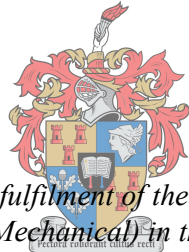


Combined Structural and Electromagnetic Analysis of Dish Reflector Antennas

by
Tristan Stuart Steele



*Thesis presented in partial fulfilment of the requirements for the degree
of Master of Engineering (Mechanical) in the Faculty of Engineering at
Stellenbosch University*

UNIVERSITEIT
iYUNIVESITHI
STELLENBOSCH
UNIVERSITY

100
1918 - 2018

Supervisor: Dr. M. P. Venter
Co-supervisor: Prof. D. B. Davidson & Dr. D. J. Ludick

March 2018

Declaration

By submitting this thesis electronically, I declare that the entirety of the work contained therein is my own, original work, that I am the sole author thereof (save to the extent explicitly otherwise stated), that reproduction and publication thereof by Stellenbosch University will not infringe any third party rights and that I have not previously in its entirety or in part submitted it for obtaining any qualification.

Date: March 2018

Copyright © 2018 Stellenbosch University
All rights reserved.



UNIVERSITEIT • STELLENBOSCH • UNIVERSITY
jou kennisvennoot • your knowledge partner

Plagiaatverklaring / Plagiarism Declaration

- 1 Plagiaat is die oorneem en gebruik van die idees, materiaal en ander intellektuele eiendom van ander persone asof dit jou eie werk is.
Plagiarism is the use of ideas, material and other intellectual property of another's work and to present it as my own.
- 2 Ek erken dat die pleeg van plagiaat 'n strafbare oortreding is aangesien dit 'n vorm van diefstal is.
I agree that plagiarism is a punishable offence because it constitutes theft.
- 3 Ek verstaan ook dat direkte vertalings plagiaat is.
I also understand that direct translations are plagiarism.
- 4 Dienooreenkomstig is alle aanhalings en bydraes vanuit enige bron (ingesluit die internet) volledig verwys (erken). Ek erken dat die woordelike aanhaal van teks sonder aanhalingstekens (selfs al word die bron volledig erken) plagiaat is.
Accordingly all quotations and contributions from any source whatsoever (including the internet) have been cited fully. I understand that the reproduction of text without quotation marks (even when the source is cited) is plagiarism.
- 5 Ek verklaar dat die werk in hierdie skryfstuk vervat, behalwe waar anders aangedui, my eie oorspronklike werk is en dat ek dit nie vantevore in die geheel of gedeeltelik ingehandig het vir bepunting in hierdie module/werkstuk of 'n ander module/werkstuk nie.
I declare that the work contained in this assignment, except where otherwise stated, is my original work and that I have not previously (in its entirety or in part) submitted it for grading in this module/assignment or another module/assignment.

Studentenommer / Student number	Handtekening / Signature
Voorletters en van / Initials and surname	Datum / Date

Abstract

Combined Structural and Electromagnetic Analysis of Dish Reflector Antennas

T. S. Steele

*Department of Mechanical and Mechatronic Engineering,
University of Stellenbosch,
Private Bag X1, Matieland 7602, South Africa.*

Thesis: MEng (Mech)

March 2018

In recent scientific endeavours, such as the Square Kilometre Array (SKA) project, significant attention (and funding) is being allocated to the design and analysis of radio telescope antenna systems, especially dish reflectors. For radio astronomy the accuracy requirements are very strict, and small changes in the shape of these antennae could result in these specifications no longer being met. In this study, a workflow process was developed to aid in the analysis of the radio telescopes while they are subject to gravitational deformation. The workflow describes procedures for performing coupled structural and electromagnetic analyses, as well as the procedures for sharing the simulation models between the respective solvers. Finite element analysis was used for the structural analysis and the method of moments and multilevel fast multipole method were used for the electromagnetic analysis. Simulations were run using the workflow process and the results compared to approximate analytical equations, where a good correlation was found. A numerical optimisation study was conducted on a simple primary-fed reflector antenna model to demonstrate how the workflow could be used in such an analysis. A case study of a real-life Cassegrain antenna was performed, and the results of these simulations were found to be similar to those of a previous study.

Uittreksel

Combined Structural and Electromagnetic Analysis of Dish Reflector Antennas

T. S. Steele

*Departement Meganiese en Megatroniese Ingenieurswese,
Universiteit van Stellenbosch,
Privaatsak X1, Matieland 7602, Suid Afrika.*

Tesis: MIng (Meg)

Maart 2018

In onlangse wetenskaplike pogings, soos die Square Kilometer Array (SKA) -projek, word aansienlike aandag (en befondsing) toegewys aan die ontwerp en analise van radioteleskoop-antennestelsels, veral reflektore. Vir radio-astronomie is die akkuraatheidsvereistes baie streng. Klein veranderinge in die vorm van hierdie antennes se gevolg kan wes dat hierdie spesifikasies nie meer nagekom word. In hierdie studie was 'n werkvloeioproses ontwikkel om te help met die analise van die radioteleskope, terwyl hulle onderworpe is aan gravitasie vervorming. Die werkstroom beskryf prosedures vir die uitvoer van gekoppelde strukturele en elektromagnetiese ontledings, asook die prosedures vir die deel van die simulasiemodelle tussen die onderskeie oplossers. Eindige element analise was gebruik vir die strukturele analise, en die metode van momente en multilevel vinnige multipol metode was gebruik vir die elektromagnetiese analise. Simulasies was uitgevoer deur die werkvloeioproses. Die resultate vergelyk met benaderde analitiese vergelykings, waar 'n goeie korrelasie gevind was. 'N numeriese optimaliseringsstudie was uitgevoer op 'n eenvoudige primêre-gevoed-reflektor antenna model om te demonstreer hoe die werkstroom gebruik kan word in so n analise. 'N Gevallestudie van 'n werklike Cassegrain-antenna was uitgevoer, en die resultate van hierdie simulاسies was gelykstaande aan dié van 'n vorige studie.

Acknowledgements

I would like to express my sincere gratitude to the following people and organisations ...

- My supervisors, Dr. M. P. Venter, Prof. D. B. Davidson, and Dr. D. J. Ludick, for all their assistance throughout the course of this project. Their knowledge and insight was invaluable.
- Altair Stellenbosch (especially Mr N. Minnaar) for all their prompt replies to queries and the assistance provided with regards running HyperMesh with TCL scripts.
- Mr C. Möller for his assistance and advice with using Stellenbosch's high performance cluster.
- SKA South Africa, the S.A. Research Chairs Initiative of the Department of Science and Technology and the S.A. National Research Foundation for the financial support granted to me.

