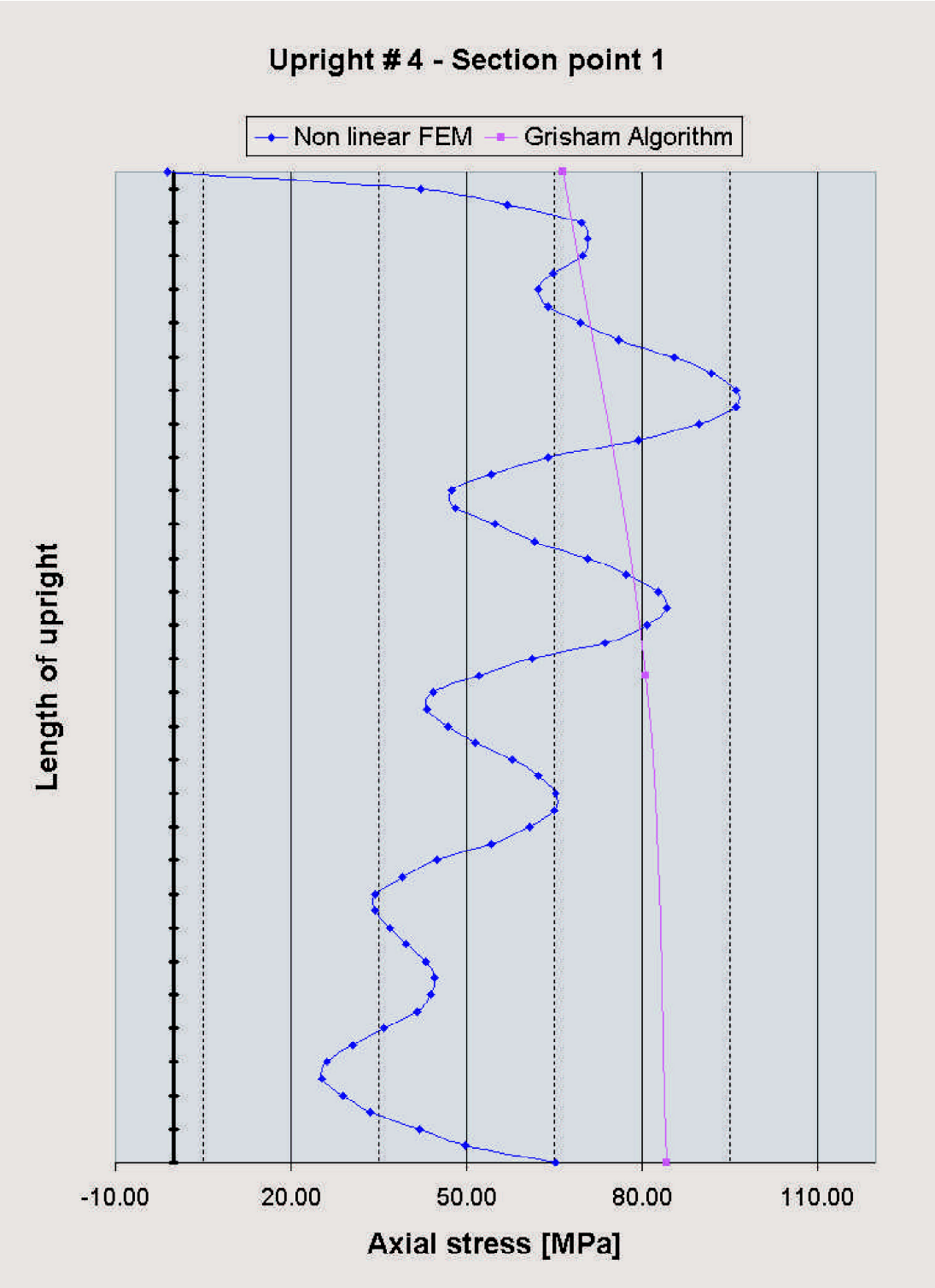


Figure 3.10: Comparative stress values in upright 4 at section point 1

University of Pretoria etd – Viljoen, A (2004)

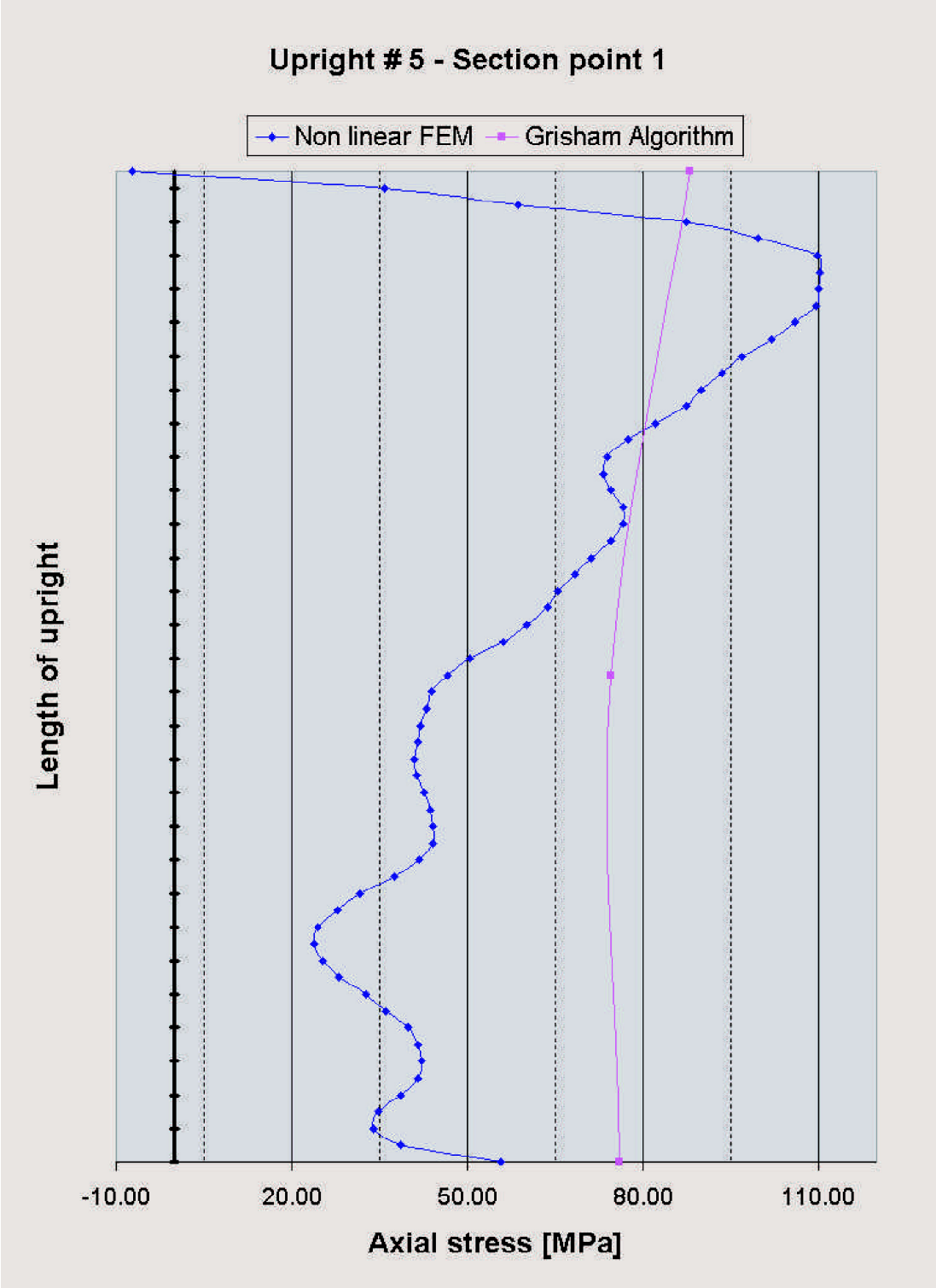


Figure 3.11: Comparative stress values in upright 5 at section point 1

University of Pretoria etd – Viljoen, A (2004)

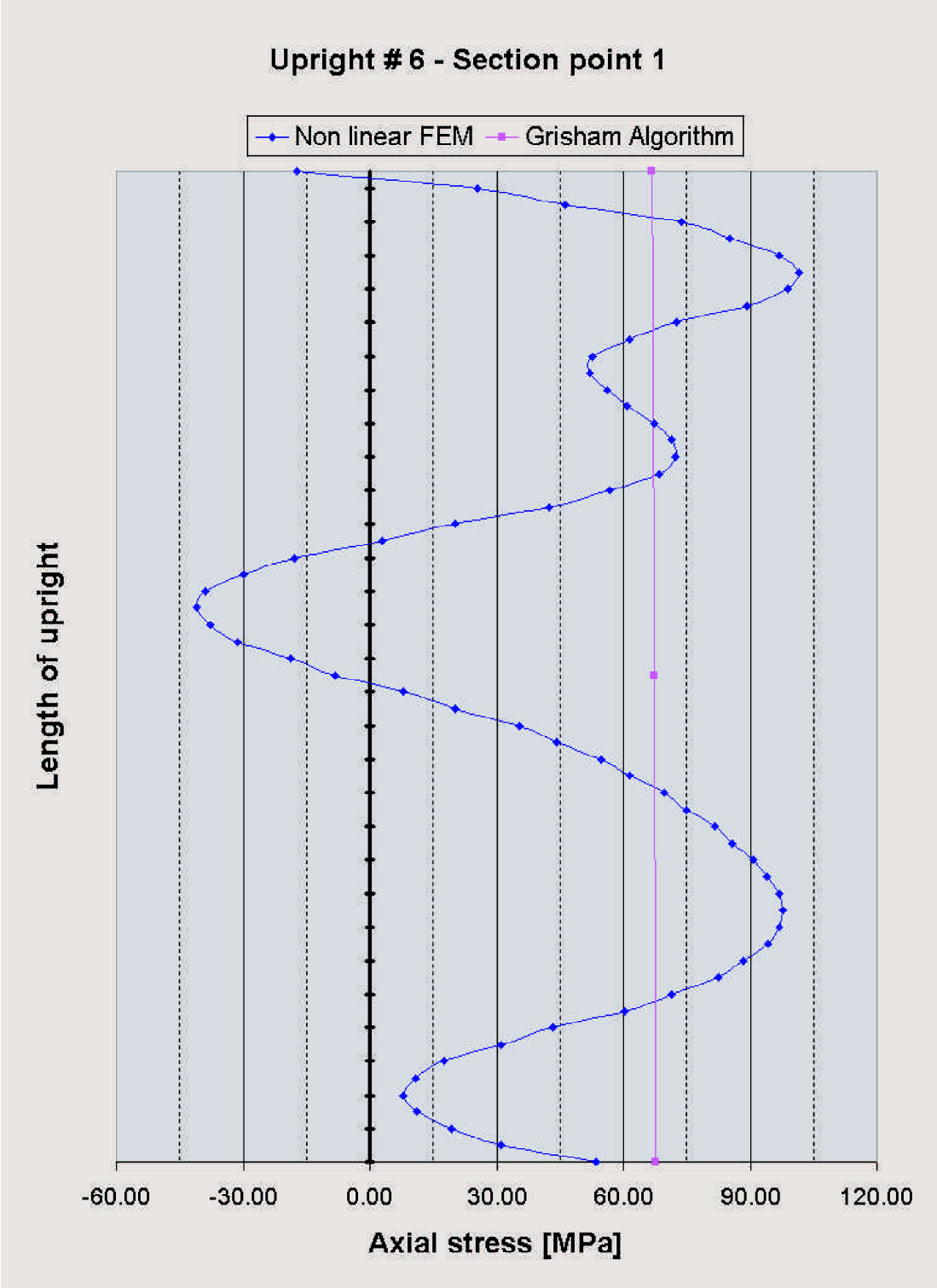


Figure 3.12: Comparative stress values in upright 6 at section point 1

University of Pretoria etd – Viljoen, A (2004)

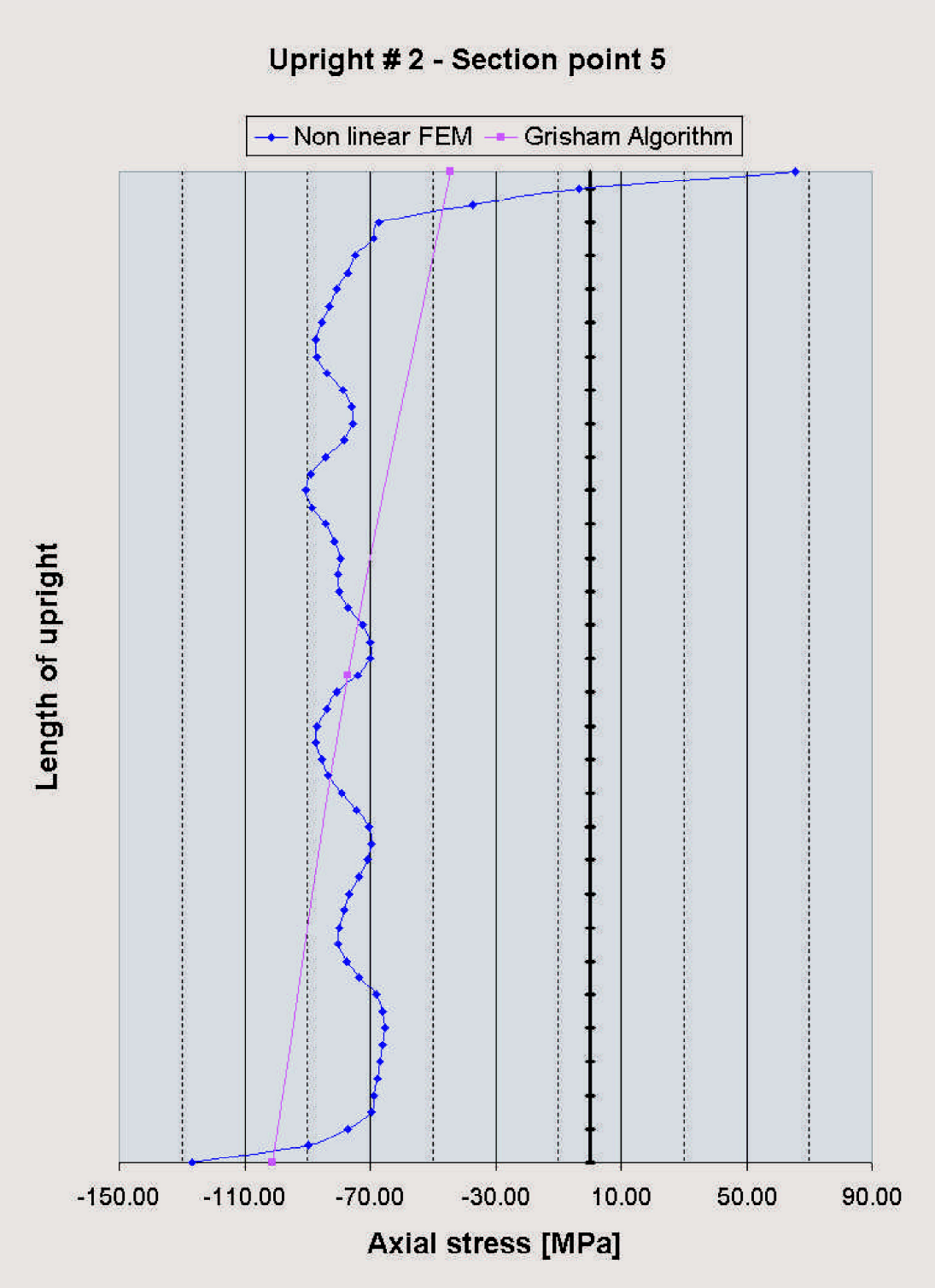


Figure 3.13: Comparative stress values in upright 2 at section point 5

University of Pretoria etd – Viljoen, A (2004)

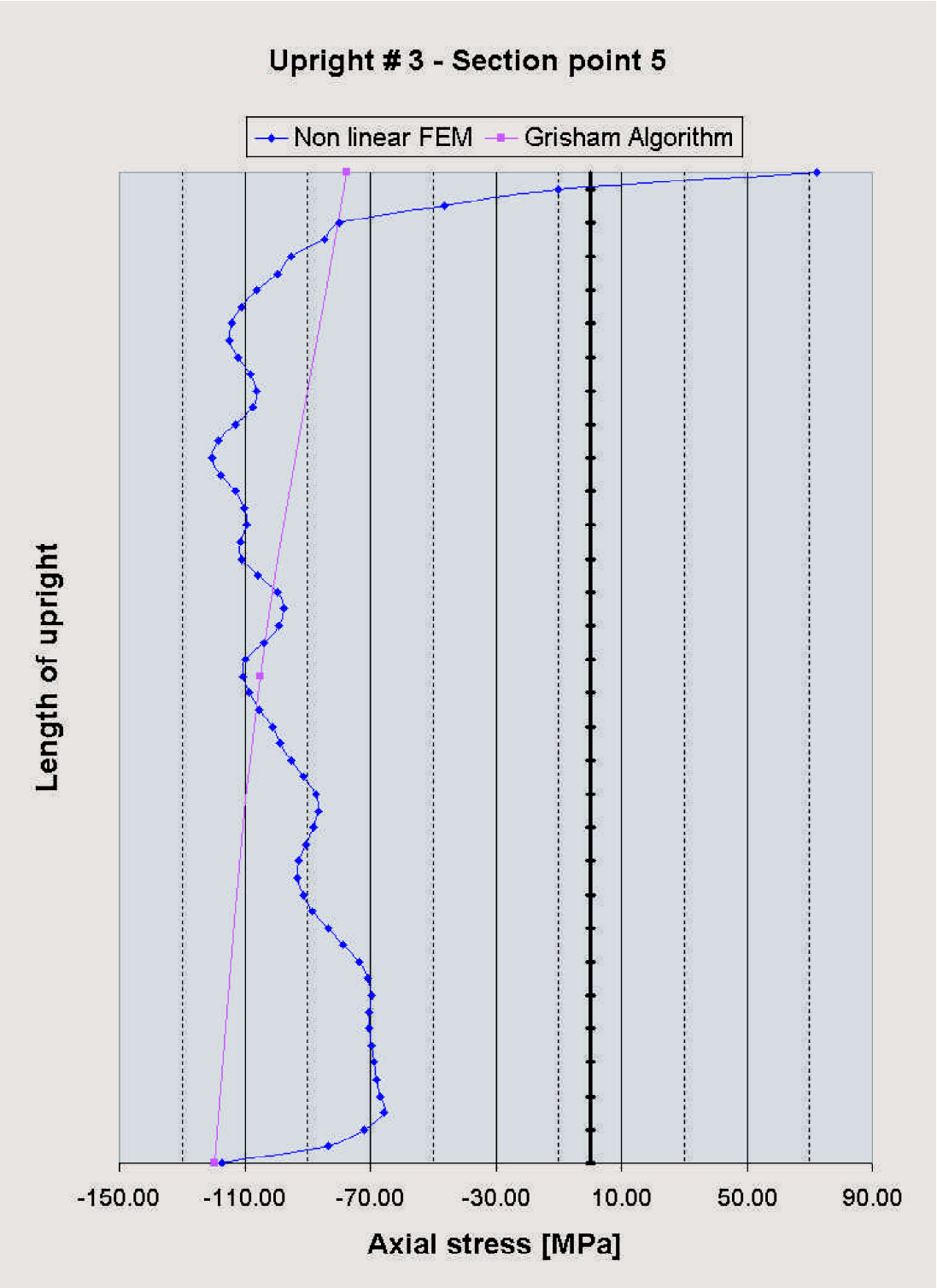


Figure 3.14: Comparative stress values in upright 3 at section point 5

University of Pretoria etd – Viljoen, A (2004)

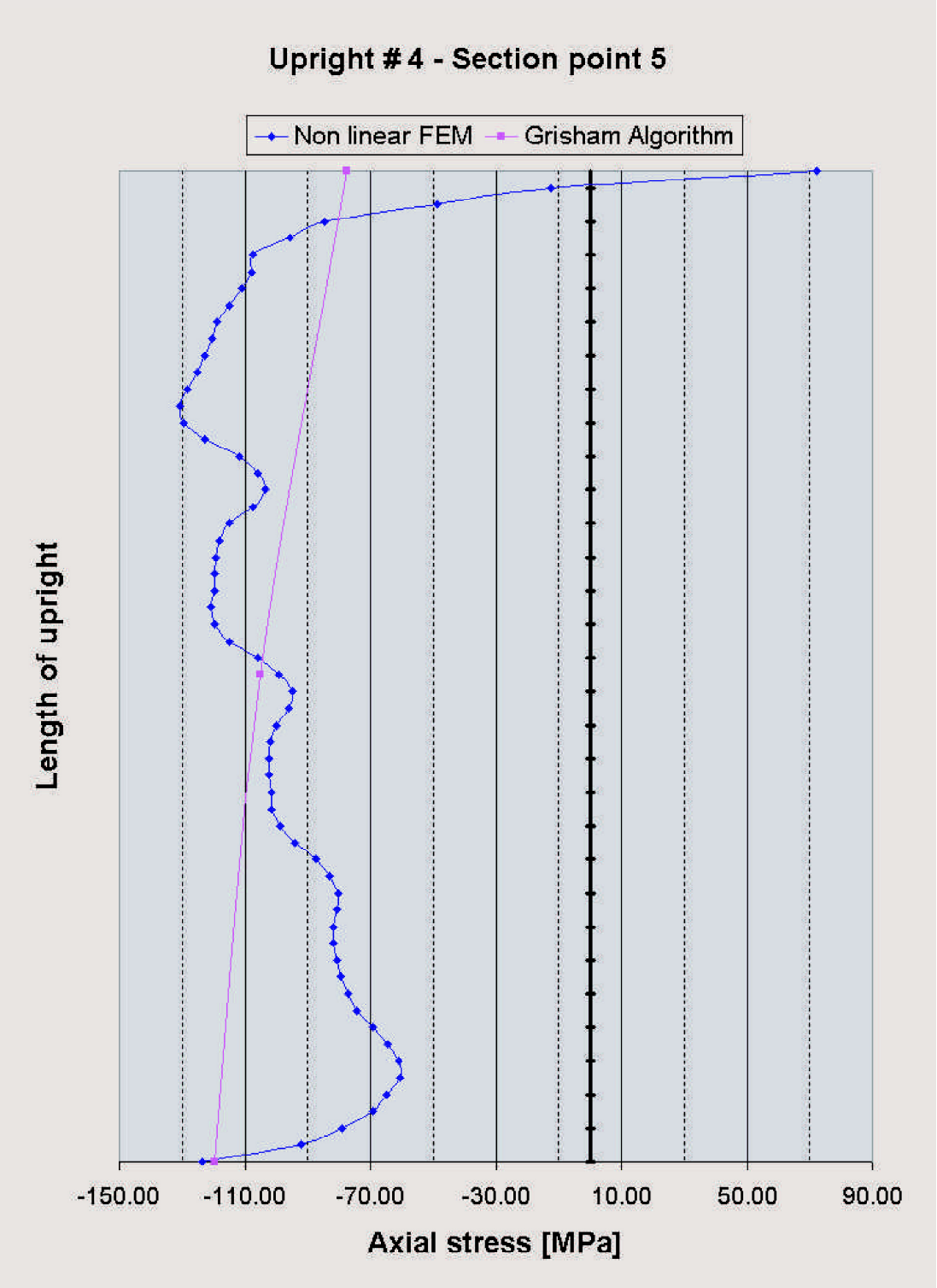


Figure 3.15: Comparative stress values in upright 4 at section point 5

University of Pretoria etd – Viljoen, A (2004)

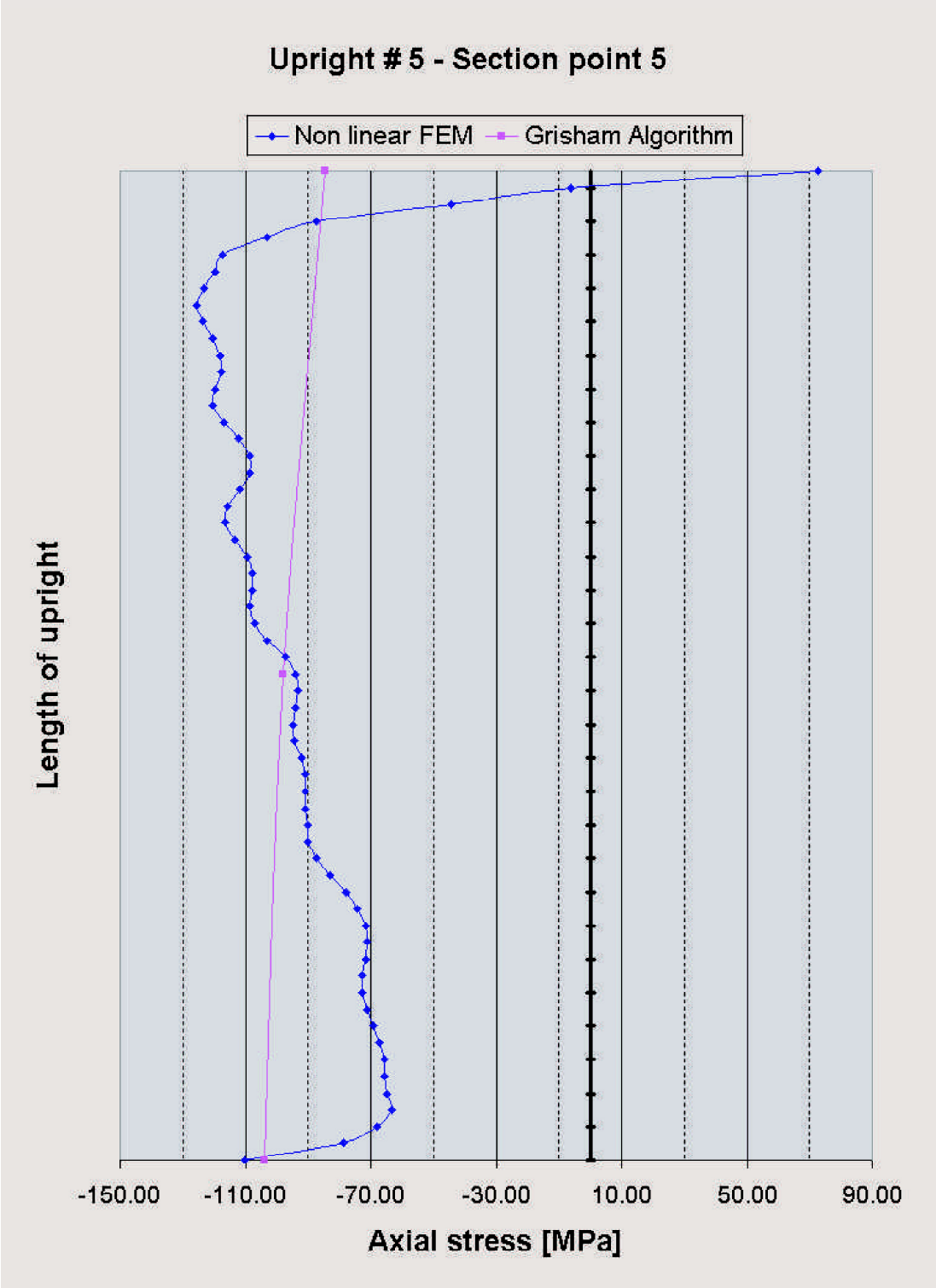


Figure 3.16: Comparative stress values in upright 5 at section point 5

University of Pretoria etd – Viljoen, A (2004)

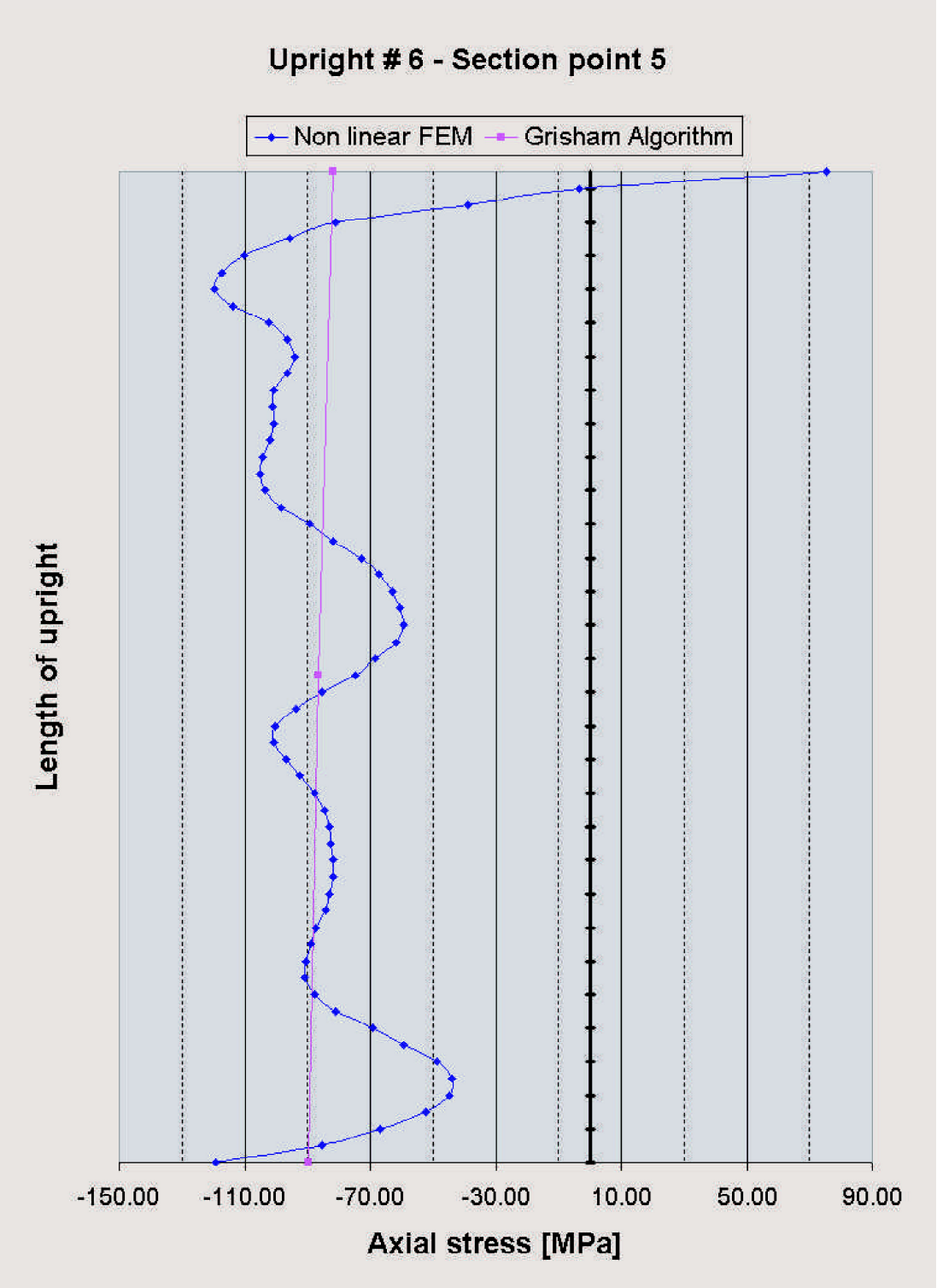


Figure 3.17: Comparative stress values in upright 6 at section point 5

University of Pretoria etd – Viljoen, A (2004)

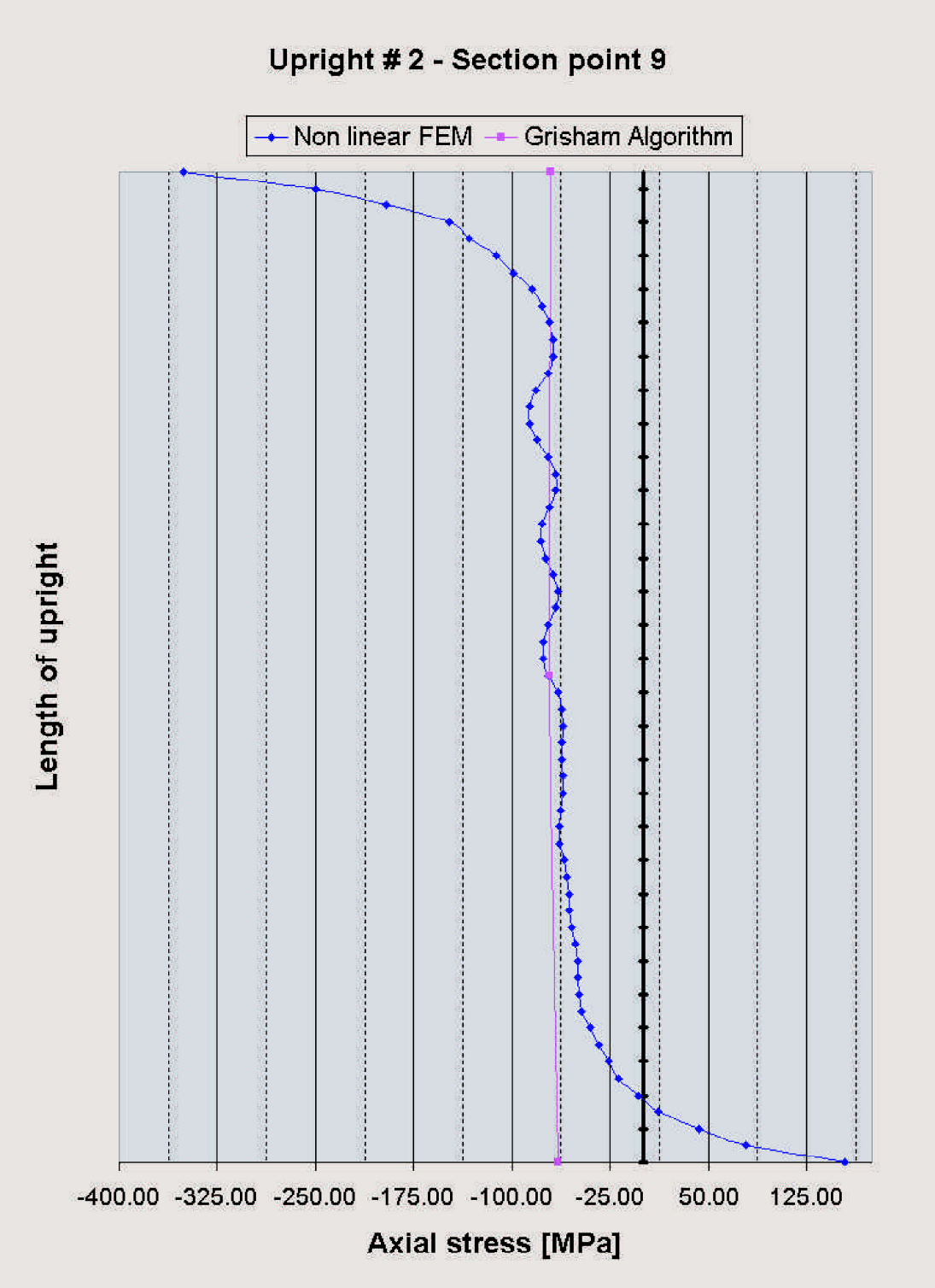


Figure 3.18: Comparative stress values in upright 2 at section point 9

University of Pretoria etd – Viljoen, A (2004)

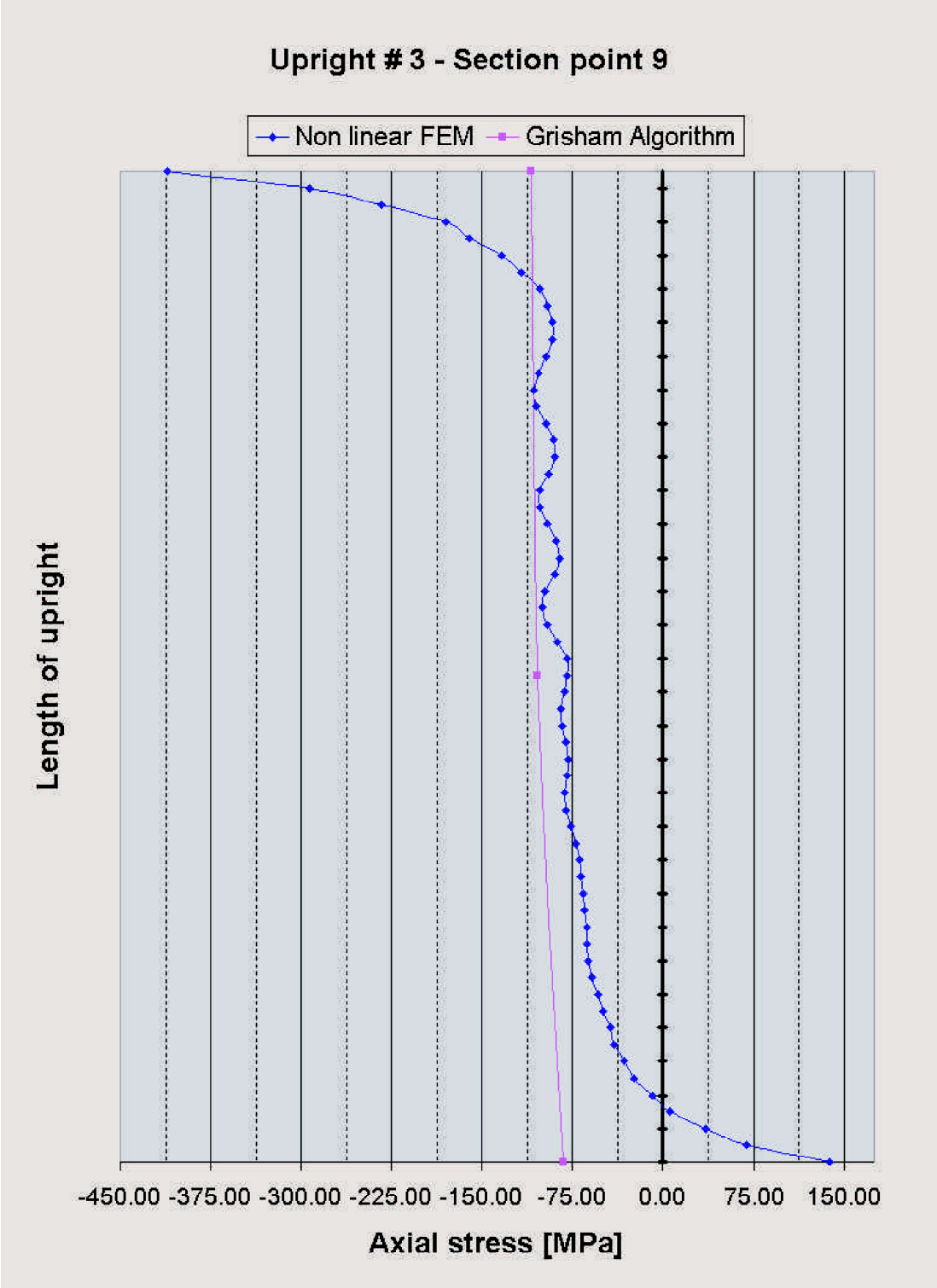


Figure 3.19: Comparative stress values in upright 3 at section point 9

University of Pretoria etd – Viljoen, A (2004)

