uS

FACULTY OF SCIENCE AND TECHNOLOGY

MASTER THESIS

Study programme / specialisation:
Marine and Subsea Technology

The spring semester, 2022

Open / ~~Confidential~~

Author: Philip Lundberg Jamissen

....
(signature author)

Course coordinator: Professor Yihan Xing

Supervisor(s): Professor Jihan Xing and Jucong Ma

Thesis title: Use of statistical methods and imperfection databank to determine the buckling strength for imperfect cylinders

Credits (ECTS): 30

Keywords: Thin walled cylinder, SST, buckling, imperfections, FEA, ACER method

Pages: 23

+ appendix: 19

Stavanger, 15.06.2022
date/year

Preface

This thesis is written as a masters thesis in the study program of Marine and Subsea Technology at the University of Stavanger in the spring semester of 2022. A project for 5 ECTs on the same topic was done in the autumn semester of 2021. This thesis builds on the same process and method, but mistakes are fixed, and the scope is widened.

It would not have been possible to write this thesis without the guidance from my supervisors or without the help from teachers, family or friends.

I would like to especially thank Yihan Xing and Yucong Ma for their excellent guidance both during the last two semesters.

A special thanks to Arvid Næss for the code to run the ACER method.

Philip L Jamisson
15.06.22

Abstract

Buckling for thin metal shells are not an easy failure mode to design structures for. The large variety of geometries, material properties, and uncertainties has med standards and best practice for designing thin metal shells very conservative with their safety factor. The design is usually based on the theoretical strength for buckling, and then, a knockdown factor, is multiplied with the strength in order to achieve a strong enough structure. For many geometries, this knockdown factor reaches a value of 0.2, therefore reducing the design strength to one fifth of the theoretical strength. Equinor is planning to create a subsea shuttle tanker (SST) in order to transport liquids to and from subsea wells. The outer hull of the SST will not be subjected to large external pressures dew to cargo tanks inside and a flooded inside. However, the shell will be very large (100 meters long and 17 meters in diameter), and must be design to withstand some pressure. This thesis will not look at the effect of stiffeners, as the main goal is to establish a method to design a safe structure that is lighter and cheaper. Using existing studies done on surface imperfections and physical tests (a study done on small nickel cylinders in the 1970's known as the A-shells), this thesis has put together both Final Element Analysis (FEA) and statistical analysis in order to achieve a knockdown factor that is larger than what is standard and best practice today. Many realisations of cylinders with different imperfections are simulated and then the result is fitted to a probability distribution. The best distribution found was the distribution in the Estimation of Extreme Values by the Average Conditional Exceedance Rate (ACER) Method. With this method, the knockdown factor could be increased from 0.238 to 0.823 for the A7-shell and from 0.497 to 0.691 for the SST.

Contents

List of Figures

List of Tables

1   Introduction

This thesis will discuss buckling of thin cylinder structures in order to improve the design of cargo carrying submarines. In this chapter, the method and motivation will descried, as well as the objective with the thesis and the thesis outline.

1.1   Problem introduction

When designing a submarine, one of the most important design considerations is hydrostatic pressure. In order to descent to 100 meters, the external hull must be able to about ten times the atmospheric pressure in order to not collapse. To achieve this, designers must take into account the most important failure modes. For submarines buckling is the failure mode that is going to occur first.

Equinor is planning on making a cargo carrying submarine called a Subsea Shuttle Tanker (SST). The intended use of the SST is to carry fluids between shore and subsea wells. Much of the design considerations of the SST is already discussed by Ma et. al. in [1]. The main element that makes this submarine possible is that the outer shell is flooded, and compartments inside the structure helps with maintaining strength and carrying the cargo. This makes it possible to have the outer shell thin, as it is not exposed to external pressure in a static environment under water.

To make the outer shell as light as possible, it would be beneficial to have the safety factor for buckling as close to the real world as possible. Today's standards are conservative when it comes to buckling. There fore, a statistical analysis of simulated cylinders with imperfections based on real world measurements can be a solution to this problem. Jamissen et. al. [2] showed that there is a possibility that this method can be made good enough that it is possible to make a design based on this method instead of current standards.

1.2   Objectives and Method

Objectives

The main object of this thesis is to check the method described above for buckling in order to achieve a cheaper and lighter design for the SST, the goal is that this method can be used on both failure modes (axial load and external pressure). A secondary object is that this method can be applied to other structures.

The method

The method is quite simple to describe, but to execute everything correctly and to the most realistic way is not a trivial matter. In short, the method in this thesis is to perform FEA analysis on imperfect cylinders, and then use the result of those simulations to perform a probabilistic analysis and determine

the buckling strength. The goal of this method is that it will be less conservative than methods used in standards today, and still be safe enough to rely on for critical structures.

1.3 Outline

It is helpful to have an overview of the thesis before staring to read it.

- **Theory** will introduce all of the methods used in this thesis, however, it will not go down in the details of how it is used in this thesis, but explain the methods in a general matter.

- **Method** will dive down in what is written about in the theory chapter, and will justify assumptions made, and explain what has been done in a more detailed manner.

- **Results** will dive down in what the results from the simulation and the statistical analysis, and discuss how the results should be interpreted and used for the conclusion.

- **Conclusion** will shortly summarise the thesis and briefly explain what was achieved by this thesis.

- **Future work** will explain what some of the problems in this thesis was, and if someone else should continue on this research path, what to do different and what to add.

- **References** is a list of references

- **Appendix** is added to the end of the thesis and contains data that would otherwise be to large and note necessary to include in the thesis. When necessary, the thesis will refer to the appendix.

2 Theory

This chapter will explain the general theory for buckling of thin walled shells and how to perform an FEA to calculate the buckling strength for the shells, how to create surface imperfection patterns from measurements and from the Fourier coefficients, as well as how to find probabilities of extreme values.

2.1 Buckling of Thin Walled Shells

Buckling of thin walled shells is a complicated area of structural engineering. This is because the buckling strength is largely dependent on dimensions, stiffeners, surface imperfections, load cases and much more. Ross [3] states that the structural strength for collapse pressure can be affected by as much as 50% by a imperfection in shape as small as 1%. This is one of the reasons that standards for buckling is very conservative.

2.1.1 Buckling Mechanics

One of the conservative methods to design a shell is to use the knockdown factor. The knockdown factor is a factor, $0 \leq \lambda \leq 1$, that is multiplied with the theoretical buckling strength for the shell. There has been a lot of physical experiments done on many shells with different R/t, notably from [4, 5, 6, 7, 8, 9, 10, 11, 12]. From these tests, an equation for the lower bound curve for the design knockdown factor has been made. The equation for the lower bound curve is shown in Equation 1 from Doup [13].

$$\lambda = \frac{P_{exp}}{P_{cl}} = 1 - 0.902 \left(1 - e^{-\frac{\sqrt{R/t}}{16}} \right) \tag{1}$$

Here, P_{exp} and P_{cl} are the lower bound knockdown factor and the theoretical buckling strength respectively. Equation 1 is plotted on top of the test results from [4, 5, 6, 7, 8, 9, 10, 11, 12] in Figure 1. The lower bound equation is a conservative knockdown factor, as is easy to see, because the majority of the test results have a real knockdown factor that is larger than the lower bound curve. Though the lower bound curve takes into account different R/t, it does not take into account the length of the cylinder. For the A-shells, R/t varies from about 800 to a little over 1000, the SST will have an R/t in the range of 170, which is considered very low, but the extreme length of the SST will also affect the buckling strength.

2.1.2 A-shells

The A-shells tested by Arbocz and Abrahamovich [4] was made using nickel electroplating, this made it possible to create shells with extremely low thickness and with a very even surface. A low thickness is desirable as buckling accrue easily when the thickness is low, as well as the testing is to fuel the research for rockets in the aerospace industry, where the R/t ratio is quite large.

2.2 Surface Imperfection Patterns

To create the surface imperfection pattern, a double Fourier series is used. From Arbocz and Abramovich [4], the double Fourier series can be seen in Equation 2 and 3, where Equation 2 is the half-wave cosine Fourier representation, and 3 is the half-wave sine Fourier representation.

$$\overline{W}(x,y) = t \sum_{i=0}^{N_1} A_{i0} \cos \frac{i\pi x}{L} + t \sum_{k=0}^{N_1} \sum_{l=1}^{N_2} \cos \frac{k\pi x}{L} \left(A_{kl} \cos \frac{ly}{R} + B_{kl} \sin \frac{ly}{R} \right) \tag{2}$$

$$\overline{W}(x,y) = t \sum_{i=0}^{N_1} C_{i0} \sin \frac{i\pi x}{L} + t \sum_{k=0}^{N_1} \sum_{l=1}^{N_2} \sin \frac{k\pi x}{L} \left(C_{kl} \cos \frac{ly}{R} + D_{kl} \sin \frac{ly}{R} \right) \tag{3}$$

Here, W is the size of the imperfection at position x and y on the surface, where $0 \leq x \leq L$ and $0 \leq y \leq 2\pi$, N_1 and N_2 are the size of the Fourier coefficient matrix A, B, C and D are the Fourier coefficients, L, R and t are the length, radius and surface thickness of the shell respectively.

The Fourier coefficient matrix given in Arbocz and Abramovich [4] was obtain after measurements of the imperfections from the A-shells before they were tested.