Contents

Declaration	i
Abstract	iii
Uittreksel	iv
Acknowledgements	v
Contents	vi
List of Figures	viii
List of Tables	xi
1 Introduction	1
1.1 Background	1
1.2 Research objectives	3
1.3 Significance of research	5
1.4 Chapter overview	5
2 Literature and Background	6
2.1 The Square Kilometre Array (SKA) Project	6
2.2 Reflector antenna radio telescopes	7
2.3 Electromagnetic performance parameters	11
2.4 Mechanical performance parameters	14
3 Combined Structural and Electromagnetic Simulation Approach	16
3.1 Solution methods used	16
3.2 Development of the workflow process	18
3.3 Workflow process	20
3.4 Feed transformation calculations	24
3.5 Verification against analytical approximations	26
3.6 Refinement of workflow	33
3.7 Summary	34

<i>CONTENTS</i>	vii
4 Optimisation Study	36
4.1 Initial optimisation investigations	36
4.2 Parametrisation of workflow	40
4.3 Optimisation objectives, variables, and constraints	41
4.4 Simulation approach	46
4.5 Results	50
4.6 Summary	54
5 Case Study: A 32 m Cassegrain Reflector	56
5.1 Antenna geometry	57
5.2 Simulation model	58
5.3 Simulation results	63
5.4 Summary	66
6 Conclusions and Recommendations	68
6.1 Summary of work done	68
6.2 Recommendation for Future Work	69
List of References	71
Appendices	74
A Applying the workflow on a high-performance cluster	A.1
B Scripts	B.1
B.1 Python script to calculate feed transformations	B.1
B.2 TCL script to export deformed mesh	B.7
B.3 TCL script for VisualDOC	B.10
C Response surface definitions	C.1
D Deformation plots of the Ghana 32 m antenna's main reflector	D.1

List of Figures

1.1	A dish reflector model before (left) and after (right) a deformation with the radiation patterns plotted below. It can be seen that there is a significant difference in these radiation patterns.	2
2.1	Examples of reflector antennas (Kildal, 2015), page 301.	8
2.2	Primary-fed reflector antenna.	9
2.3	Cassegrain reflector antenna.	9
2.4	Gregorian reflector antenna.	10
2.5	Types of antenna mounts (Garrett, 2015).	10
2.6	Beamwidth.	12
2.7	A far field pattern with the peak gain or main beam indicated as well as two of the side lobes. It can be seen that one of the side-lobes is larger than the other, this is the coma-lobe. The peak gain is also not at 0° , this offset is a pointing error.	13
3.1	Tree diagram representing the interaction between the subdivided cubes in an MLFMM simulation.	18
3.2	Dish and feed model used in the development of the workflow process.	18
3.3	Load case applied to aid in development of workflow process. The arrows are two equal forces applied at the periphery of the dish as shown	19
3.4	Antenna models used to aid in feed update automation. The different strut configurations will cause the feed to deform in predictable ways so that the focus could be on a few degrees of freedom at a time.	20
3.5	Flow diagram of the developed workflow.	21
3.6	Comparison of deformation between linear triangular elements and quadratic triangular elements. It can be seen that there is little difference between the two plots.	23
3.7	Nodes that are used in Python script to calculate the feed rotations.	24
3.8	A representation of the illumination taper for a normalised beam pattern.	27

3.9	A representation of axial defocus, the feed is moved from its original location to the location of the yellow model. This defocusing can also be in the opposite direction.	28
3.10	Simulation vs. analytical approximation for axial defocus. The simulated results lie between the approximations for uniform and fully tapered illumination. This is what was expected.	29
3.11	A representation of lateral defocus, the feed is moved from its original location to the location of the yellow model.	29
3.12	Simulation vs. analytical approximation for lateral defocus. Again the simulated fit well with what the analytical equations predict.	31
3.13	Relative Gain vs. RMS Surface Error with a Gaussian Distribution.	32
3.14	Relative Gain vs. RMS Surface Error For a Deformation.	32
3.15	A flow diagram representing how the workflow is automated.	33
4.1	Two models with different FoD ratios. The image and the left has the diameter and focal length on the model indicated. In the image on the right the FoD ratio has been increased while keeping the diameter constant.	37
4.2	Relative gain vs. gravitational load angle for various FoD ratios. It can be seen that the FoD ratio has little effect on the relative gain.	38
4.3	This represents two gravitational load cases with the dish model at 0° and 90° . The red arrow represents the equivalent force due to weight of the structure acting on its centre of mass. It can be seen that this will result in an additional moment being introduced for the model on the right (the 90° load case).	38
4.4	The simple dish model.	39
4.5	Gain vs. Backing Structure Height.	39
4.6	Primary-fed reflector antenna generated by the TCL script. This model will change between each iteration of the optimisation.	41
4.7	A far field pattern with the main beam and two side-lobes indicated. The side-lobe level is the difference between the gain of the main beam and the gain of the side-lobes. It can also be seen that the main beam is not at 0° , this is because there is a pointing error in this far field.	43
4.8	Variables of the Backing Structure.	44
4.9	Representation of moving strut locations.	46
4.10	Representation of the Face-centred Central Composite Design, design point selection for three input variables. Each dot represents a design point that will be tested.	48
4.11	Mass Response Surface.	49
4.12	Predicted optimums of input variables and the upper and lower bounds compared to the iteration of response surface. It can be seen that the input variables converge on a point.	52

4.13	A contour plot of the gain vs. thickness and t_0 of the backing structure. The coloured lines represent the boundaries determined by the constraints. In this case, the feasible region is a line where the red and blue lines overlap and is within the bounds of the green line.	52
4.14	The results of the tradeoff investigation show that there is benefit to increasing the allowable mass, but it has a diminishing return.	54
5.1	A photo of the Ghana 32 m Communication Telescope.	57
5.2	Variables used to define the Cassegrain reflector geometry.	58
5.3	This demonstrates what is meant by a mid-surface mesh. This black outline represents the solid geometry that is being approximated by the mid-surface mesh.	59
5.4	The final structural FEA model of the Ghana 32 m antenna. This is the model with an elevation angle of 0°	61
5.5	Comparison of simulation results of a Cassegrain antenna model with a feed model or a spherical mode source. The spherical mode source results in higher peak gain and a narrower HPBW. This is to be expected.	62
5.6	Comparison of the far field pattern for a solid main reflector surface and the panelled surface in the structural FEA model. It can be seen the the difference in the pattern is negligible.	64
5.7	The far field pattern for the Ghana dish at the various elevation angles as well as the far field pattern of the undeformed antenna. The most distinct difference between the patterns is the side-lobe levels.	65
D.1	Deformation plot of the main reflector at 0°	D.1
D.2	Deformation plot of the main reflector at 30°	D.2
D.3	Deformation plot of the main reflector at 90°	D.2
D.4	Deformation plot of the main reflector at -30°	D.3
D.5	Deformation plot of the main reflector at -90°	D.3

List of Tables

3.1	Gain loss functions for lateral defocus.	30
4.1	Parameterised geometric variables.	40
4.2	VisualDOC step sizes and stopping criteria.	50
4.3	Predicted optimum compared to simulation results for the first response surface optimisation.	51
4.4	Predicted optimum compared to simulation results for the reduced design space.	53
4.5	In the table, the results of simulation for the candidate model at different load angles are shown. it can be seen that for these load angles the candidate model does not have a degraded performance compared to the load case it was optimised for.	53
5.1	Geometry dimensions.	57
5.2	A comparison of the maximum and minimum deformations experienced by the main reflector for the original and the modified model. It can be seen that these values are very similar.	64
5.3	A comparison of the gain, side-lobe level, and pointing error determined in the simulations for the different elevation angle, as well as the undeformed case. It can be seen that the Gain does not change significantly, however, there is a significant drop in the side-lobe levels.	66
C.1	First response surface model coefficients	C.1
C.2	First response surface model coefficients	C.2
C.3	Third response surface model coefficients	C.2

Chapter 1

Introduction

In this section the background to the project will be discussed, the research objectives of the thesis, scope of the work, the exclusions/limitations, and the assumptions made will be stated. The significance of the research will be discussed and lastly, a chapter overview will be given.