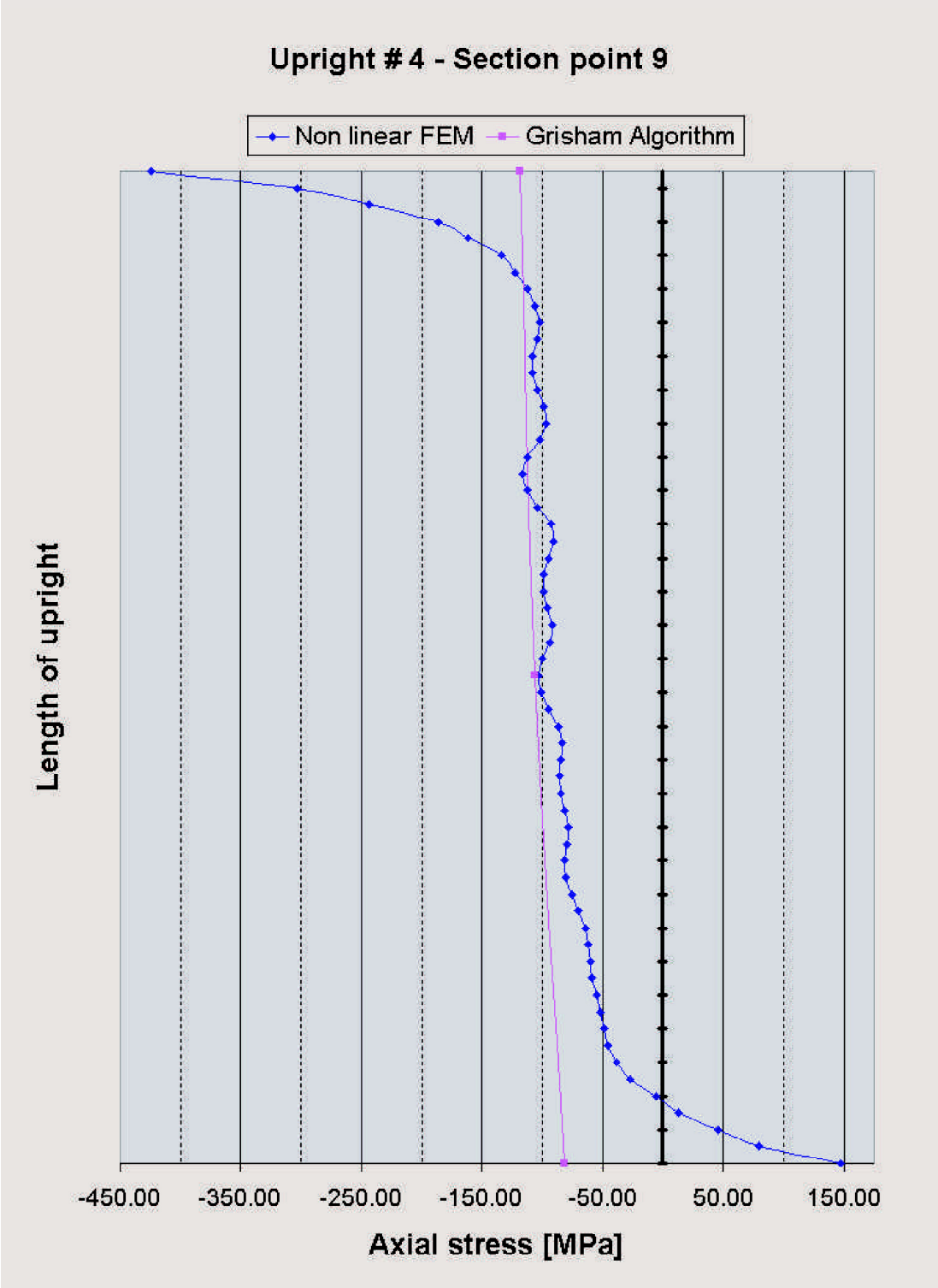


Figure 3.20: Comparative stress values in upright 4 at section point 9

University of Pretoria etd – Viljoen, A (2004)

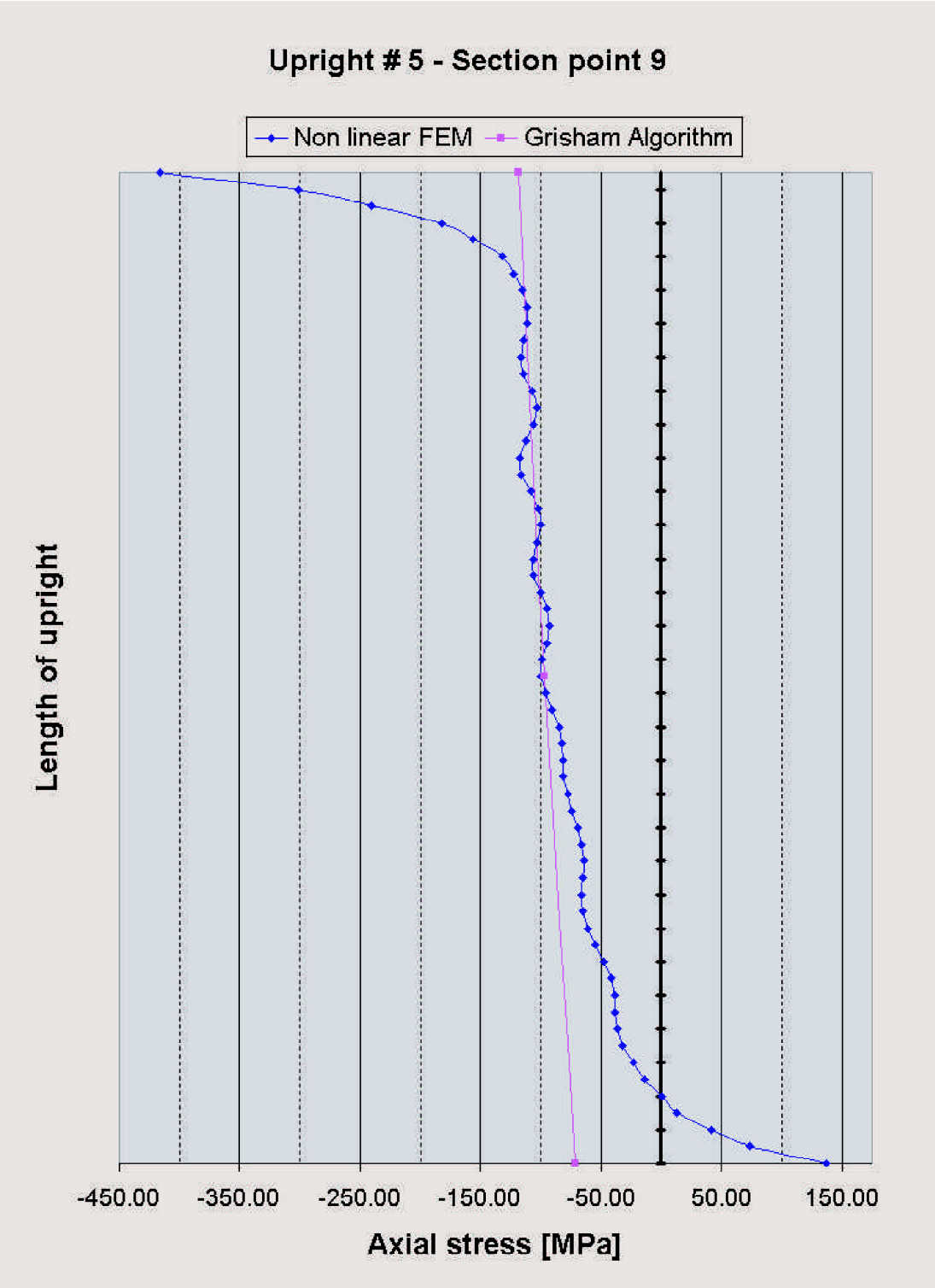


Figure 3.21: Comparative stress values in upright 5 at section point 9

University of Pretoria etd – Viljoen, A (2004)

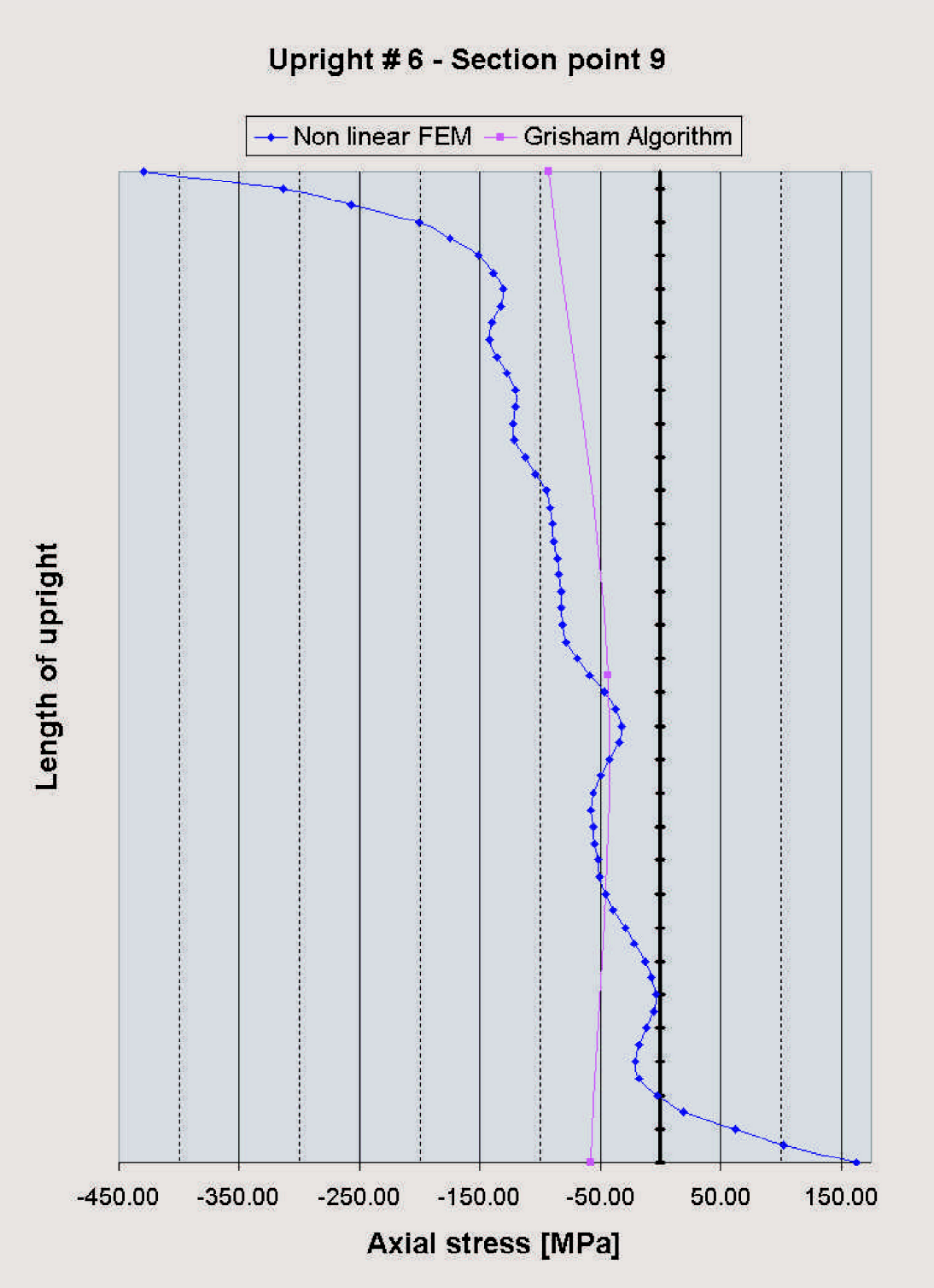


Figure 3.22: Comparative stress values in upright 6 at section point 9

University of Pretoria etd – Viljoen, A (2004)

3.5 Flange results

Table 3.2 shows the stress results in the upper and lower flanges of the non-linear finite element model compared to that of the Grisham algorithm. The correlation is reasonable, although it is noticed that the Grisham results are no longer conservative.

	Grisham Algorithm	Non-linear FEM
σ_{f_u} [MPa] (upper fibre)	-216.9	-259.4
σ_{f_u} [MPa] (lower fibre)	-255.0	-389.4
σ_{f_l} [MPa] (upper fibre)	162.9	196.7
σ_{f_l} [MPa] (lower fibre)	275.2	292.7

Table 3.2: Flange stress values at section A-A (Figure 2.1): non-linear finite element analysis

3.6 Deflection results

The displaced shape of the model, as viewed from the side, is shown in Figure 3.23. The out-of-plane displacement results of the analysis are shown in Figures 3.24 to 3.27. Figure 3.24 shows the onset of shear buckling after 13 load increments (at an applied shear load of 1552 N). Figures 3.25 and 3.26 show increased effects of shear buckling after 20 and 30 load increments into the analysis respectively. Figure 3.27 shows the final displaced shape after 102 load increments (at an applied shear load of 60 048 N).

Figure 3.28 shows a plot of the structure's displaced shape as viewed from a small angle off the mid-plane of the beam. This allows the "folds" in the webs, caused by diagonal tension effects, to be seen clearly. This is of course not visible in the linear finite element model in the Grisham algorithm. The maximum tensile principal stress in the sheet occurs parallel to these "folds" while the minimum principal stress occurs perpendicular to the "folds" and is usually in compression.

Figure 3.29 shows the displaced shape of the upper frame only (without the shell elements) so that the deformation of the upper flange and uprights can be viewed more clearly. The concave shape in the upper flange is due to the vertical components of the web stress (diagonal tension action pulling down), causing bending in the flanges between the uprights. The lower flange has a similarly displaced shape due to the diagonal tension effect.

The deflection of the structure is another variable that can be used for comparative and validation purposes between the non-linear finite element results and the Grisham algorithm results. The bottom node of the loaded end deflected 22.3 mm for the Grisham algorithm and 25.05 mm for the non-linear finite element model.

University of Pretoria etd – Viljoen, A (2004)

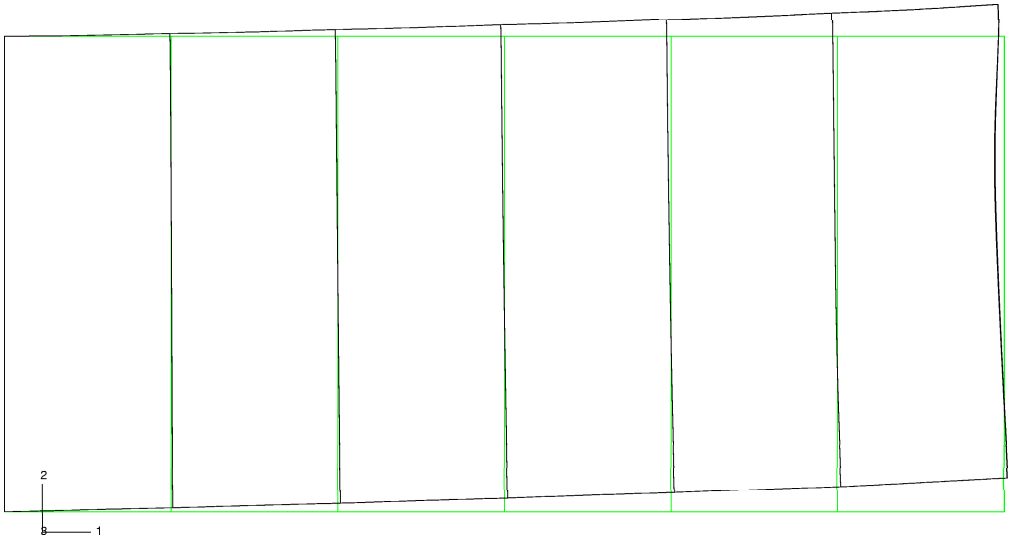


Figure 3.23: Displaced shape of beam viewed from the side

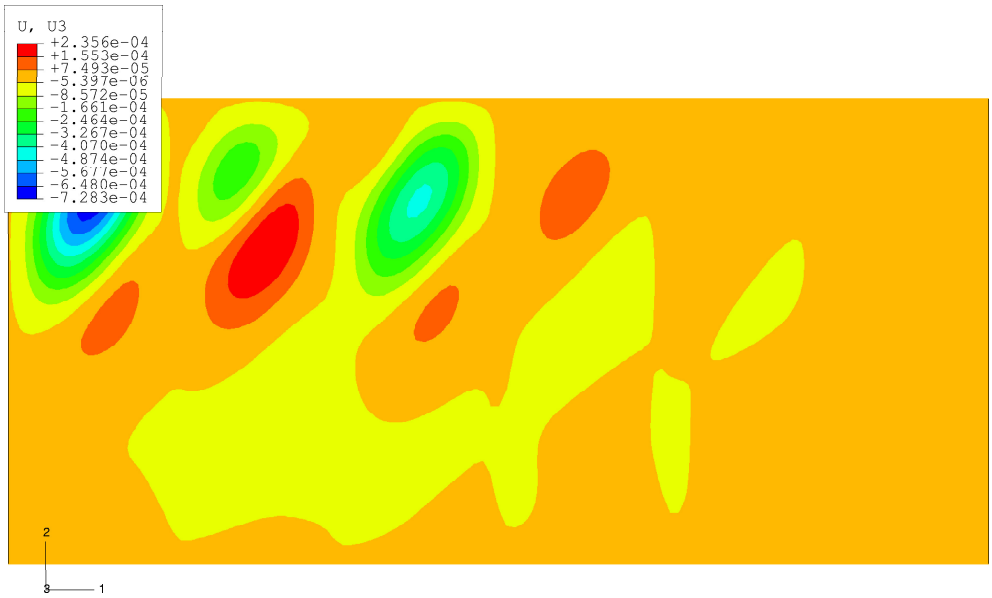


Figure 3.24: Onset of shear buckling - after 13 load increments (at an applied shear load of 1 552 N)

University of Pretoria etd – Viljoen, A (2004)

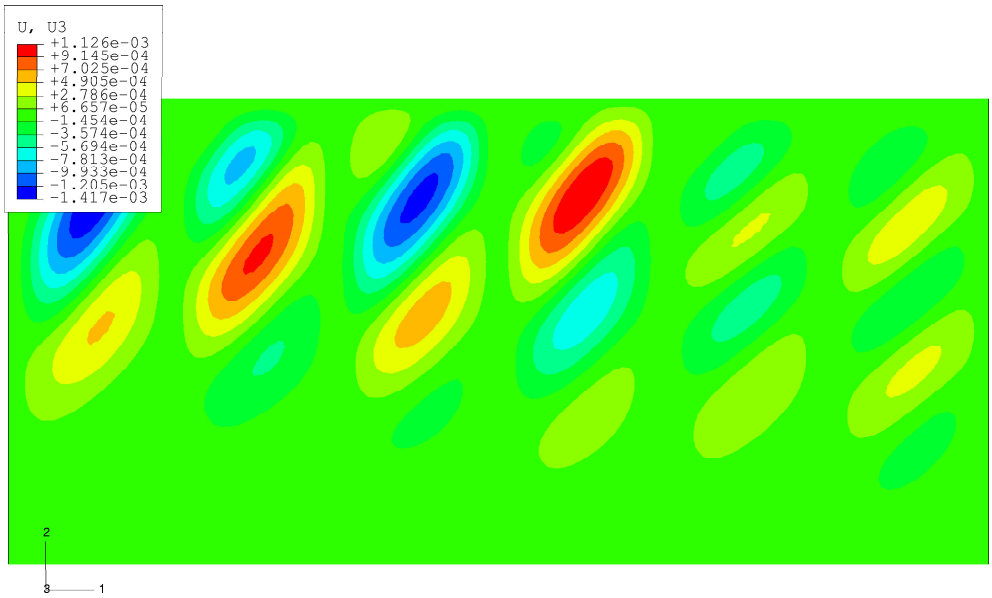


Figure 3.25: Progressive shear buckling - after 20 load increments

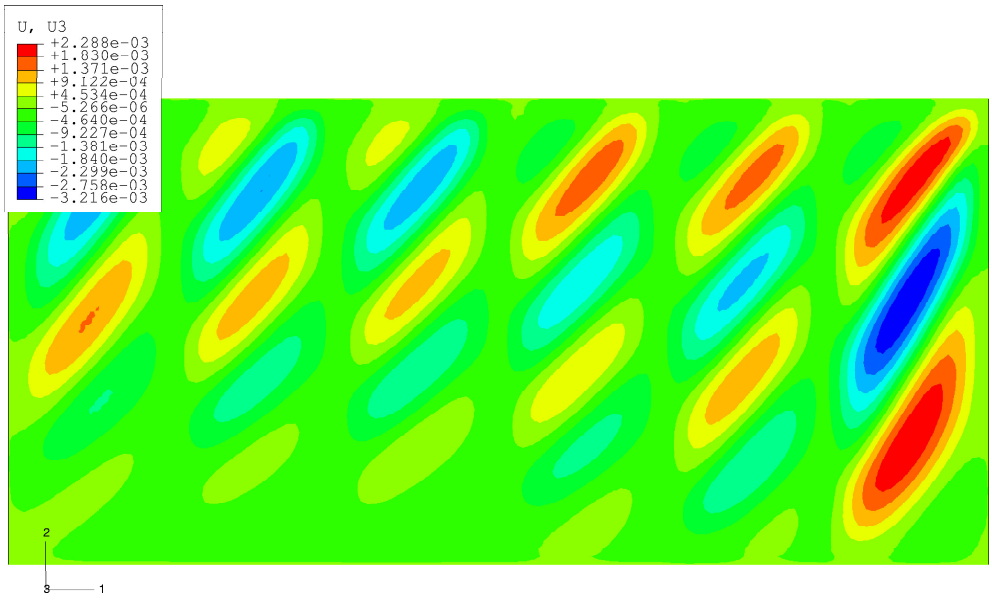


Figure 3.26: Severe shear buckling - after 30 load increments

University of Pretoria etd – Viljoen, A (2004)

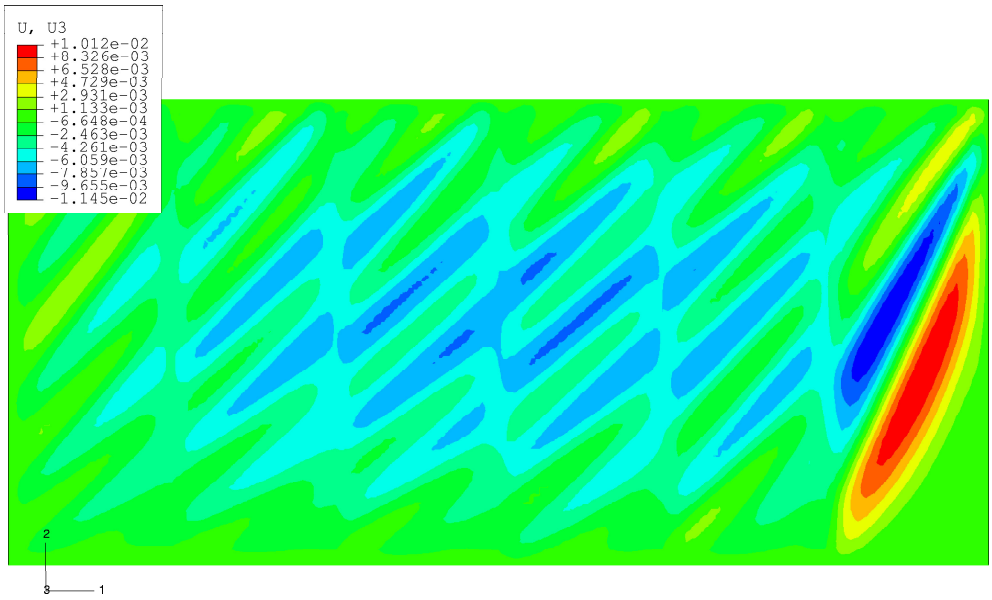


Figure 3.27: Excessive shear buckling at the end of the analysis - after 102 load increments (applied shear load of 60 048 N)

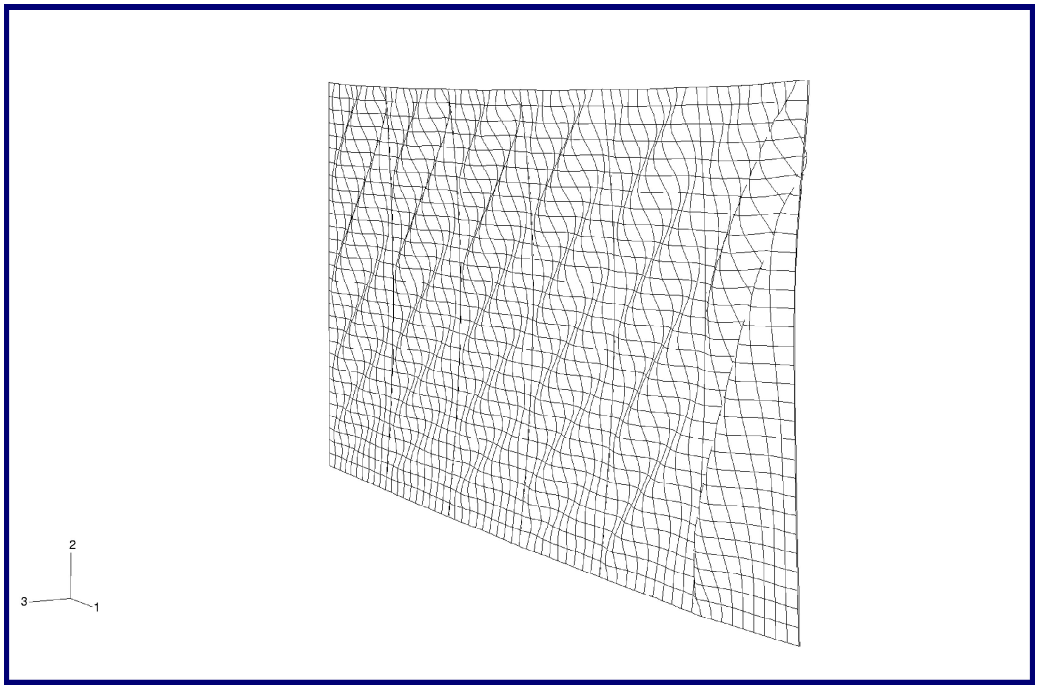


Figure 3.28: "Folds" visible in the displaced geometry

University of Pretoria etd – Viljoen, A (2004)

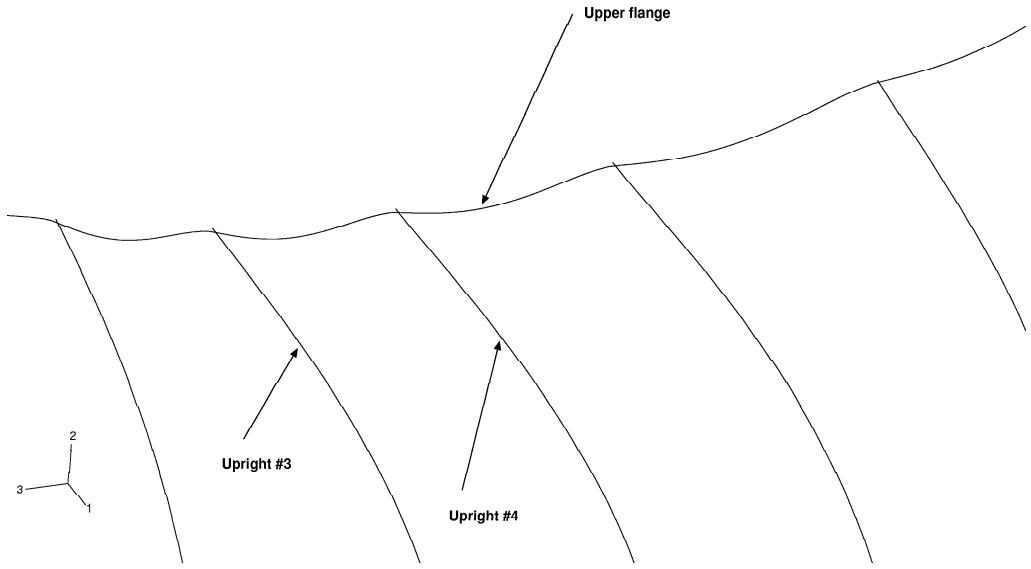


Figure 3.29: Upper flange deformation due to diagonal tension

3.7 The damping factor

For the example problem considered, the damping factor $c = 1.24 \times 10^6$ is the default value selected by the ABAQUS® finite element analysis software. The sensitivity of the factor at the onset of buckling of the structure is investigated in Table 3.3. The results show that the damping factor has a very small effect on the onset of buckling for lower values of c than the default value. Even values slightly higher have little effect and only once the damping factor is increased 1000 $\times$, is a significant effect noted. The value of the damping constant is therefore considered not to have a large influence on the results of the non-linear finite element analysis.

c	S_b
1.24×10^9	11828
1.24×10^7	1747
1.48×10^6	1552
1.24×10^6	1169
0.99×10^6	1163
1.24×10^5	1163
1.24×10^3	1163

Table 3.3: Damping factor c versus shear load at onset of buckling S_b

University of Pretoria etd – Viljoen, A (2004)

3.8 Mesh convergence study

A convergence study was done to ensure a reasonably inexpensive, yet adequately converged, finite element mesh. Table 3.4 shows that a discretization of 3×10 is sufficient. The discretizations in Table 3.4 are all selected to ensure acceptable element aspect ratios.

Mesh	τ_{cr}
1×3	2.446
3×10	2.315
5×15	2.314
10×30	2.356
20×60	2.380
30×90	2.388

Table 3.4: Mesh discretization versus critical buckling stress τ_{cr}

3.9 Run-time comparison

A run-time comparison between the Grisham algorithm and the non-linear finite element analysis is now used to indicate which method is more efficient from a computational point of view. In order to do this effectively, identical mesh discretizations are used for the two models. The results are shown in Table 3.5, where the mesh given represents the mesh for the web of *each* panel. All analyses were run on a HP C200 workstation. Although cpu times would be better, wall clock times are used in this comparison because the Grisham algorithm code can only log real time. Cpu times would give the effective time that the workstation spent on each run (process).

	Time	
Mesh	Grisham Algorithm	Non-linear FEM
3×3	31 secs	7 mins, 42 secs
10×30	30 min, 39 secs	4 hours, 59 mins

Table 3.5: Total wall-clock times for the Grisham method and non-linear finite element analysis as a function of mesh discretization

The Grisham algorithm is run with β-values which allow for the solution to converge (see Chapter 4). An initial estimate is made for the first run, after which minor adjustments are made until the solution converges. This is achieved within 2 to 3 iterations. The results indicate that the Grisham algorithm presents *an efficient and simple methodology for initial design sizing.*

University of Pretoria etd – Viljoen, A (2004)

Chapter 4

On convergence and aspect ratio

4.1 Convergence criteria

The two convergence criteria for the Grisham algorithm are:

1. the ratio of the compressive stress (σ_{x_c}, σ_{y_c}) divided by the modified critical stress ($\sigma_{x_{cr}}$, $\sigma_{y_{cr}}$) must approach one [$\frac{\sigma_{x_c}}{\sigma_{x_{cr}}} \rightarrow 1$; $\frac{\sigma_{y_c}}{\sigma_{y_{cr}}} \rightarrow 1$], and

2. the difference of the diagonal tension stress ($\sigma_{x_{DT}}$, $\sigma_{y_{DT}}$) from one iteration to the next must approach zero. [$\sigma_{x_{DT_i}}$ - $\sigma_{x_{DT_{i-1}}} \rightarrow 0$; $\sigma_{y_{DT_i}}$ - $\sigma_{y_{DT_{i-1}}} \rightarrow 0$]

Figures 4.1 and 4.2 show the convergence data for the example problem considered in Chapters 2 and 3, as a function of the iteration count (the stopping criteria are disabled). It is evident from the graphs that the stresses in the x- and y- directions, for the second convergence criterion, converge within five iterations. This is to within a 0.037 % error (taken relative to the final diagonal tension stress) and well within the 2 % margin allowed. The negative value at iteration three, on the graph for the second convergence criterion, is acceptable since convergence is not expected to be monotonic.

After another two iterations, the first convergence criterion converges to a value of one for both the x- and y- directions. This is because the first convergence criterion depends on the diagonal tension stress values. The first convergence criterion converges to within a 1.53 % error value which is again well within the 2 % margin allowed.

The diagonal tension stresses from the previous iteration are subtracted from the membrane stresses in the linear finite element stress results of the current iteration before they are used in the equations of the Grisham algorithm (σ_{x_i} - $\sigma_{x_{DT_{i-1}}}$; σ_{y_i} - $\sigma_{y_{DT_{i-1}}}$). The modified stress values appear in the equation for determining the compressive stress values of the web for each panel, which again are used in the first convergence criterion. Therefore, until the diagonal tension stress values have converged (second convergence requirement), the first convergence criterion will not be satisfied.

University of Pretoria etd – Viljoen, A (2004)

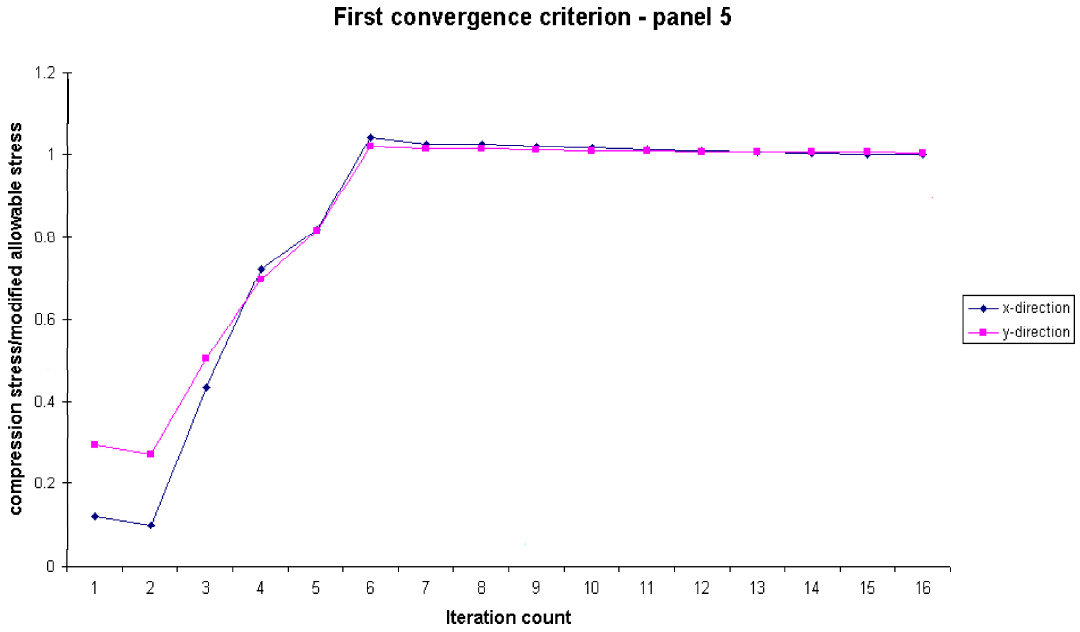


Figure 4.1: First convergence criterion for the Grisham algorithm

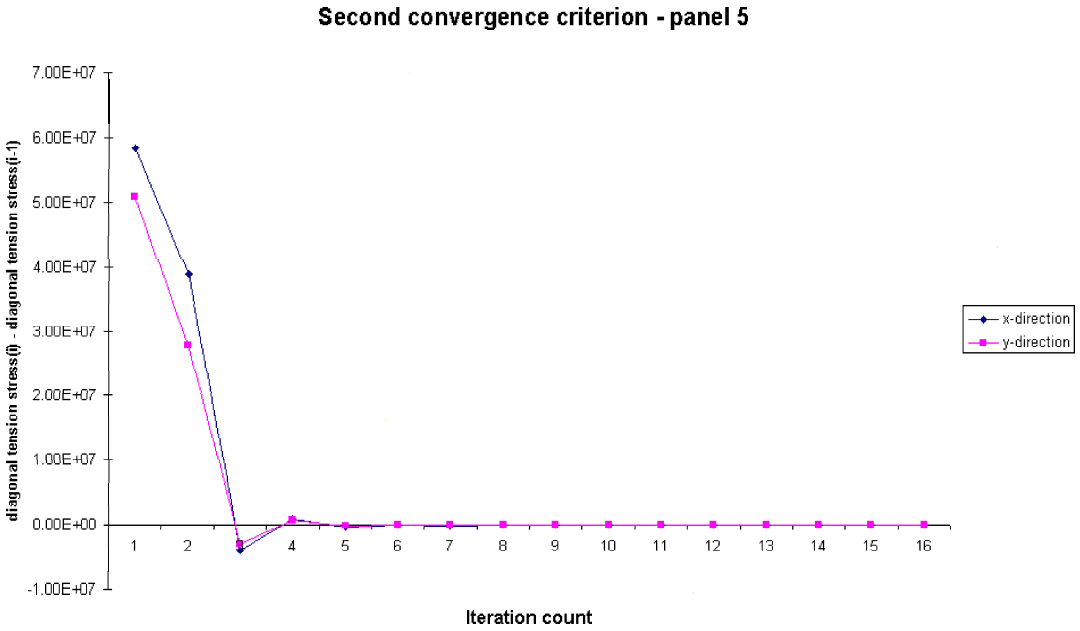


Figure 4.2: Second convergence criterion for the Grisham algorithm

University of Pretoria etd – Viljoen, A (2004)

4.2 Effect of β values on convergence

The parameters β_x and β_y are defined by Grisham [6] as the ratios of the buckled stiffness of the web in the x- and y- directions to the unbuckled stiffness of the web in the x- and y- directions respectively ($(\frac{E_{xp}}{E_p})$ and $(\frac{E_{yp}}{E_p})$) for each panel. Since the stiffness of the buckled webs are not known in advance, neither in the x- nor the y- directions initially, these values are estimates of the actual values. In the implementation of the Grisham algorithm, an initial value is assumed for both β_x and β_y for each panel. These values are then varied from iteration to iteration until the first convergence requirement approaches unity. The β values only affect the first convergence requirement (See A.29 and A.30).

When the Grisham algorithm runs through its iteration loop and the first convergence criterion approaches a value larger than unity, the relevant β value is increased and the algorithm re-run. Alternatively, when the first convergence criterion approaches a value smaller than unity, the β value is reduced, after which the analysis is re-run.

In Tables 4.1 to 4.6 the sensitivity of the β values with respect to changes in the web thickness, flange areas and upright areas are summarized. The example considered in Chapters 2 and 3 is again used for this case study. Only panels 2 to 5 are considered since panels 1 and 6 include edge effects. For the flange area investigation, both the upper and lower flanges have the same cross-sectional area.

Both the values of β_x and β_y values drop significantly when the web thickness is increased, showing that they are rather sensitive to web thickness. The β values increase only slightly when the flange and upright areas are increased, showing less sensitivity to the change of flange and upright areas. For all the tests, the values of β_x and β_y remain within $0.565 < \beta_x, \beta_y < 1.098$. These variations can all be accommodated using an *iterative, adaptive correction scheme.*

Web thickness t	Panel 2	Panel 3	Panel 4	Panel 5
0.500 [mm]	0.978	0.972	0.995	0.978
0.600 [mm]	0.960	0.958	1.018	0.962
0.635 [mm]	0.952	0.950	1.076	0.952
0.700 [mm]	0.937	0.935	1.098	0.941
0.800 [mm]	0.892	0.894	0.770	0.912
0.900 [mm]	0.854	0.850	0.822	0.872
1.000 [mm]	0.785	0.790	0.794	0.828

Table 4.1: Variation in β_x as a function of web thickness t

4.3 Effect of aspect ratio on the diagonal tension angle

To investigate the effect of panel aspect ratio on the diagonal tension angle, the aspect ratio of the example from Bruhn is now varied. The results are given in Table 4.7, and reveal a

University of Pretoria etd – Viljoen, A (2004)

Web thickness t	Panel 2	Panel 3	Panel 4	Panel 5
0.500 [mm]	0.966	0.972	0.966	0.958
0.600 [mm]	0.922	0.940	0.938	0.922
0.635 [mm]	0.898	0.928	0.930	0.912
0.700 [mm]	0.845	0.891	0.905	0.885
0.800 [mm]	0.714	0.818	0.852	0.824
0.900 [mm]	0.664	0.732	0.790	0.772
1.000 [mm]	0.565	0.641	0.696	0.688

Table 4.2: Variation in β_y as a function of web thickness t

Flange Area A_f	Panel 2	Panel 3	Panel 4	Panel 5
50.00 [mm^2]	0.848	0.840	0.838	0.830
100.00 [mm^2]	0.910	0.910	0.898	0.890
200.00 [mm^2]	0.950	0.950	0.938	0.930
243.87 [mm^2]	0.958	0.954	0.950	0.932
300.00 [mm^2]	0.966	0.964	0.952	0.938
435.48 [mm^2]	0.972	0.970	0.962	0.950
1000.00 [mm^2]	0.978	0.978	0.970	0.960
1500.00 [mm^2]	0.968	0.980	0.972	0.962
2000.00 [mm^2]	0.966	0.972	0.970	0.960

Table 4.3: Variation in β_x as a function of flange area A_f

low sensitivity to aspect ratio. This is in agreement with the numerous experimental results generated in the NACA investigation.