The surface imperfection of the A7-shell is shown in Figure 2. The x and y axis are the circumference and the length of the circle, and the z axis is the size of the imperfections with regards to the wall thickness
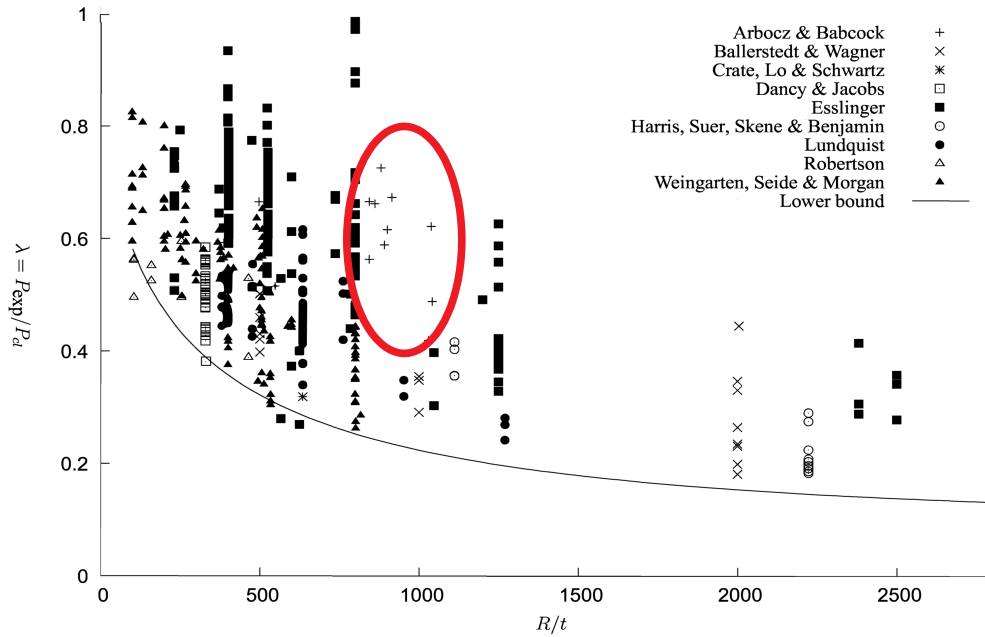
Figure 1: Results from axially compressed shells with lower bound curve (tests marked within the red ellipse is the A-shells) [14].
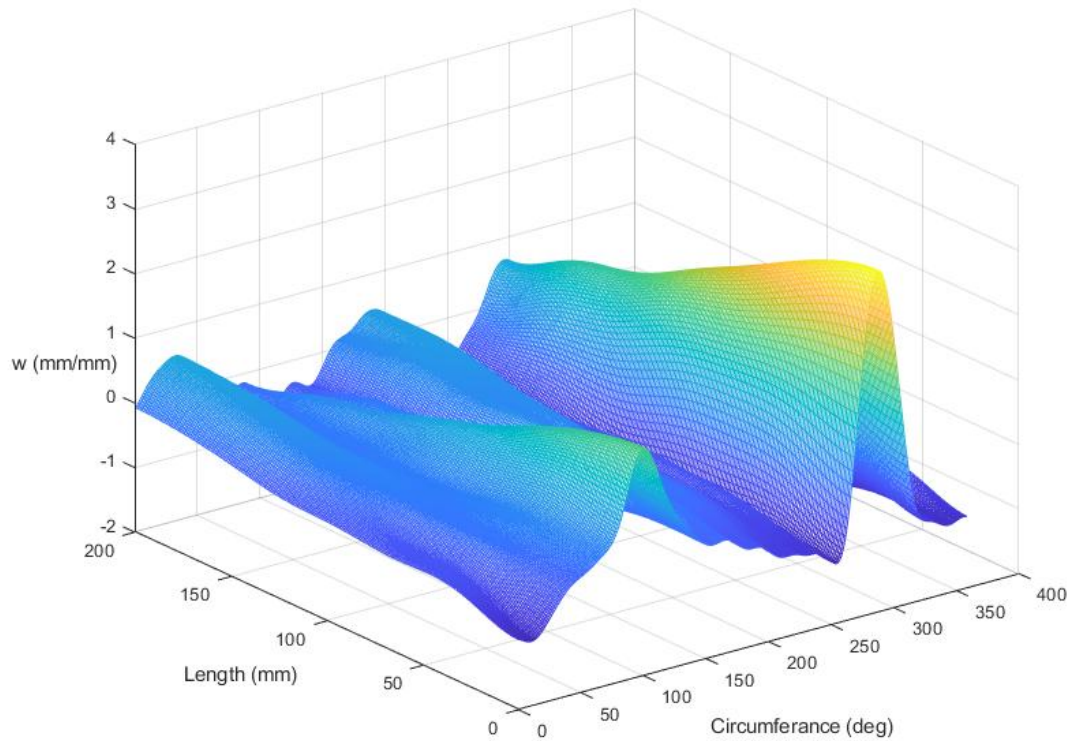
Figure 2: Surface representation of the A7-shell

of the shell. Figure 3 shows the sharpest edge of the surface representation. It is clear that there are no sharp edges in the mesh. This can help strengthening the simulated shell, as there will be less shear forces when the angle between the mesh elements is small.

2.3 FEA of Thin Walled Shells

A suitable computer application must be used in order to complete an FEA where the results can be trusted. Not only does the application support proper buckling analysis, but the model, loads and boundary conditions must be set up to reflect the real world. Another important factor for FEA is how the mesh is setup. Both mesh size and mesh shape affects the results of the simulation.

2.3.1 Computer Program

For this thesis, ANSYS Mechanical APDL was chosen as the application to run the analysis in. ANSYS is a trusted application for FEA, that has existed for many years, but the main reason that it was chosen is that UiS has licenses for the full version, as well as that ANSYS have free versions for students. This means that help is easy to get, as well as it is possible both to work from the universities computer lab,
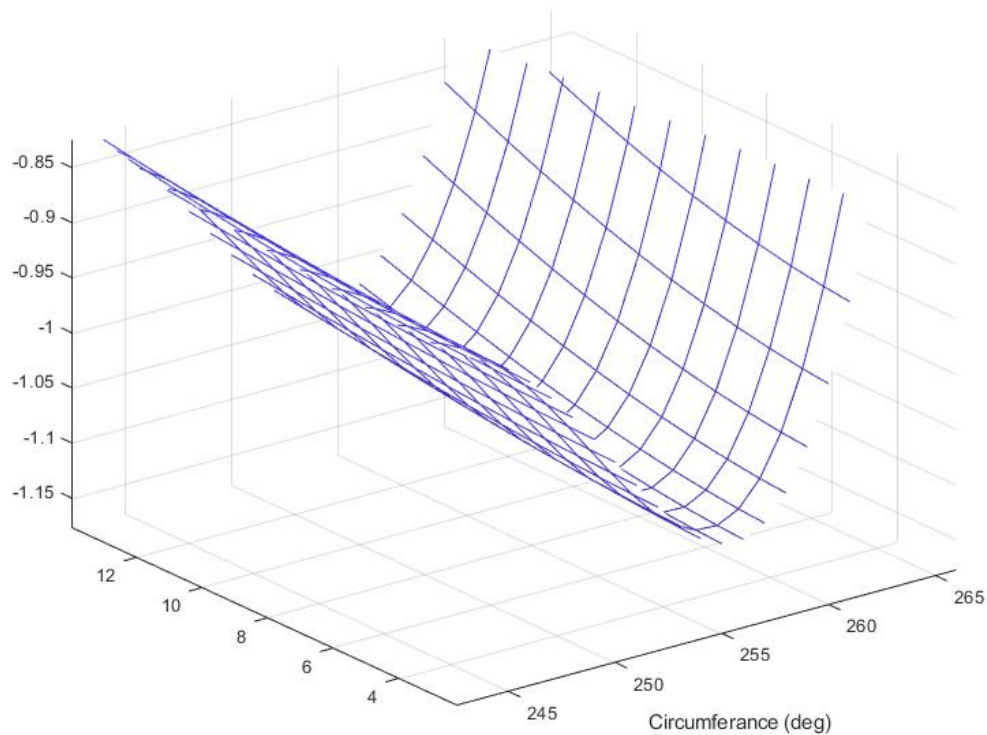
Figure 3: Surface representation of the A7-shell

as well as from home. This is important, as the Covid-19 situation was still unknown when this thesis was started.

The reason for using the APDL version of ANSYS, instead of the regular version with a modern user interface (UI) is that the APDL version makes it possible to write a script to run simulations from many different models. This is crucial, as many different surface imperfection patterns had to be simulated in order to gain enough results to run statistical analyses.

2.3.2   FEA Model

ANSYS has a programming language that lets the user create a text file with commands instead of manually clicking and typing values for each analysis, therefore making it easier to perform multiple analysis of with different

For structural models with thin shells, Shell type 181 in APDL is the best option. The other shell types have other options that is not needed, thermal conductivity, electrical conductivity or other options [15].
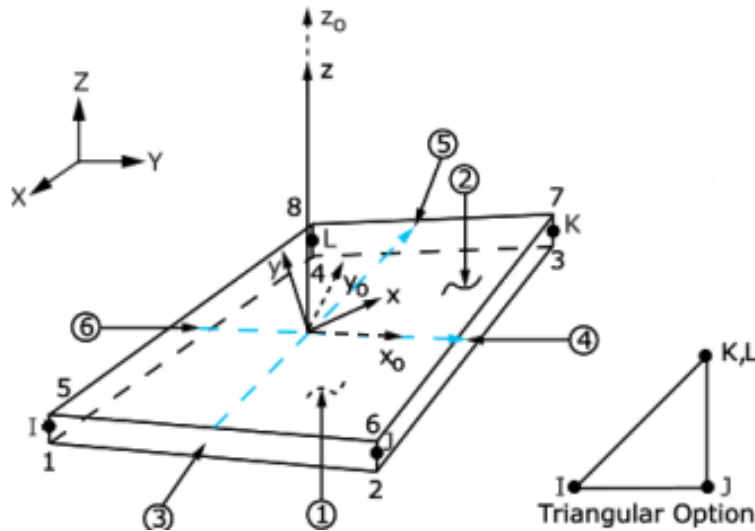
Figure 4: Representation of rectangular mesh [16].

In the documentation for APDL, shell 181, a rectangular mesh is recommended, this is because the geometry is simple, and the thickness of the shell is thin, therefore only requiring one element for the whole thickness of the shell [16]. However, a triangular mesh can have certain advantages. The main one is that for triangular mesh, all nodes lies in a single plane. In Figure 4 it is possible to see both a rectangular mesh, and a simplified triangular mesh for shell 181. It is easy to see that if one or more of the nodes is moved in the z-direction, that the element will twist.

To choose the correct mesh size is a trivial matter. The easiest way is to run a sensitivity analysis on the mesh size. This is just to start with a large mesh size, and then decrease the size until the results converges.

Boundary conditions for the model is also very important. In each end of the cylinder, there are 6 degrees of freedom that can be subjected to boundary conditions. The six boundary conditions are listed in Table 1 an overview of the different kind of boundary conditions that can be applied to a cylinder. For the most conservative result, the "relaxed" boundary conditions could be used. A problem with using the "relaxed" boundary conditions at the top of the cylinder is that it is difficult to have the top of the cylinder free in all direction when performing a physical test. Therefore, the "medium" boundary conditions are more realistic in terms of the real world tests that is possible to perform on the cylinder, as well when the shell is installed in its position on intended structure.

Table 1: Overview of different combination of boundary conditions for a cylinder (z-direction is along the length and the value in the cell represents allowable displacement for X, Y and Z, and allowable rotation for Rot_X, Rot_Y, Rot_Z, where blank cells have no restrictions).

Boundary conditions		X	Y	Z	Rot_X	Rot_Y	Rot_Z
Bottom	Strict	0	0	0	0	0	0
	Relaxed	0	0	0			
Top	Strict	0	0		0	0	0
	Medium	0	0				
	Relaxed						

2.4 Probabilities of Extreme Values

2.4.1 Gumbel Distribution

The Gumbel distribution is the most used distribution for extreme value estimations. The probability density function (PDF), cumulative density function (CDF) and the inverse CDF are shown in Equation 4, 5 and 6 respectively [17].

$$f(x) = \frac{1}{\sigma} \exp\left[-\left(\frac{x-\mu}{\sigma} + \exp\left(-\frac{x-\mu}{\sigma}\right)\right)\right] \tag{4}$$

$$F(X) = \exp\left[-\left(\exp\left(-\frac{x-\mu}{\sigma}\right)\right)\right] \tag{5}$$

$$F^{-1}(p) = \mu - \sigma \ln(-\ln p) \tag{6}$$

Here, μ is the location parameter, σ is the scale parameter and p is the probability. One disadvantage with the Gumbel distribution is that it is best suited for extreme values, where the values are greater than the average.