1.1 Background

Projects such as the Square Kilometre Array (SKA) (SKA-SA, 2012) have led to a significant focus on the design and analysis of radio telescopes. The SKA is supporting studies to help further their understanding of the effects that the operating conditions could have on their antennas. This study looks at a multidisciplinary approach, coupling structural and electromagnetic simulations. This research is a continuation of the work performed by Ludick *et al.* (2015). Their study was a combined structural and electromagnetic analysis of a large dish reflector antenna, specifically the MeerKAT antenna (SKA-SA, 2017) used by the SKA. In their work, they found that if a deformation great enough was experienced would be a significant effect on the performance, however, they concluded that the backing structure modelled was not suitable for the electromagnetic analysis and would need more work.

The current research is applicable to the design and analysis of radio telescopes, specifically dish reflector antenna systems, as the workflow developed will help antenna engineers quantify the effects of deformation on the telescope's electromagnetic performance.

There are two types of error, correlated and uncorrelated. A correlated error is an error that has some common relationship, such as a bend in a beam due to an applied load, and uncorrelated error there is no relationship in the error, so the error is random.

To date, there has been little research performed on the effects of structural deformations, a correlated error, on a radio telescope's electromagnetic performance. A common practice by antenna designers is to use formulas that try to

approximate these effects, such as Ruze's equation (Ruze, 1966) which predicts the effect of RMS surface error on peak antenna gain, or the predictions of the effects of feed defocus in Baars (2007). While these formulae are good at predicting the effects of the specific aberration on an antenna's performance they can not predict a coupling of these effects. As will be illustrated in Chapter 3, Ruze's formula is sufficient for predicting the effects of an uncorrelated error on a dish's surface but as soon as the error is correlated the results deviate.

This could be a potential problem in radio telescope design as these formulas do not account for deformations. In the design process, it is common for a required surface accuracy to be calculated by antenna engineers, using these formulas and provide that as a specification to the mechanical engineers who must then select materials with suitable properties (Azankpo, 2017). The problem here is that deformation has not been taken into account and although an antenna may meet these specified surface accuracies when manufactured, when it deforms it may no longer meet its required performance specifications. Figure 1.1 shows a dish reflector model before and after it experiences deformation, this is for a simplified model without a backing structure so the deformation is greater than would be expected. Below each model, the far field patterns can also be seen. The image on the right is the deformed model and it can be seen that it is a warped version of the image on the left, this warping is the deformation.

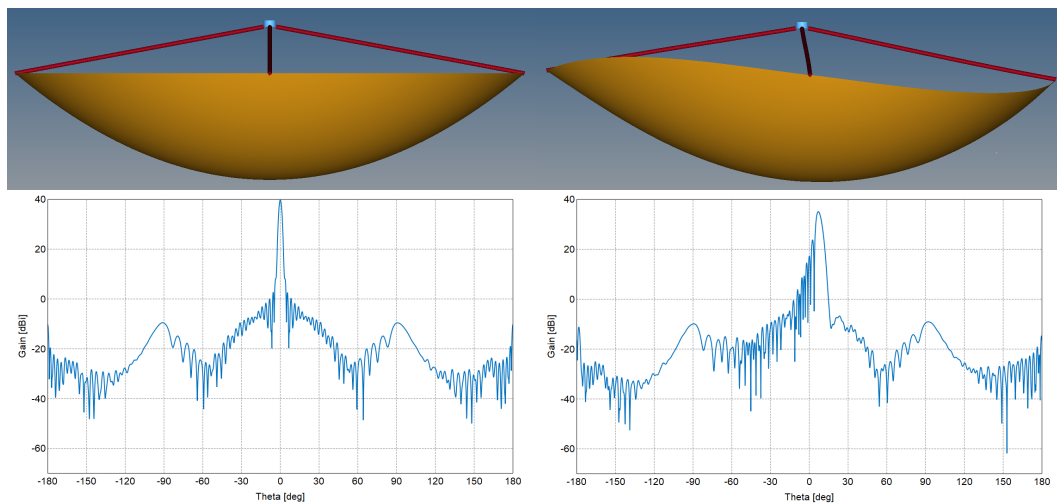


Figure 1.1: A dish reflector model before (left) and after (right) a deformation with the radiation patterns plotted below. It can be seen that there is a significant difference in these radiation patterns.

This research will show that the effects of deformation can lead to significant loss in antenna performance, and present a method for determining the effects of deformation of antenna performance. For the study the software packages; FEKO (FEKO, 2015), HyperMesh (HyperMesh, 2017), OptiStruct

(Optistruct, 2017) were selected for electromagnetic and structural analysis respectively. Parametrisation of the model was automated using TCL (TCL/TK, 2016) scripts. The entire workflow and software coupling was facilitated by bespoke Python 2.7.13 (Python, 2016) scripts.

The workflow presented here can be used to analyse an antenna under various environmental conditions, e.g. the application of gravitational, wind and thermal load conditions. The aforementioned structural load conditions lead to deformations, as will be shown later in the text. The effects of this deformation on antenna performance is quantified in this work.

1.2 Research objectives

With the increased focus on radio telescope projects such as the SKA further study into their behaviour is becoming increasingly important to facilitate the design and analysis of these structures. This study aims to develop a workflow to facilitate the analysis of radio telescope performance in terms of real-world load cases. These load cases would include gravitational, wind and thermal loading. This workflow will be applied to various test cases, including a 32 m Cassegrain reflector antenna recently added to the SKA project. The study includes the development of a numerical model of the antenna used for both the mechanical and electromagnetic modelling. Furthermore, an optimisation study is applied to illustrate the way in which a simple centre-fed dish can be improved using this combined multi-physics analysis approach.

To complete this workflow and validate it several objectives had to be met. The primary objectives of this study were:

1. Develop a process of transferring geometric information between the structural Finite Element Analysis (FEA) and Electromagnetic (EM) software.
2. Develop meaningful structural FEA models to determine the usefulness of the workflow.
3. Determine EM performance degradation due to applied loads.
4. Validate the workflow against analytical equations.
5. Perform geometry optimisations based on the full-wave combined simulation approach.
6. Perform a full structural and electromagnetic analysis of a complex antenna model.

1.2.1 Deliverables

The tasks that needed to be completed were:

1. A literature study of existing research into coupling structural and electromagnetic simulations.
2. Developing an understanding of electromagnetic and antenna analysis to a sufficient level.
3. Developing a robust method for importing and exporting geometries and meshes between solvers.
4. Developing a workflow for performing the full analysis and writing scripts to help automate this process.
5. Validating this workflow against analytical formulas that are commonly used in antenna design.
6. Performing geometry optimisations to validate that the workflow can be used to solve optimisation problems.
7. Performing an analysis of a complex antenna model to illustrate that the workflow can be applied to these complex models based on a real-life application.