University of Pretoria etd – Viljoen, A (2004)

Flange Area A_f	Panel 2	Panel 3	Panel 4	Panel 5
50.00 [mm^2]	0.902	0.894	0.894	0.896
100.00 [mm^2]	0.910	0.902	0.900	0.896
200.00 [mm^2]	0.914	0.910	0.910	0.896
243.87 [mm^2]	0.918	0.912	0.910	0.896
300.00 [mm^2]	0.924	0.914	0.910	0.904
435.48 [mm^2]	0.928	0.916	0.912	0.904
1000.00 [mm^2]	0.940	0.930	0.924	0.914
1500.00 [mm^2]	0.932	0.936	0.936	0.930
2000.00 [mm^2]	0.904	0.976	0.958	0.958

Table 4.4: Variation in β_y as a function of flange area A_f

Upright Area A_u	Panel 2	Panel 3	Panel 4	Panel 5
151.2 [mm^2]	0.952	0.950	1.076	0.952
182.9 [mm^2]	0.951	0.949	1.078	0.953
217.7 [mm^2]	0.950	0.939	1.056	0.953

Table 4.5: Variation in β_x as a function of upright area A_u

Upright Area A_u	Panel 2	Panel 3	Panel 4	Panel 5
151.2 [mm^2]	0.898	0.928	0.930	0.912
182.9 [mm^2]	0.927	0.943	0.931	0.913
217.7 [mm^2]	0.939	0.947	0.934	0.925

Table 4.6: Variation in β_y as a function of upright area A_u

L_y/L_x	α
4.000	43.1
2.330	41.2
1.500	39.5
1.000	38.2
0.667	37.1
0.428	36.5
0.250	36.1

Table 4.7: Panel aspect ratio L_y/L_x versus diagonal tension angle α

University of Pretoria etd – Viljoen, A (2004)

Chapter 5

Further validation of the Grisham algorithm

In this chapter, three further examples, some with experimental evidence, are selected from the literature in an attempt to further validate the Grisham algorithm. The first of the three examples is taken from the NACA Technical Notes [3]; the second example is taken from a report by Tsongas and Ratay: *Investigation of diagonal tension beams with very thin stiffened webs* [13] and the last example is taken from Mason *et al.*: *The application of non-linear analysis techniques to practical structural design problems* [14].

5.1 Example 1 [3]

In this example one of the test beams used in the NACA test program is analyzed. In the NACA program, about 50 beams were tested. These can be divided into three different groups: medium sized beams which are either 635 mm (25 in) or 1016 mm (40 in) deep, small, heavily loaded beams which are 308.4 mm (12 in) deep and large beams which are 1905 mm (75 in) deep. In this example, a 1016 mm (40 in) deep beam with double uprights, designated by the NACA test program as: I-40-4Da, is evaluated. Reference [4] also contains measured stress results for the uprights and measured deflections which will be compared to the results of the Grisham algorithm.

A schematic layout of the five panel test beam is shown in Figure 5.1. The dimensions are:

1. $l = 2540$ mm (100 in)
2. $h = 1094.7$ mm (43.1 in)
3. $h_e = 1051.6$ mm (41.4 in)
4. $h_c = 942.3$ mm (37.1 in)
5. $d = 508$ mm (20 in)
6. $t = 0.99$ mm (0.039 in)

University of Pretoria etd – Viljoen, A (2004)

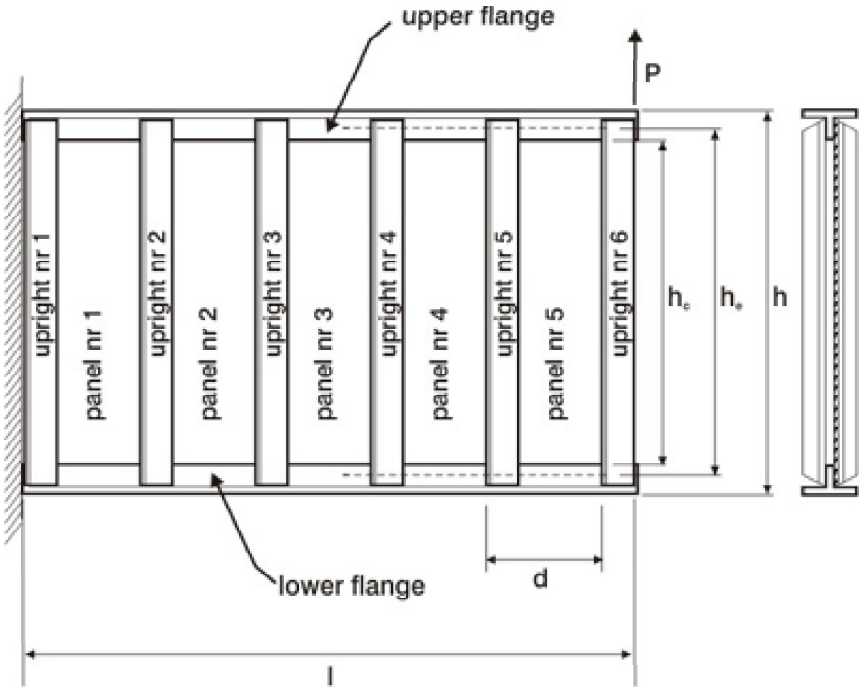


Figure 5.1: The NACA I-40-4Da test beam

7. $A_{fl} = 2293.3$ mm^2 (3.554 in^2)
8. $A_{fu} = 2293.3$ mm^2 (3.554 in^2)
9. $A_u = 227.7$ mm^2 (0.353 in^2)

The flanges are fabricated from steel while the web and uprights are fabricated from 2024-T3 alloy, with the following material properties:

1. $\sigma_{yt} = 289$ MPa (42 ksi)
2. $\sigma_{ut} = 441$ MPa (64 ksi)
3. $E_w = 72.4$ GPa (10 500 ksi)

Both flanges have a T-shaped cross-section (two angles, 76.2 mm $\times$ 76.2 mm $\times$ 7.94 mm or 3 in $\times$ 3 in $\times$ $\frac{5}{16}$ in). The uprights also have a T-shaped cross-section (two angles, 19.05 mm $\times$ 15.88 mm $\times$ 3.175 mm or $\frac{3}{4}$ in $\times$ $\frac{5}{8}$ in $\times$ $\frac{1}{8}$ in). The beam eventually failed due to column failure of the uprights at a load of 135.8 kN (30.3 kips). This load is designated the "design ultimate load" in the analysis. Two additional failure modes of the test beams are: forced crippling of the uprights and web failure.

5.1.1 The finite element model

The flanges are modelled using second order beam elements, having a T-shaped cross-section in the linear finite element analysis. The double uprights (two angles on each side of the web)

University of Pretoria etd – Viljoen, A (2004)

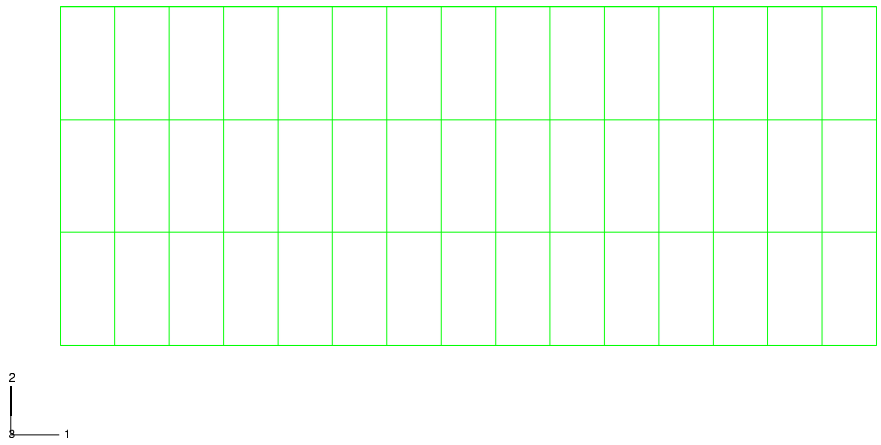


Figure 5.2: Mesh of finite element model

are also modelled with second order beam elements, having a T-shaped cross-section. The thin web is modelled with second order thin shell elements having eight nodes per element. Small displacement theory is used. The shear load is applied vertically and distributed along all the nodes at the end of the beam. Plasticity effects are not taken into account and buckling of the uprights is also not considered. NACA assumes that the web edges are clamped for calculating the critical shear and compression stress values in this example. Figure 5.2 shows the finite element mesh for the linear analysis used in the Grisham algorithm.

5.1.2 Web results

Figures 5.3 through 5.7 show the stresses in the web after completion of the analysis. Figures 5.8 and 5.9 show a vector plot of the maximum and minimum stresses in the web respectively.

In Table 5.1, the web results for the panels obtained using the iterative procedure of Grisham are compared with that of the NACA approach [3]. The correlation is good. No experimental results from the NACA tests are available for the webs. Again, the results for panels 1 and 5 are ignored since they include edge effects that are automatically included via the finite element analysis and are not accounted for in the NACA approach. The results for the three panels using Grisham's algorithm are very close to the results obtained using the NACA method.

The critical shear stress value τ_{cr} using the Grisham algorithm is $\tau_{cr} = 2.568$ MPa while the NACA procedure gives a value of $\tau_{cr} = 2.868$ MPa. The Grisham algorithm value is based on analytical relations that depend on geometry and material properties. The NACA

University of Pretoria etd – Viljoen, A (2004)

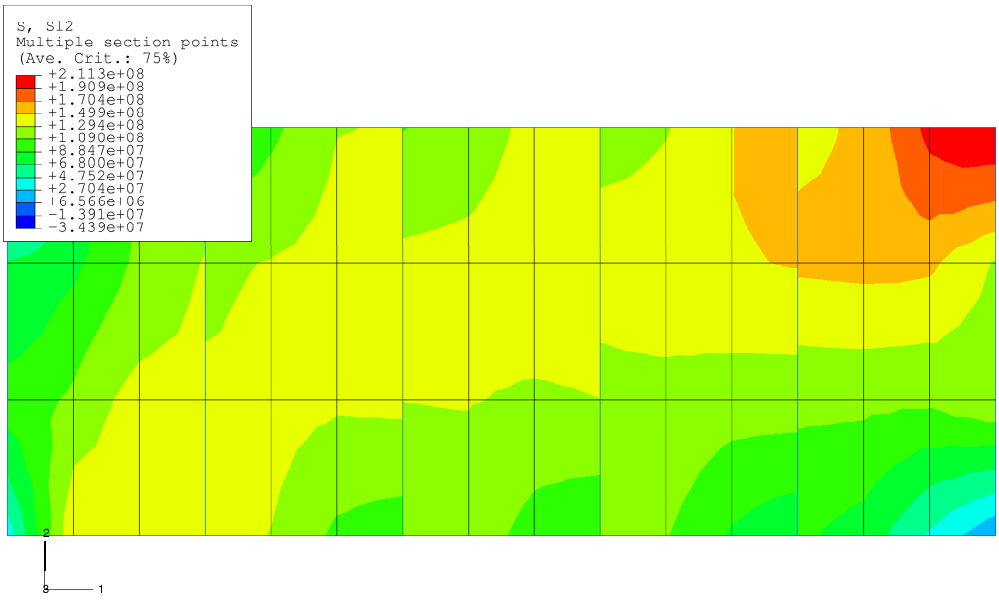


Figure 5.3: Unsmoothed shear stress distribution (τ_{xy}) in the web after the final iteration

approach uses analytical as well as empirical relations to calculate the critical shear stress. Table 5.2 shows some additional output data from the Grisham algorithm: the diagonal tension and compression-compression buckling stress values for each panel. The Modified Wagner and NACA methods do not produce values that can be compared to this data. As in the verification example, the final compressive stress values are very low. This indicates that the compressive-compressive buckling effects are insignificant in this example.

	Grisham algorithm			
	Panel 2	Panel 3	Panel 4	NACA [3]
k	0.673	0.667	0.657	0.680
α [degrees]	40.29	40.32	40.39	39.00
τ_{xy} [MPa]	127.93	128.19	128.13	129.62

Table 5.1: Web data comparison for example 1

5.1.3 Upright results

The Grisham algorithm results for the upright stresses are given in Figures 5.11 and 5.12. Again uprights number 1 and 6 are not considered because of boundary conditions. Figure 5.10 shows the position of these section points on the cross-section of the upright as well as the position of the uprights relative to the web.

University of Pretoria etd – Viljoen, A (2004)

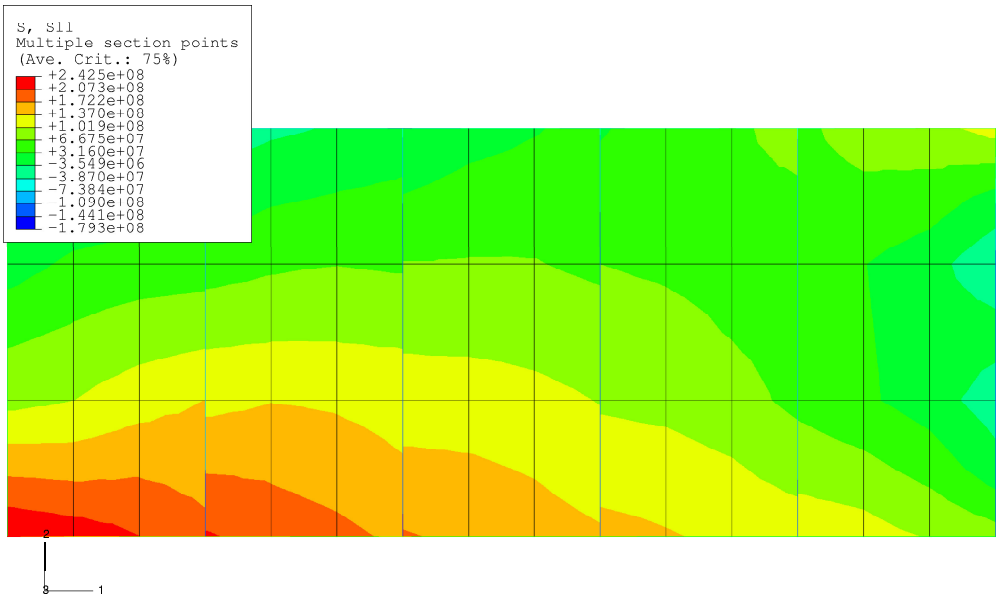


Figure 5.4: Unsmoothed normal stress (σ_x) in the web after the final iteration

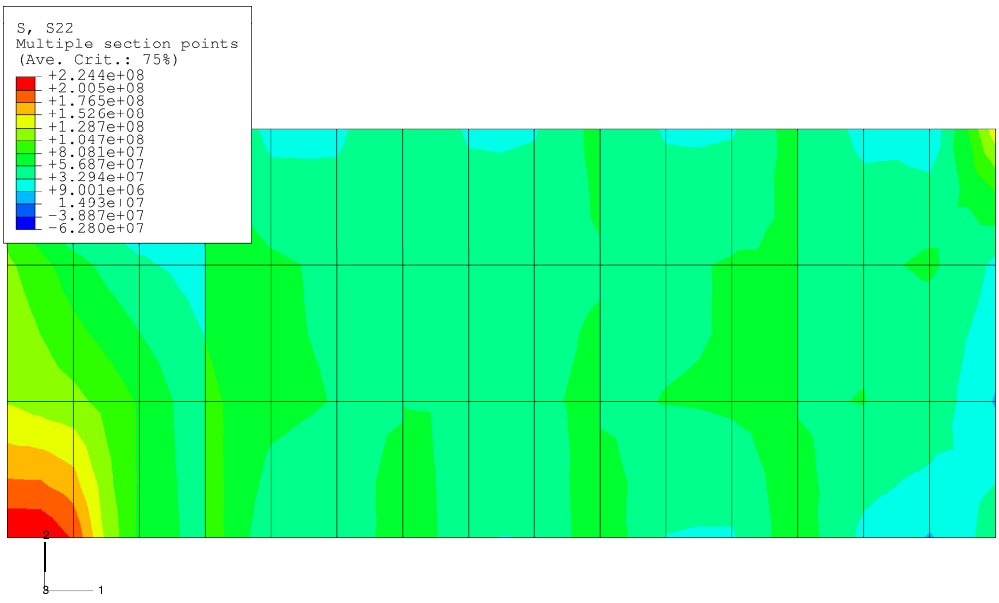


Figure 5.5: Unsmoothed normal stress (σ_y) in the web after the final iteration

University of Pretoria etd – Viljoen, A (2004)

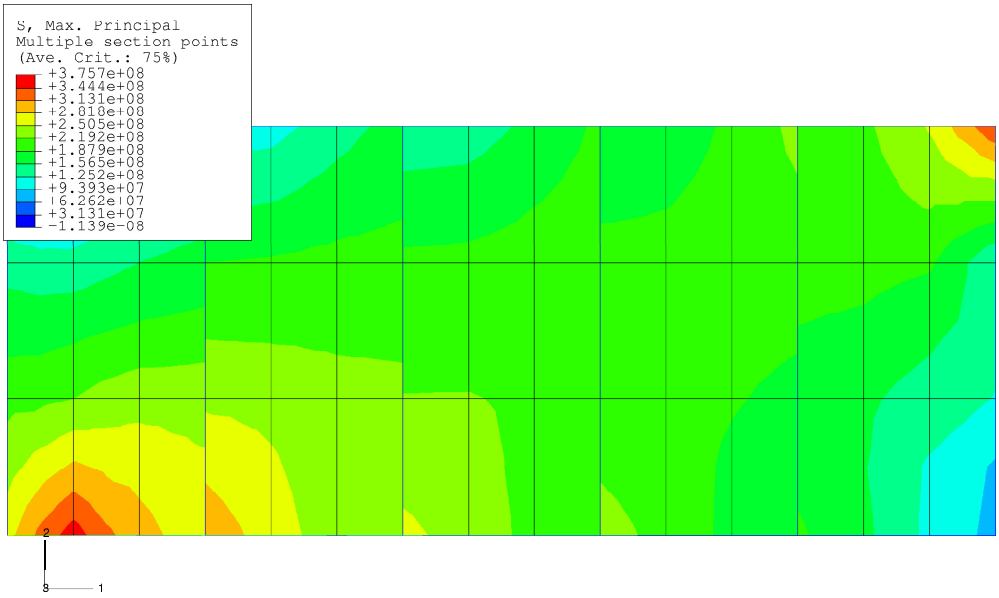


Figure 5.6: Unsmoothed maximum principal stress (σ_1) in the web after the final iteration

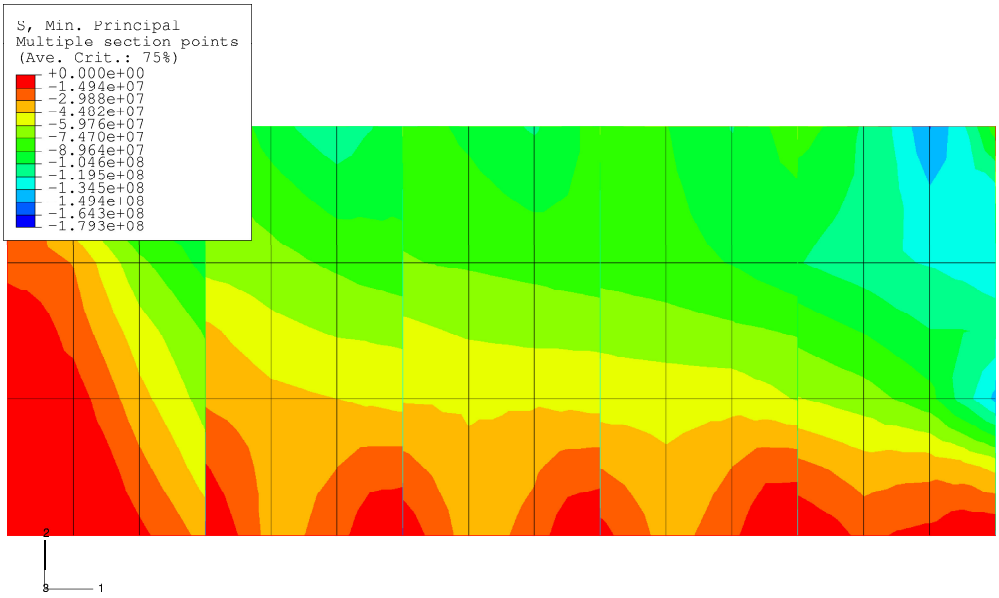


Figure 5.7: Unsmoothed minimum principal stress (σ_2) in the web after the final iteration

University of Pretoria etd – Viljoen, A (2004)

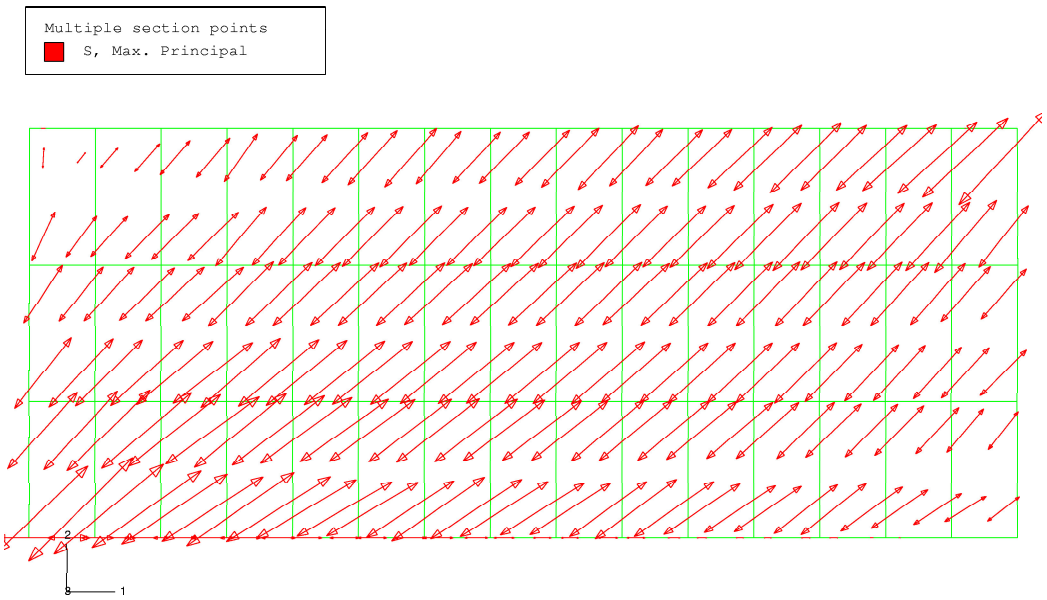


Figure 5.8: Maximum principal web stress vector plot

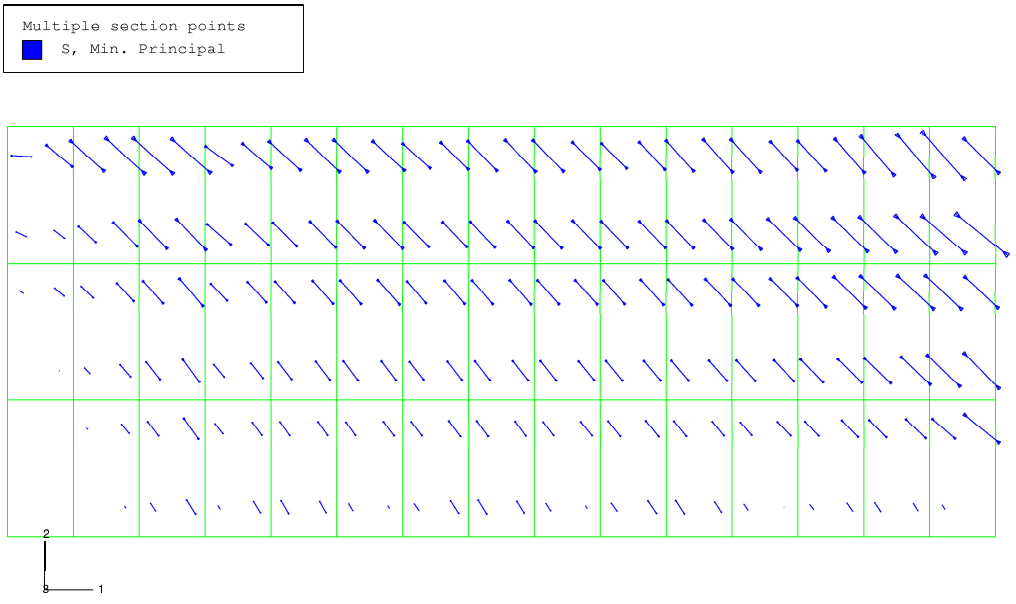


Figure 5.9: Minimum principal web stress vector plot

University of Pretoria etd – Viljoen, A (2004)

	Grisham algorithm		
	Panel 2	Panel 3	Panel 4
$\sigma_{x_{DT}}$ [MPa]	101.52	100.73	98.95
$\sigma_{y_{DT}}$ [MPa]	72.95	72.57	71.60
σ_{x_c} [MPa]	0.162	0.243	0.557
σ_{y_c} [MPa]	0.421	0.551	0.531

Table 5.2: Diagonal tension and compressive stress values calculated by the Grisham algorithm

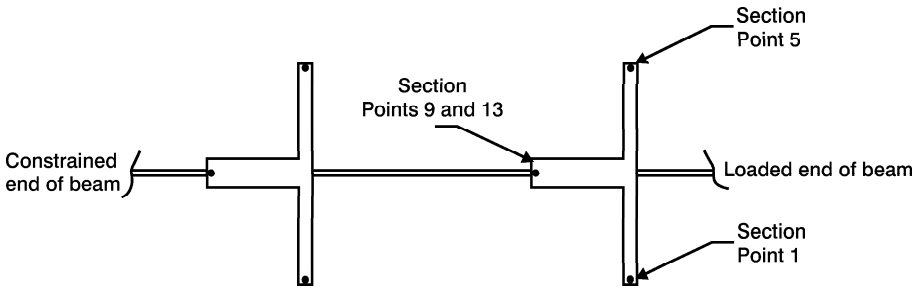


Figure 5.10: Cross-section of the upright showing the location of the section points

From the data in Figures 5.11 to 5.12 it can be seen that, for all the uprights, the values at all section points are negative and therefore in compression. The diagonal tension effects in the web tend to pull the upper and lower flanges together, loading the uprights in compression. Because there is no eccentricity here as in the verification example, and therefore no out-of-plane-bending, every point on the cross section is in compression.

The stress variation along the length of the uprights is nearly linear, increasing slightly towards the lower flange. The NACA approach, however, predicts the maximum stress value to be at a point on the neutral axis of the beam, which for this beam is half way between the two flanges.

Table 5.3 compares the results from the NACA method with the Grisham results. Measured stress values are also available from the NACA test program. The average and maximum measured upright stress as a function of load, from [4], are shown in Figure 5.13. At a load of 135.8 kN (30.3 kips) the average value is 98.6 MPa (14.3 ksi), while the maximum stress is 120.7 MPa (17.5 ksi). The average stress values for the Grisham algorithm are calculated along the length of the upright, using all the integration point values at all the section points. The measured NACA results are taken from two or three uprights in the central portion of the beam. Nine to thirteen gauge stations were used on each upright. The average measured stress in the upright is taken as the average of all the gauge readings on the beam. The Grisham algorithm results compare very well with the NACA measured results.

University of Pretoria etd – Viljoen, A (2004)

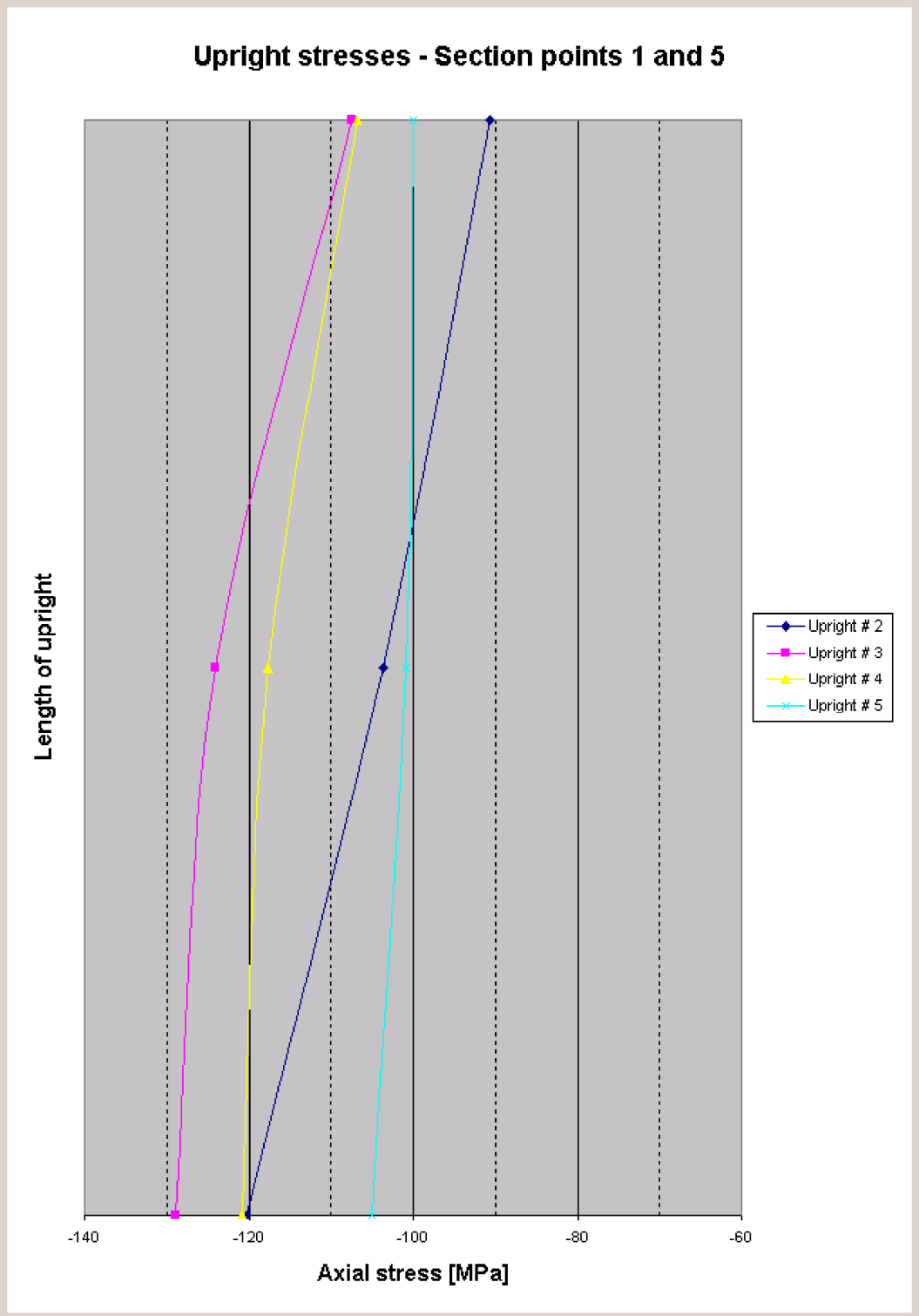


Figure 5.11: Stress along the length of the uprights at section points 1 and 5

University of Pretoria etd – Viljoen, A (2004)

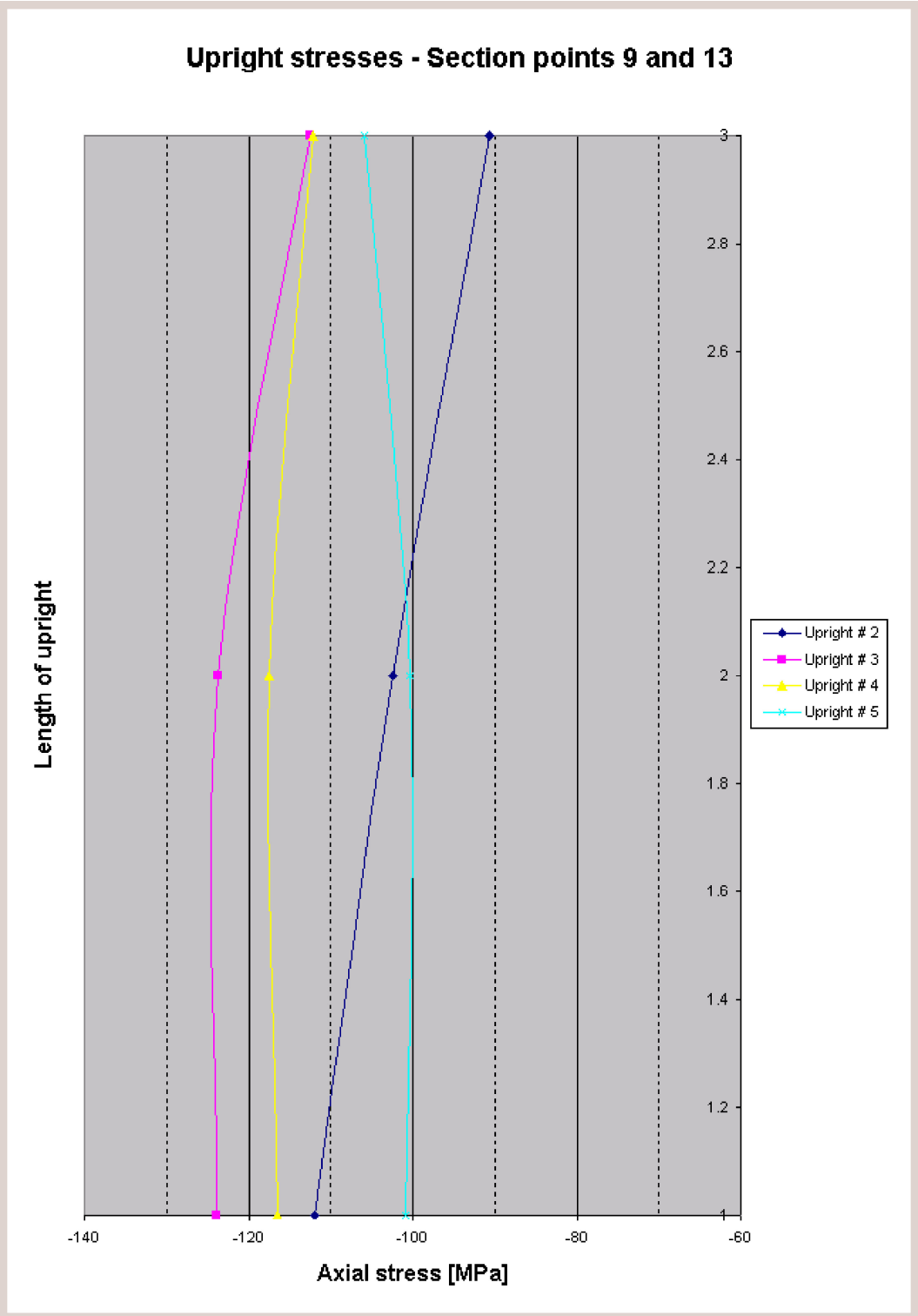


Figure 5.12: Stress along the length of the upright at section points 9 and 13

University of Pretoria etd – Viljoen, A (2004)

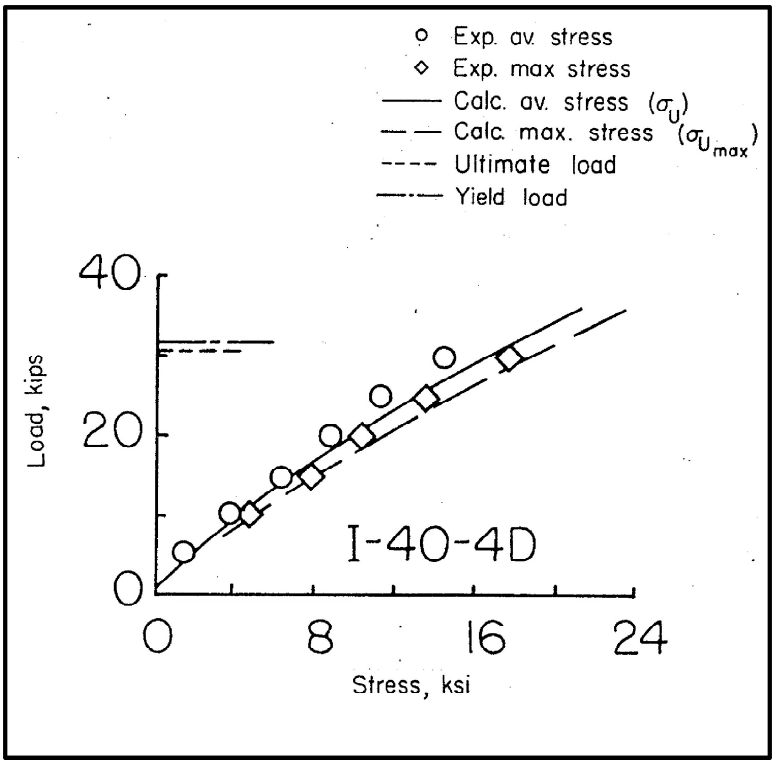


Figure 5.13: Measured upright stress data for beam I-40-4Da (Reproduced from [4])

	Grisham algorithm				NACA [3]	NACA [4]
	Upright 2	Upright 3	Upright 4	Upright 5	(theory)	(measured)
$\overline{\sigma}_u$	-94.8	-120.1	-115.2	-102.1	-116.5	-98.0
$\sigma_{u_{max}}$	-120.3	-128.9	-120.8	-105.9	-132.4	-121.0

Table 5.3: Upright stress comparison: I-40-4Da

5.1.4 Flange results

Figure 5.14 shows the bending stresses along the upper and lower flanges. The upper flange plot shows the stress in the upper fibre of the beam while the lower flange plot shows the lower fibre stress in the beam.

5.1.5 Deflection results

The experimental results for the deflection of the beam as a function of load, from Kuhn [4], are shown in Figure 5.15. From the graph the deflection at 30.3 kips is 0.85 inches which is 21.6 mm. The deflection value of the finite element analysis from the Grisham's algorithm is 28.4 mm. The displaced shape of the model at the end of the iterative analysis is shown in Figure 5.16.