2.4.2 ACER Method

Where the ACER method is superior to the Gumbel method is that it only tries to estimate the tail section of the distribution, as well as it uses many more parameters than the Gumbel distribution. Without going into all the details of how the ACER method works, the general equation for the distribution in the ACER method is shown in Equation 7, from Næss et. al. [18]. This equation is only calculated from a in the interval $\eta \leq \eta_1$, where η_1 is a tail marker that limit the amount of the data that should be used to estimate the parameters of the distribution.

$$\epsilon_k(\eta) = q_k(\eta)\left[1 + \xi_k\left(a_k(\eta - b_k)^{c_k}\right)\right]^{-\frac{1}{\xi_k}} \tag{7}$$

Here, a_k, b_k, c_k, q_k and ξ_k are parameters that is calculated using a Matlab script in order for Equation 7 to fit the data in the interval that is larger than the tail marker η_1, and η is the probability. The ACER method has the same problem as the Gumbel distribution, that is that it is made for large extremes. To get around this problem, one way of going around that problem is to inverse the results from the simulations.

3   Method

In order to calculate the the collapse strength of a thin walled shell with this method, a fitting surface imperfection pattern, an FEA model and a fitting probability distribution must be made. For the best results, all of these should be as close to the real world as possible. Arguably, the surface imperfection pattern for the SST is the most difficult to predict. Such a large and thin walled structure has not been made outside of the aerospace industry, where the demands for strength is completely different, the main differences,that the shell is standing upright and not exposed to a large external pressure. FEA models for buckling has been very accurate for many years. In most cases, the inaccuracy of simulated shells is that if a perfect shell is simulated, the analysis does not take into account for imperfections in the surface. This problem is eliminated with this method, as the shells tested are imperfect. When calculating a probability distribution for extreme cases, the Gumbel distribution is the most used. This is because it fits well in the extreme values of high and low probabilities. However, it is not always the case.

3.1   Surface imperfections

As mentioned above, it is not easy to predict the surface imperfection pattern of a structure that has not been made. Guesses can be made on account of other studies of imperfections as well as the design of the structure. From Teng and Rotter [14] an imperfection pattern of a shell made from three different sheets of metal welded together can be seen in Figure **??**. It is clearly possible to observe the larger imperfections caused by the welds in the seams. Because of the size of the SST, many more seams will be present both around the circumference and along the length. For this thesis, the seams has not been taken into account because a design of the SST has not been made, and therefore, the surface imperfection pattern has been based on another shell. Specifically the A7 shell. This is because the A-Shells is thoroughly studied in Teng and Rotter [14]. Therefore, both the A7 shell and the SST is simulated is this thesis.

The Fourier coefficients is taken from The Initial Imperfection Data Bank at the Delft University of Technology Part I (IDB) [4]. For the A7 shell, a 15 by 8 matrix for the Fourier coefficients are used. This gives enough coefficients in order to accurately create a surface representation of the imperfections.

3.2   FEA Model

Matlab was used in order to create the script that runs in ANSYS APDL. In total 2 Matlab scripts are needed in order to run the simulation in ANSYS. The first script takes the material and physical data for each shell and the Fourier coefficient matrix and uses Monte Carlo simulation in order to create a list of all nodes and their location, and then it creates a list of all elements. The second script creates many .inp files that run each, one simulation with one imperfection pattern. It also creates a file path

Table 2: Material properties and measurements.

Shell	L (m)	R (m)	t (mm)	Youngs Modulus (GPa)	Poissons Ratio
A7	0.2032	0.1016	0.114	104.11	0.3
SST	100	8.5	50	200	0.3

Table 3: mesh size, number of elements in each direction and in total.

Shell	Mesh size	Elements (length)	Elements (circumference)	Total elements
A7	1.1mm	185	580	106720
SST	2.1m	476	254	120650

and moves metadata from the simulation in order to store and make it possible to retrieve all data at a later time. At last, the script creates a .inp files that runs all other .inp files that are made. Both scripts is attached as Appendix A and B.

3.3 Simulation Inputs

For the simulation to run properly, all the inputs to ANSYS must be correct, and as close to the real world scenario as possible. The A-shells are made from electroplated nickel, and have different material properties than steel. In Table 2 is the material properties and the physical measurements of the A7-shell and the SST.

The mesh size and shape is also an essential input. As mentioned in Chapter 2.3, a rectangular mesh is best for this type of analysis. To determine the mesh size, a sensitivity study on the mesh size is needed. A sensitivity analysis on this exact model was performed by Jamissen et. al. [2]. The acceptable mesh size for the A-shells are $1mm$, and for the SST, the mesh can be $2m$ in size. The script is made such that the mesh elements are as square as possible. In Table 3, the mesh size, number of elements in each direction and in total is listed. Note that the mesh size is slightly higher in order to get under 128000 elements, which is the limit in the student version.

In order to achieve some variation in the imperfections on the shells, a variance of 0.4 is chosen to be applied to the Fourier coefficients in the Monte Carlo simulation.

The last inputs needed for the simulation is the loads and boundary conditions. When performing buckling analysis, ANSYS will output a factor that is proportional to the input load. To make it as simple as possible, a unit load of $1N$ is spread across all of the top nodes in the $-z$-direction, this will compress the cylinder. For external pressure, 1 bar is applied to the outside of the structure, and due to the way the mesh is created, all the elements are facing with the same direction outwards, thus creating only external pressure. When the simulation for the buckling are finished, the output will be the collapse load in Newton, and the collapse pressure in bar. As discussed in Chapter 2.3, it is best to lock displacement in all three directions in the bottom of the cylinder, but to keep the rotational constraint free. At the top,

as well as the force, displacement in both X and Y direction is locked. This is closest to how a cylinder will be tested in a physical test.

3.4 Analysis of Results

To have only the results of the simulation is not enough to make conclusions. After ANSYS is finished solving the simulations, the results has to be processed. To complete the storage of the results, a third Matlab script is run in order to sort the results and save it in the correct file path. This script is attached as Appendix C. When all the data is collected and stored, the probabilistic analysis can begin. Both the Gumbel distribution as well as the ACER method is used to have both a proven method and an experimental one.

For the Gumbel distribution, a simple Matlab script, performing a probability plot will solve for both μ and σ for the Gumbel distribution. The ACER method more complex to solve, but a script provided by Arvid Næss makes it possible to solve in order to get a result.

The results from both Gumbel and ACER will be compared to the results by Teng and Rotter [14], the physical test by Arbocz and Abramovich [4] and the lower bound curve.

4 Results

The results will be divided into two parts, first, the results from the FEA, and then the probabilistic calculations.

4.1 FEA Results

To get a better overview, the results from axial compression is shown first, then the results for external pressure.

4.1.1 Axial Compression

For the A7-shell, the simulated buckling strength relates to a stronger cylinder than was tested in [4], with the exact imperfections achieving a strength of $4880.1N$, this is very close to the theoretical buckling strength of 5158.6, and is equivalent to a knockdown factor of $\lambda = 0.946$. This is much higher than the knockdown factor of $\lambda = 0.589$ for the tested A7-shell.

For the SST, the theoretical buckling strength is $1.81E + 9N$ and with the exact imperfections from the A7-shell, the buckling strength is $1.46e9N$, resulting in a buckling factor of $\lambda = 0.801$.

Using a Monte Carlo method with a variation of 0.4, the coefficients in the Fourier coefficient matrix in Appendix E and the half-wave cosine Fourier series, a simulation of 20 shells had a mean result of $4500.7N$ and a standard deviation of $59.0N$ for the A7 shell. A CDF of the result is shown in Figure 5. The SST had a mean strength of $1.44E + 9N$ and a standard deviation of $4.85E + 7N$. A CDF of the result is shown in Figure 6

The shapes of the deformations are showed in 7. There is clearly a difference in where and when the cylinder buckled. This can be observed both in the difference in collapse strength and on the post-buckling geometry.

4.1.2 External Pressure

For the simulations with external pressure, the imperfections had much less effect on the results. The knockdown factor for buckling with external pressure is almost $\lambda = 1$.

Using the method from [19], better described in [20], the external collapse pressure for the A7-shell is $0.0203bar$ and it is $0.0578bar$ for the SST.

The result for the simulated shells without imperfections are $0.0255bar$ and $0.521bar$ for the A7-shell and SST respectively. For the simulation with imperfections, the A7-shell and the SST got collapse pressures of $0.0254bar$ and $0.512bar$. It is easy to observe, not only from the similarities in the simulations with and without imperfections, but also in the deformation shape in the FEA, shown

Table 4: Gumbel coefficients for A7 and SST with axial load.

Shell	μ	σ
A7	4529.9N	56.91N
SST	1.4648e+9N	5.1886e+7N

in Figure 8, that the imperfections don't affect the shells collapse pressure for buckling. For the SST, the general shape for the deformation is the same, but buckling has started on another place along the circumference because of the imperfections.

4.2 Probabilistic Calculation

It is only practical run probabilistic analysis on the simulation with axial load, as imperfections does not affect the collapse strength for external pressure.

4.2.1 Gumbel Analysis

With the Gumbel extreme value distribution, the μ and σ, was found using the Matlab script attached as Appendix D, and is listed in Table 4. The probability plot is plotted an is shown in Figure 9 and 10. The shape of the Gumbel distribution does not fit well to the simulated results for either the A7-shells or for the SST-shells. When going down to low probabilities with this distribution, the Gumbel representation will give results that are much stronger than what is the reality. This is unacceptable when designing structures.
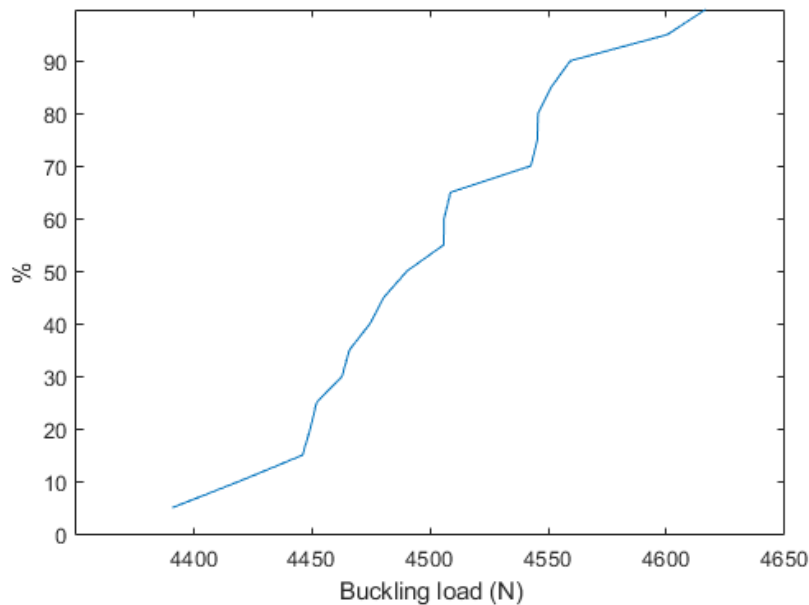
Figure 5: CDF of the simulated A7-shells.