1.2.2 Scope of work

The scope of this research and the assumptions made were as follows:

1. The analyses of the antenna models were limited to just gravitational load cases.
2. The results were not validated against experimental data as none was readily available and the simulations were more of a proof of concept.
3. Only in the initial validation of the workflow was a random surface error considered, in all other simulations the surfaces were assumed to be perfectly smooth before deformation.
4. For the optimisation, only a simple geometry was considered.
5. That the materials in the simulations were assumed to be homogeneous. The properties are continuous throughout the material.
6. The foundation that the 32 m Cassegrain reflector antenna stands upon and the rail it runs on were assumed to be perfectly flat and rigid.
7. The feed models were assumed to be rigid, as this vastly simplified the coupling of the structural and EM analyses, without significant loss of accuracy. This will be explained in Chapter 3.

1.3 Significance of research

During the design of radio telescopes, the approximations for the effect of aberrations, like surface error and feed defocusing, on the performance are used. These approximations do not take into account the effects that deformations will have on antenna performance. Also these equations cannot be used to easily couple the effects of the aberrations they are used for.

Structural deformations can have a significant effect on the performance of a radio telescope, more so than random surface error; this is shown in Chapter 3. Due to the lack of research into the effects of deformations, there are many benefits to this research. It can be seen as a pilot study into the many factors that could be affected by the dish deformation.

The workflow developed can also be used to predict if an antenna's design will still meet all design specifications for all expected load cases as they can be simulated to determine the effects of these load cases. This could allow for the more efficient design of these antennas. These antennas could also be designed to deform into their ideal shape if the load case they will experience is known, or at least closer to their ideal shape.

Regarding the pointing error that can be caused by these load cases, if the simulations are accurate enough, they can be used to aid in the calibration of the antenna to account for this pointing error.

1.4 Chapter overview

Literature and background to the project will be presented in Chapter 2. This will include explanations of some of the fundamentals of both structural and electromagnetic analysis, and reflector antenna systems that are used for radio astronomy.

The development and methodology of the workflow that is presented in this paper will be discussed in Chapter 3. Validation cases for the workflow will also be presented. This relates to objectives 1, 2, 3, and 4.

An optimisation study of a simple primary-fed reflector antenna will be presented in Chapter 4. This relates to objective 5.

A case study of a 32 m Cassegrain antenna will be discussed in Chapter 5. This entailed modifying an existing structural model of the antenna to meet the requirements of an EM analysis as well as a structural analysis. These requirements include the location of surface meshes. This relates to objective 6.

Finally, in Chapter 6 the study will be summarised and conclusions will be drawn. Recommendations for future work will also be presented.

Chapter 2

Literature and Background

2.1 The Square Kilometre Array (SKA) Project

The SKA project is an international effort with the aim to build the world's largest and most sensitive radio telescope (SKA-SA, 2012). It will operate over a wide range of frequencies, 70 MHz to 30 GHz (Dewdney *et al.*, 2009), and its size will make it 50 times more sensitive, and up to 10 000 times faster (in terms of its survey speed) than the best radio telescopes of today.

It will help scientists understand how the universe itself evolved, how stars and galaxies form and change, and what "dark matter" really is (SKA-SA, 2012).

As the name suggests the telescope array will cover a square kilometre, the estimated collecting area required is $\approx 5.7 \times 10^5 \text{ m}^2$ (Dewdney *et al.*, 2009).

There will be three different arrays built, a sparse aperture antenna array, a dense aperture antenna array and a large reflector antenna array, to analyse low, medium and high-frequency waves respectively. These ranges are 70 to 500 MHz for low frequency, 500 up to 1500 MHz for medium frequency and from as low as 0.8 to 10 GHz for high frequency. The SKA will eventually be able to produce images and other data over wavelengths from $\approx 4.3 \text{ m}$ to $\approx 1 \text{ cm}$, (Dewdney *et al.*, 2009). It is estimated that 2000 – 3000 15 m diameter reflector antennas will be needed (Dewdney *et al.*, 2009). The sparse aperture array will be built in the Murchison area of Western Australia and the other two will be built in the Karoo region of South Africa, (Dewdney *et al.*, 2009). These sites were selected due to their radio quietness, there is little electromagnetic noise due to their remote locations. Towns and cities inherently create electromagnetic noise because of radio stations, cell phones and other technology.

The reflector antennas have very small tolerance for error due to the high accuracy requirements to capture the required amount of information. Due to this the effect of the deformations experienced under expected operating

conditions on the performance of these antennas is of significant concern.

2.2 Reflector antenna radio telescopes

In the electromagnetic spectrum, there is a band of frequencies known as the radio window. This band of frequencies is generally considered to be from 30 MHz to 30 GHz. However, these limits are nominal and arbitrary so more extreme limits may consider this window to be from 2 MHz to 300 GHz (Kraus, 1966). Radio telescopes are instruments used to detect signals from extraterrestrial sources at these radio frequencies. These sources include stars, galaxies, and quasars (Kellermann, 2016). A quasar is an extremely remote celestial object, emitting exceptionally large amounts of energy in the form of light or radio waves (Quasar, 2017).

Due to the large range of frequencies that are observed the design, size, and configuration of the required antennas vary widely. The focus of this study is on reflector antennas, of which there are several types. Some examples are, primary-fed paraboloid antennas, symmetrical Cassegrain, offset primary-fed paraboloid reflector and spherical reflector antennas. Figure 2.1 shows some examples of these types of antennas. In this work, the prime-fed parabolic and Cassegrain reflectors are considered.

2.2.1 Parabolic reflector antennas

A reflector antenna is an antenna consisting of one or more reflecting surfaces and a feed system. A parabolic reflector antenna is one of these reflector antennas but where the cross-section of the primary, the largest, reflector surface is that of a paraboloid. The most common of these antennas come in two configurations, with a single reflector surface or dual reflector surfaces.

Single reflector antennas, often referred to as prime focus or primary-fed reflectors, have a large parabolic reflector surface and the feed is positioned at the focal point of this reflector. The advantage of these antennas is that they can be used across their full frequency range, however, access to the feed/receiver can be limited due to them being located high above the ground (Garrett, 2015). Figure 2.2 is a schematic example of a primary fed antenna.