University of Pretoria etd – Viljoen, A (2004)

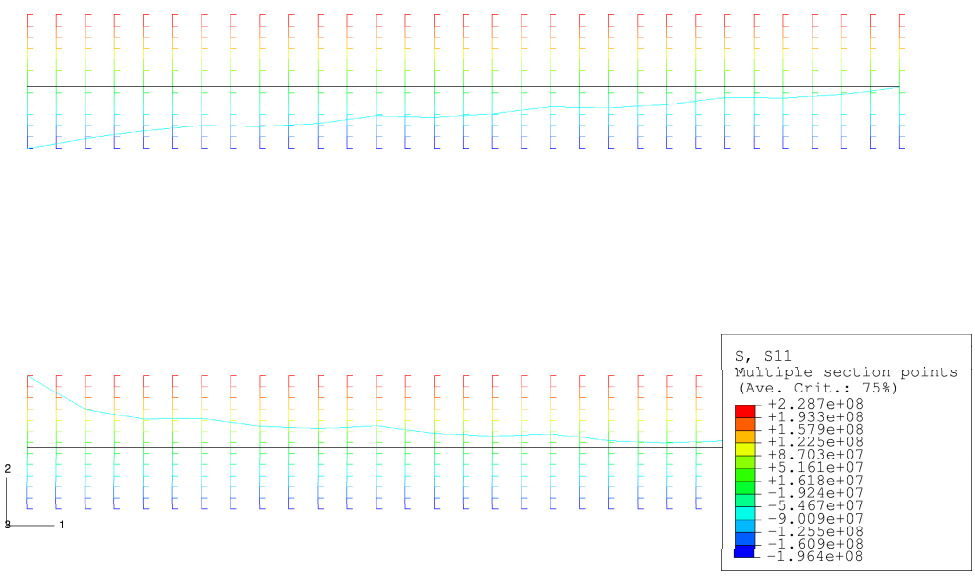


Figure 5.14: Axial stress distribution in the two flanges after the final iteration

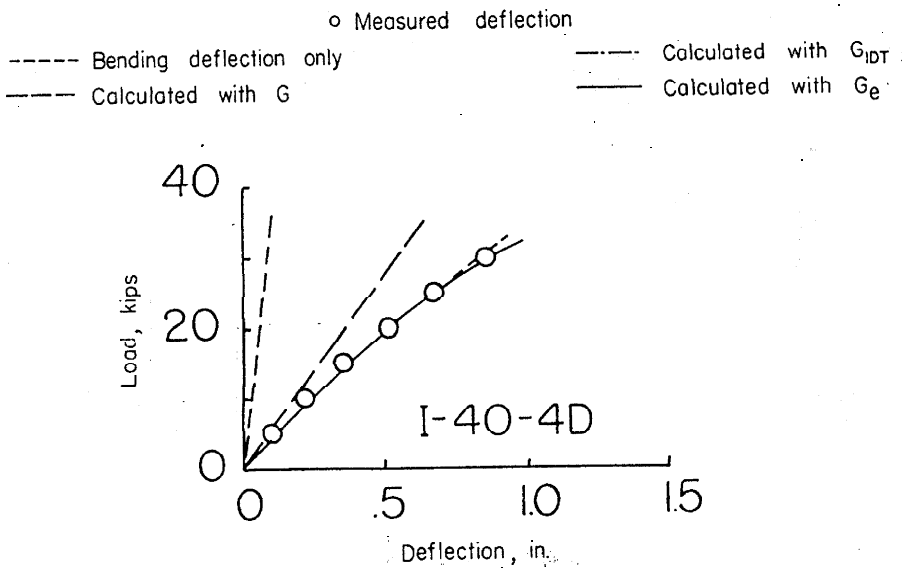


Figure 5.15: Measured deflection data for beam I-40-4Da (Reproduced from [4])

University of Pretoria etd – Viljoen, A (2004)

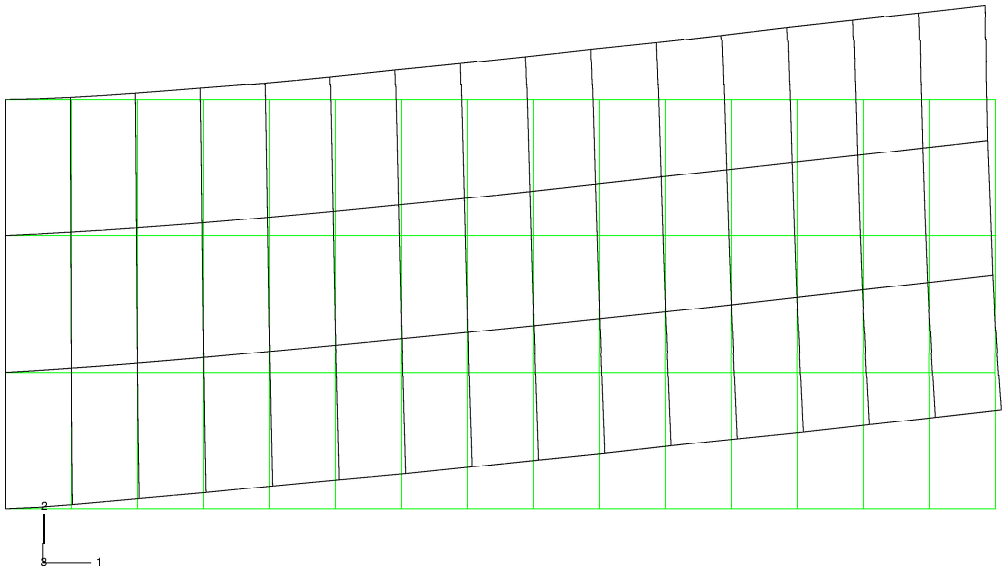


Figure 5.16: Displaced shape of the finite element model after the final iteration

5.2 Example 2 [13]

The second example, like the previous one, has experimental data that can be compared with the Grisham algorithm results (web stresses; upright stresses and shear deformation). This three panel structure, shown in Figure 5.17, has a very thin web (0.129 mm) as well as a very high aspect ratio ($\frac{L_y}{L_x}$) of 6.12, placing it outside the scope of the accepted NACA design criteria [3]. An extensive testing program (both static and fatigue) was carried out on various geometric configurations (test specimens A to M) of this structure to obtain empirical data to validate the use of the NACA design criteria for similar structures. Figure 5.18 shows a schematic layout of the test setup. Details regarding the tests can be found in [13]. This kind of deep beam structure is used in spacecraft design to ensure minimum weight. The test panels were chemically milled from 7075-T6 aluminium alloy web sheets. This allowed for very thin webs to be manufactured. The web thickness was measured, after manufacture, at various locations on the sheet and an average calculated. For test specimen C this real thickness was found to be 0.117 mm.

Test specimen C, being one of three balanced designs, having very small margins of safety for all principal modes of failure, and the one with the thinnest web, was selected for the comparative study. The specimen eventually failed in the web at a load of 44 320 N (9.96 kips). Since data at the failure load is not accurate, a load value of half the failure load was selected for the analysis.

University of Pretoria etd – Viljoen, A (2004)

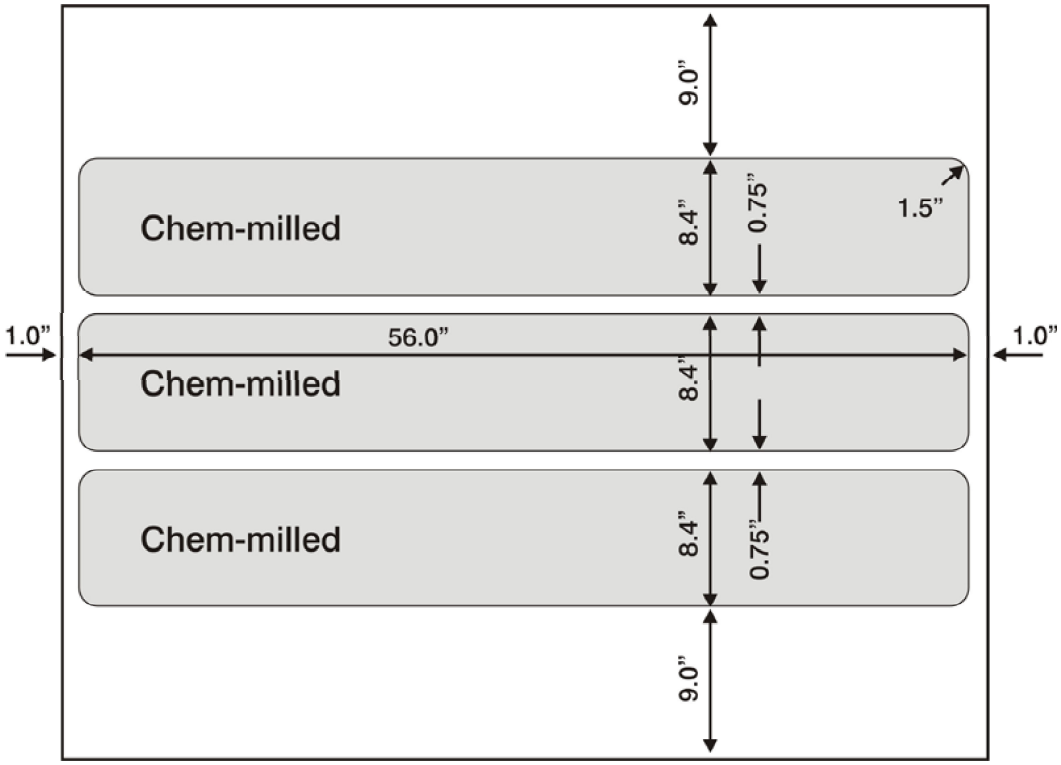


Figure 5.17: Chemically milled web sheet - test specimen C (Reproduced from [13])

5.2.1 The finite element model

The upright in this structure consists of a 'land' section that is an integral part of the web as well as a Z-shaped extruded section that is attached to the back of the 'land', resulting in an eccentric support system. The eccentricity was taken into account in the finite element model.

The measured upright stresses showed very little evidence of bending even though eccentricity is present in the structure. For this reason, second order beam elements of circular cross-section are used for the uprights in the finite element model. For simplicity, because flange stresses were not measured, the flanges are also modelled using second order beam elements of circular cross-section. Second order, thin shell elements are used for the webs. Each panel has a 3×9 web mesh density.

Figure 5.19 shows the finite element mesh for the linear model used in the Grisham algorithm. The structure is built in at the one end. At the other end, to simulate the actual testing conditions accurately, only the second degree of freedom, in which the applied load acts, is free. The applied load is evenly distributed along all the nodes.

University of Pretoria etd – Viljoen, A (2004)

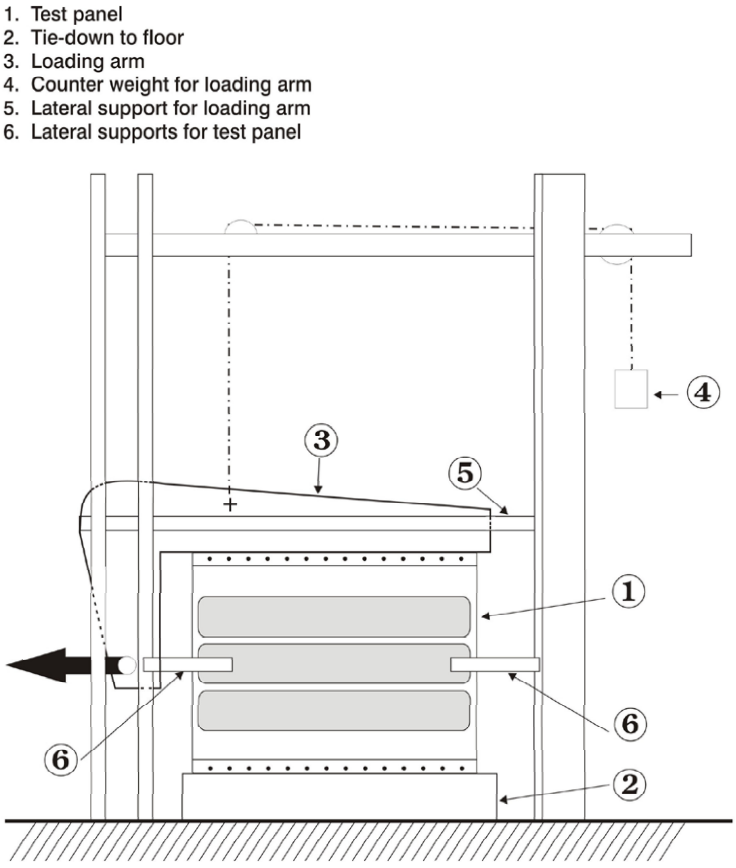


Figure 5.18: Schematic drawing of the test setup (Reproduced from [13])

University of Pretoria etd – Viljoen, A (2004)

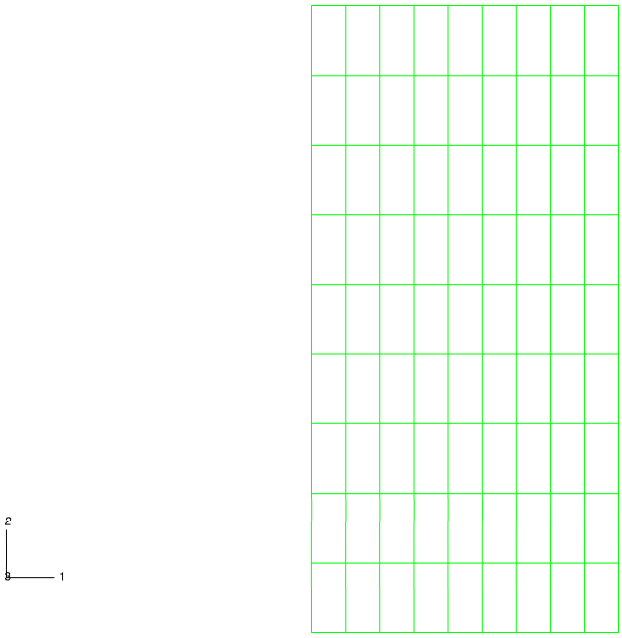


Figure 5.19: Mesh of finite element model

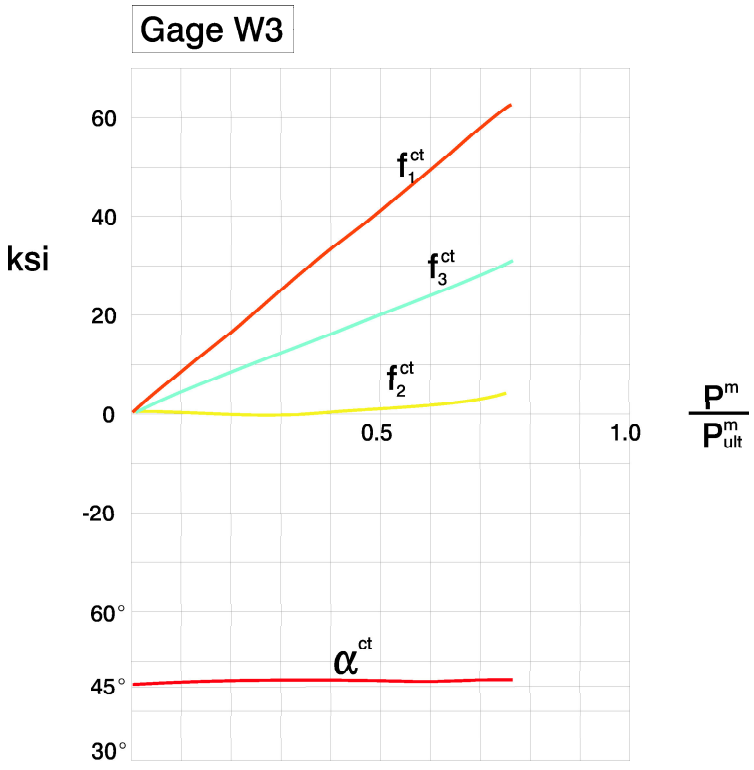


Figure 5.20: Magnitude and direction of the principal stresses in the web (Reproduced from [13])

University of Pretoria etd – Viljoen, A (2004)

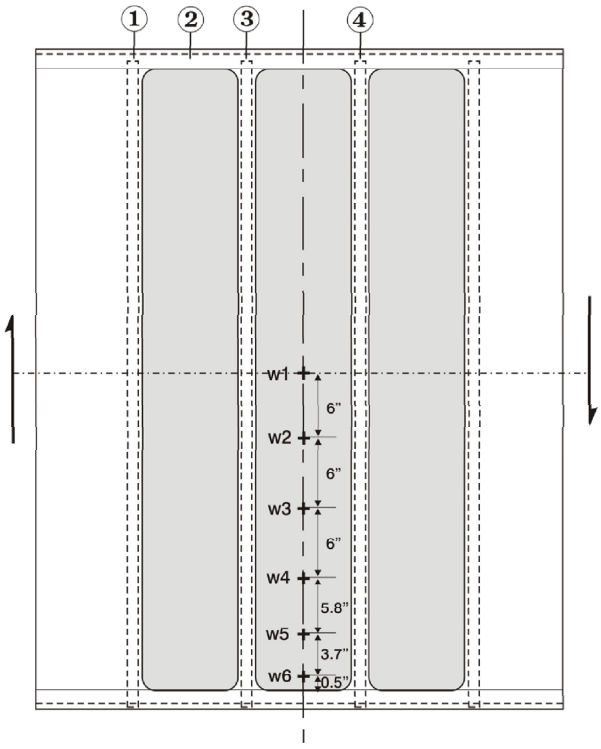


Figure 5.21: Strain and dial gauge locations on the test specimen (Reproduced from [13])

5.2.2 Web results

We now consider experimental data. Figure 5.20 shows the measured web results of the chemically milled sheet from strain gauge station 'W3'. Figure 5.21 shows the locations of the various strain gauge stations on the chemically milled web test piece. Stations 'W1' to 'W6' all had very similar results. According to Figure 5.20 the minimum principal stress (f_2^{ct} in the figure) remains close to zero for the duration of the test, while the maximum principal stress (f_1^{ct} in the figure) increases almost linearly. At half the failure load, 4.98 kips (22 160 N), which is also the load used in the finite element analysis, the maximum principal stress has a value of $\sigma_1 = 40$ ksi (275.8 MPa). The angle of major principal stress α^{ct}, as measured from the flange line and computed from gauge readings, remains just above 45 degrees for the duration of the test.

Figures 5.22 to 5.28 show the Grisham algorithm stress results in the web after the final iteration. Figure 5.25 gives a contour plot of the maximum principal stress in the web and shows this stress value in the middle bay to be in the range of $\sigma_1 = 279.0$ MPa to $\sigma_1 = 325.5$ MPa. The measured value is $\sigma_1 = 275.8$ MPa. Figure 5.26 gives a contour plot of the minimum principal stress in the web. Like the measured value this stress is compressive and only slightly less than zero in the middle bay. Figures 5.27 and 5.28 show the vector plots of the principal stresses in the web and clearly show the large difference in magnitude.

The angle of major principal stress calculated from the web results of the Grisham approach

University of Pretoria etd – Viljoen, A (2004)

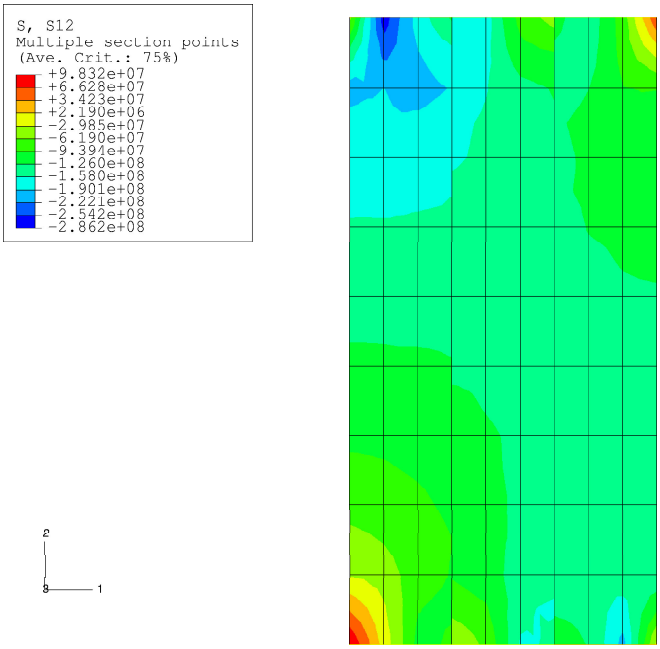


Figure 5.22: Unsmoothed shear stress (τ_{xy}) distribution in the web after the final iteration

has a value of $\alpha_{prin} = 40.2$ degrees, compared to the measured value of the chemically milled sheet $\alpha^{ct} = 45.4$ degrees.

In Table 5.4 additional output data from the Grisham algorithm is listed for the mid bay but with no measured data available for comparison. Panels 1 and 3 are ignored due to edge effects from the finite element analysis.

	Panel 2
k	0.552
α [degrees]	37.4
τ_{xy} [MPa]	-132.8
$\sigma_{x_{DT}}$ [MPa]	95.81
$\sigma_{y_{DT}}$ [MPa]	56.08
σ_{x_c} [MPa]	-3.16
σ_{y_c} [MPa]	-6.86

Table 5.4: Output data from the Grisham algorithm

The initial sheet buckling stress was impossible to determine from the tests as the extremely thin sheet would become unstable while the test piece was mounted in the machine.

University of Pretoria etd – Viljoen, A (2004)

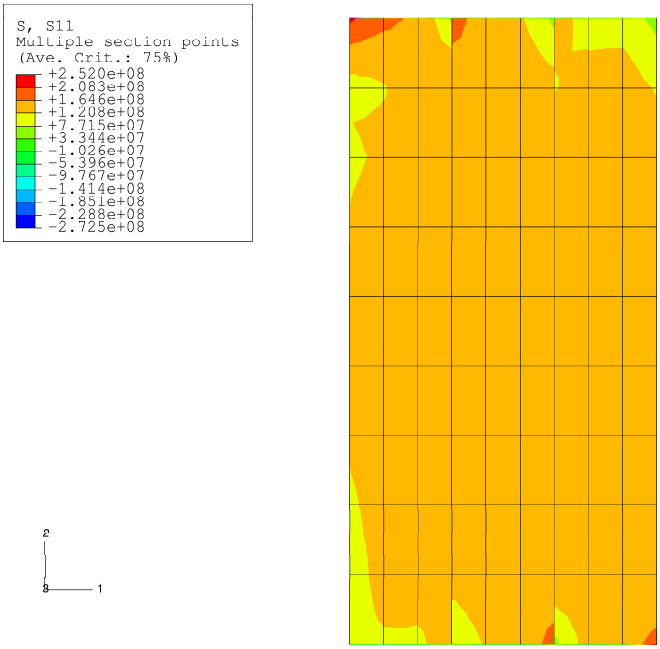


Figure 5.23: Unsmoothed normal stress (σ_x) in the web after the final iteration

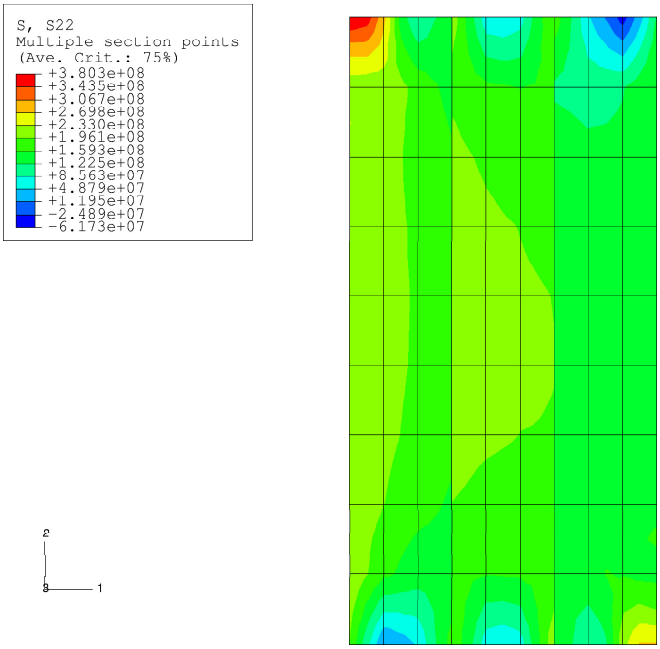


Figure 5.24: Unsmoothed normal stress (σ_y) in the web after the final iteration

University of Pretoria etd – Viljoen, A (2004)

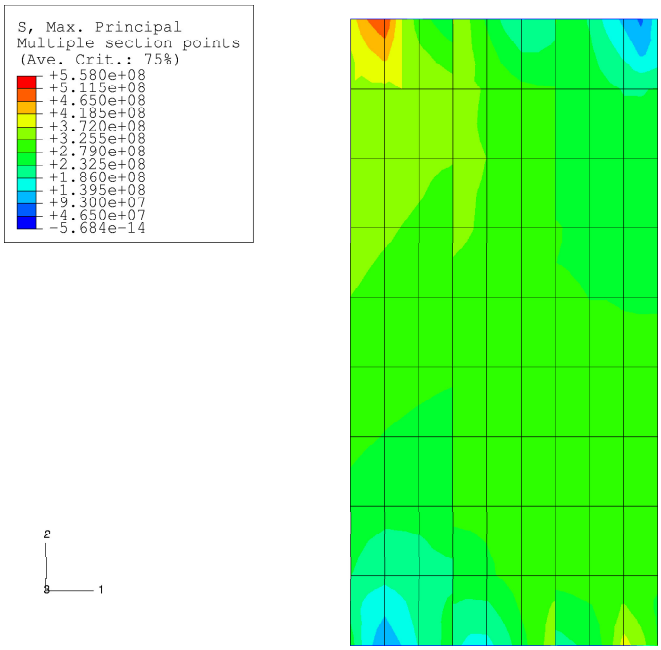


Figure 5.25: Unsmoothed maximum principal stress (σ_1) in the web after the final iteration

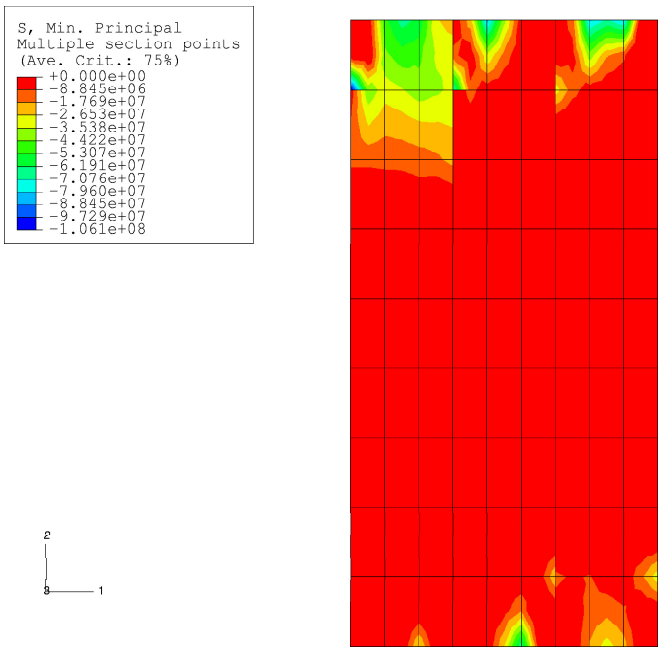


Figure 5.26: Unsmoothed minimum principal stress (σ_2) in the web after the final iteration

University of Pretoria etd – Viljoen, A (2004)

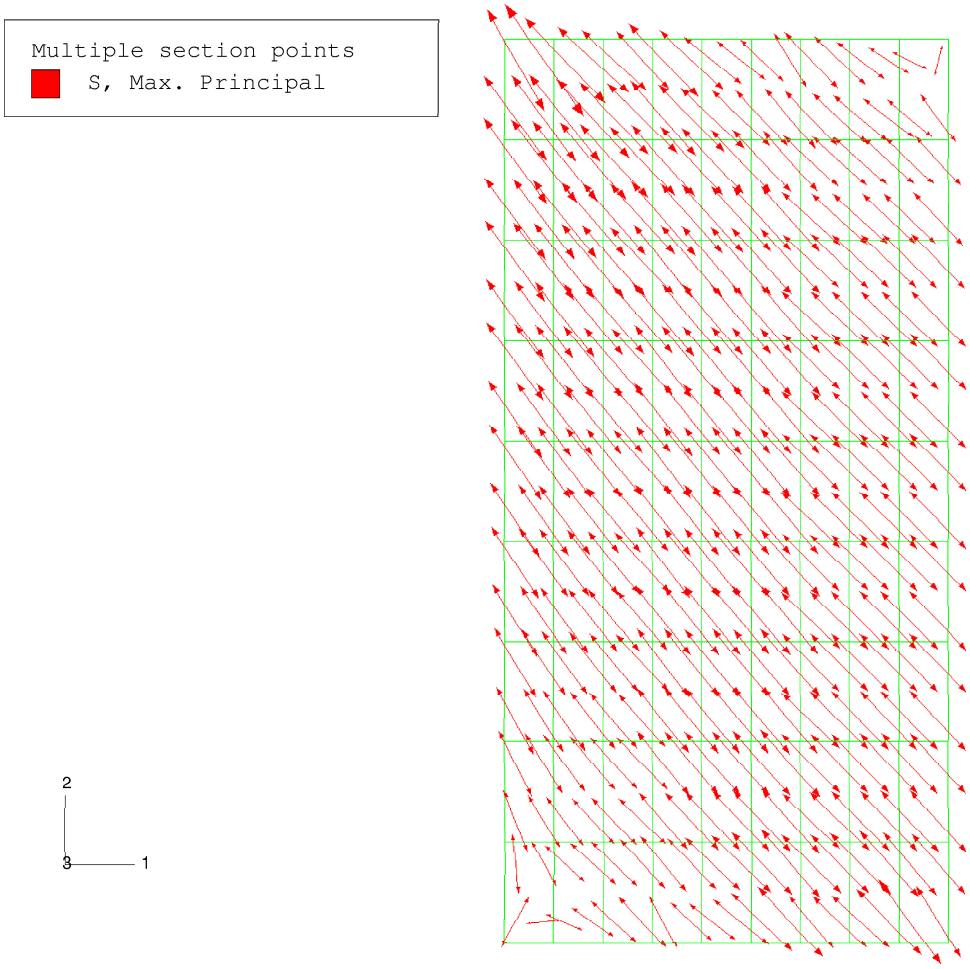


Figure 5.27: Maximum principal stress vectors in the web after the final iteration

University of Pretoria etd – Viljoen, A (2004)

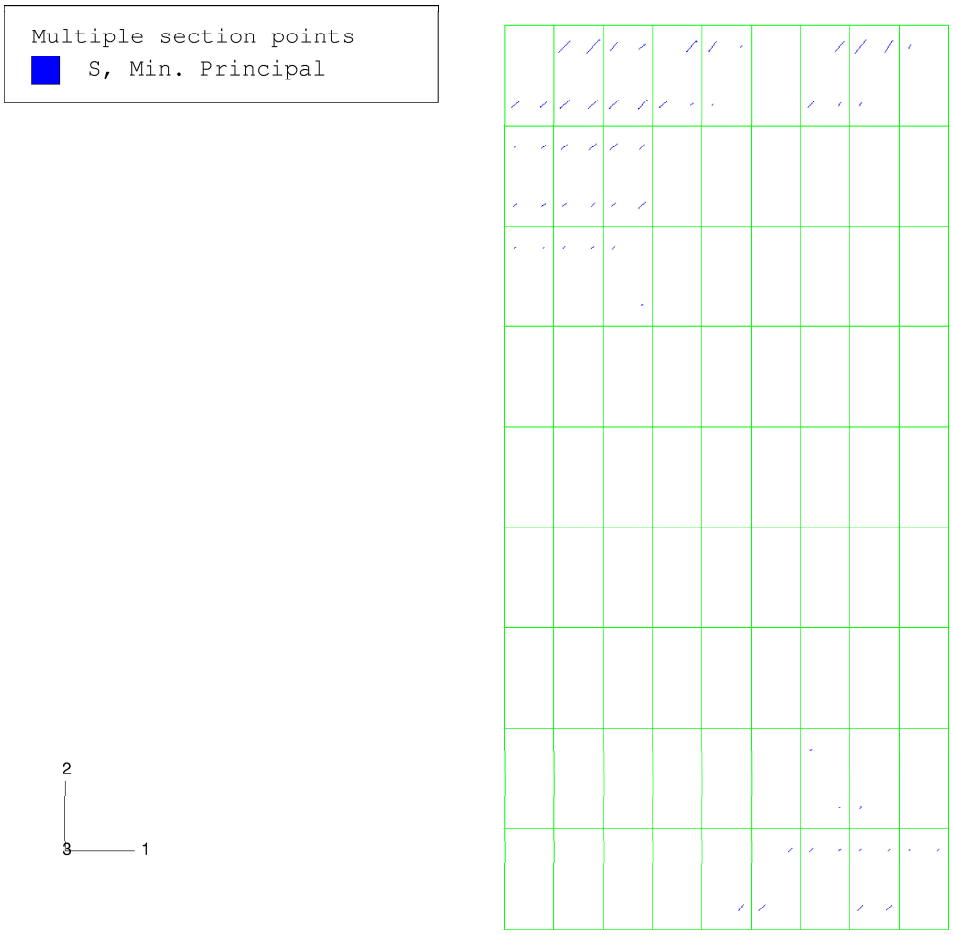


Figure 5.28: Minimum principal stress vectors in the web after the final iteration

University of Pretoria etd – Viljoen, A (2004)

5.2.3 Upright results

Axial strains in one stiffener adjacent to the middle panel were measured at three cross-sections along the length of the stiffener (See Figure 5.21). These values remained constant as the applied load increased up to about half the failure load.

Figure 5.29 shows the measured results for the longitudinal stresses at the mid-length of the stiffener. These measurements were taken on one of the uprights adjacent to the center panel. The yellow line represents the reading from strain gauge '41' which is placed on the land of the sheet. The blue line represents the reading from strain gauge '42' which is placed on the free surface of the attached stiffener leg. The orange line represents the reading from strain gauge '43' which is placed on the outside surface of the outer stiffener leg. Since the readings of these three gauges show good agreement, there are no significant bending effects, even though the upright is eccentric. Average upright stresses will therefore be compared. From the graph in Figure 5.29 the stress values at a load of 22 151 N ($\frac{P^m}{P^m_{ult}} = 0.5$ in the figure) for the three strain gauges are:

1. '41': σ_u = -9.5 ksi (-68.95 MPa)
2. '42': σ_u = -7 ksi (-48.26 MPa)
3. '43': σ_u = -7.5 ksi (-51.71 MPa)

The average value of these measured stresses is $\overline{\sigma}_u$ = -56.34 MPa. The average stress in upright number 2 of the Grisham approach is $\overline{\sigma}_u$ = -73.90 MPa. The average was calculated from all the section points at the integration points along the upright length.

5.2.4 Deflection results

Figure 5.30 shows the displaced shape of the finite element model after the final iteration. Figure 5.31 shows plots of the equivalent shear stiffness G^p_{IDT} and G^{ct}_{IDT}. G^p_{IDT} is computed using equations 31.a and 31.b in NACA TN 2661[3] and is represented by the blue dotted line. $G^{ct}_{IDT'}$ is the equivalent shear stiffness calculated from the test data. $G^{ct}_{IDT'} = \frac{P^m}{h_c \times t \times \gamma_{IDT}}$, where $\gamma_{IDT} = \frac{Gauge3 - Gauge4}{d}$. Gauges 3 and 4 are dial gauges used for measuring deflections at two locations on the test specimen. See Figure 5.21 for the locations of the dial gauges. G^{ct}_{IDT} is represented by the orange line on the graph. At $\frac{P^m}{P^m_{ult}}$ a value of $\frac{G^{ct}_{IDT}}{G} = 0.56$ is obtained. With a shear modulus of 25.98 GPa for 7075-T6 aluminium alloy, the equivalent modulus calculated from the measured data is G^{ct}_{IDT} = 14.55 GPa. Substituting the values for d and h_c into the above equations and solving for the difference between the two gauge readings, a value of Gauge 3 - Gauge 4 = 2.93 mm is obtained. In order to compare this measured difference to the finite element analysis results, displacement values at the nodes, corresponding to the locations on test specimen C were extracted and found to be: $u_{2_{gauge3}}$ = 2.81 mm and $u_{2_{gauge4}}$ = 5.14 mm which gives a difference of 2.33 mm. This is 20 % lower than the measured value of 2.93 mm.

University of Pretoria etd – Viljoen, A (2004)

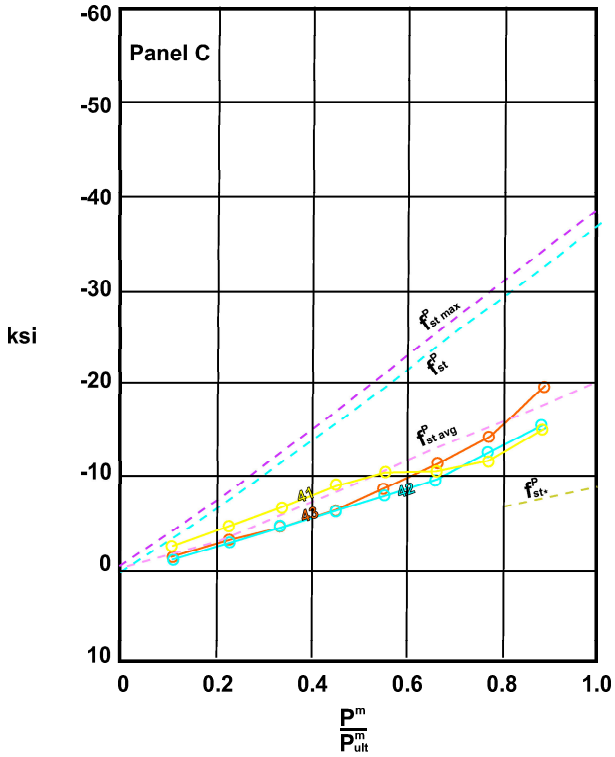


Figure 5.29: Measured longitudinal stresses at the mid-length of the stiffener (Reproduced from [13])

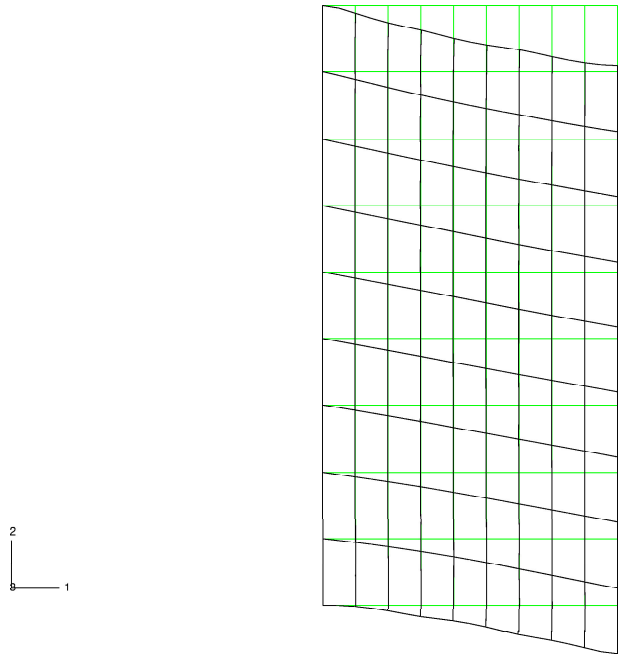


Figure 5.30: Displaced shape of the finite element model after the final iteration

University of Pretoria etd – Viljoen, A (2004)

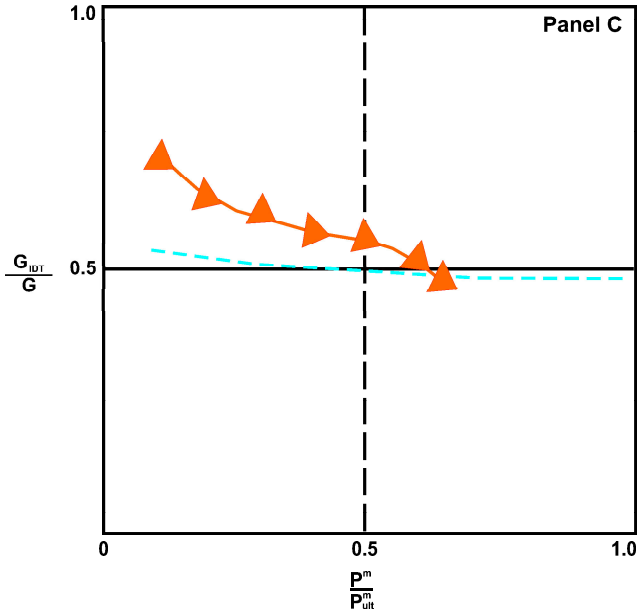


Figure 5.31: Shear stiffness variation as a function of load (Reproduced from [13])

5.3 Example 3 [14]

This example, like the two before, is a cantilever beam. Figure 5.32 gives the dimensional and geometric detail. The approach followed in [14] is to model diagonal tension by using a single model for multiple load conditions. As with the Grisham algorithm, the stiffness matrix is formulated once only. The method uses the induced strain concept in conjunction with an iterative procedure to enforce a condition of limited compressive principal stress within the thin membrane elements. Only a pure diagonal tension (PDT) condition is considered in this approach.