Figure 6: CDF of the simulated SST-shells

Table 5: 10^{-6} return level estimate with confidence interval

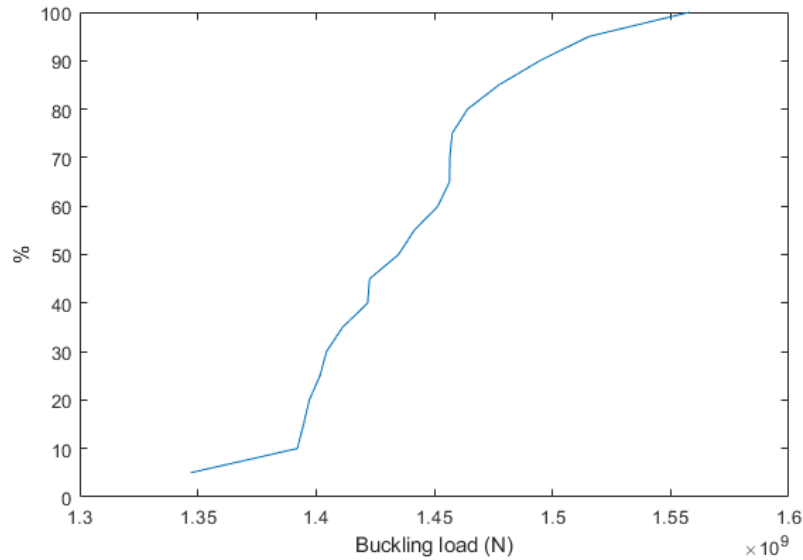| Shell	10^{-6} return level estimate	Lower confidence interval	Higher confidence interval
A7	4285N	4245N	4389N
SST	1.2739e+9N	1.2502e+9N	1.3667e+9N

Using the inverse Gumbel distribution in Equation 6, and the parameters in Table, 4 the collapse strength for the A7-shell and the SST with a failure probability of 10^{-6} are $4380N$ and $1.3286e+9N$ respectively.

4.2.2   ACER Method

For the ACER method, the result for the collapse strength with a failure probability of 10^{-6} was dependent on what tail-marker was used. After inverting the values in the results and running the Matlab script for the ACER method, when selecting the tail-marker, Figure 11 shows up. From the figure, it is possible to see that the value for the tail-marker could be set to $\eta_1 = 2.22\text{e-4}$. This ensures that the nice slope from around the middle of the graph and to the right forms the basis for script to find the best fitting parameters. For the SST, shown in Figure 12, a good tail marker is $\eta_1 = 6.9\text{e-10}$.

For these tail-markers, the results from the ACER method is shown in Figure 13 and 14, and the values for the results are shown in Table 5

4.3   Summary of Results

To summarise the result, the knockdown factor for axial can be increased for when FEA and statistical analysis is used on thin metal shells. Listed in Table 6 are the results from both the A7-shell and the
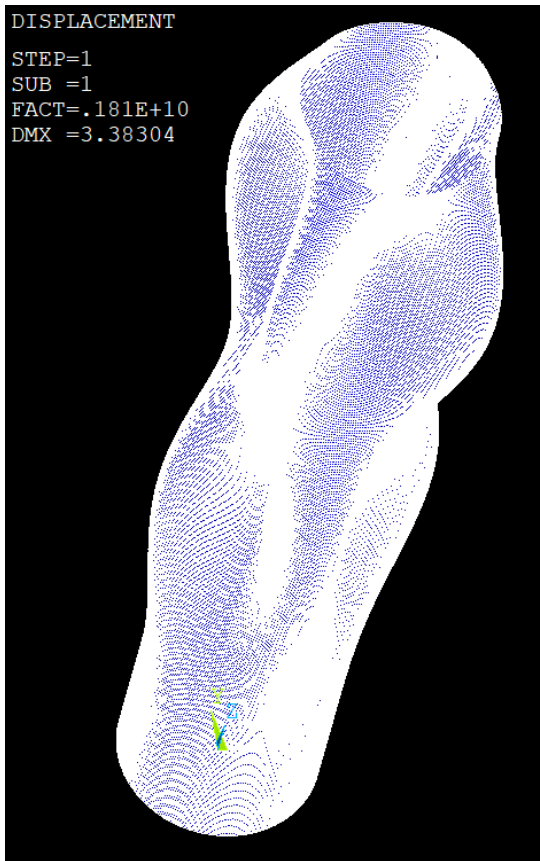
(a) SST no imperfections
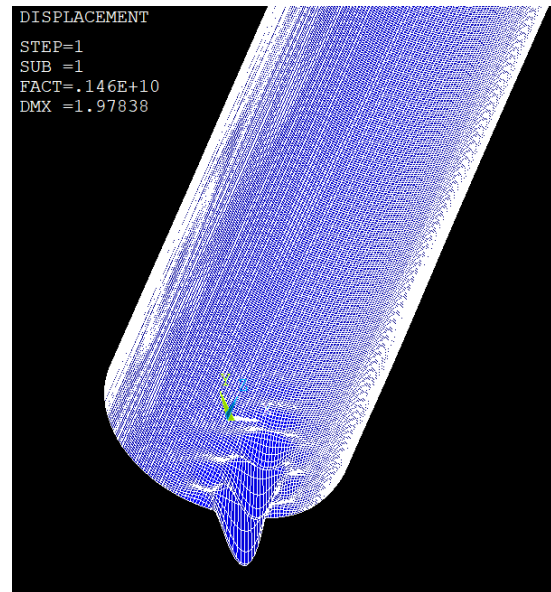
(b) SST with imperfections
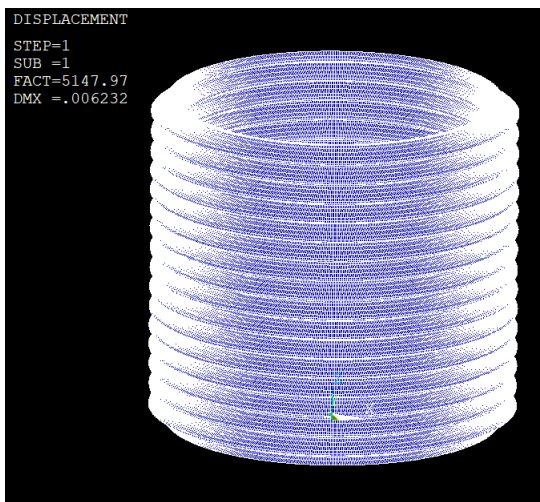
(c) A7-shell no imperfections
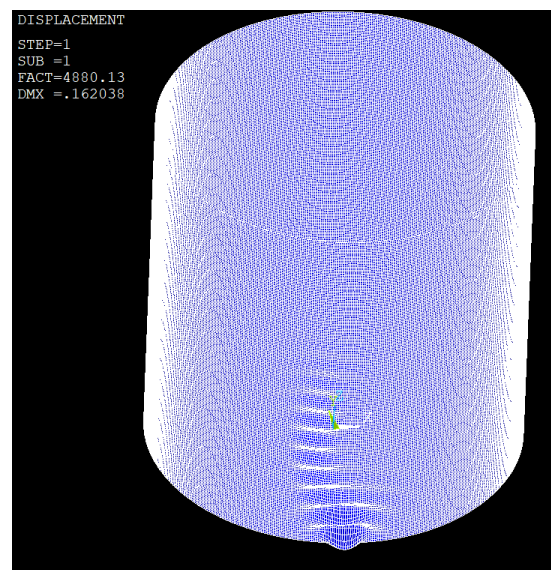
(d) A7-shell with imperfections

Figure 7: Deformed shape of cylinders that buckled because of axial load

(a) SST no imperfections

(b) SST with imperfections

(c) A7-shell no imperfections

(d) A7-shell with imperfections
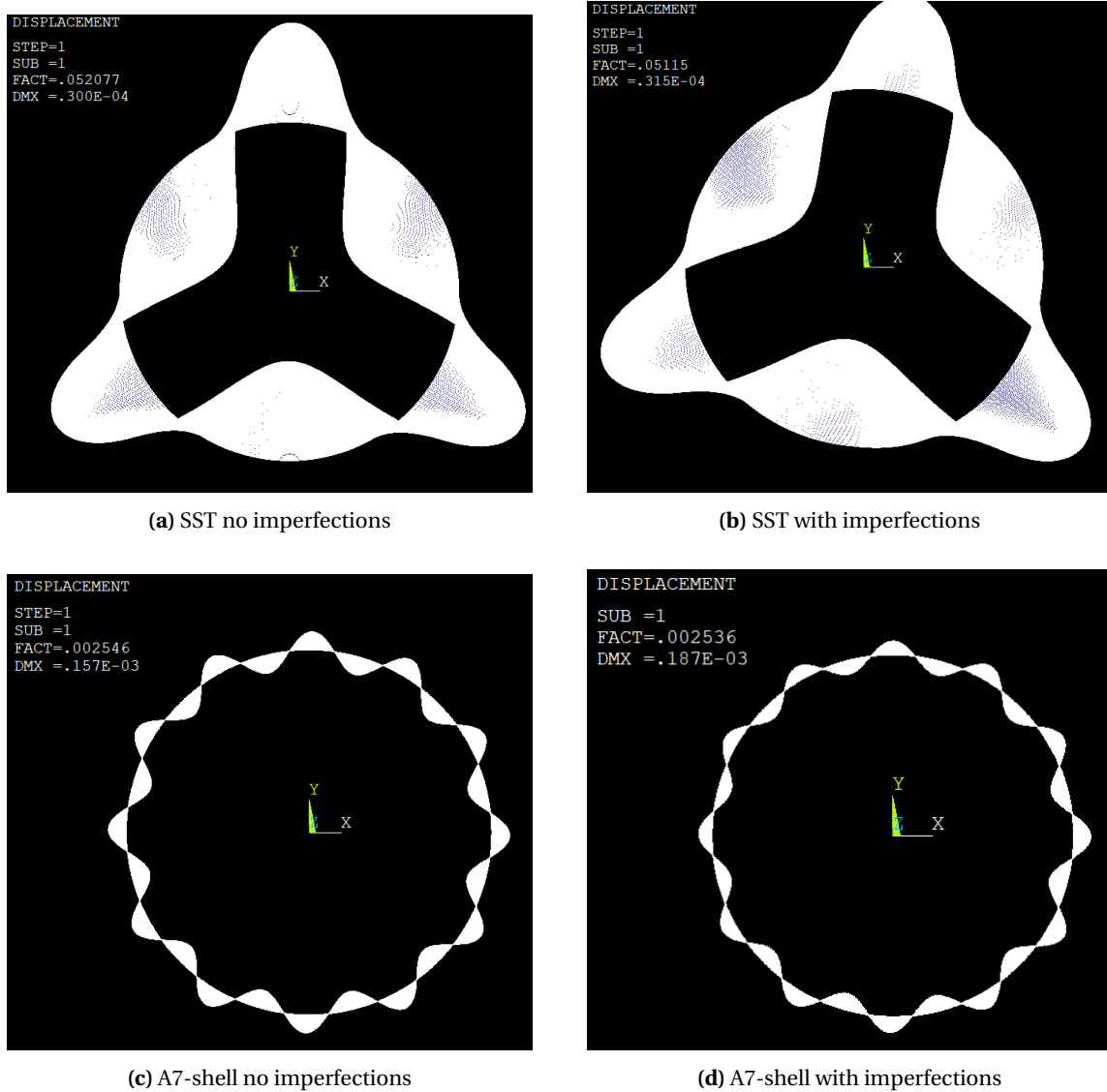
Figure 8: Deformed shape of cylinders that buckled because of external pressure (Note that the buckling factor is on order of magnitude to small compared to the correct result)

SST from Teng and Rotter [14], which used a reliability of 0.99997, Arbocz and Abramovich [4], the lower bound curve [13], the Gumbel distribution and the ACER method.