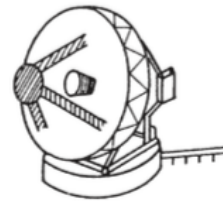
The dual reflector antennas allow for the feed/receiver to be placed closer to the ground thus providing better access. Another advantage of these dual reflector antennas is that the radiation that spills over the sub-reflector, resulting in a quantity term "spill-over" loss, is directed away from the ground. Their disadvantage is that at lower frequencies they become impractically large compared to their prime focus counter-parts (Garrett, 2015). The dual reflector antenna has a sub-reflector which is either a convex, usually hyperboloidal in shape, or concave, usually elliptical in shape. These two types are known as Cassegrain and Gregorian reflectors respectively (IEEE, 1993). In both



Primary-fed paraboloids.
Feed supported by struts (left) and self supported (right)



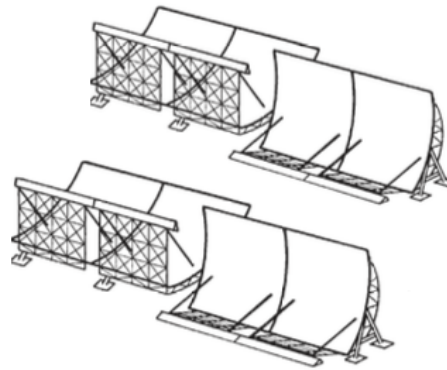
Offset primary-fed
parabolic reflector



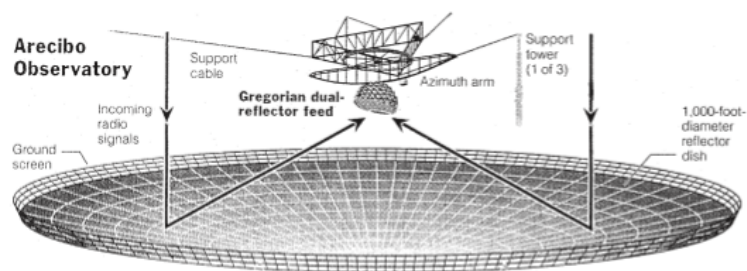
Symmetrical Cassegrain



Offset Gregorian
dual-reflector antennas



EISCAT VHF parabolic cylinder antenna



Arecibo spherical reflector with dual-reflector feed (inside enclosure)

Figure 2.1: Examples of reflector antennas (Kildal, 2015), page 301.

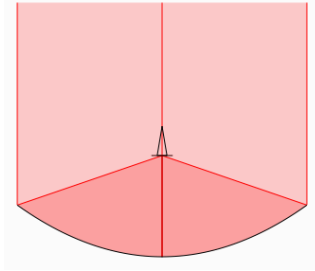


Figure 2.2: Primary-fed reflector antenna.

these cases, the feed is placed at the focal point of the sub-reflector and the sub-reflector is placed in such a way as to reflect the wave reflected by the primary reflector. Figures 2.3 and 2.4 are schematic examples of Cassegrain and Gregorian reflectors respectively.

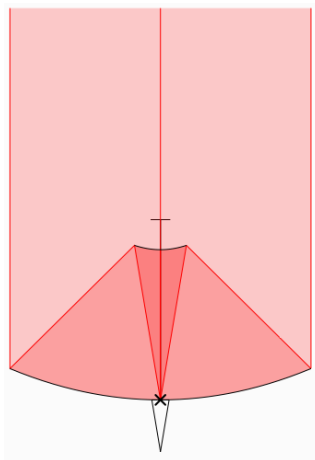


Figure 2.3: Cassegrain reflector antenna.

Offset or off-axis versions of these antennas also exist. In this case, the feed is moved to a position off-axis from the incoming signal and the shape of the reflector or sub-reflector is altered to focus the incoming radio waves at this new feed position. The reflectors still have the same type of shape as in the on-axis versions, but the reflectors are no longer rotationally symmetric. For the dual reflector systems, only the sub-reflector is altered in this way, and the sub-reflector is still located on the axis of the incoming signal.

There are dual offset reflectors in which case both the feed and sub-reflector are moved and thus both the main reflector and sub-reflector must be altered to focus on the new sub-reflector and feed locations correctly.

These offset reflectors are primarily used to reduce the effects of aperture blockage caused by the sub-reflector/feed and their support structures, and thus improves the aperture efficiency which in turn results in higher gain.

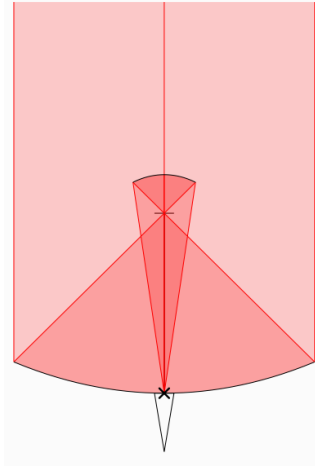


Figure 2.4: Gregorian reflector antenna.

There are disadvantages however and these include less desirable polarisation characteristics, and the asymmetry of the reflector surfaces is more complex to manufacture and thus more expensive.

2.2.2 Types of antenna mounts

There are two main methods of mounting a reflector antenna, and these are either an equatorial/polar mount which rotates around one axis, or an altitude over azimuth mount which rotate around two axes. Figure 2.5 shows these two types of mounts.

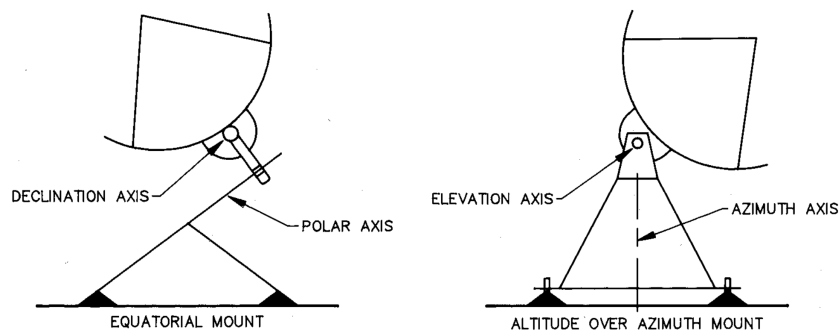


Figure 2.5: Types of antenna mounts (Garrett, 2015).

The altitude over azimuth mount is more commonly used type because they are cheaper to construct and are mechanically well-balanced. This makes them well suited for large heavy antennas. The mount allows for rotation around the azimuth/vertical axis and the elevation, horizontal axis. The main disadvantage of this mounting method is that when tracking a source in the

sky the beam rotates as the source moves across the sky, as the telescope has to be rotated around the two axes at variable rates (Garrett, 2015).

The equatorial/polar mount is designed such that the polar axis is parallel to the rotational axis of the Earth. This allows for the telescope to track a source in the sky that has diurnal motion- the apparent motion of the source due to the rotation of the Earth- by rotating at a constant rate around the polar axis. This system, however, needs heavy counterweights to counterbalance the weight of the mounted telescope and thus is less suitable for large heavy antennas. Also due to the antennas position, the accuracy of the system is more susceptible to the effects of gravity (Garrett, 2015).

2.3 Electromagnetic performance parameters

In this section, a brief summary of the electromagnetic performance parameters that were of interest in this study will be discussed.

2.3.1 Gain

The absolute gain of an antenna is described as the ratio of the radiation intensity, in a given direction, to the radiation intensity that would be obtained had the power been radiated isotropically (Balanis, 1997). This is a dimensionless term, however, it is most often presented in decibels relative to an isotropic antenna (dBi). If a direction is not stated then the gain is usually taken as in the direction of maximum radiation.