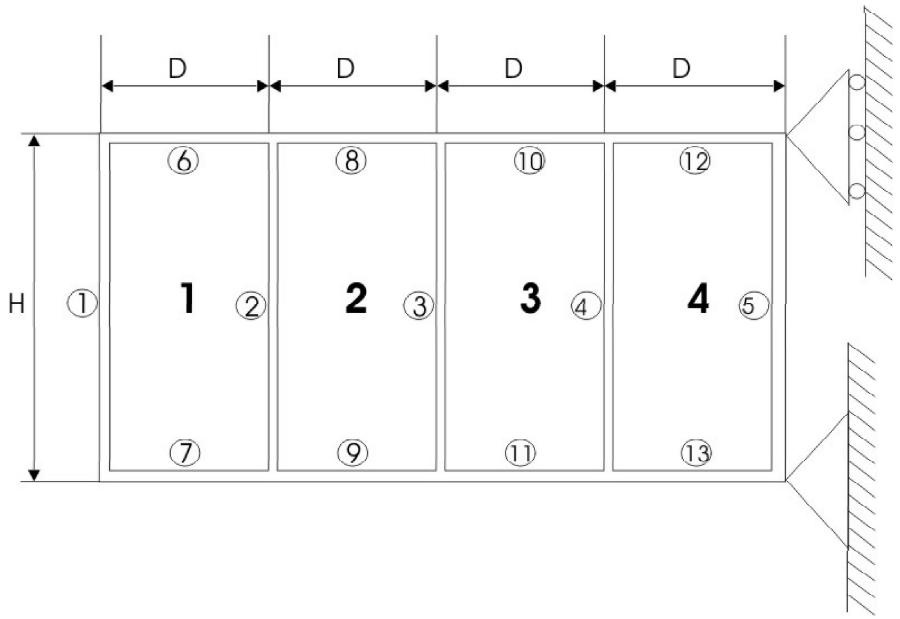
5.3.1 The finite element model

The webs are modelled using second order membrane elements while the bars are modelled using second order truss elements. A vertical load of 10 kips (44.48 kN) is applied at the free end of the beam. The mesh of the finite element model is shown in Figure 5.33. The critical buckling shear stress for this example is $\tau_{cr} = 18.66$ MPa which is much higher than that of any of the previous three examples. Figures 5.34, 5.35 and 5.36 show the stress results in the web at the end of the analysis. Figure 5.37 shows a vector plot of the maximum principal stress in the web while Figure 5.38 shows a vector plot of the minimum principal stress.

5.3.2 Web results

The only variable used in this comparative study is the diagonal tension angle. The results are shown in Table 5.5. Table 5.6 gives additional output data from the Grisham algorithm,

University of Pretoria etd – Viljoen, A (2004)



Bar areas (in)

$A_{1,6,7} = 0.3$
$A_{2\text{-}5} = 0.2$
$A_{8,9} = 0.4$
$A_{10,11} = 0.5$
$A_{12,13} = 0.6$

$T = 0.04$ in
$D = 6$ in
$H = 10$ in
$E = 10.5 \times 10^6$ PSI
$\nu = 0.3$

○ Bar

Figure 5.32: Cantilevered shear beam (Reproduced from [14])

University of Pretoria etd – Viljoen, A (2004)

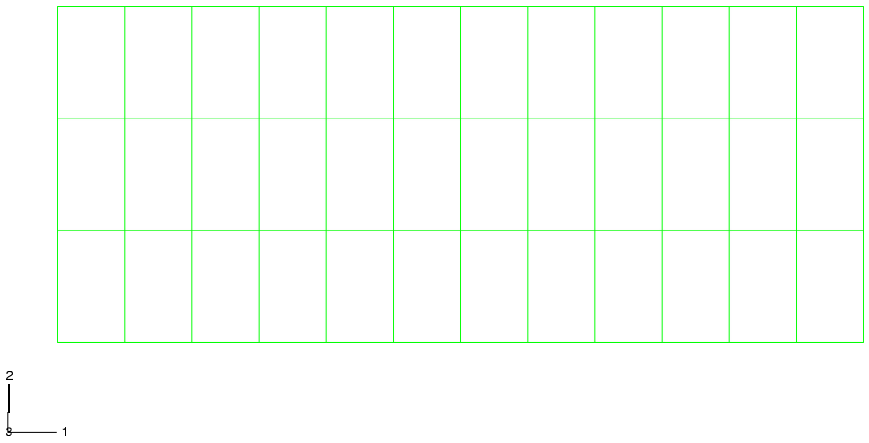


Figure 5.33: Mesh of finite element model

but with no measured data available for comparison.

	Panel 1	Panel 2	Panel 3	Panel 4
Grisham algorithm	44.37	44.00	43.77	43.70
Induced strain approach [14]	47.9	44.6	44.5	40.7
NACA approach [3]	46.9	44.3	43.6	43.0

Table 5.5: Diagonal tension angle α (degrees) comparison

5.3.3 Deflection data

The displaced shape of the beam at the end of the iterative analysis is shown in Figure 5.39.

5.4 Discussion

From the results of the example problems in this chapter as well as the results from the verification example in Chapter 2, it is evident that the Grisham algorithm yields good correlation with the worked examples and experimental data from the literature. As a tool during initial design iterations, it is certainly promising.

University of Pretoria etd – Viljoen, A (2004)

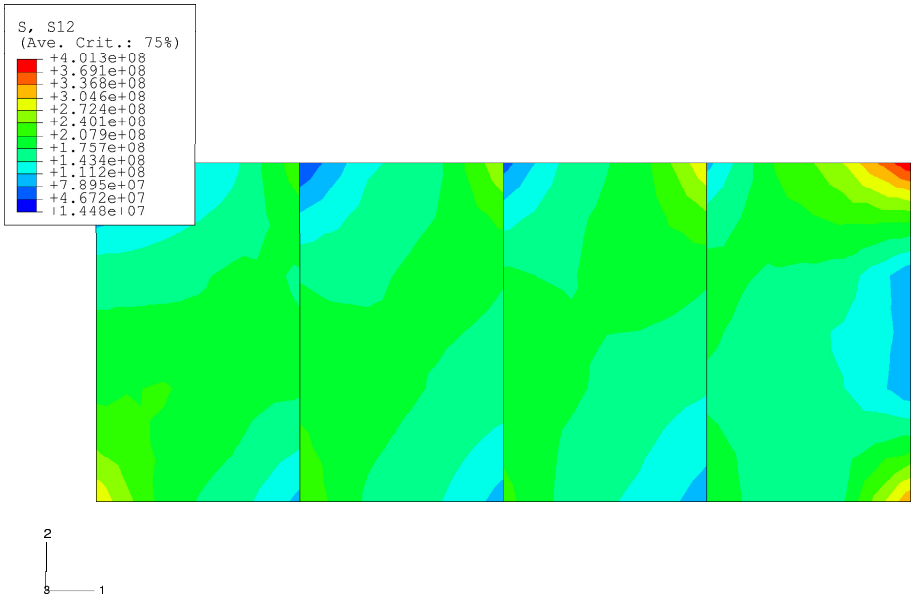


Figure 5.34: Unsmoothed shear stress (τ_{xy}) distribution in beam after the final iteration

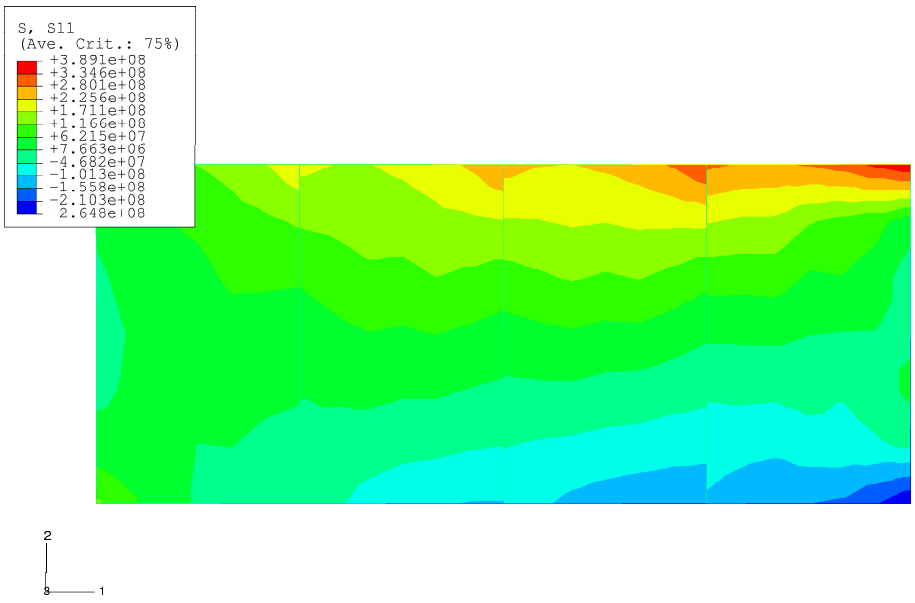


Figure 5.35: Unsmoothed normal stress (σ_x) in the beam after the final iteration

University of Pretoria etd – Viljoen, A (2004)

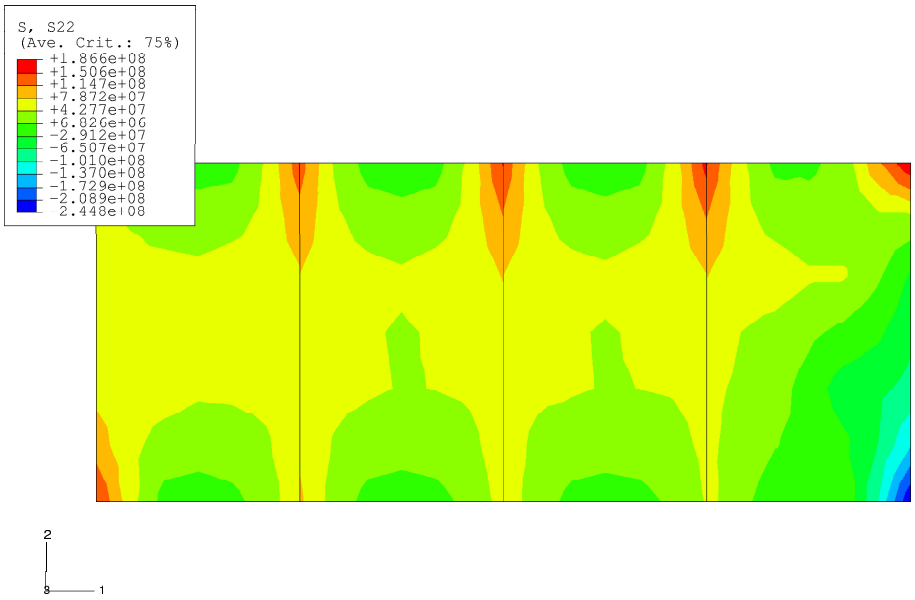


Figure 5.36: Unsmoothed normal stress (σ_y) in the beam after the final iteration

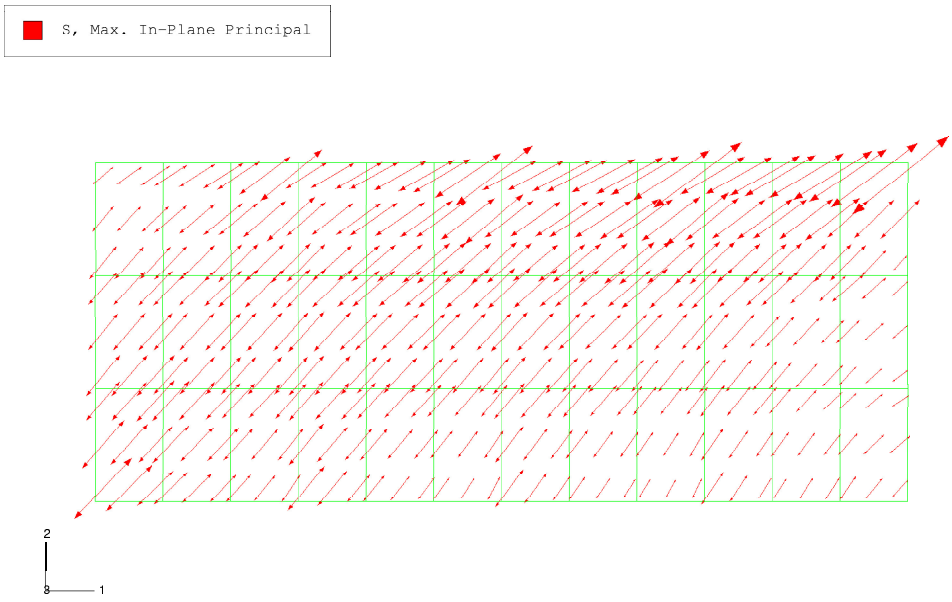


Figure 5.37: Maximum principal stress vectors in the web after the final iteration

University of Pretoria etd – Viljoen, A (2004)

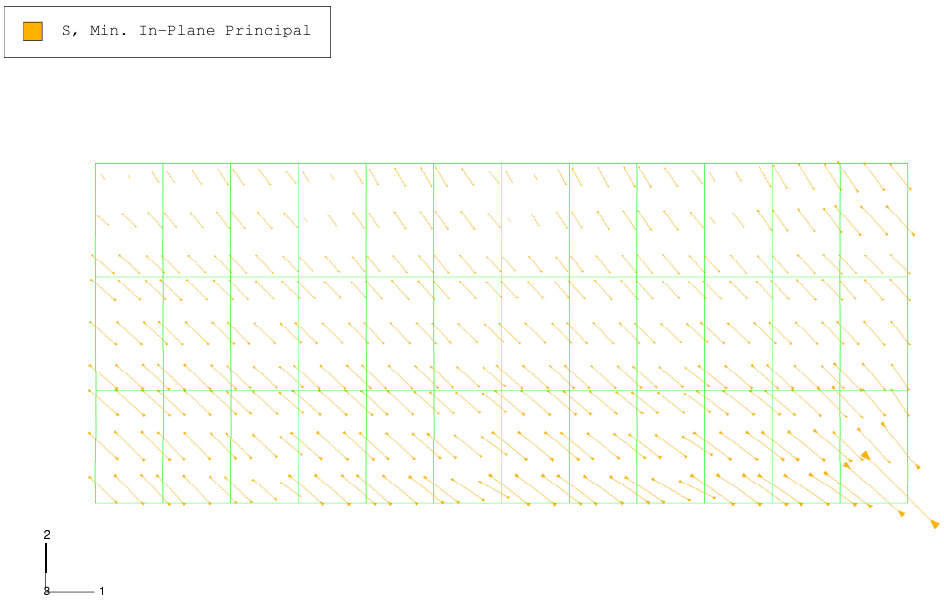


Figure 5.38: Minimum principal stress vectors in the web after the final iteration

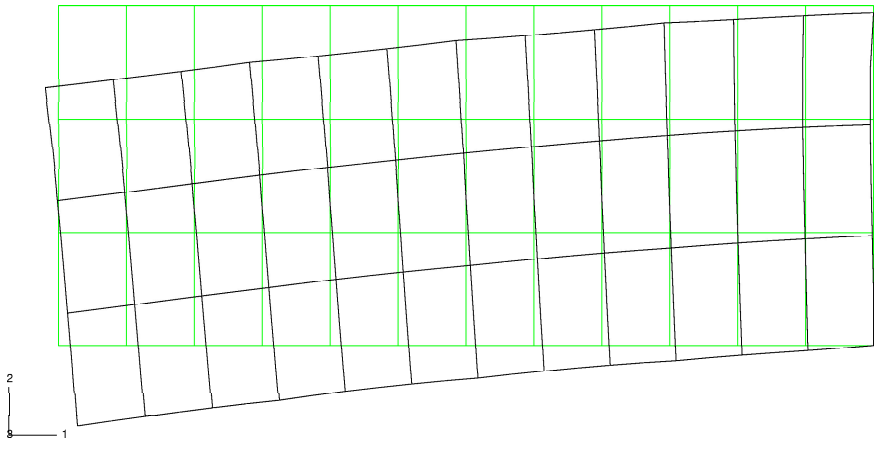


Figure 5.39: Displaced shape of the finite element model after the final iteration

University of Pretoria etd – Viljoen, A (2004)

	Grisham algorithm			
	Panel 1	Panel 2	Panel 3	Panel 4
k	0.403	0.413	0.409	0.372
τ_{xy} [MPa]	172.45	172.39	172.38	172.55
$\sigma_{x_{DT}}$ [MPa]	71.00	73.81	73.72	67.23
$\sigma_{y_{DT}}$ [MPa]	67.98	68.84	67.66	61.43
σ_{x_c} [MPa]	4.04	2.14	2.76	6.57
σ_{y_c} [MPa]	3.01	3.15	3.15	7.61

Table 5.6: Grisham algorithm web results

University of Pretoria etd – Viljoen, A (2004)

Chapter 6

Structural optimization

Grisham's algorithm is attractive in optimization, since the computational effort required per function evaluation is relatively low. The algorithm can easily be combined with a well known but simple optimization algorithm, namely the genetic algorithm (GA).

In this chapter, the example from Bruhn [8] considered in Chapter 2 is therefore optimized with respect to minimum mass, using a μ-GA with a binary representation (e.g. see Carroll [15]). While a genetic algorithm is not necessarily the best algorithm for the problem under consideration, it is selected here, as to easily allow for discrete design variables in future.

The development is as follows: Firstly, the optimal design problem is formulated. Next, the genetic algorithm is briefly outlined, whereafter the differences between a GA and a micro-genetic algorithm (μ-GA) are described. (A μ-GA is based on, and is derived from, the GA.) In the following, the presentations by Groenwold [16] and Bolton [17] are closely followed. The chapter concludes with optimal design results for the example problem.

6.1 Objective function and constraints

The optimal design problem we consider in this chapter is formulated as follows: Find the minimum f^* such that

$$f^* = f(\boldsymbol{x}^*) = \min f(\boldsymbol{x}) \tag{6.1}$$

subject to the general inequality constraints

$$g_j(\boldsymbol{x}^*) \leq 0, \ j = 1, 2, \ldots, m \tag{6.2}$$

where $\boldsymbol{x}$ is a column vector in $\mathbb{R}^n$ and f and g_j are scalar functions of the design variables $\boldsymbol{x}$. $\boldsymbol{x}$ is subject to the subsidiary conditions $x_i^l \leq x_i \leq x_i^u$, with x_i^l and x_i^u respectively representing prescribed lower and upper bounds on x_i.

(6.1) is denoted the objective function, and frequently represents mass, exposed area, cost, etc. Constraints (6.2) may represent a limit on displacement, stress, strain, magnetic flux, etc. In finding a low value for (6.1), (6.2) may never be violated. For example: when the mass of an aircraft structure is minimized, the stresses in the members may never rise above

University of Pretoria etd – Viljoen, A (2004)

a prescribed value. Else a very light structure is found, which however fails immediately upon use.

6.2 The genetic algorithm (GA)

Genetic algorithms [18] are stochastic implicit enumeration methods based on Darwinism and in particular, on the natural theory of survival of the fittest. In brief, genetic algorithms attempt to improve the fitness of designs (expressed in terms of a scalar objective function) in consecutive generations. The initial generation is populated in a random fashion with chromosomes representing possible discrete designs. The genetic operators of selection, cross-over and mutation are then used in a controlled random manner to ensure that fit parents have a high probability of passing fit genetic material to their off-spring.

Even though GA's are infamous for high computational costs (a large number of function evaluations), GA's are attractive to engineers, since their construction is quite simple. Numerous applications of GA's in engineering optimization have been presented. In addition, GA's are also suitable for implementation on massively parallel processing machines and lend themselves to be tailored according to the behavior of the objective function under consideration [16].

The most important aspects of a GA are briefly described as follows [17]:

6.2.1 Representation of design variables

Consider an initial design population, constituting of e design vectors (or 'strings' in GA jargon) $\boldsymbol{x}$, created by a random selection of the variables in the variable space for each design.

The values of the variables in the strings must be represented by a unique coding scheme. We opt for binary coding, which is powerful and frequently used.

For example: the binary string (01101) of length $l = 5$ represents the real number 22:

$$0 \cdot 2^0 + 1 \cdot 2^1 + 1 \cdot 2^2 + 0 \cdot 2^3 + 1 \cdot 2^4 = 22$$

A real value x within bounds (x_b, x_e) is represented by binary coding in the following way:

$$x = x_{bin} \cdot \frac{(x_e - x_b)}{(2^l)} + x_b \tag{6.3}$$

where

$$x_{bin} = \sum_{i=1}^{l} z_i \cdot 2^{(i-1)} \tag{6.4}$$

and z_i can be either 1 or 0 with l the binary string length.

If the objective function has several variables, then the design vector can be represented by a concatenation of the coding of each variable [19]. For example, the three dimensional design vector

$$\boldsymbol{x} = [22\ 8\ 11]$$

University of Pretoria etd – Viljoen, A (2004)

with corresponding binary code (01101); (00010); (11010) is represented as

$$\boldsymbol{X} = [011010001011010]$$

(The uppercase symbol indicates that a concatenated string is considered.)

6.2.2 Selection

The selection operation selects e strings from the current population to form the mating pool. The strings corresponding to fit objective function values have the greatest chance to be selected for mating and hence to contribute to future generations. While a large number of different selection processes are possible, only the well known expected value selection, tournament selection and ranking selection are briefly discussed in the following.

Expected value selection (roulette wheel selection)

In the expected value selection [20], also known as the roulette wheel selection, the minimization problem is converted to a maximization problem by multiplying the objective function with -1. Also, the function values must be positive and therefore a constant must be added to functions with negative values. The relative fitness p_i for each design is calculated as follows:

$$p_i = \frac{f_i}{\sum_{i=1}^{e} f_i} \qquad i = 1, 2, 3, ...e \tag{6.5}$$

where f_i denotes the function value of design i. The cumulative probability space d_j is defined as:

$$d_j = \sum_{i=1}^{j} p_i \qquad j = 1, 2, 3, ...e \tag{6.6}$$

String i is selected for the mating pool if a random number v between 0 and 1 is generated and satisfies the condition: $d_{i-1} < v \leq d_i$ with $d_0 = 0$.

Tournament selection

Tournament selection simulates the process where individuals compete for mating rights in the population [15]. In the GA, e tournaments are held between a sub group of strings chosen randomly from the existing population. The design from each tournament with the lowest function value is selected for the mating pool.

Ranking methods

After ranking the strings in ascending order according to the objective function values, the relative fitness p_i of member i is expressed as

$$p_i = \frac{t_i}{\sum_{i=1}^{e} t_i} \tag{6.7}$$

University of Pretoria etd – Viljoen, A (2004)

where

$$t_i = 2 \cdot \frac{(e+1-i)^c}{(e^2+e)} \tag{6.8}$$

c is taken as any value between 1 and 10 (typically 1) and e is the population size. The cumulative probability space d_j is constructed using the above defined relative fitness p_i and the strings are selected as in the expected value selection.

6.2.3 Crossover

After selecting e strings for the mating pool, new designs are explored by the crossover process. Crossover allows selected individuals to trade characteristics of their designs by exchanging parts of their strings. The mating pool strings are randomly grouped into pairs and a breaking point in the strings for each pair is chosen randomly. The values at the string positions after the breaking point are interchanged between the pair and the new designs are copied to the new generation. Crossover for each pair is applied with a given probability p_c, usually between 0.6 and 1. For example, when crossover is applied at the third crossover position of the following strings

$$\boldsymbol{X}_1 = [011|0100]$$

$$\boldsymbol{X}_2 = [010|1101]$$

the strings exchange the last four bits and become:

$$\boldsymbol{X}_1 = [011|1101]$$

$$\boldsymbol{X}_2 = [010|0100]$$

If crossover for a pair is not applied, then the unchanged parents are copied into the next generation. Sometimes, only one child is produced per parents. Frequently, more than one braking or cross-over point is used during crossover.

6.2.4 Mutation

The mutation operation protects against complete loss of genetic diversity by randomly changing bit values in a string. For each bit in the population a random number is generated and the bit value is changed if the random number is less than the prescribed probability of mutation p_m. For example, if mutation occurs at position four of the following string

$$\boldsymbol{X}_1 = [011\underline{0}100]$$

the string becomes

$$\boldsymbol{X}_1 = [011\underline{1}100]$$

For an integer alphabet, the mutated value can randomly be selected from the possible values from the alphabet (jump mutation), or given the value of a neighbor (creep mutation). In a binary representation, prescribing the position has a similar effect (see the example in Section 6.2.1).

Mutation typically occurs at low probability, else convergence can be impaired.

University of Pretoria etd – Viljoen, A (2004)

6.2.5 Other operators

Numerous other operators have previously been proposed. It is outside the scope of this study to name them all. However, one further operator frequently encountered is 'elitism'. It is described as follows: At any stage of the GA search, the best string found to date may (permanently) disappear from the mating pool, due to the random nature of selection. If one opts for an implementation that returns the best string found at some stage to the mating pool, (after say every a generations), this process is known as elitism. Frequently, a value of unity is used for a.

6.3 μ-GA

A μ-GA is an implementation of the GA in which rebirth replaces mutation. The basic idea is to use very small populations (say 5 individuals). It is then hoped that the small population converges very quickly, whereafter 1 or at most 2 individuals are retained using tournament selection. The remaining individuals are then killed off, and randomly repopulated, a process called rebirth.

Rebirth was first proposed by Galante [21]. The implementation of Carroll [15] is similar to that of Galante, except that Galante used larger generations than the norm in the μ-GA. In addition, Galante retained mutation, which probably is superfluous.

While the 'free lunch' theorem [22, 23] effectively prohibits a general, exhaustive comparison between the GA and the μ-GA, the latter has a secondary advantage: A reasonable solution may sometimes be obtained relatively quickly, due to the small generations typically employed. This may be desirable when evaluation of the objective function is computationally very expensive, (e.g. when using non-linear finite element analyses), and when a visual inspection of the 'current best' solution can be used to defend additional computational effort.

An inherent drawback of a μ-GA is that premature convergence is likely. Terminating a population is obvious: it can for example be done when a prescribed fraction (say 0.8) of the population has converged to a given value. However, knowing when enough restarts have been made, is not simple, and will probably be influenced by factors like computational effort and the quality of the solution found to date.

6.4 Optimization of verification example

The minimum-mass design of the stiffened shear webs is taken as the objective function, while the design variables represent the dimensions of the 6 panels. The single constraint g considered in the optimization phase is the maximum allowable prescribed post-buckled stress, expressed as

$$g = \tau_{xy}^{max} - \tau_{xy}^{pres} \tag{6.9}$$

University of Pretoria etd – Viljoen, A (2004)

where τ_{xy}^{max} represents the maximum shear stress in the webs. τ_{xy}^{pres} represents the prescribed, maximum allowable post-buckled stress. The objective function is then formulated as

$$f = \sum_{k=1}^{l} m_k + \mu\rho(g)^2 \quad (\rho > 0 \text{ and prescribed}) \tag{6.10}$$

where l represents the number of structural members, m the mass of member k and

$$\mu = \begin{cases} 0 & \text{if } g \leq 0 \\ 1 & \text{otherwise} \end{cases} \tag{6.11}$$

For the current example the constraint is simply taken as the yield stress of the material: $\sigma_{yt} = 483$ MPa for both the web material, the flange and upright material.

Further to the single considered constraint on stress, a large number of bound constraints are included to ensure validity of the solution. These constraints are all expressed in terms of the relationships given in Grisham's method, and are included in Tables 6.1 and 6.2 which tabulate the numerical results.

The example is optimized using two different sets of design variables. Firstly, four geometric variables are selected, whereafter eleven geometric variables are used. With reference to Figure 2.1, the design variables for Case 1 are:

- the upper flange area,
- the lower flange area,
- the upright area (all equal), and
- the web thickness (all equal).

For Case 2, the variables are:

- the upper flange area,
- the lower flange area,
- the area of uprights number 2 through 6 (all different), and
- the web thickness of panels number 2 through 5 (all different).

In both cases, panels one and six are not considered because of the effects of the boundary conditions.

6.4.1 μ-GA parameters

In the implementation of the μ-GA, we use a small population size of 5, and a uniform cross-over with an 80% probability. One child per pair of parents is produced, and parent selection is based on tournament selection. Elitism is included.

Eccentricity of the uprights is taken into account. The cross-sections in the example are used; T-sections for the flanges and angle sections for the uprights. During the optimization phase, the finite element mesh is never updated or changed in any way from the initial model.

6.4.2 Optimal results

For Case 1, using the μ-GA, the optimum mass is obtained as 5.515 kg after 37 generations. This gives an 11.01% saving on mass from the original mass of 6.122 kg. This is highly

University of Pretoria etd – Viljoen, A (2004)

Variable	Description	Initial value	Final value	Bounds
x_1	Lower flange area [mm^2]	243.87	201.0	$100 \leq x_1 \leq 250$
x_2	Upper flange area [mm^2]	435.48	400.4	$200 \leq x_2 \leq 450$
x_3	Upright area [mm^2]	151.21	150.4	$50 \leq x_3 \leq 170$
x_4	Web thickness [mm]	0.635	0.5188	$0.3 \leq x_4 \leq 0.9$

Table 6.1: Case 1: Optimum results using the μ-GA

Variable	Description	Initial value	Final value	Bounds
x_1	Lower flange area [mm^2]	243.9	197.7	$100 \leq x_1 \leq 250$
x_2	Upper flange area [mm^2]	435.5	396.5	$200 \leq x_2 \leq 450$
x_3	Area of upright no. 2 [mm^2]	151.2	140.1	$50 \leq x_3 \leq 170$
x_4	Area of upright no. 3 [mm^2]	151.2	142.8	$50 \leq x_4 \leq 170$
x_5	Area of upright no. 4 [mm^2]	151.2	138.5	$50 \leq x_5 \leq 170$
x_6	Area of upright no. 5 [mm^2]	151.2	141.1	$50 \leq x_6 \leq 170$
x_7	Area of upright no. 6 [mm^2]	151.2	141.9	$50 \leq x_7 \leq 170$
x_8	Panel 2: Web thickness [mm]	0.635	0.506	$0.3 \leq x_8 \leq 0.9$
x_9	Panel 3: Web thickness [mm]	0.635	0.494	$0.3 \leq x_9 \leq 0.9$
x_{10}	Panel 4: Web thickness [mm]	0.635	0.490	$0.3 \leq x_{10} \leq 0.9$
x_{11}	Panel 5: Web thickness [mm]	0.635	0.497	$0.3 \leq x_{11} \leq 0.9$

Table 6.2: Case 2: Optimum results using the μ-GA

significant in aircraft structures. At the optimum, the stresses are $\sigma_{mises} = 325.8$ MPa in the web and $\sigma_{mises} = 392.4$ MPa in the flanges.

The stress results of the optimized design for Case 1 are plotted in Figures 6.1, 6.2 and 6.3. Figures 6.4 and 6.5 show vector plots of the maximum and minimum principal stresses in the web.

For Case 2, the optimum mass is obtained as 5.366 kg after 23 generations. This gives a 14.08% saving on mass from the original mass of 6.122 kg. This is very similar to the four variable case. At the optimum, the stresses are $\sigma_{mises} = 307.7$ MPa in the web and $\sigma_{mises} = 427.8$ MPa in the flanges.

It is noted that neither Case 1 nor Case 2 are converged, since neither the stress constraint nor the variable bounds are active in either case. Case 2 does produce a more optimal result.

University of Pretoria etd – Viljoen, A (2004)

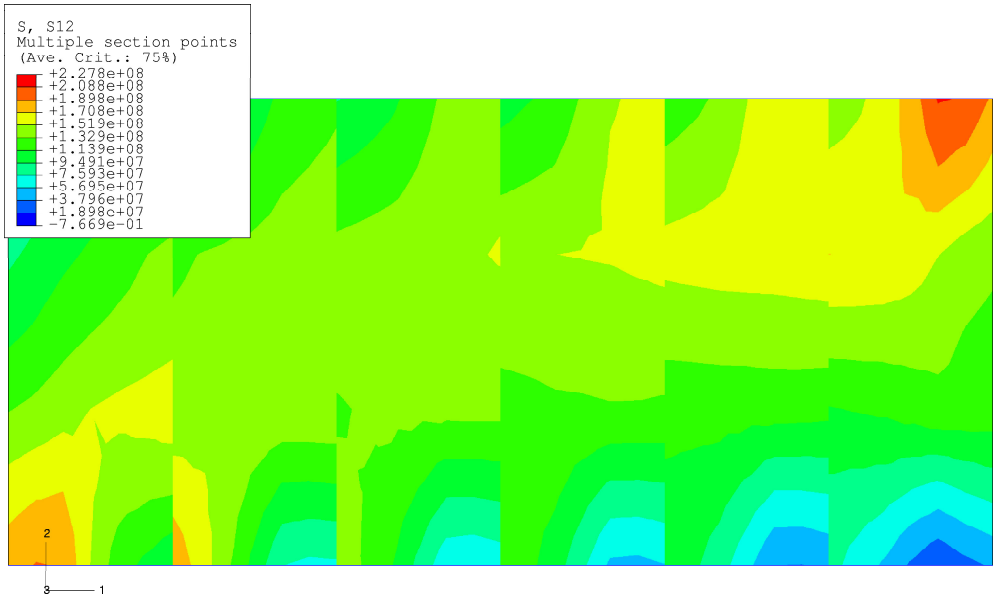


Figure 6.1: Unsmoothed shear stress distribution (τ_{xy}) in the web after the final iteration

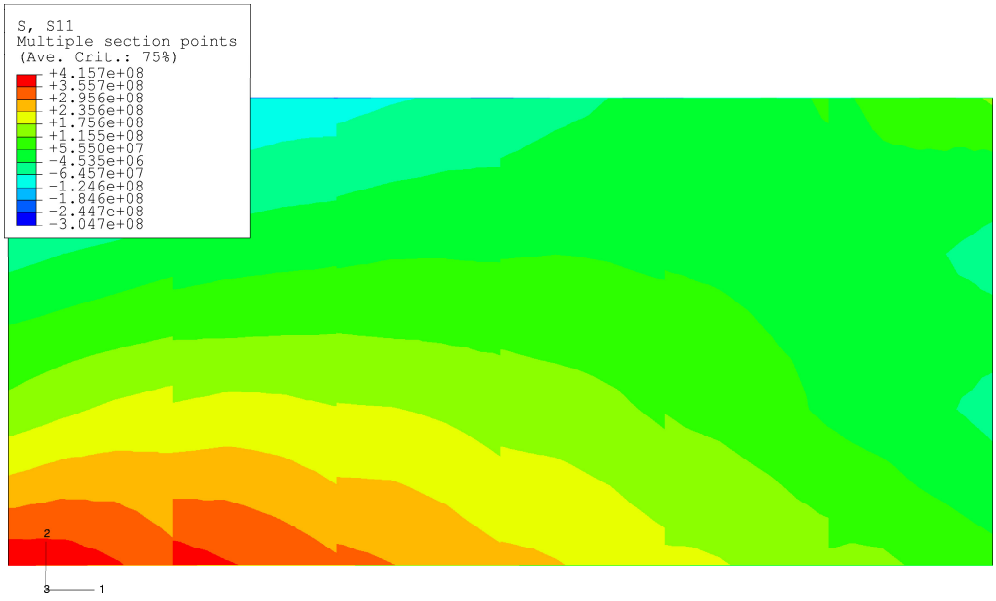


Figure 6.2: Unsmoothed normal stress (σ_x) in the web after the final iteration

University of Pretoria etd – Viljoen, A (2004)

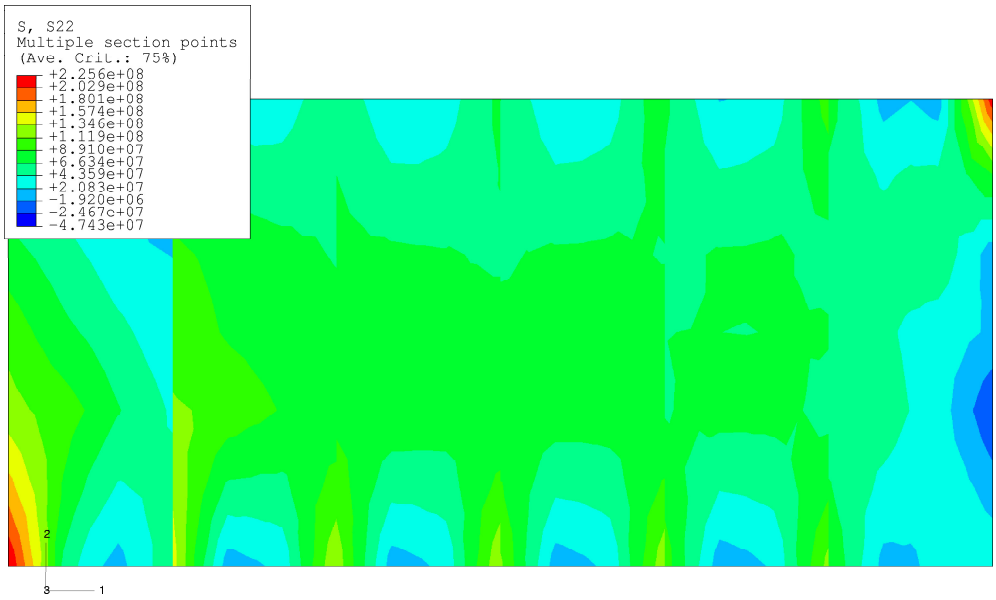


Figure 6.3: Unsmoothed normal stress (σ_y) in the web after the final iteration

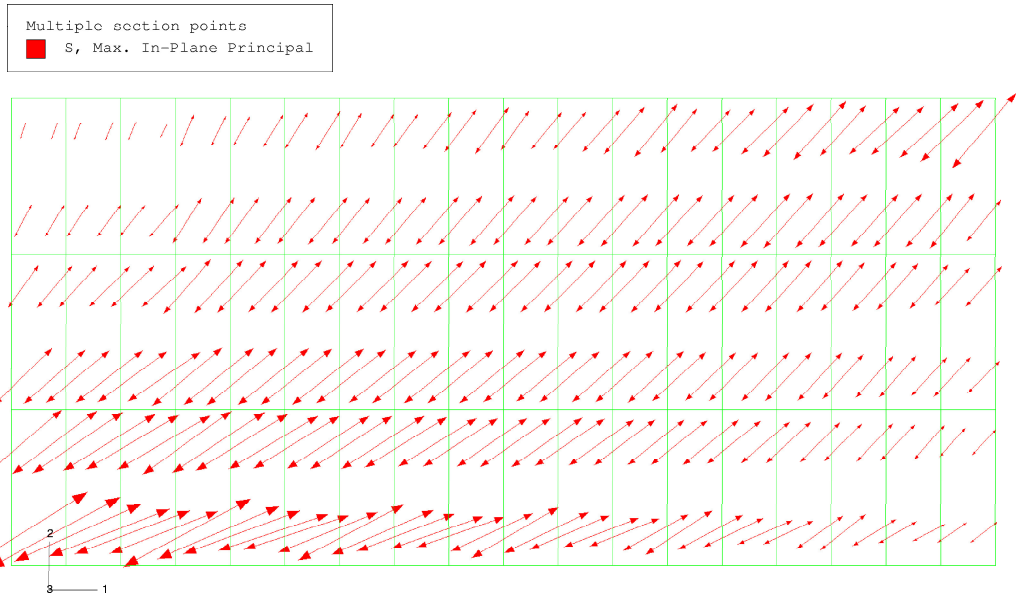


Figure 6.4: Maximum principal web stress vector plot after the final iteration

University of Pretoria etd – Viljoen, A (2004)

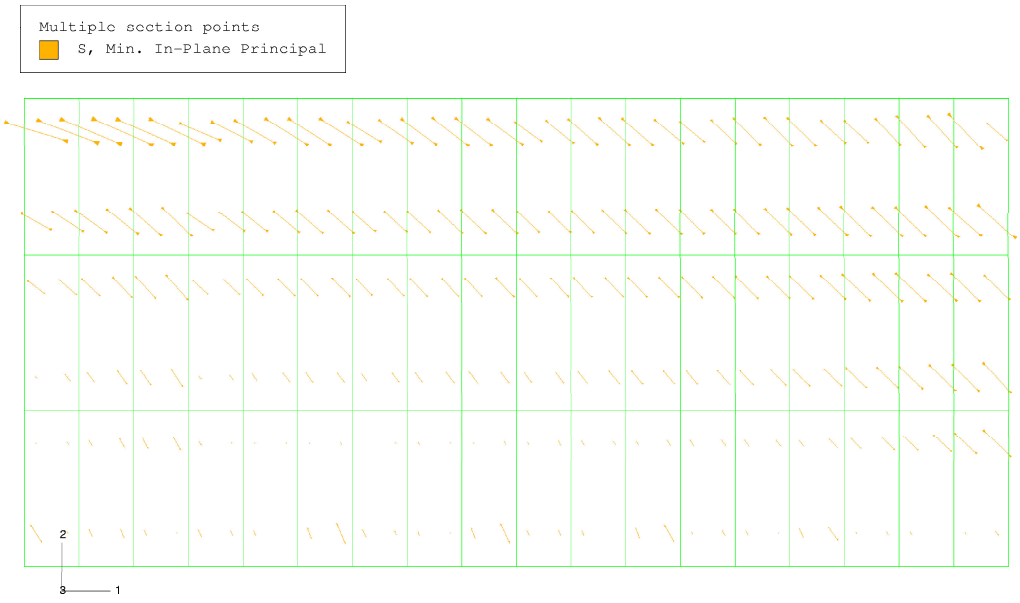


Figure 6.5: Minimum principal web stress vector plot after the final iteration

University of Pretoria etd – Viljoen, A (2004)

Chapter 7

Closure

7.1 Summary of contributions

In this thesis, the iterative post-buckling algorithm proposed by Grisham was successfully implemented and tested. The implementation was verified using various examples from the literature, for some of which experimental results were also available. For most examples, comparisons with results obtained using the modified Wagner and NACA approaches were also performed. In addition, the results obtained using the Grisham algorithm were compared with the results of a complete non-linear finite element analysis. The algorithm was then used in an optimization exercise to determine the minimum mass of the structure using an implementation of a genetic algorithm.