From Table 6, it is possible to observe that the method of doing and FEA on the imperfections can drastically increase the knockdown factor. Though, the result for the exact imperfections that the A7-shell resulted in a larger collapse strength than then the physical test, method is promising.

The imperfections does not seem to affect the buckling factor for cylinders exposed to external pressure.
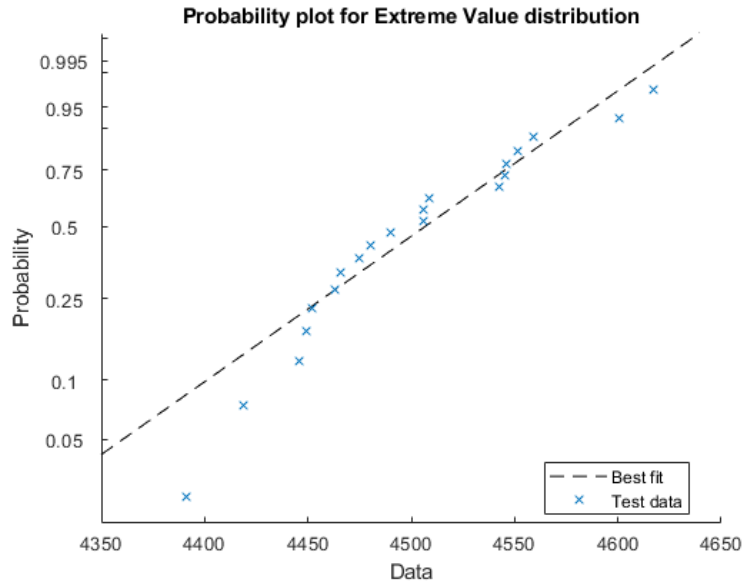
Figure 9: Gumbel probability plot for the simulated A7-shells.
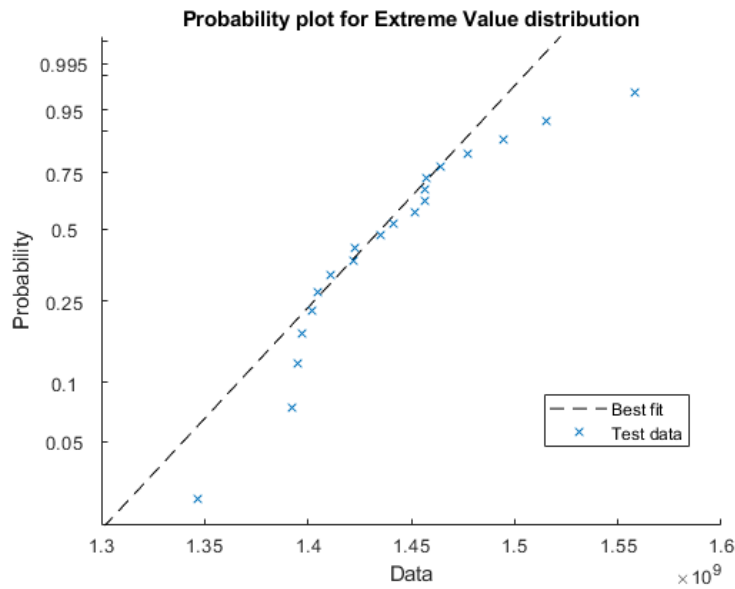
Figure 10: Gumbel probability plot for the simulated SST-shells.

Table 6: Knockdown factor for simulation and tests, numbers in first row are failure probability

Shell	Teng and Rotter $3*10^{-5}$	Physical test by Arbocz and Abramovich	Lover bound curve	Gumbel distribution $1*10^{-6}$ ($3*10^{-5}$)	ACER method $1*10^{-6}$ ($3*10^{-5}$)
A7	0.54	0.589	0.238	0.849 (0.852)	0.823 (0.830)
SST	N/A	N/A	0.497	0.734 (0.742)	0.691 (0.705)
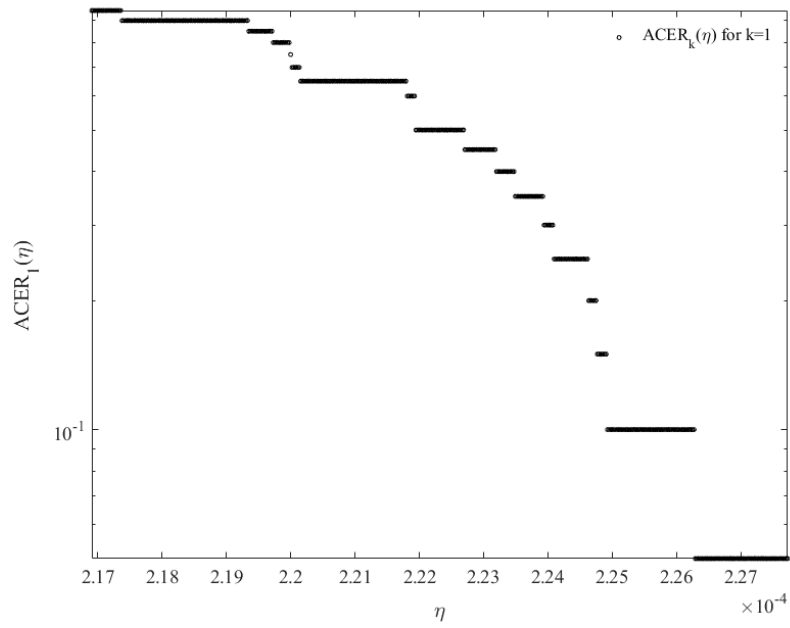
Figure 11: $\eta_1 = 2.22$e-4 is a good tail-marker.
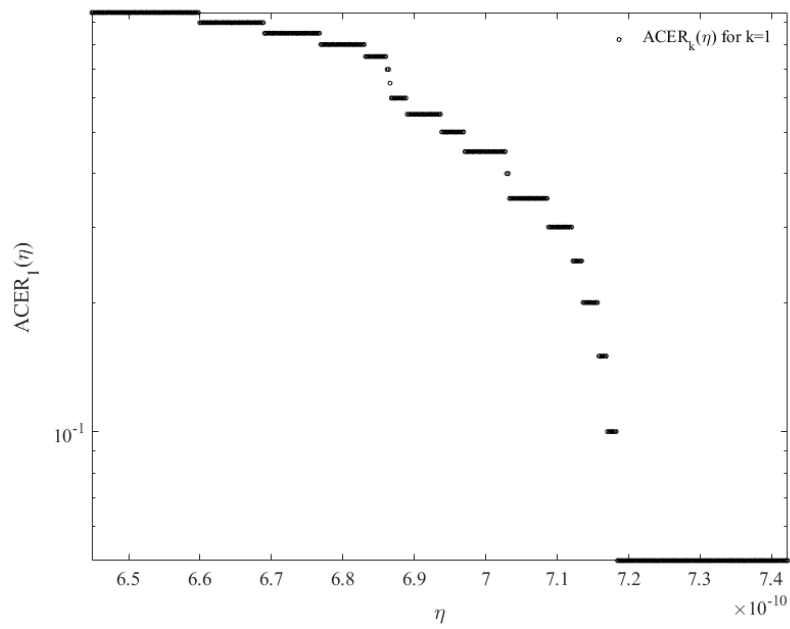
Figure 12: $\eta_1 = 6.9$e-10 is a good tail-marker.
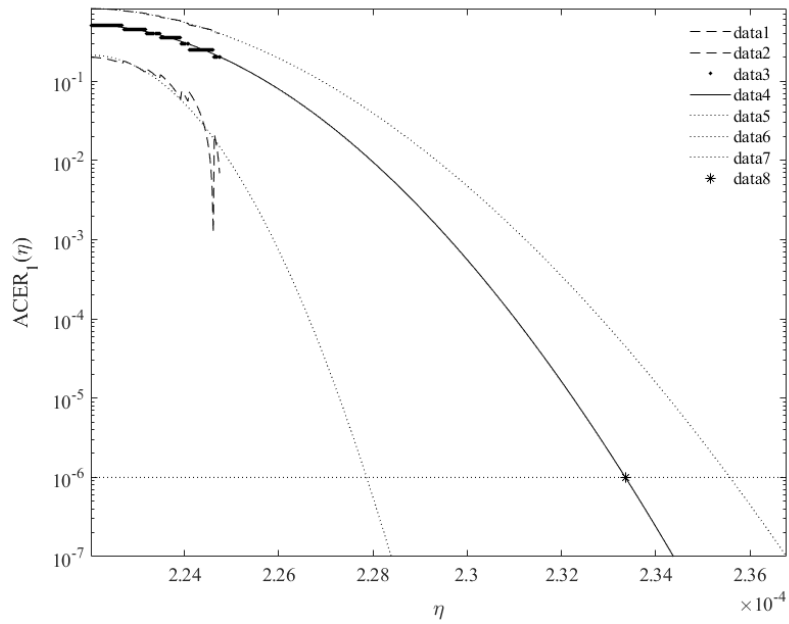
Figure 13: ACER distribution for the A7-shell
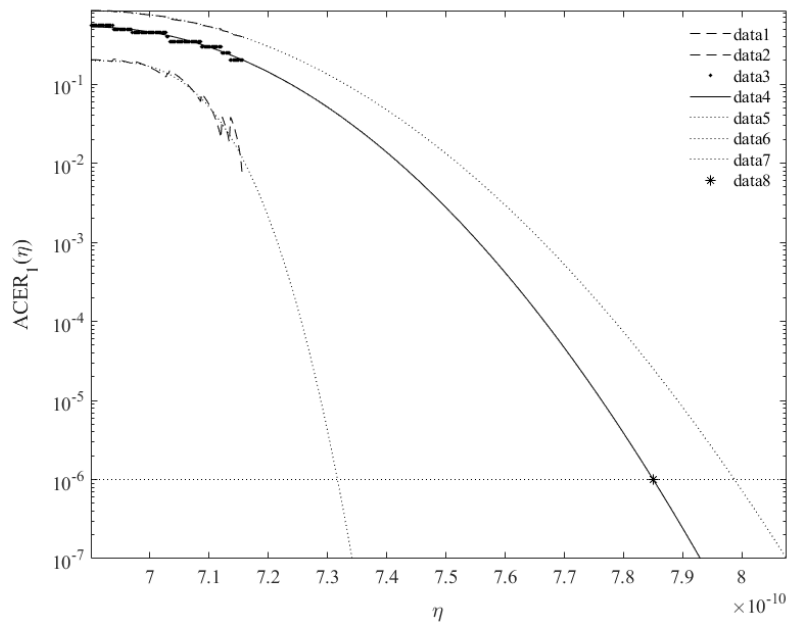
Figure 14: ACER distribution for the SST.