The theoretical gain of a reflector antenna is,

$$G = \eta_T \left(\frac{\pi D}{\lambda} \right)^2 \quad (2.1)$$

where D is the diameter of the main reflector, λ is the wavelength, and η_T is the total efficiency of the antenna. This efficiency is the accumulation of a number of subterms,

$$\eta_T = \eta_a \eta_S \eta_f \eta_c \quad (2.2)$$

which are the aperture efficiency, spillover efficiency, feed mismatch, and the reflector and feed conductor losses (Bird, 2015). A commonly used figure of merit for receiving antennas is the gain, G over the system noise temperature, T_{sys} (Kildal, 2015). The larger this ratio, the higher the sensitivity of the system. It was decided to focus on maximising the gain in this study.

2.3.2 Half Power Beamwidth

The half-power beamwidth (HPBW) is the angular separation between points in the radiation pattern where the magnitude has decreased by 50% or 3dB

from the peak of the main beam. The HPBW represents the area in which the relative power is more than 50% of the peak power (Minoli, 2009). Figure 2.6 is a simple representation of a radiation pattern and shows the HPBW as well as the First-Null Beamwidth (FNBW), which isn't of interest in this study.

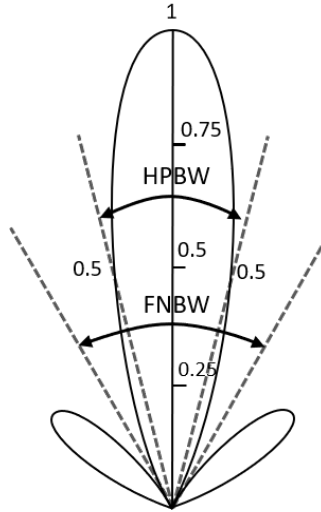


Figure 2.6: Beamwidth.

This figure shows the power normalized to the peak of the main beam, and is shown as an absolute value as opposed to in dB. This is the area in which a receiving antenna will pick up the most information or is most sensitive. The HPBW can be approximated by,

$$HPBW = k \frac{\lambda}{D} \quad (2.3)$$

where λ is the wavelength, D is the diameter of the parabolic dish and k is a factor that depends on the shape of the reflector. For a typical reflector, k equals 70 for a HPBW in degrees and 1.22 for it in radians. It is desirable to have a HPBW be as small as possible as this will improve the angular resolution of the antenna. Angular resolution is the minimum angular separation at which two equal targets can be separated when at the same range (Wolff, 2014).

2.3.3 Side-lobe level

A side lobe is a local maximum in the radiation pattern, and the main lobe is the global maximum. In Figure 2.7 the main beam, or peak gain is indicated as well as two of the side lobes. Side-lobe level refers to the difference between the gains of these side lobes and the gain of the main lobe (Balanis, 1997). The side-lobe level of the largest side-lobe is of most interest. The largest side-lobe is typically the one of the first side-lobes to either side of the main beam. In a

symmetrical radiation pattern, these side-lobe levels are the same. What is of interest here is keeping the side-lobe levels large enough that there is negligible interference in the results.

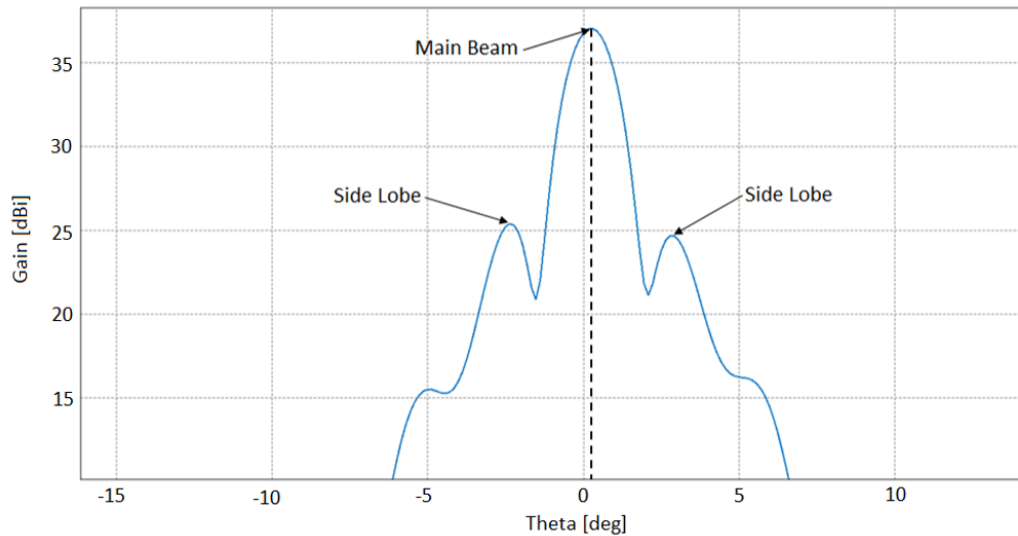


Figure 2.7: A far field pattern with the peak gain or main beam indicated as well as two of the side lobes. It can be seen that one of the side-lobes is larger than the other, this is the coma-lobe. The peak gain is also not at 0° , this offset is a pointing error.

The side lobes introduce interference into the results of the antenna by picking up radiation in the directions they are pointing, the main lobe/beam is pointed at the desired point and only the radiation from this point is of interest. This is why it is important that the side-lobe levels are all sufficiently large so that the interference they add will be negligible. A level of 20 dB is often a good target value as this represents the peak side-lobe gain that is at least 100 times smaller than the peak gain (Balanis, 1997).

2.3.4 Coma-lobe level

When a feed is displaced laterally there is a squinting on the far field pattern. This leads to the side lobe levels deforming, and the side lobe on one side of the pattern becomes larger. This is known as the coma-lobe (Baars, 2007). The coma-lobe level is then the side lobe level of this coma-lobe. In Figure 2.7 the larger side lobe on the left is the coma-lobe.

2.3.5 Pointing error

Pointing error refers to the angle at which the main beam of the antenna is pointing away from the intended direction. In Figure 2.7 the dashed line

indicated the pointing error in the far field pattern. This error needs to be very small otherwise the main beam could completely miss the target it is meant to be aiming at. At an astronomical scale, even a fraction of a degree can be significant. An example of how small these errors need to be in radio astronomy applications, the MeerKAT antenna has a maximum allowable pointing error of 25 arc-seconds or about 0.007° (SKA-SA, 2016).

2.4 Mechanical performance parameters

A brief summary of the mechanical performance parameters that were of interest will be discussed in this section.

2.4.1 Root Mean Squared surface error

In antenna design, it is common to specify a required Root Mean Squared (RMS) surface error. This surface error refers to deviations from the ideal surface that are created during the manufacture of the dish surfaces and the assembly of the antenna. These errors are unavoidable but they can be limited. This is an uncorrelated error.

RMS surface error can also refer to a weighted average of the surface error induced by some deformation to the antenna. In this case, it is a correlated error. The equation for calculating RMS surface error is,

$$\epsilon = \sqrt{\frac{\sum(x - x_1)^2}{n}} \quad (2.4)$$

where x is the path length from the focal point of the dish to some point on the ideal dish and x_1 is the path length to the same point but on the deformed dish, and n is the total number of points where this is done.