7.1.1 Software developed

The Grisham algorithm was programmed in FORTRAN77 and is now available for use at the CSIR[1]. All the linear finite element models were analyzed using ABAQUS®. To achieve this, portable FORTRAN77 subroutines were developed within the ABAQUS® environment to allow for the extraction of data and interaction with the Grisham algorithm.

7.2 Evaluation of the Grisham algorithm

7.2.1 Comparison with the Wagner, modified Wagner and NACA approaches

For the verification example studied in Chapter 2, the Grisham algorithm gave comparable results to those obtained using the Wagner and NACA approaches, with the Grisham algorithm being less conservative. The web results compared very well. The diagonal tension factor k was calculated to be between 2.5% and 6.0% lower (depending on the panel studied).

[1]See also Appendix B for a listing of the source code.

University of Pretoria etd – Viljoen, A (2004)

The diagonal tension angle α of 42.2 degrees is roughly halfway between the Wagner and NACA predictions, with the total shear stress in the web being 6% lower than that predicted by NACA.

The average upright stress results varied between 12% and 38% below that of the NACA predictions, depending on the position of the upright. NACA again was some 6% lower than the Wagner prediction. For the maximum stress values in the uprights, the Grisham algorithm results were between 13% and 32% lower than that calculated using NACA (with the Wagner approach not producing a maximum stress value). The flange stress results with the Grisham algorithm resulted in far lower values than both the NACA and Wagner approaches (up to as much as 138% difference). This may be accounted for by the secondary bending stresses (σ_{f_u}, σ_{f_l}) caused by diagonal tension. When ignored, the correlation improved dramatically (within 44%).

7.2.2 Comparison with a full non-linear analysis

To further validate the Grisham algorithm theoretically, a non-linear finite element analysis was performed for the verification example. While no diagonal tension factor is obtained from the non-linear finite element analysis, a qualitative inspection reveals that the principle stresses in the non-linear analysis agree well with the diagonal tension angle calculated using the Grisham algorithm.

The web critical buckling shear stress (τ_{cr}) calculated with the non-linear finite element analysis results is slightly higher and less conservative than that of the other methods (Wagner, NACA and Grisham). The agreement between the non-linear analysis and the Grisham web stress results is within 22%.

The highest average upright stress value is found in Upright 4 (the middle of the structure) and reduces in magnitude by up to 35%, to both ends. This is the case with both the non-linear analysis and the Grisham algorithm. Since the linear finite element analysis used in the Grisham algorithm does not take into account all the intricacies of the non-linear finite element analysis, the best way to compare the results in the uprights is probably to refer to averaged values. The general trend of the two sets of data compare very well, the non-linear finite element results being lower and probably less conservative than the Grisham algorithm results. The flange results compare reasonably well although the Grisham results are no longer conservative (within 16%, except for the upper flange values which are 30% lower than that of the non-linear finite element analysis results). The deflection results compare reasonably well, with the Grisham algorithm tip deflection being 11% lower than that of the non-linear finite element analysis.

7.2.3 Computational effort

The comparative study revealed that the Grisham algorithm is much more efficient than a non-linear finite element analysis. A run time comparison on an HP C200 workstation showed that for a coarse mesh discretization, the Grisham algorithm ran 14 $\times$ faster than a comparable non-linear finite element analysis. For a fine mesh discretization, the Grisham

University of Pretoria etd – Viljoen, A (2004)

algorithm ran 10 × faster. This result seems influenced by external loads on the workstation.

For the non-linear finite element analysis, a refined mesh is a requirement, as to allow for adequate representation of the buckling modes (viz. an element characteristic length should probably at least be some 3-5 times smaller than the wavelength of the buckling mode). The Grisham algorithm on the other hand is an approximate method that requires far less accuracy in the finite element model.

Hence an advantage in computational efficiency of at least 10 × for most problems seems reasonable.

7.2.4 Stopping criteria and parameters

The two convergence criteria used in the Grisham algorithm both typically converge within five iterations to within a 2% margin. This seems quite efficient and realistic.

The method has two parameters (β values) which are related to the stiffness of the buckled webs. They are not known *a priori*. When the algorithm starts off, they are approximated by initial estimates. The β values reveal little sensitivity to the flange or upright dimensions, but notable sensitivity to the thickness of the webs in the structure. Nevertheless, the β values can easily be adjusted using an iterative procedure and do not effectively impair convergence.

7.2.5 Further verification

Three further examples taken from the literature were used to validate the Grisham algorithm; the first two with experimental results. The Grisham algorithm results compared reasonably well with the experimental values, providing further proof of the validity of the procedure.

7.3 Structural optimization

The verification example was successfully optimized for mass, using a micro-genetic algorithm (μ-GA). Using only four design variables, an 11.01% saving in mass was achieved. Using eleven design variables, a 14.08% saving was achieved.

7.4 Recommendation

Based on

1. the demonstrated accuracy of the Grisham algorithm, as well as
2. the computational efficiency as compared to a full non-linear finite element analysis,

University of Pretoria etd – Viljoen, A (2004)

it is recommended that the Grisham algorithm be used in particular during initial design iterations and optimization analyses.

After initial design iterations are complete, final designs can still be evaluated using a full non-linear finite element analysis if desired. This will allow for a single expensive non-linear finite element analysis to suffice during the design or optimization process.

7.5 Future work

The Grisham algorithm can relatively easily be extended to provide for box structures, curved panels and composite materials. It is noted that the computational edge of the Grisham algorithm can be expected to increase as the complexity of the structures analyzed increases (since non-linear finite element analyses will become increasingly expensive).

In addition, the use of discrete design variables during optimization is of interest. This is already provided for in selecting a genetic algorithm for the infrastructure.

University of Pretoria etd – Viljoen, A (2004)

Bibliography

[1] H. Wagner. Ebene Blechwandtrüger mit sehr dünnen Stegbleich. *Zeitschrift für Flugtechnik und Motorluftschiffahrt*, 20, 1929.

[2] H. Wagner. Flat sheet metal girders with very thin webs. Technical Report TM 604, 605 and 606, NACA, Washington D.C., U.S.A., February 1931.

[3] P. Kuhn, J.P. Peterson, and L.R. Levin. A summary of diagonal tension. Part 1: Methods of analysis. Technical Report TN 2661, NACA, Washington D.C., U.S.A., May 1952.

[4] P. Kuhn, J.P. Peterson, and L.R. Levin. A summary of diagonal tension. Part 2: Experimental evidence. Technical Report TN 2662, NACA, Washington D.C., U.S.A., May 1952.

[5] R.M. Mello, R.E. Sherrer, and M.D. Musgrove. Intermediate diagonal tension field shear beam development for the Boeing SST. *J. Aircraft*, 9:470–476, 1972.

[6] A.F. Grisham. A method for including post-buckling of plate elements in the internal loads analysis of any complex structure idealized using finite element analysis methods. In *Proc. 19th Structures, Structural Dynamics and Materials Conference*, Bethesda, MD., U.S.A., April 1978. Paper no. 78-515.

[7] J.M. Fehrenbach. Elements pour le calcul des structures d'avions. *Ecole Nationale Superieure de l'Aeronautique et de l'Espace*, 10 Avenue Edouard Belin, Toulouse, 1974.

[8] E.F. Bruhn. *Analysis and Design of Flight Vehicle Structures.* Tri-State Offset Co., U.S.A., 1973.

[9] Paul Kuhn. *Stresses in Aircraft and Shell Structures.* McGraw-Hill Book Co., U.S.A., 1956.

[10] D.J. Peery and J.J. Azar. *Aircraft Structures.* McGraw-Hill Book Co., U.S.A., 1982.

[11] E. Riks. The application of Newton's method to the problem of elastic stability. *J. Applied Mech.*, 39:1060–1066, 1972.

[12] G.A Wempner. Discrete approximations related to nonlinear theories in solids. *Int. J. Solids Struct.*, 7:1581–1599, 1971.

University of Pretoria etd – Viljoen, A (2004)

[13] A.G. Tsongas and R.T. Ratay. Investigation of diagonal tension beams with very thin stiffened webs. Technical Report N69-32379, Grumman Aerospace Corporation, Bethpage, New York, U.S.A., July 1969.

[14] P.W. Mason, T. Balderes, and H. Armen. The application of nonlinear analysis techniques to practical structural design problems. *Computers and Structures*, 16:549–562, 1983.

[15] D.L. Carroll. Chemical laser modeling with genetic algorithms. *AIAA Journal*, 34, No. 2:338–346, 1996.

[16] A.A. Groenwold, N. Stander, and J.A. Snyman. A regional genetic algorithm for the discrete optimal design of truss structures. *Int. J. Num. Meth. Eng.*, 44:749–766, 1999.

[17] H.P.J. Bolton. Parallel competing algorithms in global optimization. Master's thesis, University of Pretoria, Department of Mechanical Engineering, 2000.

[18] David E. Goldberg. *Genetic Algorithms in Search, Optimization, and Machine Learning.* Addison-Wesley Publishing Co., The University of Alabama, 1989.

[19] E. Potgieter. A genetic algorithm for the discrete structural optimization of laminated plates. Master's thesis, University of Pretoria, South Africa, Department of Mechanical Engineering, 1997.

[20] D.E. Goldberg. *Genetic Algorithms in Search, Optimization and Machine learning.* Addison-Wesley, MA, 1989.

[21] M. Galante. Genetic algorithms as an approach to optimize real-world trusses. *Int. J. Num. Meth. Eng.*, 39:361–382, 1996.

[22] D. Wolpert and W. Macready. No free lunch theorems for optimization. *IEEE Trans. Evol. Comput.*, 1:67–82, 1997.

[23] T.M. English. Evaluation of evolutionary and genetic operators: No free lunch. In L.J. Fogel, P.J. Angeline, and T. Baeck, editors, *Evolutionary Programming V: Proc. Fifth Annual Conference on Evolutionary Programming*, pages 163–169, San Diego, CA, USA, February 1995.

University of Pretoria etd – Viljoen, A (2004)

Appendix A

Summary of Grisham's equations

A summary of the equations in Grisham's paper is given below. Note that for all equations, the buckling stress is positive.

First, the modified shear buckling allowable stress is evaluated:

$$\tau_{xy_{cr}} = \tau_{xy_{cr_0}} \sqrt{1 - \frac{\sigma_x}{\sigma_{x_{cr_0}}} - \frac{\sigma_y}{\sigma_{y_{cr_0}}}} \tag{A.1}$$

If the linear finite element solution indicates a tension-tension stress state, else for tension-compression and compression-compression the following equation is used:

$$\tau_{xy_{cr}}^2 + \frac{\tau_{xy_{cr_0}}^2 (\sigma_x \sigma_{y_{cr_0}} + \sigma_y \sigma_{x_{cr_0}})}{|\tau_{xy}| \sigma_{x_{cr_0}} \sigma_{y_{cr_0}}} \tau_{xy_{cr}} - \tau_{xy_{cr_0}}^2 = 0 \tag{A.2}$$

Once this is done, the modified membrane buckling stresses are determined:

$$\frac{|\tau_{xy}|}{|\tau_{xy_{cr}}|} = \frac{\sigma_x}{\sigma_{x_{cr}}} = \frac{\sigma_y}{\sigma_{y_{cr}}} \tag{A.3}$$

Once these have been calculated an interaction equation is used to determine if the web buckles:

$$\frac{\sigma_{x_{cr}}}{\sigma_{x_{cr_0}}} + \frac{\sigma_{y_{cr}}}{\sigma_{y_{cr_0}}} + \left(\frac{\tau_{xy_{cr}}}{\tau_{xy_{cr_0}}}\right)^2 = 1 \tag{A.4}$$

If the web buckles, the diagonal tension factor, angle and stresses in the x- and y-directions can be calculated:

University of Pretoria etd – Viljoen, A (2004)

$$k = \tanh(0.5 \log \frac{\tau_{xy}}{\tau_{xy_{cr}}}) \tag{A.5}$$

The diagonal tension angle is calculated using the following equations and a series of successive approximations:

$$\tan^2 \alpha = \frac{\epsilon - \epsilon_f}{\epsilon - \epsilon_u} \tag{A.6}$$

where:

$$\epsilon_f = \frac{H_1}{\tan \alpha} \tag{A.7}$$

$$\epsilon_u = H_2 \tan \alpha \tag{A.8}$$

$$\epsilon = \frac{H_3}{\sin 2\alpha} + H_4 \sin 2\alpha \tag{A.9}$$

$$H_1 = \frac{-k|\tau_{xy}|}{E_f(\frac{A_f}{L_y t} + 0.5(1-k))} \tag{A.10}$$

$$H_2 = \frac{-k|\tau_{xy}|}{E_u(\frac{A_u}{L_x t} + 0.5(1-k))} \tag{A.11}$$

$$H_3 = \frac{2k|\tau_{xy}|}{E_w} \tag{A.12}$$

$$H_4 = (1 - k + \mu - \mu k)(\frac{|\tau_{xy}|}{E_w}) \tag{A.13}$$

A_f and A_u are half of the summed stiffener areas in the x- and y-directions respectively.

The diagonal tension stress to be developed in the plate following shear buckling is:

$$\sigma_{x_{DT}} = k|\tau_{xy}| \cot \alpha \tag{A.14}$$

$$\sigma_{y_{DT}} = k|\tau_{xy}| \tan \alpha \tag{A.15}$$

The shear strain of the plate in its post-buckled state is:

University of Pretoria etd – Viljoen, A (2004)

$$\gamma_{xy} = [\frac{1-k}{G_w} + \frac{k}{G_{pdt}}]\tau_{xy} \tag{A.16}$$

where:

$$\frac{1}{G_{pdt}} = \frac{1}{2G_w(1+\mu)}\{\frac{4}{\sin^2 2\alpha} + \frac{E_w L_x t \tan^2\alpha}{E_u(A_u + 0.5L_x t(1-k))} + \frac{E_w L_y t \cot^2\alpha}{E_f(2A_f + 0.5L_y t(1-k))}\} \tag{A.17}$$

Defining:

$$\frac{1}{G_{pdt}} = \frac{1}{G_w}\theta \tag{A.18}$$

then:

$$\gamma_{xy} = (\frac{1}{G_w} + \frac{k(\theta-1)}{G_w})\tau_{xy} \tag{A.19}$$

where the second term is the shear deformation of the plate in its post-buckled state.

Therefore, the set of pre-strains required to induce the diagonal tension stresses in the plate is:

$$\epsilon_{x_{DT}} = -\frac{1}{E_w}(\frac{E_w L_y t}{A_f E_f} + 1)(\sigma_{x_{DT}} - \mu\sigma_{y_{DT}}) \tag{A.20}$$

$$\epsilon_{y_{DT}} = -\frac{1}{E_w}(\frac{E_w L_x t}{A_u E_u} + 1)(\sigma_{y_{DT}} - \mu\sigma_{x_{DT}}) \tag{A.21}$$

$$\gamma_{xy_{DT}} = \frac{k(\theta-1)}{G_w})\tau_{xy} \tag{A.22}$$

The compressive stresses in the buckled plate are:

$$\sigma_{x_c} = \frac{C_2C_3[\sigma_x - \sigma_{x_{cr}}]L + \beta_x\mu C_4[\sigma_y - \sigma_{y_{cr}}]L}{C_1C_2} \tag{A.23}$$

$$\sigma_{y_c} = \frac{C_1C_4[\sigma_y - \sigma_{y_{cr}}]L + \beta_y\mu C_3[\sigma_x - \sigma_{x_{cr}}]L}{C_1C_2} \tag{A.24}$$

University of Pretoria etd – Viljoen, A (2004)

where:

L is a factor to control the rate at which compressive buckling is incorporated and has a value between '0' and '1'.

$$C_1 = 1 + \frac{L_y t E_w \beta_x}{A_f E_f} \tag{A.25}$$

$$C_2 = 1 + \frac{L_x t E_w \beta_y}{A_u E_u} \tag{A.26}$$

$$C_3 = 1 - \beta_x \tag{A.27}$$

$$C_4 = 1 - \beta_y \tag{A.28}$$

and

$$\beta_x = \frac{E_{w_x}}{E_w} \tag{A.29}$$

$$\beta_y = \frac{E_{w_y}}{E_w} \tag{A.30}$$

The pre-strains corresponding to these changes in stress due to compression buckling are:

$$\epsilon_{x_c} = -\frac{1}{E_w}(1 + \frac{L_y t E_w}{A_f E_f})(\sigma_x - \mu\sigma_y)|\beta_x + \beta_x\beta_y - 1| \tag{A.31}$$

$$\epsilon_{y_c} = -\frac{1}{E_w}(1 + \frac{L_x t E_w}{A_u E_u})(\sigma_y - \mu\sigma_x)|\beta_y + \beta_x\beta_y - 1| \tag{A.32}$$

$$\gamma_{xy_c} = 0.0 \tag{A.33}$$

The total change in strain for the first iteration is the sum of the diagonal tension strain and the compressive strain:

$$\triangle\epsilon_x = \epsilon_{x_{DT}} + \epsilon_{x_c} \tag{A.34}$$

$$\triangle\epsilon_y = \epsilon_{y_{DT}} + \epsilon_{y_c} \tag{A.35}$$

University of Pretoria etd – Viljoen, A (2004)

$$\triangle\gamma_{xy} = \frac{k(\theta - 1)}{G_w})\tau_{xy} \tag{A.36}$$

These strains then become the pre-strains in the finite element model, for the next iteration. For the second and succeeding iterations, all the above equations are used. The diagonal tension stress must first be removed from the finite element stress though. The total strain for the n-th iteration is:

$$\epsilon^n_{x_{total}} = \epsilon^{n-1}_x + \triangle\epsilon^n_x \tag{A.37}$$

$$\epsilon^n_{y_{total}} = \epsilon^{n-1}_y + \triangle\epsilon^n_y \tag{A.38}$$

$$\gamma_{xy_{total}} = \frac{k^n(\theta^n - 1)}{G_w}\tau^n_{xy} \tag{A.39}$$

University of Pretoria etd – Viljoen, A (2004)

Appendix B

Source code of the software developed

The Grisham Algorithm was implemented by coding the procedure in FORTRAN 77. The program consists of a main routine and numerous subprograms, each fulfilling a different task. An additional postprocessing program was also coded to extract data from the linear finite element analysis results in the ABAQUS® environment which could then be read by the main routine.

The code generates a flat structure of np_x by np_y panels in the x- and y- directions respectively as specified by the user. Each panel has stiffener members around its perimeter which may have totally different cross-sectional areas for each member. Once the number of elements per flange and the elements per upright are chosen, they remain the same for all flanges and all uprights.

The program also makes provision for all possible configurations of boundary conditions for the web buckling critical values (all sides simply supported; all sides fixed; 2 horizontal sides fixed and 2 vertical sides simply supported; 2 horizontal sides simply supported and 2 vertical sides fixed).

The web can be modelled using either shell or membrane elements. The flanges and uprights can be modelled using beam or truss elements. Elements can be first or second order. Only buckling of the web is considered, buckling of the uprights is not taken into account.

University of Pretoria etd – Viljoen, A (2004)


```
C
C     VARIABLE DEFINITIONS: (DOUBLE PRECISION)
C
C     ALFA      Diagonal tension angle;  positive when measured
C               from the local x- axis using the right hand rule
C     ALFA11    Expansion coefficient in the x- direction
C     ALFA22    Expansion coefficient in the y- direction
C     ALFA12    Expansion coefficient - shear
C     AX        Half the summed stiffener areas in the x- direction
C               of each panel
C     AY        Half the summed stiffener areas in the y- direction
C               of each panel
C-----------------------------------------------------------------------
C     B         Shortest side of plate (minimum of LX or LY)
C     BETAX/Y   Parameter used to indicate whether pre- or post-buckling
C               modulus of elasticity is applicable in the x- or y-
C               direction of the plate/sheet
C     BBETAX/Y Coefficients used in incremental change in strain equations
C               (second and succeeding iterations) to assist convergence of
C               previously compressively buckled plates that have become
C               unbuckled
C     BNF1      Beam nodal force - component 1
C     BNF2      Beam nodal force - component 2
C     BNF3      Beam nodal force - component 3
C     BUCK      The interaction equation used for bi-axial compression
C               and shear buckling
C-----------------------------------------------------------------------
C     C1-C4     Variables used in calculating the compressive stresses in
C               a plate
C     CONV1X    Convergence requirement 1
C     CONV1Y    Convergence requirement 1
C     CONV2X    Convergence requirement 2
C     CONV2Y    Convergence requirement 2
C-----------------------------------------------------------------------
C     DFX12     Difference in total nodal force between points 1 and 2
C               of the panel in the x-direction (normal to side)
C     DFX23     Difference in total nodal force between points 2 and 3
C               of the panel in the x-direction (normal to side)
C     DFX34     Difference in total nodal force between points 3 and 4
C               of the panel in the x-direction (normal to side)
C     DFX41     Difference in total nodal force between points 4 and 1
C               of the panel in the x-direction (normal to side)
C     DFY12     Difference in total nodal force between points 1 and 2
C               of the panel, in the y-direction (// to side)
C     DFY23     Difference in total nodal force between points 2 and 3
```

University of Pretoria etd – Viljoen, A (2004)


```
C              of the panel, in the y-direction (// to side)
C     DFY34    Difference in total nodal force between points 3 and 4
C              of the panel, in the y-direction (// to side)
C     DFY41    Difference in total nodal force between points 4 and 1
C              of the panel, in the y-direction (// to side)
C-----------------------------------------------------------------------
C     ENODE    Eccentricity of uprights !! (not flanges) - node to node
C     ECC      Eccentricity of uprights - centroid to centroid
C     EP       Modulus of Elasticity of plate
C     EQN1X    Equation used in first test for BETAs
C     EQN1Y    Equation used in first test for BETAs
C     EQN2X    Equation used in second test for BETAs
C     EQN2Y    Equation used in second test for BETAs
C     EQN3X    Equation used in third test for BETAs
C     EQN3Y    Equation used in third test for BETAs
C     ERR      % error made while calculating successive approximations of the
C              diagonal tension angle (ALFA)
C     ES       Minimum % error allowed in calculating root of
C              equation (subroutine)
C     EXB      Modulus of Elasticity of beam in x- direction
C     EXC      Strain in the x-direction due to compressive loading
C              only
C     EXDT     Strain in the x- direction due to diagonal tension
C     EXP      Effective Modulus of Elasticity of buckled plate in x-
C              direction
C     EXT      Total strain in the x- direction [compressive loading +
C              diagonal tension]
C     EXYDT    Shear strain due to diagonal tension
C     EXYT     Total shear strain
C     EYB      Modulus of Elasticity of beam in y-direction
C     EYC      Strain in the y- direction due to compressive loading
C              only
C     EYDT     Strain in the y- direction due to diagonal tension
C     EYP      Effective Modulus of Elasticity of buckled plate in y-
C              direction
C     EYT      Total strain in the y- direction[compressive loading +
C              diagonal tension]
C-----------------------------------------------------------------------
C     FLA      Flange T-section dimension
C     FLB      Flange T-section dimension
C     FLT1     Flange T-section dimension
C     FLT2     Flange T-section dimension
C-----------------------------------------------------------------------
C     GP       Shear modulus of plate
C-----------------------------------------------------------------------
```

University of Pretoria etd – Viljoen, A (2004)


```
C     H1-H4    Variables used to calculate the diagonal tension angle in
C              a plate
C-----------------------------------------------------------------------
C     IXX      Moment of Inertia of upright about an axis through its own
C              centroid and parallel to the web plane
C-----------------------------------------------------------------------
C     K        Diagonal tension factor
C     KS       Shear buckling coefficient for a flat plate
C     KX       Buckling coefficient in x-direction for a flat plate
C     KY       Buckling coefficient in y-direction for a flat plate
C-----------------------------------------------------------------------
C     L        Factor between 0.0 and 1.0 to control the rate at which
C              buckling is incorporated in the solution
C     L1       if LX >= LY then L1 = LY and L2 = LX
C     L2       if LY > LX then L1 = LX and L2 = Ly
C     L12      Plate side length; between corner nodes 1 and 2
C     L23      Plate side length; between corner nodes 2 and 3
C     L34      Plate side length; between corner nodes 3 and 4
C     L41      Plate side length; between corner nodes 4 and 1
C     LPNX     Length between nodes in the x-direction
C     LPNY     Length between nodes in the y-direction
C     LTOTX    Total length of model in x-direction
C     LTOTY    Total length of model in y-direction
C     LX       Length of plate in x-direction
C     LY       Length of plate in y-direction
C-----------------------------------------------------------------------
C     NCX      Nodal x-coordinate generated within program
C     NCY      Nodal y-coordinate generated within program
C     NCZ      Nodal z-coordinate generated within program
C     NF1T     Total force at node - component 1 read from ABAQUS output
C     NF2T     Total force at node - component 2 read from ABAQUS output
C     NF3T     Total force at node - component 3 read from ABAQUS output
C-----------------------------------------------------------------------
C     POISS    Poisson's ratio
C-----------------------------------------------------------------------
C     RHO      Radius of gyration of the upright area
C-----------------------------------------------------------------------
C     SNF1     Shell nodal force - component 1
C     SNF2     Shell nodal force - component 2
C     SNF3     Shell nodal force - component 3
C     SX       Average panel normal stress in the x-direction
C     SX12     Normal stress to plate edge between nodes 1 and 2
C     SX23     Normal stress to plate edge between nodes 2 and 3
C     SX34     Normal stress to plate edge between nodes 3 and 4
C     SX41     Normal stress to plate edge between nodes 4 and 1
```

University of Pretoria etd – Viljoen, A (2004)


```
C     SXC       Total compressive stress acting on plate in x-direction
C     SXCR      Modified buckling allowable stress in x- direction
C     SXCR0     Critical buckling stress in x-direction (geometry only)
C     SXDT      Diagonal tension stress in plate following buckling
C     SXEFF     Effective (resultant) tensile/compressive stress applied to
C               panel/sheet calculated from panel/sheet nodal
C               forces [x-direction]
C     SXY       Average panel shear stress
C     SXY12     Shear stress alongside plate edge between nodes 1 and 2
C     SXY23     Shear stress alongside plate edge between nodes 2 and 3
C     SXY34     Shear stress alongside plate edge between nodes 3 and 4
C     SXY41     Shear stress alongside plate edge between nodes 4 and 1
C     SXYEFF    Effective (resultant) shear stress applied to panel/sheet
C               calculated form panel/sheet nodal forces [same value on each
C               of the 4 sides]
C     SXYCR     Modified buckling shear allowable stress
C     SXYCR0    Critical shear buckling stress (geometry only)
C     SY        Average panel normal stress in the y-direction
C     SYC       Total compressive stress acting on plate in y-direction
C     SYCR      Modified buckling allowable stress in y- direction
C     SYCR0     Critical buckling stress in y-direction (geometry only)
C     SYDT      Diagonal tension stress in plate following buckling
C     SYEFF     Effective (resultant) tensile/compressive stress applied to
C               panel/sheet calculated from panel/sheet nodal
C               forces [x-direction]
C-----------------------------------------------------------------------
C     T         Plate/sheet thickness
C     TDB       Buckling equation rounded to four decimal places value
C     THETA     Diagonal tension angle
C-----------------------------------------------------------------------
C     UPRA      Angle cross-section dimension
C     UPRT      Angle cross-section thickness dimension
C     UPRY      Angle cross-section dimension
C-----------------------------------------------------------------------
C     XALFA     Guestimate of diagonal tension angle in method of successive
C               approximations to determine ALFA
C     XN        The root of the equation (subroutine)
C     XR1       Initial guess of root of equation (subroutine)
C     XR2       Initial guess of root of equation (subroutine)
C-----------------------------------------------------------------------
C     YCENT     Centroid position of angled upright
C-----------------------------------------------------------------------
C
C
C     VARIABLE DEFINITIONS: (INTEGERS)
```

University of Pretoria etd – Viljoen, A (2004)


```
C
C     BENF     Beam element node force array used with node forces
C     BNNNF    Beam node number for node force data
C     BTOT     Total number of panels that are not buckled at the end
C              of the analysis
C-----------------------------------------------------------------------
C     C2O      Constant that = 2 when second order elements are used
C-----------------------------------------------------------------------
C     ELBH     Horizontal beam element set
C     ELBHN    Horizontal beam element number
C     ELBV     Vertical beam element set
C     ELBVN    Vertical beam element number
C     ELS      Shell element set
C     ELSN     Shell element number
C-----------------------------------------------------------------------
C     FN       File number for each panel
C     FN       FIle number for each panel
C-----------------------------------------------------------------------
C     IC1,2..  Counters
C     ICB      Beam element counter
C     ICBH     Horizontal beam element counter
C     ICBV     Vertical beam element counter
C     ICS      Shell element counter
C     IM       Maximum number of iterations allowed to determine root of
C              equation (subroutine)
C     ITN      Iteration number
C-----------------------------------------------------------------------
C     K1,K2..  Constants
C-----------------------------------------------------------------------
C     NALL     All the nodes in the linear FEA model
C     NBEPP    Total number of beam elements associated with each panel
C     NEX      Number of elements per panel/sheet along the x-axis
C     NEY      Number of elements per panel/sheet along the y-axis
C     NFLS     Number of flange element sets
C     NI       Same as ITER
C     NINCY    Node number increment per row along the y-axis
C     NIPPX    Node increment per panel/sheet in the x-direction
C     NIPPY    Node increment per panel/sheet in the y-direction
C     NNXMAX   Maximum node number value in the x-direction
C     NNYMAX   Maximum node number value in the y-direction
C     NONX     Total number of nodes along the x-axis
C     NONY     Total number of nodes along the y-axis
C     NP       Number of panels in structure
C     NPX      Number of panels/sheets along the x-axis
C     NPY      Number of panels/sheets along the y-axis\
```

University of Pretoria etd – Viljoen, A (2004)


```
C     NRBNF    Number of records in the beam element node forces file
C     NSEPP    Total number of shell elements associated with each panel
C     NTOTAL   Total number of nodes generated
C-----------------------------------------------------------------------
C     PCNN     Panel corner node number
C     PNLBE    Panel beam elements (panel number,panel beam element
C              number - random)
C     PNLNN    Panel corner node numbers [4] (panel number,panel corner node
C              number - clockwise; must be clockwise and start at 0,0,0
C              so that stresses in each direction can be calculated
C              correctly)
C     PNLSE    Panel shell elements (panel number,panel shell element
C              number - random)
C-----------------------------------------------------------------------
C     SHENF    Shell element node force array used with node forces
C     SHNNNF   Shell node number for node force data
C-----------------------------------------------------------------------
C     YINC     Node number increment per row along the y-axis;
C              = relevant NINCY value
C-----------------------------------------------------------------------
C
C***********************************************************************
C              VARIABLE DEFINITIONS:  (CHARACTERS)
C
C     COMMA    A comma
C     UPRE     Upright eccentricity presence (=NOE or ECC)
C
C***********************************************************************
C***********************************************************************
C
C               MAIN PROGRAM
C
C***********************************************************************
C***********************************************************************
C
C
      DOUBLE PRECISION ALFA,
     +AX,AY,
     +B,BETAX(301),BETAY(301),BBETAX,BBETAY,
     +EP,ERR,EXB,EXP,EYB,EYP,ECC(301),ENODE(301),
     +EQN1X,EQN1Y,EQN2X,EQN2Y,EQN3X,EQN3Y,
     +FLA(301),FLB(301),FLT1(301),FLT2(301),
     +GP,
     +H1,H2,H3,H4,
     +IXX(301),
```

University of Pretoria etd – Viljoen, A (2004)


```
     +K,KS,KX,KY,
     +L,L1,L2,LPNX,LPNY,LTOTX,LTOTY,LX,LY,
     +POISS,PI,
     +RHO(301),
     +SXCR0,SYCR0,SXYCR0,
     +T(301),THETA,TDB,
     +UPRA(301),UPRT(301),UPRY(301),
     +XALFA,
     +YCENT(301),
     +SXAV,SYAV,SXYAV,SX,SY,SXY,BB,SXYCR1,SXYCR2,SXYCR,
     +SXCR,SYCR,SXCRI(301),SYCRI(301),SXMOD,SYMOD,
     +BUCK,EPSX,EPSY,EPS,
     +SXDT(301),SYDT(301),EXDT,EYDT,EXYDT,C1,C2,C3,C4,
     +SXC,SYC,EXC,EYC,EXT(301),EYT(301),EXYT,
     +SXPS(301),SYPS(301),SXYPS(301),
     +SXDTI(301),SYDTI(301),SXDTII(301),SYDTII(301),
     +NCX(1000),NCY(1000),NCZ(1000),
     +S11(1000),S22(1000),S12(1000),
     +SXAVE(301),SYAVE(301),SXYAVE(301),
     +SXAVP(301),SYAVP(301),SXYAVP(301),
     +SXTE(1000),SYTE(1000),SXYTE(1000),
     +SXTP(1000),SYTP(1000),SXYTP(1000),
     +K1X,K1Y,K1XP,K1YP,K2X,K2Y,K2XP,K2YP,K1XN(301),K1YN(301),
     +K1XNN(301),K1YNN(301),K2XN(301),K2YN(301),K1XPP,K1YPP,
     +K2XPP,K2YPP,K2XNN(301),K2YNN(301),K2XPPP,K2YPPP,
     +K1XVAL(301),K1YVAL(301),
     +K1XNC(301),K1YNC(301),PVAL,
     +UPRRAD(301),FLRAD(301),AFL(301),AUPR(301),AUPRE(301)
C
      INTEGER BHTOT,BVTOT,C20,
     +CONA(301),CONB(301),CONC(301),COND(301),
     +BCONA(301),BCONB(301),
     +CTOT,BCTOT,
     +FN,FN2,
     +ITN,
     +NBEPP,NP,NFLS,NRBNF,NINCY,
     +NSEPP,NUPRS,
     +SEC,
     +TSTART,TSTOP,
     +YINC,
     +ELE(1000),
     +ELBH(1000,1000),ELBHN(1000),ELBV(1000,1000),ELBVN(1000),
     +ELSN(1000),
     +IPT(1000),SPT(1000),
     +NALL(1000),
```

University of Pretoria etd – Viljoen, A (2004)


```
     +PCNN(1000),ELS(1000,1000),
     +PNLNN(1000,6),PNLSE(1000,20),PNLBE(1000,20),
     +K1XID(301),K1YID(301),
     +BPL(301),BTOT,BPID(301),
     +CONAT,CONBT,CONCT,CONDT,
     +IBCMAX,IFAC
C
      CHARACTER*(1)COMMA
      CHARACTER*(3)UPRE
C
      OPEN(10,FILE='bukl.dat')
      OPEN(21,FILE='conv1.dat')
      OPEN(12,FILE='inputdata.dat')
      OPEN(13,FILE='conv2.dat')
      OPEN(31,FILE='pnl1-x.dat')
      OPEN(32,FILE='pnl2-x.dat')
      OPEN(33,FILE='pnl3-x.dat')
      OPEN(34,FILE='pnl4-x.dat')
      OPEN(35,FILE='pnl5-x.dat')
      OPEN(36,FILE='pnl6-x.dat')
      OPEN(41,FILE='pnl1-y.dat')
      OPEN(42,FILE='pnl2-y.dat')
      OPEN(43,FILE='pnl3-y.dat')
      OPEN(44,FILE='pnl4-y.dat')
      OPEN(45,FILE='pnl5-y.dat')
      OPEN(46,FILE='pnl6-y.dat')
C
C ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C
C     INPUT DATA:
C
C ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C
C
      NPX=6
      NPY=1
      NEX=3
      NEY=3
      LX=0.254
      LY=0.7254
      NP=NPX*NPY
C
      DO 63 I=1,NP+1
       UPRA(I)=0.0254
       UPRT(I)=0.003175
```

University of Pretoria etd – Viljoen, A (2004)


```
63    CONTINUE
C
      UPRE='ECC'
C
      DO 64 J=1,NP
       T(J)=0.000635
       FLRAD(J)=0.00881
       FLRAD(NP+J)=0.01177
64    CONTINUE
C
      EXB=71.0E9
      EYB=71.0E9
      EP=72.4E9
      GP=26.92E9
      POISS=0.3
      C2O=2
C
      COMMA=','
      PVAL=0.0
C
      IBCMAX=60
      PI=3.14159265359
C
C +++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C
C     PROGRAM STARTS
C
C +++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C
      CALL TIME(SEC)
        TSTART=SEC
        WRITE(10,10)TSTART
10      FORMAT('TSTART = ',I20)
C
C            geometric calculations for structure
C
      LTOTX=LX*NPX
      LTOTY=LY*NPY
      NNXMAX=C2O*NPX*NEX+1
      NONX=NNXMAX
      NONY=C2O*NPY*NEY+1
      NIPPX=C2O*NEX
      NBEPP=2*NEX+2*NEY
      NSEPP=NEX*NEY
      LPNX=LTOTX/(NONX-1)
```