5   Conclusion

To conclude this thesis the method of using imperfections and FEA are the future for designing large thin metal shells. The shells using this method is shown to have a higher knockdown factor, therefore increasing the buckling load, and consequently reducing both weight and cost. The ACER method is a good tool to perform statistical analysis, as it more easily adapts to the distribution of test results, resulting in a closer fit than the Gumbel distribution.

For cylinders exposed to external pressure, the imperfections does not affect the strength of the cylinder in any way that can not be within the margin of error, therefore conventional methods should be used when designing for external pressure. This result is a bit strange, as mentioned in 2.1 and by [3], an imperfection that disturbs the geometry by as 1% can decrease the buckling strength by as much as 50%. One reason that the buckling strength is not affected with the imperfections in the A7-shell is that because of the Fourier representation only has a limited amount of coefficients, there is a limit to how sharp the imperfections can get, see Figure 3 in Chapter 2.2, and that sharper imperfections will affect the buckling strength more.

6   Further work and recommendations

Although the process used in this thesis did not get concrete results for the design of the SST outer shell, the method is really promising. The next focus areas of others, when trying to continue this research should be, but is not limited to; creating a model to run FEA on an outer shell with stiffeners, creating another set of Fourier coefficients that more randomly creates the imperfection surface or finding a better probability distribution to fit the lower end extremes of the FEA results.

The use of shell stiffeners drastically increase the buckling strength of the outer shell. Because of the internal tanks of the SST, an analysis of ring stiffeners alone will give a very conservative results, as the internal structure will provide extra support to the ring stiffeners.

This thesis used only Fourier coefficients from the A7 shell made random with Monte Carlo simulations in order to make the imperfections surfaces. To create a more realistic set of coefficients for the production process of a 17m diameter shell as well as not create only one general shape for the imperfections is a good idea in order to make room for more different imperfections. One last thing to take into account with imperfections is to look at the effect of small to large dents on the buckling strength of the shell, as it is not uncommon for containers, anchors or other objects to fall to the sea bed. If the SST is hit with an object, it can cause massive damage as well as collapse.

Only Gumbel extreme value distribution and the ACER method was used to create an estimate for the weakest allowable buckling strength for the SST, neither of whom was a distribution that fitted as well enough to create a results that can be trusted enough to create a design from.

References

[1] Yucong Ma et al. "Baseline design of a subsea shuttle tanker system for liquid carbon dioxide transportation". In: *Ocean Engineering* 240.109891 (2021).

[2] P. L. Jamissen, Ma. J, and Y Xing. "PROBABILISTIC DESIGN OF THIN-WALLED CYLINDRICAL STRUCTURES FOR APPLICATION IN LARGE CARGO SUBMARINES". In: *OMAE2022* 79485 (2022).

[3] C.T. F. Ross. *Pressure Vessels External Pressure Technology*. Vol. 2. Woodhead Publishing, 2011.

[4] J. Arbocz and H. Abramovich. "The Initial Imperfection Data Bank at the Delft University of Technology Part I". In: *TU Delft Department of Aerospace Engineering* (1979).

[5] W. Ballerstedt and H. Wagner. "Versuche über die Festigkeit dünner unversteifter Zylinder unter Schub-und Längskräften". In: *Luftfahrtforschung* 13 (1936), pp. 309–312.

[6] H. Lo, E. B. Crate, and B. Schwartz. "Buckling of Thin-walled Cylinders Under Axial Compression and Internal Pressure. - NACA Report 1027". In: *NASA Langley Research Center* (1951).

[7] R. Dancy and D. Jacobs. "The initial imperfection data bank at the Delft University of Technology, Part II." In: *TU Delft Department of Aerospace Engineering* (1998).

[8] M. Esslinger, B. Geier, and U. Heidemann. "Comments on the paper 'some complementsto the eccs design code concerning isotropic cylinders' given in the 'preliminary report'for the Liege colloquim". In: *Deutsches Zentrum fur Luft- und Raumfahrt - Braunschweig - Germany* (1977).

[9] L. A. Harris et al. "The stability of thin-walled unstiffened circular cylinders under axial compression including the effects of internal pressure". In: *Journal of the Aeronautical Sciences* 24 (1957), pp. 587–596.

[10] E.E. Lundquist. "Strength tests of thin-walled duralumin cylinders in compression - NACA Report 473". In: *NASA Langley Research Center* (1933).

[11] A. Robertson. "The strength of tubular struts". In: *Proceedings of the Royal Society of London* 121 A (1928), pp. 558–585.

[12] P. Seide, V. I. Weingarten, and E. J. Morgan. "The development of design criteria for elastic stability of thin shell structures". In: *TRW Space Technology Labs Los Angeles CA* (1960).

[13] M.R. Doup. "Probabilistic analysis of the buckling of thin-walled shells using an imperfection database and a two-mode analysis. Ir. Thesis (also Memorandum M-808)". In: *Delft University of Technology - Faculty of Aerospace Engineering - The Netherlands* (1997).

[14] J.G. Teng and J.M. Rotter. *Buckling of thin metal shells*. Vol. 1. Spon Press, 2001.

[15] Department of Applied Mechanics - Faculty of Mechanical Engineering - Budapest University of Technology & Economics. *ANSYS APDL external documentation*. URL: `https://www.mm.bme.hu/~gyebro/files/ans_help_v182/ans_elem/`. (Accessed 13.06.2022).

[16] Department of Applied Mechanics - Faculty of Mechanical Engineering - Budapest University of Technology & Economics. *ANSYS APDL external documentation*. URL: `https://www.mm.`

`bme.hu/~gyebro/files/ans_help_v182/ans_elem/Hlp_E_SHELL181.html`. (Accessed 03.02.2022).

[17]   Charles Zaiontz. *Gumbel Distribution*. URL: `https://www.real-statistics.com/other-key-distributions/gumbel-distribution/`. (Accessed 10.06.2022).

[18]   A. Næss, O. Gaidai, and O. Karpa. "Estimation of Extreme Values by the Average Conditional Exceedance Rate Method". In: *Journal of Probability and Statistics* 2013 (2013).

[19]   US Department of Commerce. "Stress Analysis Manual". In: *Air Force Flight Dynamics Laboratory Wright-Patterson AFB - OH* (1986).

[20]   Abbott Aerospace Canada LTD. *Analysis and design of Composite and metallic flight vehicle Structures*. URL: `https://www.abbottaerospace.com/aa-sb-001/15-local-stability-isotropic-materials/15-4-buckling-specific-cases/15-4-1-buckling-of-thin-cylindrical-shells/`. (Accessed 14.06.2022).

Appendices

Appendix A

Create surface imperfection and node coordinates

```matlab
clc
% clf
clear
load('..\OFF600\Data\Coefficients.mat')
```

```matlab
%
% SST       - 0
% A7        - 1
% A12       - 2
% B1        - 3
% HO        - 4
% Custom    - 5


Shell= 0 ; %See table above
Imp=1; %See table above
Thickness_override = 0; % If 0, no, if 1, yes
Thickness = 0.00005;
Perfect= 0 ; %If 0, imperfect, if 1, perfect
Variance = 0.4; %Set to zero for exact imperfections
Num_Sim=20;
Imperfectionscalar=1;
```

```matlab
if Shell == 0

    % SST
    t=0.05;         %Thickness
    R=17/2;          %Radius
    L=100;           %Lenght
    Dia=2*R;

    Meshsize=0.21;
    Div_Len= round(L/Meshsize);
    Div_Cir= round(2*pi*R/Meshsize);

    E=2e11;
    v=0.3;

%     Div_Len=200;        %No of divisions on length
%     Div_Cir=80;         %No of divisions on circumferance

elseif Shell == 1

    % A-7
    t=0.000114;         %Thickness
    R=0.1016;           %Radius
    L=0.2032;            %Lenght
    Dia=2*R;

    E=1.0411e11;
```

```matlab
    v=0.3;

    Meshsize=0.0011;
    Div_Len= round(L/Meshsize);
    Div_Cir= round(2*pi*R/Meshsize);
%      Div_Len=200;           %No of divisions on length
%      Div_Cir=360;             %No of divisions on circumferance



elseif Shell == 2

    % A-12
    t=0.0001204;         %Thickness
    R=0.1016;            %Radius
    L=0.20955;           %Lenght
    Dia=2*R;

    E=1.048e11;
    v=0.3;

    Meshsize=0.002;
    Div_Len= round(L/Meshsize);
    Div_Cir= round(2*pi*R/Meshsize);
%    Div_Len=200;           %No of divisions on length
%    Div_Cir=360;             %No of divisions on circumferance



elseif Shell == 3

    % B-1
    t=0.000205;          %Thickness
    R=0.1016;            %Radius
    L=0.19685;           %Lenght
    Dia=2*R;

        E=1.065e11;
    v=0.3;

    Meshsize=0.002;
    Div_Len= round(L/Meshsize);
    Div_Cir= round(2*pi*R/Meshsize);
%    Div_Len=200;           %No of divisions on length
%    Div_Cir=360;             %No of divisions on circumferance


elseif Shell == 4

    % HO
    t=0.0006426;          %Thickness
    R=0.9462;            %Radius
    L=2.7432;             %Lenght
```

```matlab
    Dia=2*R;

        E=6.895e10;
    v=0.3;

    Meshsize=0.02;
    Div_Len= round(L/Meshsize);
    Div_Cir= round(2*pi*R/Meshsize);
     Div_Len=200;           %No of divisions on length
     Div_Cir=360;            %No of divisions on circumferance


elseif Shell == 5

    %Custom
    t=0.000114;          %Thickness
    R=0.1016;            %Radius
    L=0.2032;            %Lenght
    Dia=2*R;

        E=1.065e11;
    v=0.3;

    Meshsize=0.002;

    Div_Len= round(L/Meshsize);
    Div_Cir= round(2*pi*R/Meshsize);
%    Div_Len=200;           %No of divisions on length
%    Div_Cir=360;            %No of divisions on circumferance

else
    disp('Error! Wrong shell')
    return
end

if Imp == 0

elseif Imp == 1

    MA=table2array(A7A);
    MB=table2array(A7B);

elseif Imp == 2

    MA=table2array(A12A);
    MB=table2array(A12B);

elseif Imp == 3

        MA=table2array(B1A);
    MB=table2array(B1B);

elseif Imp == 4
```

```matlab
    MA=table2array(HOA);
    MB=table2array(HOB);

else
    disp('Error ! Wrong imperfection')
return

end

if Thickness_override == 1
t=Thickness;
end
```