2.4.2 Deformation

Deformation refers to the amount a structure changes when it is loaded, generally given in total displacement. This total displacement is generally given as a scalar value with units metres. Figure 1.1 shows a dish reflector model before and after it experiences deformation, this has been exaggerated to make the deformation clear. Deformations can have a significant effect on the performance of an antenna as the reflector surface. For instance, the antenna can deform in such a way that the focal point changes and thus is no longer at the point where the feed is located, or parts of the incoming radiation could simply be scattered away.

2.4.3 Structural stiffness

Structural stiffness is a structure's ability to resist deformations under load. A stiffer structure will be more resistant to deformations and thus be less likely to change from its original shape. This stiffness is a product of both the material's properties and its geometry. This stiffness will vary locally throughout the structure as the geometry changes.

2.4.4 Stress

Stress is the internal distribution of forces within a material to balance and react to the loads that are applied to it (Figliola and Beasley, 2011). The units for stress are Pascal. The equation for stress is,

$$\sigma = F/A \quad (2.5)$$

where F is the force in N applied over an area, and A is that area in m^2 . Material properties which denote the strength of a material are generally given as stresses with which the material is expected to fail.

2.4.5 Strain

Strain is a dimensionless value that denotes the amount of stretch or percentage elongation a material experiences. The formula for strain is,

$$\varepsilon = \frac{\Delta l}{l} \quad (2.6)$$

where ε is the strain, Δl is the elongation in m , and l is the original length over which the elongation is experienced. If the strain is large enough the structure will experience a permanent deformation. This means that when the load applied to it is removed it will not return to its original shape but remain partially deformed, or in the extreme case will fracture.

Chapter 3

Combined Structural and Electromagnetic Simulation Approach

In this chapter, the workflow that has been developed will be discussed. The processes used to develop the workflow and the workflow process itself will be explained. The procedures used to validate the workflow will be described and the results will be presented. This workflow was developed for use with HyperMesh (HyperMesh, 2017) as the Finite Element (FE) preprocessor, OptiStruct (OptiStruct, 2017) as the FE solver, and FEKO (FEKO, 2015) as the electromagnetic (EM) solver. Python 2.7.13 (Python, 2016) was used as well to write a variety of scripts.

3.1 Solution methods used

In this section, the solution methods used to solve the structural analysis and electromagnetic analysis will be briefly explained.

3.1.1 Solution method used for structural analysis

For the structural analysis finite element analysis (FEA), also known as finite element method (FEM) is used. This is a method for numerical solution of a field problem. In this case, the distributions of displacements and stresses throughout the structure are of interest.

FEA is an analysis method in which a field variable is approximated by connecting simple interpolation functions. These interpolation functions are defined over a small region called a finite element (Cook *et al.*, 2001). These finite elements are connected at nodes and they share the same field value at these nodes. These nodes are where loads and boundary conditions are imposed.

In general, the objective of FEA for structural analysis is to solve the matrix equation,

$$\mathbf{K}\mathbf{d} = \mathbf{r} \quad (3.1)$$

where \mathbf{K} is the structure's stiffness matrix, \mathbf{d} is a vector of the nodal displacements and \mathbf{r} is the load on the nodes.

3.1.2 Solution method used for electromagnetic analysis

For the electromagnetic analysis, the method of moments (MoM) is used. In the MoM, the radiating or scattering structure is replaced by equivalent currents. This process entails discretising the structure's geometry into a number of small elements. These elements can be one-dimensional wire segments, two-dimensional triangular surface elements, three-dimensional tetrahedral elements or a three-dimensional grid. This will depend on the geometry being modelled and the applied formulation. Each of these elements then has a related basis function that defines simple approximations for the spatial variation of the unknown quantity (Davidson, 2011). In the case of MoM the unknown quantity is the surface current. A matrix equation,

$$\mathbf{Z}\mathbf{i} = \mathbf{v} \quad (3.2)$$

is then derived. \mathbf{Z} is a $N \times N$ impedance matrix that accounts for the self and mutual interactions between all the basis functions, where N is the total number of basis functions. \mathbf{i} is a vector that represents the unknown surface current distribution, and \mathbf{v} is related to the applied excitation. From this, it can be seen that the memory requirements for this method are of $\mathcal{O}(N^2)$ and the computational runtime is of $\mathcal{O}(N^3)$ as MoM is a direct method.

For this reason, it was decided to use the multilevel fast multipole method (MLFMM). This method approximates the impedance matrix \mathbf{Z} and solves for the solution vector, \mathbf{i} , by iterating until a solution is found. First the entire computational space is enclosed in a cube. This cube is then subdivided into eight smaller cubes and this is repeated recursively until the edge length of the smallest cube is approximately a quarter wavelength (van Tonder and Jakobus, 2005). Only non-empty cubes are stored. The interaction between each element and the centre of the cube it is in is determined and this is repeated recursively for each cube and the centre of the larger cube it is in. This is represented by the tree diagram in Figure 3.1. The smallest nodes represent the elements that make up the structure and the line represent an interaction that is calculated.

This leads to a memory requirement of $\mathcal{O}(N \log N)$ and run times of $\mathcal{O}(N \log^2 N)$ (Song *et al.*, 1997). Which for large values of N results in significant reduction in memory requirement and computational runtime.

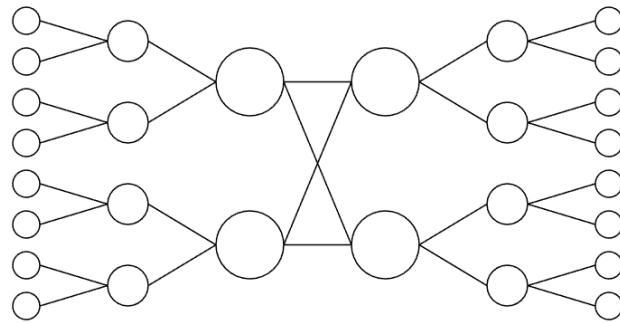


Figure 3.1: Tree diagram representing the interaction between the subdivided cubes in an MLFMM simulation.

3.2 Development of the workflow process

To develop the workflow a simple dish and feed model was initially developed, Figure 3.2 shows this model. This model had all the parameters required to perform the EM simulation defined, such as the operating frequency and the waveguide source that powers the feed. The simple model was chosen so that the focus could be on the development of the workflow process. The model was used to develop the methods for "integrating" the structural analyses and EM analyses together, allowing for a robust method to perform a structural analysis and then an EM analysis on the deformed structure. This antenna has a main reflector diameter of 12 m and all EM simulations were performed at a frequency of 1 GHz. The main dish was assumed to be made of steel.