University of Pretoria etd – Viljoen, A (2004)


```
      LPNY=LTOTY/(NONY-1)
      NINCY=NNXMAX
      NNYMAX=C20*NPY*NEY+1
      NTOTAL=NNXMAX*NNYMAX
      NIPPY=C20*NEY*NINCY
C
      DO 67 I=1,NPX*NPY+NPX
       AFL(I)=PI*FLRAD(I)**2
67    CONTINUE
C
C             Geometric properties of uprights calculated
C
      DO 14 I=1,NPX+1
       AUPR(I)=(UPRA(I)-UPRT(I))*UPRT(I)+UPRA(I)*UPRT(I)
       YCENT(I)=((UPRA(I)-UPRT(I))*UPRT(I)*UPRT(I)/2.0+
     +          UPRA(I)*UPRT(I)*UPRA(I)/2.0)/((UPRA(I)-
     +          UPRT(I))*UPRT(I)+UPRA(I)*UPRT(I))
       UPRY(I)=UPRA(I)-YCENT(I)
       IXX(I)=(1.0/3.0)*(UPRT(I)*UPRY(I)**3+UPRA(I)*
     +        (UPRA(I)-UPRY(I))**3-(UPRA(I)-UPRT(I))*
     +        (UPRA(I)-UPRY(I)-UPRT(I))**3)
       RHO(I)=SQRT(IXX(I)/AUPR(I))
       IF(I.EQ.1)THEN
        ECC(I)=YCENT(I)+T(I)/2.0
        ENODE(I)=-(T(I)/2.0+UPRT(I)/2.0)
       ELSE IF(I.EQ.NPX+1)THEN
        ECC(I)=YCENT(I)+T(I-1)/2.0
        ENODE(I)=-(T(I-1)/2.0+UPRT(I)/2.0)
       ELSE
        ECC(I)=YCENT(I)+(T(I-1)+T(I))/4.0
        ENODE(I)=-((T(I-1)+T(I))/4.0+UPRT(I)/2.0)
       ENDIF
       AUPRE(I)=AUPR(I)/(1+(ECC(I)/RHO(I))**2)
      WRITE(10,15)I,AUPR(I),YCENT(I),IXX(I),RHO(I),
     +ECC(I),AUPRE(I),ENODE(I)
15    FORMAT(/,'    Upright No',I3,/,
     +'AUPR =  ',G15.5,/,
     +'YCENT = ',G15.5,'       IXX =    ',G20.5,/,
     +'RHO =   ',G15.5,'       ECC =    ',G15.5,/,
     +'AUPRE = ',G15.5,'       ENODE =  ',G15.5)
14    CONTINUE
C
      CALL FNODES(NPX,NPY,NEX,NEY,C20,LTOTX,LTOTY,UPRE,
     +ENODE,NALL,NCX,NCY,NCZ,NTOT,NTFLAT)
      PRINT *,' Finished subroutine fnodes !@!!!!'
```

University of Pretoria etd – Viljoen, A (2004)


```
C
      CALL ELEMENTS(NPX,NPY,NEX,NEY,C2O,UPRE,NALL,ELS,ELBH,
     +ELBV,ELBVN,ELBHN,ELSN,PNLSE,ICS,ICBH,ICBV,NTFLAT)
      PRINT *,' Finished subroutine ele !!!!'
C
C          Iterative loop to determine best value for BETAX and BETAY
C          for each panel to satisfy convergence !
C
      DO 650 I=1,NP
      BETAX(I)=0.85
      BETAY(I)=0.85
650   CONTINUE
C
      DO 3500 IBC=1,IBCMAX
C
      PRINT *,'IBC = ',IBC
      WRITE(10,651)IBC
      WRITE(13,651)IBC
651   FORMAT(/,' %%%%%%%%%%% Loop',I3,' for the BETAs',
     +' convergence iterations ! %%%%%%%%%%%%%',/)
C
      DO 655 I=1,NP
       BPL(I)=0
       BPID(I)=0
       SXCRI(I)=0.0
       SYCRI(I)=0.0
       SXDTI(I)=0.0
       SYDTI(I)=0.0
       SXDTII(I)=0.0
       SYDTII(I)=0.0
       K1XN(I)=0.0
       K1YN(I)=0.0
       K1XNN(I)=0.0
       K1YNN(I)=0.0
       K2XN(I)=0.0
       K2YN(I)=0.0
       K2XNN(I)=0.0
       K2YNN(I)=0.0
       K1XVAL(I)=0
       K1YVAL(I)=0
       CONA(I)=0
       CONB(I)=0
       CONC(I)=0
       COND(I)=0
655   CONTINUE
```

University of Pretoria etd – Viljoen, A (2004)


```
C
C *************** ITERATION LOOP STARTS HERE ***********************
C
      DO 658 FN=31,36,1
      WRITE(FN,657)
657   FORMAT(/,'IBC ITN K1X=SXC/SXCR',4X,'SXC',3X,'SXCR=SXYCR/SXY',
     +4X,'SXYCR',8X,'SXY',6X,'SX-SXCR',7X,'SY-SYCR',8X,'SX',8X,'SXAV',
     +6X,'SXDT(I)')
658   CONTINUE
      DO 661 FN2=41,46,1
      WRITE(FN2,662)
662   FORMAT(/,'IBC ITN K1Y=SYC/SYCR',4X,'SYC',3X,'SYCR=SXYCR/SXY',
     +4X,'SXYCR',8X,'SXY',6X,'SY-SYCR',7X,'SX-SXCR',8X,'SY',8X,'SYAV',
     +6X,'SYDT(I)')
661   CONTINUE
C
      DO 3000 ITN=1,20
C
      WRITE(10,692)ITN
      WRITE(21,692)ITN
692   FORMAT(/,'####################  Iteration ',I2,
     +'  ########################',/)
C
      DO 690 I=1,NP
       SXTP(I)=0.0
       SYTP(I)=0.0
       SXYTP(I)=0.0
       SXAVP(I)=0.0
       SYAVP(I)=0.0
       SXYAVP(I)=0.0
690   CONTINUE
C
      DO 691 J=1,ICS
       SXTE(J)=0.0
       SYTE(J)=0.0
       SXYTE(J)=0.0
       SXAVE(J)=0.0
       SYAVE(J)=0.0
       SXYAVE(J)=0.0
691   CONTINUE
C
      CALL FEMINP(NPX,NPY,NEX,NEY,C20,ICS,ICBH,ICBV,
     +NTFLAT,NTOT,PNLSE,ITN,UPRE,T,SXPS,SYPS,SXYPS,NALL,
     +NCX,NCY,NCZ,ELS,ELSN,ELBHN,ELBH,ELBVN,ELBV)
C
```

University of Pretoria etd – Viljoen, A (2004)


```
      CALL SYSTEM("abq58 job=femmodel interactive")
C
      CALL SYSTEM("strdata.x")
C
C           read in the stresses for the panel elements
C
      NRBNF=0
      WRITE(12,1082)
1082  FORMAT(/,2X'ELE',3X,'INTGR PT',9X,'SECT PT',11X,'S11',11X,'S22',
     +3X,'S12')
      OPEN(4,FILE='panelstress.txt')
      DO 1080 I=1,10000000
       READ(4,*,END=1081)ELE(I),IPT(I),SPT(I),S11(I),S22(I),S12(I)
       WRITE(12,*)ELE(I),IPT(I),SPT(I),S11(I),S22(I),S12(I)
1080  CONTINUE
1081  NRBNF=I-1
      PRINT *,'NRBNF =',NRBNF
      CLOSE(4)
C
      DO 1090 I=1,ICS
      ICES=0
       WRITE(10,1092)ELSN(I)
1092   FORMAT(/,'Element Number:',I4,/,'-------------------',
     + /,8X,'Integr pt',
     + 6X,'S11',17X,'S22',17X,'S12')
       DO 1095 J=1,NRBNF
         IF(ELSN(I).EQ.ELE(J))THEN
         SXTE(I)=SXTE(I)+S11(J)
         SYTE(I)=SYTE(I)+S22(J)
         SXYTE(I)=SXYTE(I)+S12(J)
         ICES=ICES+1
         WRITE(10,1096)IPT(J),S11(J),S22(J),S12(J)
1096       FORMAT(10X,I4,6X,F18.3,2X,F18.3,2X,F18.3)
         ENDIF
1095  CONTINUE
       WRITE(10,1099)ICES,SXTE(I),SYTE(I),SXYTE(I)
1099   FORMAT(/,'TOTAL:',4X,I4,10X,F18.3,2X,F18.3,2X,F18.3)
        SXAVE(I)=SXTE(I)/ICES
        SYAVE(I)=SYTE(I)/ICES
        SXYAVE(I)=SXYTE(I)/ICES
        WRITE(10,1098)SXAVE(I),SYAVE(I),SXYAVE(I)
1098    FORMAT(/,'AVERAGE:',12X,F18.3,2X,F18.3,2X,F18.3,/)
1090  CONTINUE
C
      DO 1120 I=1,NP
```

University of Pretoria etd – Viljoen, A (2004)


```
       WRITE(10,1122)I
1122   FORMAT(/,'Panel Number: ',I3,/,'-----------------',/,
     + 9X,'SXAVE',16X,'SYAVE',13X,
     +'SXYAVE')
       DO 1130 J=1,NSEPP
       DO 1131 K=1,ICS
        IF(PNLSE(I,J).EQ.ELSN(K))THEN
        SXTP(I)=SXTP(I)+SXAVE(K)
        SYTP(I)=SYTP(I)+SYAVE(K)
        SXYTP(I)=SXYTP(I)+SXYAVE(K)
        WRITE(10,1140)SXAVE(K),SYAVE(K),SXYAVE(K)
1140    FORMAT(20X,F18.3,2X,F18.3,2X,F18.3)
        ENDIF
1131   CONTINUE
1130   CONTINUE
       WRITE(10,1125)SXTP(I),SYTP(I),SXYTP(I)
1125   FORMAT(/,'TOTAL:',18X,F18.3,2X,F18.3,2X,F18.3)
        SXAVP(I)=SXTP(I)/NSEPP
        SYAVP(I)=SYTP(I)/NSEPP
        SXYAVP(I)=SXYTP(I)/NSEPP
        WRITE(10,1128)SXAVP(I),SYAVP(I),SXYAVP(I)
1128    FORMAT(/,'AVERAGE:',12X,F18.3,2X,F18.3,2X,F18.3,/)
1120   CONTINUE
C
C          calculate average panel stress
C
       NPC=0
       BTOT=0
       CTOT=0
       CONAT=0
       CONBT=0
       CONCT=0
       CONDT=0
       DO 1163 IJ=1,NP
        K1XNC(IJ)=0
        K1YNC(IJ)=0
        K1XID(IJ)=0
        K1YID(IJ)=0
1163   CONTINUE
       WRITE(13,1166)IBC,ITN
1166   FORMAT('IBC = ',I4,/,'ITN = ',I4)
C
C          Individual panel loop !
C
       XFAC1=0.0
```

University of Pretoria etd – Viljoen, A (2004)


```
      IFAC=0
      DO 3002 I=1,NP
      WRITE(10,1161)I,ITN
1161  FORMAT(/,1X,'******************** Analyzing Panel No',I3,
     +' / Iteration No',I3,' *************************',/)
C
      SXAV=SXAVP(I)
      SYAV=SYAVP(I)
      SXYAV=SXYAVP(I)
      WRITE(10,1162)SXAV,SYAV,SXYAV
1162  FORMAT('SXAV = ',F18.3,'   SYAV = ',F18.3,'   SXYAV = ',F18.3)
C
      WRITE(*,1031)AFL(I),AFL(I+NPX)
      WRITE(10,1031)AFL(I),AFL(I+NPX)
1031  FORMAT('AFL(I) = ',F15.10,'    AFL(I+NPX) = ',F15.10)
       AX=0.5*(AFL(I)+AFL(I+NPX))
       IF(UPRE.EQ.'ECC')THEN
        AY=AUPRE(I)
       ELSE
        AY=0.5*(AUPR(I+IFAC)+AUPR(I+1+IFAC))
       ENDIF
       IFAC=INT(I/NPX)
      WRITE(10,1021)AX,AY,IFAC
1021  FORMAT('AX = ',F15.10,'    AY = ',F15.10,/,'IFAC = ',I4)
C
C         Buckling coefficients
C
      KX=1.985*((LY**2)/(LX**2))+0.941*(LY/LX)+6.31
      KY=1.985*((LX**2)/(LY**2))+0.941*(LX/LY)+6.31
      IF(LX.GE.LY)THEN
         according to ILENGTH requirements !
       L1=LY
       L2=LX
       ELSE
       L1=LX
       L2=LY
      ENDIF
      KS=5.21*((L1**2)/(L2**2))+0.14*(L1/L2)+8.05
C
      WRITE(10,FMT=1020)KX,KY,KS,L1,L2
1020  FORMAT('KX = ',F15.9,/,'KY = ',F15.9,/,'KS = ',
     +F15.9,/,'L1 = ',F15.9,/,'L2 = ',F15.9)
      IF(LX.GE.LY)THEN
       B=LY
       ELSE
```

University of Pretoria etd – Viljoen, A (2004)


```
       B=LX
      ENDIF
      WRITE(10,FMT=1042)B
1042  FORMAT('B = ',F15.9)
C
      SXCR0=KX*EP*((T(I)/LY)**2)
      SYCR0=KY*EP*((T(I)/LX)**2)
      SXYCR0=KS*EP*((T(I)/B)**2)
      WRITE(10,FMT=1045)SXCR0,SYCR0,SXYCR0
1045  FORMAT('SXCR0 = ',F15.3,3X,'SYCR0 = ',F15.3,3X,'SXYCR0 = ',F15.3)
C
      L=1.0
      IF(ITN.EQ.1)THEN
C
      SX=-1.0*SXAV
      SY=-1.0*SYAV
      SXY=SXYAV
      WRITE(10,1200)SX,SY,SXY
1200  FORMAT('SX = ',F18.3,'  SY = ',F18.3,'  SXY = ',F18.3)
C
      IF(SX.LT.0.0.AND.SY.LT.0.0)THEN
      SXYCR=SXYCR0*((1-SX/SXCR0-SY/SYCR0)**0.5)
      ELSE
      BB=((SXYCR0**2)*(SX*SYCR0+SY*SXCR0))/(ABS(SXY)*SXCR0*SYCR0)
      SXYCR1=(-BB+(BB**2+4*SXYCR0**2)**0.5)/2.0
      SXYCR2=(-BB-(BB**2+4*SXYCR0**2)**0.5)/2.0
      WRITE(10,1210)BB,SXYCR1,SXYCR2
1210  FORMAT('BB = ',F18.3,/,'SXYCR1 = ',F18.3,/,
     +'SXYCR2 = ',F18.3)
      IF(ABS(SXYCR1).GE.ABS(SXYCR2))THEN
       SXYCR=SXYCR1
      ELSE
       SXYCR=SXYCR2
      ENDIF
      ENDIF
      SXCR=SX*(ABS(SXYCR)/ABS(SXY))
      SYCR=SY*(ABS(SXYCR)/ABS(SXY))
      WRITE(10,1220)SXYCR,SXCR,SYCR
1220  FORMAT('SXYCR= ',F18.3,'  SXCR= ',F18.3,'  SYCR= ',F18.3)
C
      BUCKT1=SXCR/SXCR0
      BUCKT2=SYCR/SYCR0
      BUCKT3=(SXYCR/SXYCR0)**2
       WRITE(10,1215)BUCKT1,BUCKT2,BUCKT3
1215   FORMAT('BUCKT1 = ',F20.10,/,'BUCKT2 = ',F20.10,/,
```

University of Pretoria etd – Viljoen, A (2004)


```
     + 'BUCKT3 = ',F20.10)
       BUCK=SXCR/SXCR0+SYCR/SYCR0+(SXYCR/SXYCR0)**2
       TDB=NINT(BUCK*10000.0)/10000.0
       WRITE(10,1222)BUCK,TDB
1222   FORMAT('Buckling Equation = ',F20.10,/,'TDB =',F20.4)
C
       IF(TDB.LT.1.0)THEN
         BPL(I)=BPL(I)+1
       WRITE(10,1224)I
1224   FORMAT('    Panel no',I2,' does not buckle !',/)
       GOTO 3002
       ENDIF
C
       K=TANH(0.5*LOG10(ABS(SXY)/ABS(SXYCR)))
C
       H1=(-K*ABS(SXY))/(EXB*(2.0*AX/LY/T(I)+0.5*(1-K)))
       H2=(-K*ABS(SXY))/(EYB*(AY/LX/T(I)+0.5*(1-K)))
       H3=ABS(SXY)*2.0*K/EP
       H4=(1-K+POISS-POISS*K)*(ABS(SXY)/EP)
       WRITE(10,1230)K,H1,H2,H3,H4
1230   FORMAT('K = ',F7.5,/,
     +'H1 = ',F15.11,'  H2 = ',F15.11,'  H3 = ',F15.11,
     +'  H4 = ',F15.11)
C
       XALFA=PI/4.0
1240   WRITE(10,1245)XALFA
1245   FORMAT('XALFA = ',F10.7)
       EPSX=H1/TAN(XALFA)
       EPSY=H2*TAN(XALFA)
       EPS=H3/SIN(2*XALFA)+H4*SIN(2*XALFA)
       WRITE(10,1250)EPSX,EPSY,EPS
1250   FORMAT('EPSX = ',F10.8,'   EPSY = ',F10.8,'   EPS = ',F10.8)
C
       IF((EPS-EPSX).LT.0.0.AND.(EPS-EPSY).GT.0.0)THEN
        PRINT *,' Diagonal tension angle cannot be calculated !'
        GOTO 3001
       ELSE IF((EPS-EPSX).GT.0.0.AND.(EPS-EPSY).LT.0.0)THEN
        PRINT *,' Diagonal tension angle cannot be calculated !'
        GOTO 3001
       ELSE IF((EPS-EPSY).EQ.0.0)THEN
        PRINT *,' Diagonal tension angle cannot be calculated !'
        GOTO 3001
       ENDIF
       ALFA=ATAN(((EPS-EPSX)/(EPS-EPSY))**0.5)
       ERR=(ABS(ALFA-XALFA))*100.0
```

University of Pretoria etd – Viljoen, A (2004)


```
      WRITE(10,1255)ALFA,ERR
1255  FORMAT('ALFA = ',F10.7,' % ERR = ',F6.3)
      IF(ERR.GE.0.1)THEN
       XALFA=ALFA
       GOTO 1240
      ENDIF
      WRITE(10,1256)I,ALFA*180.0/3.141596
1256  FORMAT(/,'   Diagonal Tension Angle for Panel',I3,' = ',F6.3,
     +' degrees !',/)
C
      SXDT(I)=K*ABS(SXY)/TAN(ALFA)
      SYDT(I)=K*ABS(SXY)*TAN(ALFA)
      WRITE(10,1260)SXDT(I),SYDT(I)
1260  FORMAT('SXDT(I) = ',F18.3,'SYDT(I) = ',F18.3)
C
      EXDT=(-1.0/EP)*(EP*LY*T(I)/2.0/AX/EXB+1)*(SXDT(I)-POISS*SYDT(I))
      EYDT=(-1.0/EP)*(EP*LX*T(I)/AY/EYB+1)*(SYDT(I)-POISS*SXDT(I))
      WRITE(10,1265)EXDT,EYDT
1265  FORMAT('EXDT = ',F13.10,'  EYDT = ',F13.10)
C
      THETA=(1.0/(2*(1.0+POISS)))*(4.0/(SIN(2*ALFA))**2+
     +(EP*LX*T(I)*(TAN(ALFA))**2)/(EYB*(AY+(0.5*LX*T(I))*(1-K)))+
     +(EP*LY*T(I)/(TAN(ALFA))**2)/(EXB*(2.0*AX+(0.5*LY*T(I))*(1-K))))
      EXYDT=K*(THETA-1)*SXY/GP
C
      C1=1+LY*T(I)*EP*BETAX(I)/2.0/AX/EXB
      C2=1+LX*T(I)*EP*BETAY(I)/AY/EYB
      C3=1-BETAX(I)
      C4=1-BETAY(I)
      WRITE(10,1270)THETA,EXYDT,BETAX(I),BETAY(I),C1,C2,C3,C4
1270  FORMAT('THETA = ',F10.6,'   EXYDT = ',F13.10,/,
     +'BETAX(I) = ',F10.6,'   BETAY(I) = ',F10.6,/,
     +'C1 = ',F10.6,'  C2 = ',F10.6,'  C3 = ',F10.6,'  C4 = ',F10.6)
C
      SXC=(C2*C3*(SX-SXCR)*L+BETAX(I)*POISS*C4*(SY-SYCR)*L)/C1/C2
      SYC=(C1*C4*(SY-SYCR)*L+BETAY(I)*POISS*C3*(SX-SXCR)*L)/C1/C2
      WRITE(10,1275)SXC,SYC
1275  FORMAT('SXC = ',F18.3,'   SYC = ',F18.3)
C
      EXC=(-1.0/EP)*(1+LY*T(I)*EP/2.0/AX/EXB)*(SXC-POISS*SYC)*
     +(ABS(BETAX(I)+BETAX(I)*BETAY(I)-1))
      EYC=(-1.0/EP)*(1+LX*T(I)*EP/AY/EYB)*(SYC-POISS*SXC)*
     +(ABS(BETAY(I)+BETAX(I)*BETAY(I)-1))
      WRITE(10,1280)EXC,EYC
1280  FORMAT('EXC = ',F13.10,'   EYC = ',F13.10)
```

University of Pretoria etd – Viljoen, A (2004)


```
C
      EXT(I)=EXDT+EXC
      EYT(I)=EYDT+EYC
      EXYT=EXYDT
      WRITE(10,1285)EXT(I),EYT(I),EXYT
1285  FORMAT('EXT(I) = ',F13.10,'   EYT(I) = ',F13.10,
     +'   EXYT = ',F13.10)
C
      SXPS(I)=-EP*(EXT(I)+POISS*EYT(I))/(1-POISS**2)
      SYPS(I)=-EP*(EYT(I)+POISS*EXT(I))/(1-POISS**2)
      SXYPS(I)=-EP*EXYT/(2*(1+POISS))
      WRITE(10,1290)SXPS(I),SYPS(I),SXYPS(I)
1290  FORMAT('SXPS(I) = ',F18.3,'   SYPS(I) = ',F18.3,
     +'   SXYPS = ',F18.3)
C
      IF(ITN.GE.2)THEN
      SXMOD=SXAV-SXDTI(I)
      SYMOD=SYAV-SYDTI(I)
      WRITE(10,1363)SXMOD,SYMOD
1363  FORMAT('SXMOD = ',F18.3,'  SYMOD = ',F18.3)
C
      SX=-1.0*SXMOD
      SY=-1.0*SYMOD
      SXY=SXYAV
      WRITE(10,1360)SX,SY,SXY
1360  FORMAT('SX = ',F18.3,'  SY = ',F18.3,'  SXY = ',F18.3)
C
      IF(SX.LT.0.0.AND.SY.LT.0.0)THEN
      SXYCR=SXYCR0*((1-SX/SXCR0-SY/SYCR0)**0.5)
      ELSE
      BB=((SXYCR0**2)*(SX*SYCR0+SY*SXCR0))/(ABS(SXY)*SXCR0*SYCR0)
      SXYCR1=(-BB+(BB**2+4*SXYCR0**2)**0.5)/2.0
      SXYCR2=(-BB-(BB**2+4*SXYCR0**2)**0.5)/2.0
      WRITE(10,1370)BB,SXYCR1,SXYCR2
1370  FORMAT('BB = ',F18.3,/,'SXYCR1 = ',F18.3,/,
     +'SXYCR2 = ',F18.3)
      IF(ABS(SXYCR1).GE.ABS(SXYCR2))THEN
       SXYCR=SXYCR1
      ELSE
       SXYCR=SXYCR2
      ENDIF
      ENDIF
      SXCR=SX*(ABS(SXYCR)/ABS(SXY))
      SYCR=SY*(ABS(SXYCR)/ABS(SXY))
      WRITE(10,1380)SXYCR,SXCR,SYCR
```

University of Pretoria etd – Viljoen, A (2004)


```
1380  FORMAT('SXYCR= ',F18.3,'  SXCR= ',F18.3,'  SYCR= ',F18.3)
C
      BUCKT1=SXCR/SXCR0
      BUCKT2=SYCR/SYCR0
      BUCKT3=(SXYCR/SXYCR0)**2
       WRITE(10,1376)BUCKT1,BUCKT2,BUCKT3
1376   FORMAT('BUCKT1 = 'F20.10,/,'BUCKT2 = 'F20.10,/,
     + 'BUCKT3 = ',F20.10)
      BUCK=SXCR/SXCR0+SYCR/SYCR0+(SXYCR/SXYCR0)**2
      TDB=NINT(BUCK*10000.0)/10000.0
      WRITE(10,1382)BUCK,TDB
1382  FORMAT('Buckling Equation = ',F20.10,/,'TDB = ',F20.4)
C
      IF(TDB.LT.1.0)THEN
       BPL(I)=BPL(I)+1
      WRITE(10,1384)I
1384  FORMAT('    Panel no',I2,' does not buckle !',/)
      GOTO 3002
      ENDIF
C
      K=TANH(0.5*LOG10(ABS(SXY)/ABS(SXYCR)))
C
      H1=(-K*ABS(SXY))/(EXB*(2.0*AX/LY/T(I)+0.5*(1-K)))
      H2=(-K*ABS(SXY))/(EYB*(AY/LX/T(I)+0.5*(1-K)))
      H3=ABS(SXY)*2.0*K/EP
      H4=(1-K+POISS-POISS*K)*(ABS(SXY)/EP)
      WRITE(10,1400)K,H1,H2,H3,H4
1400  FORMAT('K = ',F7.5,/,
     +'H1 = ',F15.11,'  H2 = ',F15.11,'  H3 = ',F15.11,
     +'  H4 = ',F15.11)
C
      XALFA=3.141596/4.0
1450  WRITE(10,1420)XALFA
1420  FORMAT('XALFA = ',F10.7)
      EPSX=H1/TAN(XALFA)
      EPSY=H2*TAN(XALFA)
      EPS=H3/SIN(2*XALFA)+H4*SIN(2*XALFA)
      WRITE(10,1430)EPSX,EPSY,EPS
1430  FORMAT('EPSX = ',F10.8,'  EPSY = ',F10.8,'  EPS = ',F10.8)
C
      IF((EPS-EPSX).LT.0.0.AND.(EPS-EPSY).GT.0.0)THEN
       PRINT *,' Diagonal tension angle cannot be calculated !'
       GOTO 3001
      ELSE IF((EPS-EPSX).GT.0.0.AND.(EPS-EPSY).LT.0.0)THEN
       PRINT *,' Diagonal tension angle cannot be calculated !'
```

University of Pretoria etd – Viljoen, A (2004)


```
       GOTO 3001
      ELSE IF((EPS-EPSY).EQ.0.0)THEN
       PRINT *,' Diagonal tension angle cannot be calculated !'
       GOTO 3001
      ENDIF
      ALFA=ATAN(((EPS-EPSX)/(EPS-EPSY))**0.5)
      ERR=(ABS(ALFA-XALFA))*100.0
      WRITE(10,1440)ALFA,ERR
1440  FORMAT('ALFA = ',F10.7,'  % ERR = ',F6.3)
      IF(ERR.GE.0.1)THEN
      XALFA=ALFA
      GOTO 1450
      ENDIF
      WRITE(10,1460)I,ALFA*180.0/3.141596
1460  FORMAT(/,'   Diagonal Tension Angle for Panel',I3,' = ',F6.3,
     +' degrees !',/)
C
      SXDT(I)=K*ABS(SXY)/TAN(ALFA)
      SYDT(I)=K*ABS(SXY)*TAN(ALFA)
      WRITE(10,1500)SXDT(I),SYDT(I)
1500  FORMAT('SXDT(I) = ',F18.3,'   SYDT(I) = ',F18.3)
C
C             Tests 1 to 5 in the Grisham algorithm
C
      BBETAX=0.0
      BBETAY=0.0
C
2400  EXDT=(-1.0/EP)*(EP*LY*T(I)/2.0/AX/EXB+1)*(SXDT(I)-(1-BBETAX)*
     +SXDTI(I)-POISS*(SYDT(I)-(1-BBETAX)*SYDTI(I)))
      EYDT=(-1.0/EP)*(EP*LX*T(I)/AY/EYB+1)*(SYDT(I)-(1-BBETAY)*
     +SYDTI(I)-POISS*(SXDT(I)-(1-BBETAY)*SXDTI(I)))
      WRITE(10,2540)EXDT,EYDT
2540  FORMAT('EXDT = ',F13.10,'  EYDT = ',F13.10)
C
      THETA=(1.0/(2*(1.0+POISS)))*(4.0/(SIN(2*ALFA))**2+
     +(EP*LX*T(I)*(TAN(ALFA))**2)/(EYB*(AY+(0.5*LX*T(I))*(1-K)))+
     +(EP*LY*T(I)/(TAN(ALFA))**2)/(EXB*(2.0*AX+(0.5*LY*T(I))*(1-K))))
      EXYDT=K*(THETA-1)*SXY/GP
      WRITE(10,2560)THETA,EXYDT
2560  FORMAT('THETA = ',F10.6,'  EXYDT = ',F13.10)
C
      C1=1+LY*T(I)*EP*BETAX(I)/2.0/AX/EXB
      C2=1+LX*T(I)*EP*BETAY(I)/AY/EYB
      C3=1-BETAX(I)
      C4=1-BETAY(I)
```

University of Pretoria etd – Viljoen, A (2004)


```
      WRITE(10,2580)C1,C2,C3,C4,BETAX(I),BETAY(I)
2580  FORMAT('C1 = ',F10.6,'   C2 = ',F10.6,'   C3 = ',F10.6,
     +'   C4 = ',F10.6,/,'BETAX(I) = ',F10.6,'   BETAY(I) = ',F10.6)
C
      SXC=(C2*C3*(SX-SXCR)*L+BETAX(I)*POISS*C4*(SY-SYCR)*L)
     +/C1/C2
      SYC=(C1*C4*(SY-SYCR)*L+BETAY(I)*POISS*C3*(SX-SXCR)*L)
     +/C1/C2
      WRITE(10,2600)SXC,SYC
2600  FORMAT('SXC = ',F18.3,'  SYC = ',F18.3)
C
      EXC=(-1.0/EP)*(1+LY*T(I)*EP/2.0/AX/EXB)*(SXC-POISS*SYC)*
     +(ABS(BETAX(I)+BETAX(I)*BETAY(I)-1))
      EYC=(-1.0/EP)*(1+LX*T(I)*EP/AY/EYB)*(SYC-POISS*SXC)*
     +(ABS(BETAY(I)+BETAX(I)*BETAY(I)-1))
      WRITE(10,2620)EXC,EYC
2620  FORMAT('EXC = ',F13.10,'   EYC = ',F13.10)
C
      EXT(I)=(1-BBETAX)*EXT(I)+EXDT+EXC
      EYT(I)=(1-BBETAY)*EYT(I)+EYDT+EYC
      EXYT=EXYDT
      WRITE(10,2640)EXT(I),EYT(I),EXYT
2640  FORMAT('EXT = ',F13.10,'   EYT = ',F13.10,'   EXYT = 'F13.10)
C
      SXPS(I)=-EP*(EXT(I)+POISS*EYT(I))/(1-POISS**2)
      SYPS(I)=-EP*(EYT(I)+POISS*EXT(I))/(1-POISS**2)
      SXYPS(I)=-EP*EXYT/(2*(1+POISS))
C
      WRITE(10,2660)SXPS(I),SYPS(I),SXYPS(I)
2660  FORMAT('SXPS(I) = ',F18.3,'   SYPS(I) = ',F18.3,
     +'   SXYPS(I) = ',F18.3)
      ENDIF
C
C         calculate convergence parameters
C
      WRITE(10,2680)SXC,SXCR,SYC,SYCR
2680  FORMAT('SXC = ',F18.3,'   SXCR = ',F18.3,/,
     +'SYC = ',F18.3,'   SYCR = ',F18.3)
      IF(ABS(SXCR).LT.1.0E-5)THEN
      WRITE(21,2682)
2682  FORMAT('  K1X  --> INF ;   K1XP  --> 0')
      ELSE IF(ABS(SXC).LT.1.0E-5)THEN
      WRITE(21,2684)
2684  FORMAT('  K1X  --> 0 ;   K1XP  --> INF')
      ELSE
```

University of Pretoria etd – Viljoen, A (2004)


```
      K1X=SXC/SXCR
      K1XP=(K1X-K1XN(I))*100.0/K1X
      K1XPP=(K1XN(I)-K1XNN(I))*100.0/K1XN(I)
      WRITE(10,2687)K1X,K1XP,K1XPP
2687  FORMAT('K1X = ',F10.3,5X,'K1XP = ',F10.3,5X,'K1XPP = ',
     +F10.3)
      ENDIF
C
      IF(ABS(SYCR).LT.1.0E-5)THEN
      WRITE(21,2686)
2686  FORMAT('  K1Y  --> INF ;   K1YP  --> 0')
      ELSE IF(ABS(SYC).LT.1.0E-5)THEN
      WRITE(21,2688)
2688  FORMAT('  K1Y  --> 0 ;   K1YP  --> INF')
      ELSE
      K1Y=SYC/SYCR
      K1YP=(K1Y-K1YN(I))*100.0/K1Y
      K1YPP=(K1YN(I)-K1YNN(I))*100.0/K1YN(I)
      WRITE(10,2689)K1Y,K1YP,K1YPP
2689  FORMAT('K1Y = ',F10.3,5X,'K1YP = ',F10.3,5X,'K1XPP = ',
     +F10.3)
      ENDIF
C
      FN=I+30
      WRITE(FN,2662)IBC,ITN,K1X,SXC,SXCR,SXYCR,SXY,SX-SXCR,SY-SYCR,
     +SX,SXAV,SXDT(I)
2662  FORMAT(2I3,10G12.5)
C
      FN2=I+40
      WRITE(FN2,2663)IBC,ITN,K1Y,SYC,SYCR,SXYCR,SXY,SY-SYCR,SX-SXCR,
     +SY,SYAV,SYDT(I)
2663  FORMAT(2I3,10G12.5)
C
2704  WRITE(10,2700)SXDT(I),SXDTI(I),SYDT(I),SYDTI(I)
2700  FORMAT('SXDT(I) = ',F18.3,'   SXDTI(I) = ',F18.3,/,
     +'SYDT(I) = ',F18.3,'   SYDTI(I) = ',F18.3)
      K2X=SXDT(I)-SXDTI(I)
      K2Y=SYDT(I)-SYDTI(I)
      K2XP=(K2X/SXDT(I))*100.0
      K2YP=(K2Y/SYDT(I))*100.0
      K2XPP=(K2XN(I)/SXDTI(I))*100.0
      K2YPP=(K2YN(I)/SYDTI(I))*100.0
      K2XPPP=(K2XNN(I)/SXDTII(I))*100.0
      K2YPPP=(K2YNN(I)/SYDTII(I))*100.0
      WRITE(10,2710)I,K1X,K1XP,K1Y,K1YP,K1XPP,K1YPP,
```

University of Pretoria etd – Viljoen, A (2004)


```
     +K2X,K2XP,K2Y,K2YP,K2XPP,K2YPP,K2XPPP,K2YPPP
      WRITE(21,2710)I,K1X,K1XP,K1Y,K1YP,K1XPP,K1YPP,
     +K2X,K2XP,K2Y,K2YP,K2XPP,K2YPP,K2XPPP,K2YPPP
2710  FORMAT('Panel No ',I2,/,'K1X = ',F18.3,'(',F10.3,' %)',
     +5X,'K1Y = ',F18.3,' (',F10.3,' %)',/,24X,' (',F10.3,' %)',
     +29X,' (',F10.3,' %)',/,
     +'K2X = ',F18.3,'(',F10.3,' %)',5X,'K2Y = ',
     +F18.3,' (',F10.3,' %)',/,24X,' (',F10.3,' %)',29X,
     +' (',F10.3,' %)',/,24X,' (',F10.3,' %)',29X,
     +' (',F10.3,' %)')
C
      SXCRI(I)=SXCR
      SYCRI(I)=SYCR
      SXDTII(I)=SXDTI(I)
      SYDTII(I)=SYDTI(I)
      SXDTI(I)=SXDT(I)
      SYDTI(I)=SYDT(I)
      WRITE(10,2720)SXCRI(I),SYCRI(I),SXDTI(I),SYDTI(I),
     +SXDTII(I),SYDTII(I)
2720  FORMAT('SXCRI(I) = ',F18.3,'     SYCRI(I) = ',F18.3,/,
     +'SXDTI(I) = ',F18.3,'     SYDTI(I) = ',F18.3,/,
     +'SXDTII(I) = ',F18.3,'     SYDTII(I) = ',F18.3)
      K1XNN(I)=K1XN(I)
      K1YNN(I)=K1YN(I)
      K1XN(I)=K1X
      K1YN(I)=K1Y
      K2XNN(I)=K2XN(I)
      K2YNN(I)=K2YN(I)
      K2XN(I)=K2X
      K2YN(I)=K2Y
C
      WRITE(10,2726)K1XNN(I),K1YNN(I),K1XN(I),K1YN(I)
2726  FORMAT('K1XNN(I) = ',F10.3,5X,'K1YNN(I) = ',F10.3,/,
     +'K1XN(I) = ',F10.3,5X,'K1YN(I) = ',F10.3)
      WRITE(10,2728)K2XNN(I),K2YNN(I),K2XN(I),K2YN(I)
2728  FORMAT('K2XNN(I) = ',F18.3,5X,'K2YNN(I) = ',F18.3,/,
     +'K2XN(I) = ',F18.3,5X,'K2YN(I) = ',F18.3)
C
C         Test for convergence:
C
      IF(ITN.GE.3)THEN
      IF(ABS(K1XP).LT.5.0.AND.ABS(K1XPP).LT.5.0)THEN
      WRITE(13,2730)K1X,I,ITN
2730    FORMAT(4X,'K1X converges to',F10.3,' for plate ',I3,
     +  ' !! (ITN=',I3,')')
```

University of Pretoria etd – Viljoen, A (2004)


```
        K1XVAL(I)=K1X
       CONA(I)=1
       ELSE IF(CONA(I).NE.1)THEN
        K1XNC(I)=K1X
        K1XID(I)=1
        WRITE(13,2735)K1XNC(I),K1XID(I),I
2735    FORMAT('K1XNC(I) =',F15.7,'    K1XID(I) =',I3,'    I =',I3)
       ENDIF
       IF(ABS(K1YP).LT.5.0.AND.ABS(K1YPP).LT.5.0)THEN
       WRITE(13,2740)K1Y,I,ITN
2740     FORMAT(4X,'K1Y converges to',F10.3,' for plate ',I3,
     +  ' !! (ITN=',I3,')')
         K1YVAL(I)=K1Y
       CONB(I)=1
       ELSE IF(CONB(I).NE.1)THEN
        K1YNC(I)=K1Y
        K1YID(I)=1
        WRITE(13,2745)K1YNC(I),K1YID(I),I
2745    FORMAT('K1YNC(I) =',F15.7,'    K1YID(I) =',I3,'     I =',I3)
       ENDIF
       IF(ABS(K2XP).LT.2.0.AND.ABS(K2XPP).LT.2.0.AND.ABS(K2XPPP).
     +LT.2.0)THEN
       WRITE(13,2780)I,ITN
2780     FORMAT(4X,'K2X converges for plate ',I3,' !! (ITN=',
     +  I3,')')
       CONC(I)=1
       ENDIF
       IF(ABS(K2YP).LT.2.0.AND.ABS(K2YPP).LT.2.0.AND.ABS(K2YPPP).
     +LT.2.0)THEN
       WRITE(13,2790)I,ITN
2790     FORMAT(4X,'K2Y converges for plate ',I3,' !! (ITN=',
     +  I3,')')
       COND(I)=1
       ENDIF
       ENDIF
         CONAT=CONAT+CONA(I)
         CONBT=CONBT+CONB(I)
         CONCT=CONCT+CONC(I)
         CONDT=CONDT+COND(I)
3002   CONTINUE
C
       IF(CONAT+CONBT+CONCT+CONDT.EQ.4*NP)THEN
        NOIT=ITN
        WRITE(13,3003)NOIT
3003    FORMAT('NOIT = ',I3)
```