Half wave cosine

```matlab
SDA=MA*Variance;                        %Standard deviation
SDB=MB*Variance;                        %Standard deviation

sz=size(MA);



for m=1:Num_Sim

    if Perfect == 0

        A=zeros(sz);
        B=zeros(sz);
        for i=1:sz(1)                   %Monte Carlo simulation on coeficcients
        for j=1:sz(2)
            A(i,j)=( randn() * SDA(i,j) ) + MA(i,j);
            B(i,j)=( randn() * SDB(i,j) ) + MB(i,j);
        end
        end
        W2=zeros(Div_Len,Div_Cir);



        for x=1:Div_Len        %Half-wave sine
            for y=1:Div_Cir
                for i=1:sz(1)
                        W2(x,y)=W2(x,y)+A(i,1)*cos((i-1)*pi*x/Div_Len);
                end

                for i=1:sz(1)
                    for j=2:sz(2)
                        W2(x,y)=W2(x,y)+ (cos((i-1)*pi*x/Div_Len)*(A(i,j)*cos((j-
1)*y/Div_Cir*2*pi)+B(i,j)*sin((j-1)*y/Div_Cir*2*pi)));
                    end

                end
            end
```

```matlab
        end

    W2=W2*Imperfectionscalar;
```

Node

```matlab
        n = 1;
        Node=zeros(Div_Cir*Div_Len,5);
        for i = 1:Div_Len              %Only from i=1, not i=0
            z = i/Div_Len*L;
            for j = 1:Div_Cir
                x = (R+t*W2(i,j))*cos(j/Div_Cir*2*pi);
                y = (R+t*W2(i,j))*sin(j/Div_Cir*2*pi);
                Node(n,1) = n;
                Node(n,2) = x;
                Node(n,3) = y;
                Node(n,4) = z;
                n = n + 1;
            end
        end
```

Node Without imperfection

```matlab
        elseif Perfect ==1

        n = 1;
        Node=zeros(Div_Cir*Div_Len,5);
        for i = 1:Div_Len              %Only from i=1, not i=0
            z = i/Div_Len*L;
            for j = 1:Div_Cir
                x = (R)*cos(j/Div_Cir*2*pi);
                y = (R)*sin(j/Div_Cir*2*pi);
                Node(n,1) = n;
                Node(n,2) = x;
                Node(n,3) = y;
                Node(n,4) = z;
                n = n + 1;
            end
        end

    else
        break
    end
```

Elements

```matlab
Ele=zeros((Div_Len-1)*Div_Cir,5);
```

```matlab
n = 1;
for i = 1:(Div_Len-1)
    for j = 1:Div_Cir
        Ele(n,1) = n;
        Ele(n,2) = n;
        Ele(n,5) = n + Div_Cir;
        n = n + 1;
    end
end

for i = 0:Div_Len-2
    Ele(1+i*(Div_Cir):Div_Cir-1+i*(Div_Cir),3) = Ele(2+i*(Div_Cir):Div_Cir+i*(Div_Cir),2);
    Ele(Div_Cir+i*(Div_Cir),3) = Ele(1+i*(Div_Cir),2);
    Ele(1+i*(Div_Cir):Div_Cir-1+i*(Div_Cir),4) = Ele(2+i*(Div_Cir):Div_Cir+i*(Div_Cir),5);
    Ele(Div_Cir+i*(Div_Cir),4) = Ele(1+i*(Div_Cir),5);
end

 save(['..\OFF600\Mat\' num2str(m) '.mat'],'Node','Ele');

end

UnitLoad=1/Div_Cir;

FE_mesh.E = E;
FE_mesh.v = v;
FE_mesh.Variance = Variance;
FE_mesh.Dia = Dia;
FE_mesh.L = L;
FE_mesh.Div_Cir = Div_Cir;
FE_mesh.Div_Len = Div_Len;
FE_mesh.UnitLoad = UnitLoad;
FE_mesh.t = t;
FE_mesh.Num_Sim = Num_Sim;
FE_mesh.Perfect = Perfect;
FE_mesh.Shell = Shell;

save('..\OFF600\Data\FE_mesh.mat','FE_mesh');

load('..\OFF600\Data\ser.mat','serial')
serial = serial +1;

save('..\OFF600\Data\ser.mat','serial');
```

Published with MATLAB® R2021b

Appendix B

Create .inp file for ANSYS

```matlab
clear all;


load('..\OFF600\Data\FE_mesh.mat','FE_mesh')
load('..\OFF600\Data\ser.mat','serial')

for m=1:FE_mesh.Num_Sim

    load(['..\OFF600\Mat\' num2str(m) '.mat'],'Node','Ele');

fid = fopen(sprintf('%ia%i.inp',serial,m), 'w');

fprintf(fid, '\n');

fprintf(fid, '\n');

fprintf(fid, '/TITLE,file%i \n',m);
fprintf(fid, '\n');

fprintf(fid, '/PREP7\n');
fprintf(fid, '\n');

for i = 1:size(Node,1)
    fprintf(fid, 'N,%i,%d,%d,%d,,,,\n', Node(i,1),...
        Node(i,2),Node(i,3),Node(i,4));
end
fprintf(fid, '\n');

fprintf(fid, 'ET,1,SHELL181\n');
fprintf(fid, '\n');

fprintf(fid, 'MPTEMP,,,,,,,,  \n');
fprintf(fid, 'MPTEMP,1,0  \n');

fprintf(fid, 'MPDATA,EX,1,,%d  \n',FE_mesh.E);

fprintf(fid, 'MPDATA,PRXY,1,,%d \n',FE_mesh.v);
fprintf(fid, '\n');

fprintf(fid, 'sect,1,shell,,Shell \n');
fprintf(fid, 'secdata, %d,1,0.0,3   \n',FE_mesh.t);
fprintf(fid, 'secoffset,MID   \n');
fprintf(fid, 'seccontrol,,,, , , ,\n');
fprintf(fid, '\n');

for i = 1:size(Ele,1)
    fprintf(fid, 'EN,%i,%i,%i,%i,%i\n', Ele(i,1),...
        Ele(i,2),Ele(i,3),Ele(i,4),Ele(i,5));
end
fprintf(fid, '\n');
```

```
fprintf(fid, 'elist,all,,,0,0 \n');
fprintf(fid, 'TYPE,   1   \n');
fprintf(fid, 'MAT,        1\n');
fprintf(fid, 'REAL,   \n');
fprintf(fid, 'ESYS,       0   \n');
fprintf(fid, 'SECNUM,   1 \n');
fprintf(fid, '\n');

fprintf(fid, '/SOLU\n');
fprintf(fid, '\n');

fprintf(fid, 'ANTYPE,0\n');
fprintf(fid, '\n');

fprintf(fid, '\n');
```

Pressure

```
% fprintf(fid, 'ESEL,ALL,,,,,,\n');
% fprintf(fid, 'SFE,ALL,1,PRES, ,-1000000, , ,  \n');
```

Force

```
fprintf(fid, 'FLST,2,5,1,ORDE,2 \n');
fprintf(fid, 'FITEM,2,%i   \n',size(Node,1)-FE_mesh.Div_Cir+1);      %From
fprintf(fid, 'FITEM,2,%i \n',-size(Node,1));                         %To node
fprintf(fid, '!* \n');
fprintf(fid, '/GO \n');
fprintf(fid, 'F,P51X,FZ,%f  \n',-FE_mesh.UnitLoad);
fprintf(fid, ' \n');
```

Boundary conditions Fixed Bottom

```
fprintf(fid, 'FLST,2,5,1,ORDE,2  \n');
fprintf(fid, 'FITEM,2,1 \n');
fprintf(fid, 'FITEM,2,%i \n',-FE_mesh.Div_Cir);
fprintf(fid, '!*  \n');
fprintf(fid, '/GO \n');
fprintf(fid, 'D,P51X, ,0, , , ,UX,UY,UZ, , , \n');
fprintf(fid, ' \n');
```

Boundary conditions All Bottom

```
% fprintf(fid, 'FLST,2,5,1,ORDE,2  \n');
% fprintf(fid, 'FITEM,2,1 \n');
```

```
% fprintf(fid, 'FITEM,2,%i \n',-FE_mesh.Div_Cir);
% fprintf(fid, '!* \n');
% fprintf(fid, '/GO \n');
% fprintf(fid, 'D,P51X,UX, , , , ,UY,UZ,ROTX,ROTY,ROTZ, \n')
% fprintf(fid, ' \n');
```

Boundary conditions Fixed Top

```
fprintf(fid, 'FLST,2,5,1,ORDE,2 \n');
fprintf(fid, 'FITEM,2,%i    \n',size(Node,1)-FE_mesh.Div_Cir+1);      %From
fprintf(fid, 'FITEM,2,%i \n',-size(Node,1));                          %To node
fprintf(fid, '!* \n');
fprintf(fid, '/GO  \n');
fprintf(fid, ' D,P51X,UX,0, , , ,UY,,,, ,   \n');
```

Boundary conditions All Top

```
%fprintf(fid, 'FLST,2,5,1,ORDE,2 \n');
%fprintf(fid, 'FITEM,2,%i    \n',size(Node,1)-FE_mesh.Div_Cir+1);      %From
%fprintf(fid, 'FITEM,2,%i \n',-size(Node,1));                          %To node
%fprintf(fid, '!* \n');
%fprintf(fid, '/GO  \n');
%fprintf(fid, ' D,P51X,UX,0, , , ,UY,ROTX,ROTY,ROTZ, ,   \n');
%fprintf(fid, ' \n');
```

Simulation

```
fprintf(fid, 'FINISH   \n');
fprintf(fid, '/PREP7   \n');


fprintf(fid, '/NOPR    \n');
fprintf(fid, 'KEYW,PR_SET,1    \n');
fprintf(fid, 'KEYW,PR_STRUC,1  \n');
fprintf(fid, 'KEYW,PR_THERM,0  \n');
fprintf(fid, 'KEYW,PR_FLUID,0  \n');
fprintf(fid, 'KEYW,PR_ELMAG,0  \n');
fprintf(fid, 'KEYW,MAGNOD,0    \n');
fprintf(fid, 'KEYW,MAGEDG,0    \n');
fprintf(fid, 'KEYW,MAGHFE,0    \n');
fprintf(fid, 'KEYW,MAGELC,0    \n');
fprintf(fid, 'KEYW,PR_MULTI,0  \n');
fprintf(fid, '/GO  \n');

fprintf(fid, '! /COM,    \n');
fprintf(fid, '! /COM,Preferences for GUI filtering have been set to display: \n');
fprintf(fid, '! /COM,  Structural     \n');
```

```
fprintf(fid, 'FINISH   \n');
fprintf(fid, '/SOL    \n');
fprintf(fid, 'ANTYPE,0 \n');
fprintf(fid, 'PSTRES,1 \n');
fprintf(fid, '! /STATUS,SOLU \n');
fprintf(fid, 'SOLVE     \n');

fprintf(fid, 'FINISH   \n');
fprintf(fid, '/SOLUTION    \n');
fprintf(fid, 'ANTYPE,1 \n');

fprintf(fid, 'BUCOPT,LANB,1,0,0,CENTER \n');
fprintf(fid, '! /STATUS,SOLU \n');
fprintf(fid, 'SOLVE     \n');
fprintf(fid, 'FINISH    \n');
fprintf(fid, '/POST1    \n');
fprintf(fid, 'SET,FIRST     \n');
```

Output results

```
fprintf(fid, 'FINISH  \n');
fprintf(fid, '/POST26 \n');
fprintf(fid, 'FILE,''file'',''rst'',''.''  \n');
fprintf(fid, '/UI,COLL,1  \n');
fprintf(fid, 'NUMVAR,200  \n');
fprintf(fid, 'SOLU,191,NCMIT  \n');
fprintf(fid, 'STORE,MERGE \n');
fprintf(fid, 'FILLDATA,191,,,,1,1 \n');
fprintf(fid, 'REALVAR,191,191 \n');
fprintf(fid, '! Save time history variables to file test/12345678.csv\n');
fprintf(fid, '*CREATE,scratch,gui \n');
fprintf(fid, '*DEL,_P26_EXPORT\n');
fprintf(fid, '*DIM,_P26_EXPORT,TABLE,1,1  \n');
fprintf(fid, 'VGET,_P26_EXPORT(1,0),1 \n');
fprintf(fid, '/OUTPUT,''%i'',''csv'',''Raw\\%i''\n',m,serial);
fprintf(fid, '*VWRITE,''TIME''  \n');
fprintf(fid, '%%C  \n');
fprintf(fid, '*VWRITE,_P26_EXPORT(1,0) \n');
fprintf(fid, '%%G  \n');
fprintf(fid, '/OUTPUT,TERM \n');
fprintf(fid, '*END\n');
fprintf(fid, '/INPUT,scratch,gui  \n');


fclose(fid);

movefile(sprintf('%ia%i.inp',serial,m),'..\OFF600\Model');
```

end

```
load('..\OFF600\Data\ser.mat','serial')
load('..\OFF600\Data\FE_mesh.mat','FE_mesh')
fid = fopen(sprintf('%i-%s.inp',serial,date), 'w');
fprintf(fid, '/BATCH \n');
fprintf(fid, '\n');
fprintf(fid, '/TITLE,OneFile \n');
fprintf(fid, '\n');
q=char(39);
str=['/INPUT' ',' '''%ia%i''' ',' '''inp''' ',' '''../Model'''];
for m=1:FE_mesh.Num_Sim;
fprintf(fid, str,serial,m);
fprintf(fid, '\n');
end

fclose(fid);

mkdir('..\OFF600\Ansyspath\Raw',sprintf('%i',serial));
movefile(sprintf('%i-%s.inp',serial,date),'..\OFF600\Ansyspath');
movefile('..\OFF600\Data\FE_mesh.mat',sprintf('..\\OFF600\\Ansyspath\\Raw\\%i\\FE_mesh_%i.mat',se
rial,serial));
```

Published with MATLAB® R2021b

Appendix C

Read results from ANSYS simulation

```matlab
load('..\OFF600\Data\ser.mat','serial')
serial=270;        %Override serial
load(sprintf('..\\OFF600\\Ansyspath\\Raw\\%i\\FE_mesh_%i.mat',serial,serial))
```

```matlab
%Override
% FE_mesh.Div_Cir=80;
% FE_mesh.Div_Len=200;
% FE_mesh.Dia=0.1016*2;

%Override
% FE_mesh.Num_Sim=20;

Res=zeros(FE_mesh.Num_Sim,2);
```

```matlab
for i=1:size(Res,1);
    A=importdata(sprintf('..\\OFF600\\Ansyspath\\Raw\\%i\\%i.csv',serial,i));

    Res(i,1)=A.data;
    Res(i,2)=i*100/FE_mesh.Num_Sim;
end

Res=sort(Res);
```

```matlab
delete(findall(gcf,'type','annotation'))
Mean=mean(Res(:,1));
SD=std(Res(:,1));
PL=plot(Res(:,1),Res(:,2));
xlabel('Buckling load (N)');
ylabel('%');
Text={[sprintf('Coefficients: SD= %f * Mean', FE_mesh.Variance)],[sprintf('Number of simulations
= %i',FE_mesh.Num_Sim)],[sprintf('Mean = %.6g (N)', Mean)],[sprintf('Standard deviation = %.6g
(N)',SD)],[sprintf('Lowest 10^-3 = %.6g (N)',Mean-(4*SD))]};
title(sprintf('CDF with mesh size: %.3g m * %.3g m
',FE_mesh.L/FE_mesh.Div_Len,FE_mesh.Dia*pi/FE_mesh.Div_Cir));
annotation('textbox', [0.15, 0.75, 0.36, 0.15], 'String', Text);

if exist(sprintf('..\\OFF600\\Results\\Results-%i.xlsx',serial)) ==2;

else
    xlswrite(sprintf('..\\OFF600\\Results\\Results-%i.xlsx',serial),Res);
    saveas(PL,sprintf('..\\OFF600\\Ansyspath\\Raw\\%i\\Fig_%i.fig',serial,serial));
end
```

Published with MATLAB® R2021b

Appendix D

Create a probability plot, as well as get Gumbel parameters

```matlab
clear

load('..\OFF600\Data\ser.mat','serial')
serial=270;

load(sprintf('..\\OFF600\\Ansyspath\\Raw\\%i\\FE_mesh_%i.mat',serial,serial))

prob=readtable(sprintf('..\\OFF600\\Results\\Results-%i.xlsx',serial));
prob=prob.Var1;

dist=fitdist(prob,'ExtremeValue');
mu=dist.mu;
sigma=dist.sigma;
x=1430000000;


Lin=(1/FE_mesh.Num_Sim):(1/FE_mesh.Num_Sim):1;

y=evpdf(x,mu,sigma);


ev =sigma^-1*exp((x-mu)/sigma)*exp(-exp((x-mu)/sigma));

% test2=proflik('extremevalue',)

PL=figure
probplot('extreme value',prob(:,1))
legend('Best fit','Test data','Location','best')

saveas(PL,sprintf('..\\OFF600\\Ansyspath\\Raw\\%i\\Probplot_%i.fig',serial,serial));
```

Published with MATLAB® R2021b

Appendix E

Table 7: Fourier coefficients for A7 shell

A_{kl} coefficients

	L=	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
k=	0	0.018	-0.005	-0.391	-0.456	0.098	0.19	0.106	-0.044	0.002	0.036	-0.049	0.027	0.006	-0.004	-0.006
k=	1	0.067	0.007	-0.446	-0.052	0.054	0.077	0.018	-0.045	-0.044	-0.005	0.04	0.029	0.005	0.02	-0.005
k=	2	-0.016	0.021	0.021	0.024	-0.005	0.011	-0.005	0.007	-0.027	-0.029	0.033	-0.012	-0.008	0.006	0.003
k=	3	-0.018	-0.003	-0.043	-0.01	0.005	0.007	0.012	-0.008	-0.021	0.003	0.011	-0.018	-0.002	0.004	0.003
k=	4	-0.04	-0.001	0.006	0.001	0.003	0.003	0.005	-0.002	-0.008	0.003	-0.002	-0.007	0	0	0.003
k=	5	-0.003	-0.009	-0.022	-0.003	0.002	0.004	0.002	-0.003	-0.002	0	0.003	0	-0.002	0.002	0
k=	6	-0.032	0	0.003	0.004	-0.003	0	0	0	-0.002	-0.002	0	0	0	0	0
k=	7	-0.005	-0.005	-0.008	-0.002	0	0.001	0.003	-0.002	-0.003	0.002	0	0	0.001	0.002	0.001

B_{kl} coefficients

	L=	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
k=	0		-0.002	-0.241	0.155	0.51	0.132	-0.165	-0.077	0.008	0.018	-0.03	0.006	0.016	-0.028	0.013
k=	1		-0.002	-0.342	-0.036	0.128	-0.007	-0.143	-0.035	0.076	0.035	-0.024	0.032	0	-0.024	0.008
k=	2		0.005	-0.001	-0.005	-0.015	-0.021	-0.026	-0.042	0.007	0.008	0.012	-0.011	-0.007	0.031	-0.01
k=	3		0.011	-0.038	-0.012	0.013	0.006	-0.028	-0.013	0.013	0.003	0	-0.009	0.009	0.014	-0.025
k=	4		0.003	0.004	0	-0.002	-0.001	-0.014	-0.007	0.003	0.002	0	-0.006	0.009	0	-0.006
k=	5		-0.003	-0.017	0.003	0.006	0	-0.007	-0.003	0.003	0.003	-0.001	0	0	0	-0.003
k=	6		0.003	0.001	0	-0.003	-0.002	-0.004	-0.004	0	0	0	-0.001	0	0.002	0
k=	7		0.004	-0.007	-0.002	0.004	0	-0.008	0	0.003	0	0	0	0.002	0	-0.003

C_{kl} coefficients

	L=	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
k=	1	0.037	-0.016	-0.506	-0.588	0.127	0.236	0.136	-0.059	0.015	0.057	-0.076	0.04	0.011	-0.008	-0.009
k=	2	0.066	0.009	-0.348	-0.038	0.042	0.06	0.008	-0.033	-0.026	-0.006	0.028	0.034	0.005	0.014	-0.006
k=	3	0.03	0.013	-0.149	-0.17	0.035	0.084	0.037	-0.011	-0.015	-0.008	0.006	0.006	-0.003	0.002	-0.002
k=	4	0.012	0.008	-0.156	-0.022	0.02	0.027	0.013	-0.018	-0.027	0	0.018	-0.003	0	0.007	0
k=	5	0.006	0	-0.082	-0.099	0.025	0.048	0.026	-0.01	-0.012	0.003	-0.003	-0.001	0	0	0
k=	6	0.011	0	-0.1	-0.012	0.012	0.018	0.006	-0.01	-0.014	-0.002	0.011	0.001	-0.002	0.005	-0.001
k=	7	0.007	-0.003	-0.052	-0.061	0.013	0.03	0.016	-0.005	-0.006	0	-0.001	0.001	0	0	0

D_{kl} coefficients

	L=	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
k=	1		-0.005	-0.305	0.198	0.653	0.177	-0.196	-0.078	0.007	0.02	-0.043	0.013	0.026	-0.05	0.022
k=	2		-0.007	-0.264	-0.024	0.1	-0.009	-0.103	-0.023	0.056	0.027	-0.02	0.031	-0.005	-0.028	0.02
k=	3		0	-0.103	0.06	0.199	0.04	-0.077	-0.058	0.007	0.012	-0.003	-0.002	-0.002	0.011	0
k=	4		0.009	-0.124	-0.021	0.045	0.002	-0.059	-0.019	0.03	0.011	-0.007	0.004	0.007	0.003	-0.012
k=	5		-0.001	-0.055	0.033	0.113	0.026	-0.051	-0.031	0.005	0.008	-0.004	-0.005	0.009	0	-0.003
k=	6		-0.002	-0.079	-0.006	0.028	0	-0.032	-0.012	0.017	0.009	-0.005	0.004	0.001	-0.002	-0.004
k=	7		0	-0.037	0.023	0.007	0.016	-0.031	-0.021	0.003	0.004	-0.002	-0.002	0.003	0.002	0