University of Pretoria etd – Viljoen, A (2004)


```
       GOTO 3006
      ELSE
       NOIT=ITN
      ENDIF
C
3000  CONTINUE
C
3006  WRITE(13,3007)ITN
3007  FORMAT('ITN = ',I5)
      BCTOT=0
      DO 3005 I=1,NP
       BCONA(I)=0
       BCONB(I)=0
3005  CONTINUE
C
      IF(IBC.EQ.IBCMAX)THEN
       WRITE(13,3206)IBC
3206   FORMAT('Maximum number of iterations reached - ',I3,' - still',
     + ' no convergence !')
       GOTO 3001
      ENDIF
C
      DO 3051 I=1,NP
C
C          x-BETA values
C
C          modify the BETAs when the solution does not buckle
C
       IF(BPL(I).EQ.NOIT)THEN
       WRITE(13,3011)I,BETAX(I)
3011   FORMAT('Panel no',I3,' converges to an unbuckled state !',/,
     + 'The BETAX value will therefore remain as is:',/,
     + 'BETAX(I) = ',F10.7)
       ELSE IF(BPL(I).LT.NOIT.AND.BPL(I).GT.0.AND.CONA(I).EQ.0)THEN
       WRITE(13,3012)I,BPL(I)
3012    FORMAT('Panel number',I3,'  buckled less than 12 times (BPL=',
     +  I3,');',/,'probably does not converge to an unbuckled state !')
       WRITE(13,3013)I,BETAX(I)
3013    FORMAT('Old BETAX(I) value for panel no',I3,' is:',F10.7)
        BETAX(I)=BETAX(I)+0.02
       IF(BETAX(I).GE.1.0)BETAX(I)=0.001
       WRITE(13,3014)I,BETAX(I)
3014    FORMAT('New BETAX(I) value for panel no',I3,' is:',F10.7)
C
C          modify the BETAs when the solution does not converge
```

University of Pretoria etd – Viljoen, A (2004)


```
C
      ELSE IF(K1XID(I).EQ.1.AND.K1XNC(I).LT.1.0)THEN
      WRITE(13,3050)I,BETAX(I)
3050   FORMAT('K1X of panel no',I3,' did not converge !',/,
     +  'Old BETAX(I) value for panel',I3,' is: ',F10.7)
      BETAX(I)=BETAX(I)-0.016
      IF(BETAX(I).LE.0.0)BETAX(I)=0.001
      WRITE(13,3055)I,BETAX(I)
3055   FORMAT('New BETAX(I) value for panel',I3,' is: ',F10.7)
      ELSE IF(K1XID(I).EQ.1.AND.K1XNC(I).GT.1.0)THEN
      WRITE(13,3060)I,I,BETAX(I)
3060   FORMAT('K1X of panel no',I3,' did not converge !',/,
     +  'Old BETAX(I) value for panel',I3,' is: ',F10.7)
      BETAX(I)=BETAX(I)+0.02
      IF(BETAX(I).GE.1.0)BETAX(I)=0.999
      WRITE(13,3065)I,BETAX(I)
3065   FORMAT('New BETAX(I) value for panel',I3,' is: ',F10.7)
C
C          modify the BETAs when the solution converges
C
      ELSE IF(CONA(I).EQ.1.AND.K1XVAL(I).LT.0.8)THEN
      WRITE(13,3015)I,BETAX(I)
3015   FORMAT('Old BETAX(I) value for panel',I3,' is: ',F10.7)
      BETAX(I)=BETAX(I)-0.02
       WRITE(13,3020)I,BETAX(I)
3020   FORMAT('New BETAX(I) value for panel',I3,' is: ',F10.7)
      ELSE IF(CONA(I).EQ.1.AND.K1XVAL(I).GE.0.8.AND.
     + K1XVAL(I).LT.0.95)THEN
      WRITE(13,3021)I,BETAX(I)
3021   FORMAT('Old BETAX(I) value for panel',I3,' is: ',F10.7)
      BETAX(I)=BETAX(I)-0.002
       WRITE(13,3022)I,BETAX(I)
3022   FORMAT('New BETAX(I) value for panel',I3,' is: ',F10.7)
      ELSE IF(CONA(I).EQ.1.AND.K1XVAL(I).GT.1.2)THEN
       WRITE(13,3025)I,BETAX(I)
3025   FORMAT('Old BETAX(I) value for panel',I3,' is: ',F10.7)
      BETAX(I)=BETAX(I)+0.02
       WRITE(13,3030)I,BETAX(I)
3030   FORMAT('New BETAX(I) value for panel',I3,' is: ',F10.7)
      ELSE IF(CONA(I).EQ.1.AND.K1XVAL(I).GT.1.05.AND.
     + K1XVAL(I).LE.1.2)THEN
       WRITE(13,3031)I,BETAX(I)
3031   FORMAT('Old BETAX(I) value for panel',I3,' is: ',F10.7)
      BETAX(I)=BETAX(I)+0.002
       WRITE(13,3032)I,BETAX(I)
```

University of Pretoria etd – Viljoen, A (2004)


```
3032    FORMAT('New BETAX(I) value for panel',I3,' is: ',F10.7)
       ELSE IF(CONA(I).EQ.1.AND.K1XVAL(I).GE.0.95.AND.
     + K1XVAL(I).LE.1.05)THEN
        BCONA(I)=1
        WRITE(13,3040)I,BETAX(I),BCONA(I)
3040    FORMAT('BETAX(I) is the correct value for K1X to converge',
     +  ' for plate ',I3,'(BETAX(I) = ',F10.7,')',/,'BCONA = ',I3)
C
C          modify BETAs when none of the above apply
C
       ELSE
        WRITE(13,3041)I,BETAX(I)
3041    FORMAT('Default change - Old BETAX(I) value for panel',I3,
     +  ' is: ',F10.7)
        BETAX(I)=BETAX(I)+0.03
         IF(BETAX(I).GE.1.0)BETAX(I)=0.001
        WRITE(13,3044)I,BETAX(I)
3044    FORMAT('Default change - New BETAX(I) value for panel',I3,
     +  ' is: ',F10.7)
       ENDIF
C
C         y-BETA values
C
C         modify the BETAs when the solution does not buckle
C
       IF(BPL(I).EQ.NOIT)THEN
       WRITE(13,2791)I,BETAY(I)
2791   FORMAT('Panel no',I3,' converges to an unbuckled state !',/,
     + 'The BETAY value will therefore remain as is:',/,
     + 'BETAY(I) = ',F10.7)
        BTOT=BTOT+1
       BPID(I)=1
       ELSE IF(BPL(I).LT.NOIT.AND.BPL(I).GT.0.AND.CONB(I).EQ.0)THEN
        WRITE(13,3128)I,BPL(I)
3128    FORMAT('Panel number',I3,'  buckled less than 12 times (BPL=',
     +  I3,');',/,'probably does not converge to an unbuckled state !')
         WRITE(13,3129)I,BETAY(I)
3129     FORMAT('Old BETAY(I) value for panel no',I3,' is:',F10.7)
       BETAY(I)=BETAY(I)+0.02
       IF(BETAY(I).GE.1.0)BETAY(I)=0.001
         WRITE(13,3131)I,BETAY(I)
3131     FORMAT('New BETAY(I) value for panel no',I3,' is:',F10.7)
C
C         modify BETAs when solution does not converge
C
```

University of Pretoria etd – Viljoen, A (2004)


```
      ELSE IF(K1YID(I).EQ.1.AND.K1YNC(I).LT.1.0)THEN
       WRITE(13,3110)I,I,BETAY(I)
3110   FORMAT('K1Y of panel no',I3,' did not converge !',/,
     +  'Old BETAY(I) value for panel',I3,' is: ',F10.7)
      BETAY(I)=BETAY(I)-0.016
      IF(BETAY(I).LE.0.0)BETAY(I)=0.001
       WRITE(13,3115)I,BETAY(I)
3115   FORMAT('New BETAY(I) value for panel',I3,' is: ',F10.7)
      ELSE IF(K1YID(I).EQ.1.AND.K1YNC(I).GT.1.0)THEN
       WRITE(13,3120)I,I,BETAY(I)
3120   FORMAT('K1Y of panel no',I3,' did not converge !',/,
     +  'Old BETAY(I) value for panel',I3,' is: ',F10.7)
      BETAY(I)=BETAY(I)+0.02
      IF(BETAY(I).GE.1.0)BETAY(I)=0.999
       WRITE(13,3125)I,BETAY(I)
3125   FORMAT('New BETAY(I) value for panel',I3,' is: ',F10.7)
C
C          modify BETAs when solution converges
C
      ELSE IF(CONB(I).EQ.1.AND.K1YVAL(I).LT.0.8)THEN
       WRITE(13,3075)I,BETAY(I)
3075   FORMAT('Old BETAY(I) value for panel',I3,' is: ',F10.7)
      BETAY(I)=BETAY(I)-0.02
       WRITE(13,3080)I,BETAY(I)
3080   FORMAT('New BETAY(I) value for panel',I3,' is: ',F10.7)
      ELSE IF(CONB(I).EQ.1.AND.K1YVAL(I).GE.0.8.AND.
     + K1YVAL(I).LT.0.95)THEN
       WRITE(13,3081)I,BETAY(I)
3081   FORMAT('Old BETAY(I) value for panel',I3,' is: ',F10.7)
      BETAY(I)=BETAY(I)-0.002
       WRITE(13,3082)I,BETAY(I)
3082   FORMAT('New BETAY(I) value for panel',I3,' is: ',F10.7)
      ELSE IF(CONB(I).EQ.1.AND.K1YVAL(I).GT.1.2)THEN
       WRITE(13,3085)I,BETAY(I)
3085   FORMAT('Old BETAY(I) value for panel',I3,' is: ',F10.7)
      BETAY(I)=BETAY(I)+0.02
       WRITE(13,3090)I,BETAY(I)
3090   FORMAT('New BETAY(I) value for panel',I3,' is: ',F10.7)
      ELSE IF(CONB(I).EQ.1.AND.K1YVAL(I).GT.1.05.AND.
     + K1YVAL(I).LE.1.2)THEN
       WRITE(13,3091)I,BETAY(I)
3091   FORMAT('Old BETAY(I) value for panel',I3,' is: ',F10.7)
      BETAY(I)=BETAY(I)+0.002
       WRITE(13,3092)I,BETAY(I)
3092   FORMAT('New BETAY(I) value for panel',I3,' is: ',F10.7)
```

University of Pretoria etd – Viljoen, A (2004)


```
       ELSE IF(CONB(I).EQ.1.AND.K1YVAL(I).GE.0.95.AND.
     + K1YVAL(I).LE.1.05)THEN
        BCONB(I)=1
        WRITE(13,3100)I,BETAY(I),BCONB(I)
3100    FORMAT('BETAY(I) is the correct value for K1Y to converge',
     +  ' for panel ',I3,'(BETAY(I) = ',F10.7,')',/,'BCONB = ',I3)
C
C          modify BETAs when none of the above apply
C
       ELSE
        WRITE(13,3101)I,BETAY(I)
3101    FORMAT('Default change - Old BETAY(I) value for panel',I3,
     +  ' is: ',F10.7)
        BETAY(I)=BETAY(I)+0.03
         IF(BETAY(I).GE.1.0)BETAY(I)=0.001
        WRITE(13,3104)I,BETAY(I)
3104    FORMAT('Default change - New BETAY(I) value for panel',I3,
     +  ' is: ',F10.7)
       ENDIF
      BCTOT=BCTOT+BCONA(I)+BCONB(I)
      WRITE(13,3180)BTOT,BCTOT
3180  FORMAT('BTOT = ',I4,'     BCTOT = ',I4)
3051  CONTINUE
C
      IF(BTOT.NE.NP.AND.BCTOT.EQ.2*(NP-BTOT))THEN
       WRITE(13,3512)
3512   FORMAT('BETAs adjusted successfully - all no 1 convergence',
     + ' criteria is satisfied !')
       IF(CONCT.EQ.NP)THEN
       WRITE(13,3513)
3513   FORMAT('The Diagonal tension stress requirement in the x-dir',/,
     + 'is satisfied (no 2 requirement)')
       ELSE
       WRITE(13,3514)
3514   FORMAT('Requirement no 2 is not satisfied in x-dir !')
       ENDIF
       IF(CONDT.EQ.NP)THEN
       WRITE(13,3516)
3516   FORMAT('The diagonal tension stress requirement in the y-dir',/,
     +  'is satisfied (no 2 requirement)')
       ELSE
       WRITE(13,3517)
3517   FORMAT('Requirement no 2 is not satisfied in y-dir !')
       ENDIF
       GOTO 3001
```

University of Pretoria etd – Viljoen, A (2004)


```
      ELSE IF(BTOT.EQ.NP)THEN
       WRITE(13,3518)
3518   FORMAT('No panels buckled !')
       GOTO 3001
      ENDIF
C
3500  CONTINUE
C
      CALL TIME(SEC)
      TSTOP=SEC
      WRITE(10,3505)TSTOP
3505  FORMAT('TSTOP = ',I20)
C
      CLOSE(10)
      CLOSE(20)
      CLOSE(21)
      CLOSE(13)
      CLOSE(12)
      CLOSE(31)
      CLOSE(32)
      CLOSE(33)
      CLOSE(34)
      CLOSE(35)
      CLOSE(36)
      CLOSE(41)
      CLOSE(42)
      CLOSE(43)
      CLOSE(44)
      CLOSE(45)
      CLOSE(46)
C
      STOP
      END




C*****************************************************************
C
      SUBROUTINE FNODES(NPX,NPY,NEX,NEY,C20,LTOTX,LTOTY,UPRE,
     +SENODE,SNALL,SNCX,SNCY,SNCZ,SNTOT,SNTFLAT)
C
C*****************************************************************
C            Subroutine to generate all nodes for the
```

University of Pretoria etd – Viljoen, A (2004)


```
C            finite element model
C*****************************************************************
C
      DOUBLE PRECISION LTOTX,LTOTY,LPNX,LPNY,
     +SNCX(1000),SNCY(1000),SNCZ(1000),SENODE(301)
      INTEGER YINC,SNALL(1000),C20,NPX,NPY,NEX,NEY,SNTOT,SNTFLAT
      CHARACTER*(3)UPRE
C
      OPEN(20,FILE='nodedata.dat')
C
      NNXMAX=C20*NPX*NEX+1
      NONX=NNXMAX
      NONY=C20*NPY*NEY+1
      NIPPX=C20*NEX
      NBEPP=2*NEX+2*NEY
      NSEPP=NEX*NEY
      LPNX=LTOTX/(NONX-1)
      LPNY=LTOTY/(NONY-1)
      NINCY=NNXMAX
      NNYMAX=C20*NPY*NEY+1
      NTOTAL=NNXMAX*NNYMAX
      NIPPY=C20*NEY*NINCY
       WRITE(10,30)NNXMAX,NNYMAX,NONX,NONY,NIPPX,NIPPY,
     + NBEPP,NSEPP,LPNX,LPNY
30     FORMAT('NNXMAX = ',I5,/,'NNYMAX = ',I5,/,
     + 'NONX = ',I5,/,'NONY = ',I5,/,'NIPPX = ',I5,/,'NIPPY = ',I5,
     + /,'NBEPP = ',I5,/,'NSEPP = ',I5,/,'LPNX = ',F10.4,/,
     + 'LPNY = ',F10.4)
C
C                generate nodes in one plane
C
      NCOUNT=0
      YINC=1
       DO 120 J=1,NONY
       DO 100 I=1,NONX
       K=1
         IF(J.EQ.1)THEN
          NCOUNT=NCOUNT+1
          SNALL(NCOUNT)=NCOUNT
          SNCX(NCOUNT)=0.0+LPNX*(I-1)
          SNCY(NCOUNT)=0.0
          SNCZ(NCOUNT)=0.0
          WRITE(20,101)SNALL(NCOUNT),I,J,NCOUNT,YINC,SNCX(NCOUNT),
     +    SNCY(NCOUNT),SNCZ(NCOUNT)
101       FORMAT('SNALL(NCOUNT) = ',I4,'  I = ',I4,'  J = ',I4,
```

University of Pretoria etd – Viljoen, A (2004)


```
     +    '  NCOUNT = ',I4,'  YINC = ',I4,'  SNCX(NCOUNT) = ',F10.4,
     +    '  SNCY(NCOUNT) = ',F10.4,'   SNCZ(NCOUNT) = ',F10.4)
          GOTO 100
         ELSE
          NCOUNT=NCOUNT+1
          SNALL(NCOUNT)=NCOUNT
          SNCX(NCOUNT)=0.0+LPNX*(I-1)
          SNCY(NCOUNT)=SNCY(NCOUNT)+(J-1)*LPNY
          SNCZ(NCOUNT)=0.0
          WRITE(20,102)SNALL(NCOUNT),I,J,NCOUNT,YINC,SNCX(NCOUNT),
     +    SNCY(NCOUNT),SNCZ(NCOUNT)
102       FORMAT('SNALL(NCOUNT) = ',I4,'  I = ',I4,'  J = ',I4,
     +    '  NCOUNT = ',I4,'  YINC = ',I4,'  SNCX(NCOUNT) = ',F10.4,
     +    '  SNCY(NCOUNT) = ',F10.4,'   SNCZ(NCOUNT) = ',F10.4)
         ENDIF
100    CONTINUE
       YINC=NINCY
120   CONTINUE
      SNTFLAT=NCOUNT
C
C         generate nodes in second plane for upright eccentricity
C
      IF(UPRE.EQ.'ECC')THEN
      YINC=1
       DO 115 J=1,NONY
        NUPRC=0
       DO 110 I=1,NONX,C2O*NEX
          NUPRC=NUPRC+1
         IF(J.EQ.1)THEN
          NCOUNT=NCOUNT+1
          SNALL(NCOUNT)=NCOUNT
          SNCX(NCOUNT)=0.0+LPNX*(I-1)
          SNCY(NCOUNT)=0.0
          SNCZ(NCOUNT)=SENODE(NUPRC)
          WRITE(20,105)SNALL(NCOUNT),I,J,NCOUNT,YINC,SNCX(NCOUNT),
     +    SNCY(NCOUNT),SNCZ(NCOUNT)
105       FORMAT('SNALL(NCOUNT) = ',I4,'  I = ',I4,'  J = ',I4,
     +    '  NCOUNT = ',I4,'  YINC = ',I4,'  SNCX(NCOUNT) = ',F10.4,
     +    '  SNCY(NCOUNT) = ',F10.4,'   SNCZ(NCOUNT) = ',F10.4)
          GOTO 110
         ELSE
          NCOUNT=NCOUNT+1
          SNALL(NCOUNT)=NCOUNT
          SNCX(NCOUNT)=0.0+LPNX*(I-1)
          SNCY(NCOUNT)=SNCY(NCOUNT)+(J-1)*LPNY
```

University of Pretoria etd – Viljoen, A (2004)


```
          SNCZ(NCOUNT)=SENODE(NUPRC)
          WRITE(20,106)SNALL(NCOUNT),I,J,NCOUNT,YINC,SNCX(NCOUNT),
     +    SNCY(NCOUNT),SNCZ(NCOUNT)
106       FORMAT('SNALL(NCOUNT) = ',I4,'  I = ',I4,'  J = ',I4,
     +    '  NCOUNT = ',I4,'  YINC = ',I4,'  SNCX(NCOUNT) = ',F10.4,
     +    '  SNCY(NCOUNT) = ',F10.4,'   SNCZ(NCOUNT) = ',F10.4)
         ENDIF
110    CONTINUE
        YINC=NINCY
115    CONTINUE
       SNTOT=NCOUNT
       PRINT *,'SNTOT = ',SNTOT
      ELSE
       SNTOT=NCOUNT
       PRINT *,'SNTOT = ',SNTOT
      ENDIF
C
       CLOSE(20)
       RETURN
       END




C*****************************************************************
C
      SUBROUTINE ELEMENTS(NPX,NPY,NEX,NEY,C2O,UPRE,NALL,
     +SELS,SELBH,SELBV,SELBVN,SELBHN,SELSN,SPNLSE,
     +ICSS,ICBHS,ICBVS,NTFLAT)
C
C*****************************************************************
C           Subroutine to generate all elements for the
C           finite element model
C*****************************************************************
C
      INTEGER NALL(1000),NPX,NPY,NEX,NEY,C2O,NNXMAX,
     +SELS(1000,1000),SELBH(1000,1000),SELBV(1000,1000),
     +SELBVN(1000),SELBHN(1000),SELSN(1000),SPNLSE(1000,20),
     +ICSS,ICBHS,ICBVS,NTFLAT
      CHARACTER*(3) UPRE
      PRINT *,'NPX =',NPX
      PRINT *,'NPY =',NPY
      PRINT *,'NEX =',NEX
      PRINT *,'NEY =',NEY
```

University of Pretoria etd – Viljoen, A (2004)


```
      PRINT *,'C20 =',C20
      PRINT *,'UPRE =',UPRE
C
      OPEN(20,FILE='element-data.dat')
C
      NNXMAX=C20*NPX*NEX+1
      NONX=NNXMAX
      NONY=C20*NPY*NEY+1
C
C           generate the shell elements
C
      ICSS=0
      NINC=0
        DO 169 I=1,NEY*NPY,1
        NINC=(I-1)*NONX*C20
        DO 171 NCOUNT=1+NINC,(NONX-C20)+NINC,C20
          ICSS=ICSS+1
          SELSN(ICSS)=ICSS
          SELS(ICSS,1)=NALL(NCOUNT)
          SELS(ICSS,2)=NALL(NCOUNT+C20)
          SELS(ICSS,3)=NALL(NCOUNT+NONX*C20+C20)
          SELS(ICSS,4)=NALL(NCOUNT+NONX*C20)
          SELS(ICSS,5)=NALL(NCOUNT+C20/2)
          SELS(ICSS,6)=NALL(NCOUNT+NONX+C20)
          SELS(ICSS,7)=NALL(NCOUNT+NONX*C20+C20/2)
          SELS(ICSS,8)=NALL(NCOUNT+NONX)
        WRITE(20,181)ICSS,SELS(ICSS,1),ICSS,SELS(ICSS,2),ICSS,
     +  SELS(ICSS,3),
     +  ICSS,SELS(ICSS,4),ICSS,SELS(ICSS,5),ICSS,SELS(ICSS,6),ICSS,
     +  SELS(ICSS,7),ICSS,SELS(ICSS,8),I
181     FORMAT('SELS(',I3,',1) = ',I5,'  SELS(',I3,',2) = ',I5,
     +  '  SELS(',I3,',3) = ',I5,'  SELS(',I3,',4) = ',I5,/,
     +  'SELS(',I3,',5) = ',I5,'  SELS(',I3,',6) = ',I5,
     +  '  SELS(',I3,',7) = ',I5,'  SELS(',I3,',8) = ',I5,/,
     +  'I = ',I3)
171     CONTINUE
169     CONTINUE
C
C           create element sets (shells) related to panels/sheets
C
      IC4=0
      IC5=0
      WRITE(20,*)'  '
      DO 530 IL=1,NEY
      IC3=0
```

University of Pretoria etd – Viljoen, A (2004)


```
      DO 505 I=1,NEY*NPY,NEY
       DO 500 J=1,NEX*NPX,NEX
         IC3=IC3+1
         DO 510 IK=1,NEX
          IC4=IK+NEX*(IL-1)
          IC5=IC5+1
          SPNLSE(IC3,IC4)=SELSN(IC5)
          WRITE(20,520)IC3,IC4,SPNLSE(IC3,IC4)
520       FORMAT(1X,'SPNLSE(',I2,',',I2,') = ',I3)
510      CONTINUE
500    CONTINUE
       IC5=IC5+NEX*NPX*(NEY-1)
505   CONTINUE
      IC5=NEX*NPX*IL
530   CONTINUE
C
C           generate the beam elements - horizontal
C
       NINC=0
       ICBHS=0
       ICB=ICSS
       DO 194 I=1,NPY+1,1
       NINC=(I-1)*NONX*C20*NEY
       DO 192 NCOUNT=1+NINC,NONX-C20+NINC,C20
        ICB=ICB+1
        ICBHS=ICBHS+1
          SELBHN(ICBHS)=ICB
          SELBH(ICBHS,1)=NALL(NCOUNT)
          SELBH(ICBHS,2)=NALL(NCOUNT+C20/2)
          SELBH(ICBHS,3)=NALL(NCOUNT+C20)
          WRITE(20,191)ICBHS,SELBHN(ICBHS),ICBHS,SELBH(ICBHS,1),ICBHS,
     +    SELBH(ICBHS,2),ICBHS,SELBH(ICBHS,3),I,J
191       FORMAT('SELBHN(',I3,') = ',I5,'  SELBH(',I3,',1) = ',I5,
     +    '  SELBH(',I3,',2) = ',I5,'  SELBH(',I3,',3) = ',I5,/,
     +    'I = ',I5,'  J = ',I5)
192    CONTINUE
194    CONTINUE
C
C           generate the beam elements - vertical - no eccentricity
C
      IF(UPRE.EQ.'NOE')THEN
      ICBVS=0
      NINC=0
      DO 210 I=1,NPX+1,1
       NINC=(I-1)*NEX*C20
```

University of Pretoria etd – Viljoen, A (2004)


```
      DO 220 NCOUNT=1+NINC,NONX*NEY*C20+NINC,NONX*C20
        ICB=ICB+1
        ICBVS=ICBVS+1
          SELBVN(ICBVS)=ICB
          SELBV(ICBVS,1)=NALL(NCOUNT)
          SELBV(ICBVS,2)=NALL(NCOUNT+NONX)
          SELBV(ICBVS,3)=NALL(NCOUNT+NONX*C20)
          WRITE(20,211)ICBVS,SELBVN(ICBVS),ICBVS,SELBV(ICBVS,1),ICBVS,
     +    SELBV(ICBVS,2),ICBVS,SELBV(ICBVS,3),I,J,ICB
211       FORMAT('SELBVN(',I3,') = ',I5,'  SELBV(',I3,',1) = ',I5,
     +    '  SELBV(',I3,',2) = ',I5,'  SELBV(',I3,',3) = ',I5,/,
     +    'I = ',I5,'  J = ',I5,'  ICB = ',I5)
220    CONTINUE
210    CONTINUE
C
C           generate the beam elements - vertical - with eccentricity
C
      ELSE IF(UPRE.EQ.'ECC')THEN
      ICBVS=0
      NINC=0
      DO 240 I=1,NPX+1,1
       NINC=I-1
      DO 235 NCOUNT=NTFLAT+1+NINC,NTFLAT+NPY*NEY*C20*(NPX+1)+NINC,
     +(NPX+1)*C20
        ICB=ICB+1
        ICBVS=ICBVS+1
          SELBVN(ICBVS)=ICB
          SELBV(ICBVS,1)=NALL(NCOUNT)
          SELBV(ICBVS,2)=NALL(NCOUNT+NPX+1)
          SELBV(ICBVS,3)=NALL(NCOUNT+(NPX+1)*C20)
          WRITE(20,230)ICBVS,SELBVN(ICBVS),ICBVS,SELBV(ICBVS,1),ICBVS,
     +    SELBV(ICBVS,2),ICBVS,SELBV(ICBVS,3),I,J,ICB
230       FORMAT('SELBVN(',I3,') = ',I5,'  SELBV(',I3,',1) = ',I5,
     +    '  SELBV(',I3,',2) = ',I5,'  SELBV(',I3,',3) = ',I5,/,
     +    'I = ',I5,'  J = ',I5,'  ICB = ',I5)
235    CONTINUE
240    CONTINUE
      ENDIF
C
      CLOSE(20)
C
      RETURN
      END
```

University of Pretoria etd – Viljoen, A (2004)


```
C************************************************************
C
      SUBROUTINE FEMINP(NPX,NPY,NEX,NEY,C2O,ICS,ICBH,ICBV,
     +NTFLAT,NTOT,PNLSE,ITN,UPRE,T,SXPS,SYPS,SXYPS,NALL,
     +NCX,NCY,NCZ,ELS,ELSN,ELBHN,ELBH,ELBVN,ELBV)
C
C************************************************************
C          Subroutine to write the ABAQUS input deck
C************************************************************
C
      DOUBLE PRECISION SXPS(301),SYPS(301),SXYPS(301),
     +NCX(1000),NCY(1000),NCZ(1000),T(301)
      INTEGER ITN,ELS(1000,1000),ELSN(1000),ELBHN(1000),
     +ELBH(1000,1000),ELBVN(1000),ELBV(1000,1000),
     +NALL(1000),PNLSE(1000,20),C2O
      CHARACTER*(3)UPRE
      CHARACTER*(1)COMMA
C
      OPEN(30,FILE='femmodel.inp')
C
      COMMA=','
      NP=NPX*NPY
      NSEPP=NEX*NEY
      NONX=C2O*NPX*NEX+1
      NONY=C2O*NPY*NEY+1
C
      WRITE(30,700)
700   FORMAT('*PREPRINT,HISTORY=NO,ECHO=NO,CONTACT=NO',/,
     +'*RESTART,WRITE,FREQ=1')
      IF(ITN.GE.2)THEN
      WRITE(30,705)
705   FORMAT('*INITIAL CONDITIONS,TYPE=STRESS')
      DO 703 I=1,NP
      WRITE(30,704)I,COMMA,SXPS(I),COMMA,SYPS(I),COMMA,
     +SXYPS(I),COMMA
704   FORMAT('E-PL',I3,A1,G20.12,A1,G20.12,A1,G20.12,A1)
703   CONTINUE
      ENDIF
C
C          write nodal coordinates (includes eccentricity nodes)
C
      WRITE(30,706)
706   FORMAT('*NODE,NSET=N-ALL')
```

University of Pretoria etd – Viljoen, A (2004)


```
        DO 266 NCOUNT=1,NTOT
          WRITE(30,FMT=710)NALL(NCOUNT),COMMA,NCX(NCOUNT),COMMA,
     +    NCY(NCOUNT),COMMA,NCZ(NCOUNT)
710       FORMAT(I7,A1,G15.8,A1,G15.8,A1,G15.8)
266     CONTINUE
C
C          generate node sets for BEAM MPC
C
      NINC=0
      WRITE(30,761)
761   FORMAT('*NSET,NSET=N-SUPR')
      DO 764 I=1,NONY,1
       NINC=(I-1)*NONX
      DO 771 NCOUNT=1+NEX*C20+NINC,NONX+NINC,NEX*C20
          WRITE(30,765)NALL(NCOUNT),COMMA
765       FORMAT(I7,A1)
771     CONTINUE
764     CONTINUE
      NINC=0
      WRITE(30,781)
781   FORMAT('*NSET,NSET=N-ECC')
        DO 784 I=1,NONY,1
         NINC=(I-1)*(NPX+1)
        DO 783 NCOUNT=NTFLAT+2+NINC,NTFLAT+NPX+1+NINC,1
        WRITE(30,788)NALL(NCOUNT),COMMA
788       FORMAT(I7,A1)
783     CONTINUE
784     CONTINUE
C
C          write shell elements
C
      WRITE(30,720)
720   FORMAT('*ELEMENT,TYPE=S8R5,ELSET=E-MEMB')
       DO 280 J=1,ICS
         WRITE(30,721)ELSN(J),COMMA,ELS(J,1),COMMA,ELS(J,2),
     +   COMMA,ELS(J,3),COMMA,ELS(J,4),COMMA,ELS(J,5),COMMA,
     +   ELS(J,6),COMMA,ELS(J,7),COMMA,ELS(J,8)
721      FORMAT(I6,A1,I6,A1,I6,A1,I6,A1,I6,A1,I6,A1,I6,A1,I6,A1,I6)
280    CONTINUE
C
C          generate seperate element sets for each panel (shell elements only)
C          seperate panel thicknesses
C          - max no of panels:  999
C
      DO 725 I=1,NP
```

University of Pretoria etd – Viljoen, A (2004)


```
      IF(I.LE.9)THEN
       WRITE(30,754)I
754    FORMAT('*ELSET,ELSET=E-PL',I1)
      ELSE IF(I.GE.10.AND.I.LE.99)THEN
       WRITE(30,752)I
752    FORMAT('*ELSET,ELSET=E-PL',I2)
      ELSE
       WRITE(30,753)I
753    FORMAT('*ELSET,ELSET=E-PL',I3)
      ENDIF
       DO 727 J=1,NSEPP
        WRITE(30,728)PNLSE(I,J),COMMA
728     FORMAT(I7,A1)
727    CONTINUE
725   CONTINUE
C
C         create beam elements
C
      WRITE(30,730)
730   FORMAT('*ELEMENT,TYPE=B32,ELSET=E-BHORI')
        DO 290 J=1,ICBH
          WRITE(30,FMT=731)ELBHN(J),COMMA,ELBH(J,1),COMMA,
     +    ELBH(J,2),COMMA,ELBH(J,3)
731       FORMAT(I6,A1,I6,A1,I6,A1,I6)
290     CONTINUE
      WRITE(30,740)
740   FORMAT('*ELEMENT,TYPE=B32,ELSET=E-BVERT')
        DO 300 J=1,ICBV
          WRITE(30,FMT=741)ELBVN(J),COMMA,ELBV(J,1),COMMA,
     +    ELBV(J,2),COMMA,ELBV(J,3)
741       FORMAT(I6,A1,I6,A1,I6,A1,I6)
300     CONTINUE
C
C         create seperate element sets for each upright so that
C         max no of upright sets = 999
C
      BVTOT=0
      NUPRS=ICBV/NEY
      WRITE(10,801)NUPRS
801   FORMAT('NUPRS = ',I3)
      DO 800 I=1,NUPRS
      IF(I.LE.9)THEN
       WRITE(30,810)I
810    FORMAT('*ELSET,ELSET=E-UP',I1)
      ELSE IF(I.GE.10.AND.I.LE.99)THEN
```

University of Pretoria etd – Viljoen, A (2004)


```
       WRITE(30,811)I
811    FORMAT('*ELSET,ELSET=E-UP',I2)
      ELSE
       WRITE(30,812)I
812    FORMAT('*ELSET,ELSET=E-UP',I3)
      ENDIF
        DO 805 J=1,NEY
         BVTOT=BVTOT+1
         WRITE(30,815)ELBVN(BVTOT),COMMA
815      FORMAT(I6,A1)
805     CONTINUE
800   CONTINUE
C
      WRITE(30,808)
808   FORMAT('*NSET,NSET=N-ENDC,ELSET=E-UP7')
C
C          create seperate element sets for each part of the flange
C          between two uprights; max no of flange sets = 999
C
      BHTOT=0
      NFLS=ICBH/NEX
      WRITE(10,840)NFLS
840   FORMAT('NFLS = ',I3)
      DO 850 I=1,NFLS
      IF(I.LE.9)THEN
       WRITE(30,845)I
845    FORMAT('*ELSET,ELSET=E-FL',I1)
      ELSE IF(I.GE.10.AND.I.LE.99)THEN
       WRITE(30,846)I
846    FORMAT('*ELSET,ELSET=E-FL',I2)
      ELSE
       WRITE(30,847)I
847    FORMAT('*ELSET,ELSET=E-FL',I3)
      ENDIF
        DO 855 J=1,NEX
         BHTOT=BHTOT+1
         WRITE(30,835)ELBHN(BHTOT),COMMA
835      FORMAT(I6,A1)
855     CONTINUE
850   CONTINUE
C
C          create seperate element sets for the upper and lower flange
C
      WRITE(30,742)
742   FORMAT('*ELSET,ELSET=E-BH1')
```

University of Pretoria etd – Viljoen, A (2004)


```
       DO 743 I=1,NEX*NPX
        WRITE(30,745)ELBHN(I),COMMA
745     FORMAT(I6,A1)
743    CONTINUE
      WRITE(30,746)
746   FORMAT('*ELSET,ELSET=E-BH2')
       IF(NEX*NPX.EQ.1)THEN
       IY=2
      ELSE
      IY=NEX*NPX+1
       ENDIF
       DO 747 J=IY,NEX*NPX*(NPY+1)
        WRITE(30,748)ELBHN(J),COMMA
748     FORMAT(I6,A1)
747    CONTINUE
      WRITE(30,813)
813   FORMAT('*NSET,NSET=N-UPFL,ELSET=E-BH2',/,
     +'*NSET,NSET=N-LOWFL,ELSET=E-BH1')
C
C          write shell section properties for all shells
C
      DO 880 I=1,NP
      IF(I.LE.9)THEN
       WRITE(30,882)I,T(I),COMMA
882    FORMAT('*SHELL SECTION,ELSET=E-PL',I1,',MATERIAL=ALU-S',
     +/,G12.7,A1)
      ELSE IF(I.GE.10.AND.I.LE.99)THEN
       WRITE(30,884)I,T(I),COMMA
884    FORMAT('*SHELL SECTION,ELSET=E-PL',I2,',MATERIAL=ALU-S',
     +/,G12.7,A1)
      ELSE
       WRITE(30,886)I,T(I),COMMA
886    FORMAT('*SHELL SECTION,ELSET=E-PL',I3,',MATERIAL=ALU-S',
     +/,G12.7,A1)
      ENDIF
880   CONTINUE
C
      WRITE(30,755)
755   FORMAT('*BEAM SECTION,ELSET=E-BH1,MATERIAL=ALU-B,SECTION=I',/,
     +'0.0055,0.02937,0.0571,0.0,0.00318,0.0,0.00238',/,
     +'0.0,0.0,-1.0',/,
     +'*BEAM SECTION,ELSET=E-BH2,MATERIAL=ALU-B,SECTION=I',/,
     +'0.02952,0.0381,0.0,0.05874,0.0,0.00519,0.00397',/,
     +'0.0,0.0,-1.0',/,
     +'*BEAM SECTION,ELSET=E-BVERT,MATERIAL=ALU-B,SECTION=L',/,
```

University of Pretoria etd – Viljoen, A (2004)


```
     +'0.0254,0.0254,0.003175,0.003175',/,
     +'0.0,0.0,-1.0',/,
     +'*MATERIAL,NAME=ALU-B',/,'*ELASTIC',/,'71.0E9,0.3',/,
     +'*MATERIAL,NAME=ALU-S',/,'*ELASTIC',/,'72.4E9,0.3',/,
     +'*BOUNDARY',/,'1,1,6',/,'38,1,6',/,'75,1,6',/,
     +'112,1,6',/,'149,1,6',/,'186,1,6',/,'223,1,6')
C
      IF(UPRE.EQ.'ECC')THEN
       WRITE(30,FMT=759)
759    FORMAT('*MPC',/,'BEAM,N-SUPR,N-ECC')
      ENDIF
C
      WRITE(30,760)
760   FORMAT('*STEP',/,'*STATIC')
C
      WRITE(30,FMT=762)
C
762   FORMAT('*CLOAD',/,'37,2,8578.3',/,'74,2,8578.3',/,
     +'111,2,8578.3',/,'148,2,8578.3',/,'185,2,8578.3',/,
     +'222,2,8578.3',/,'259,2,8578.3')
C
      WRITE(30,FMT=770)
770   FORMAT('*EL FILE',/,'S,SINV',/,'*NODE FILE',/,'COORD')
      WRITE(30,FMT=780)
780   FORMAT('*EL PRINT,TOTALS=YES',/,'S,MISES',/,'SP',/,'*END STEP')
      ENDFILE(UNIT=30)
C
      CLOSE(30)
C
      RETURN
      END
```