To be able to investigate the effects of deformation on electromagnetic performance, that deformation needs to be represented in the EM model. By

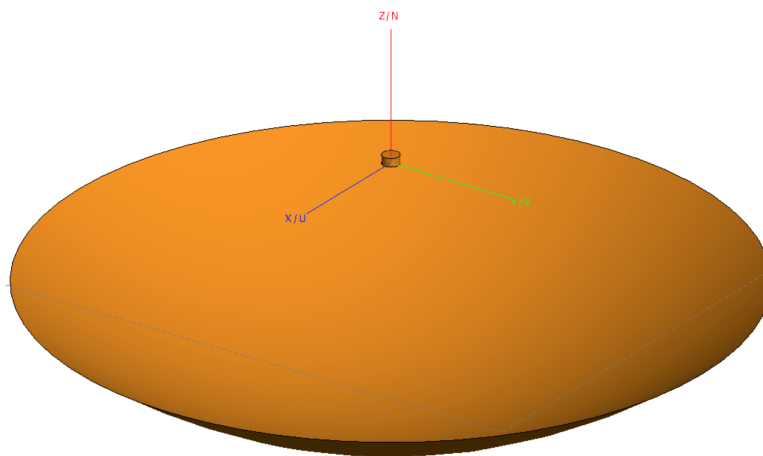


Figure 3.2: Dish and feed model used in the development of the workflow process.

using the same mesh between the structural model and the EM model there is greater certainty that deformation calculated by the structural FEA solver is accurately captured in the EM model.

First, the methods for exporting/importing the meshes and geometries between the two software packages were investigated. In HyperMesh the model was loaded with two equal forces along the periphery of the dish pointing inwards, as shown in Figure 3.3. This was to cause some deformation in the dish so that a deformed mesh could be obtained. This is an unlikely load case, but it is known in what manner the dish would deform and the deformations would be obvious. It was found that the most reliable formats for the exporting/importing of the meshes and geometries were the STL and STEP formats respectively. In this phase of the development only the dish was allowed to deform. This was for the sake of simplicity as only the exporting/importing methods were of interest.

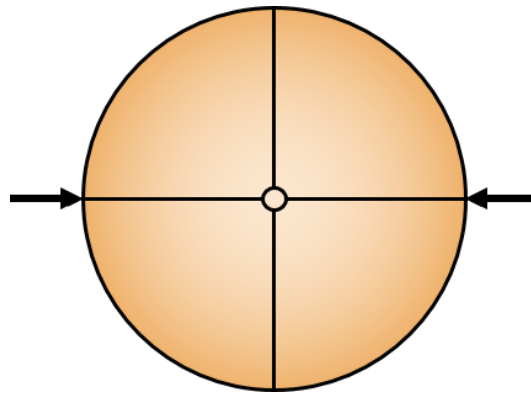


Figure 3.3: Load case applied to aid in development of workflow process. The arrows are two equal forces applied at the periphery of the dish as shown

In reality, the feed will deform in some way due to it being mounted to the same structure as the dish surface. It was assumed that the feed would only translate and rotate as a rigid body. This was achieved by giving the feed model an artificially larger Young's modulus and shear modulus in the structural model. This assumption was made because the feeds used in reflector antennas are often made of significantly thicker material than the rest of the antenna and thus it is expected to deform significantly less, it is also much smaller which will also lead to there being less deformation. This assumption made the updating of the feed's position and orientation significantly easier in the EM model as its rotation and translation could easily be calculated. A problem that was encountered here was that if the feed was not an ideal cylindrical shape the definition of its source would fail and the EM simulation would not run. Thus it was decided to have a predefined FEKO model of the feed and update that

as opposed to attempting to create a new feed model to match the deformed mesh in the structural model.

To start automating the feed update procedure four support struts were added to the model to connect the feed to the dish, thus causing the feed to displace, in the z-direction only, when the dish was deformed. Two scripts were written, one in Python and the other in EDITFEKO, to transform the feed and its waveguide port in the FEKO model by the same amount determined in the structural simulations. The functionality of these scripts will be explained in the next section.

Once the displacement of the feed in the z-direction had been automated, the model was changed slightly to allow for new translations and rotations to be included with this. This was implemented incrementally until all six degrees of movement could be automated. This was implemented as follows;

- Only one strut was used to connect the feed to the dish resulting in the feed displacing in the y and z-directions and rotating about the x-axis.
- A second strut was added so that the feed would displace in all three directions and rotate about the x and y-axes.
- Lastly four struts were used again; however, this time a torque was added to the feed so that it would rotate about the z-axis. This also resulted in a displacement in the z-direction.

Figure 3.4 shows the models used to aid in the creation of the feed update automation. After these steps had been performed the feed could be updated for all translations and rotations it may experience during deformation of the dish.

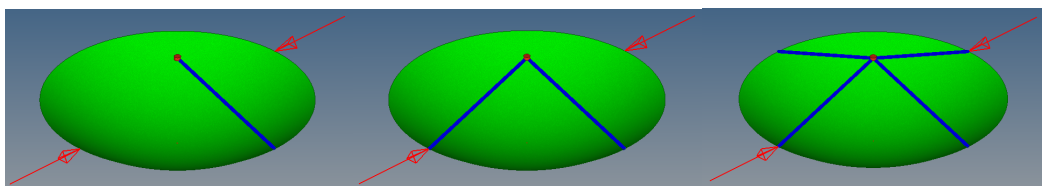


Figure 3.4: Antenna models used to aid in feed update automation. The different strut configurations will cause the feed to deform in predictable ways so that the focus could be on a few degrees of freedom at a time.

3.3 Workflow process

The workflow developed is shown in a flow diagram in Figure 3.5. In this workflow, the starting point is a complete FEKO model and thus before the structural analysis can be performed the model needs to be exported/imported

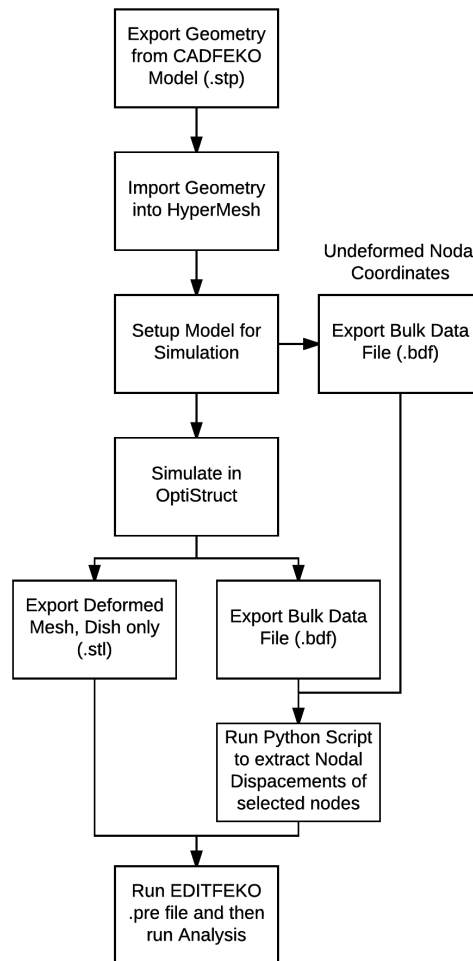


Figure 3.5: Flow diagram of the developed workflow.

into the structural analysis software. This can be performed in one of two ways, the figure only shows one of these. These are to export/import the geometry, as is shown in the figure, or to export/import the mesh. The only difference these two methods have on the final result is the mesh that is used in the simulations, which is not significant at all. This allows for either software package to create the mesh so the package with the better meshing algorithms can be used, if one is in fact better. In this study, it was decided to export the geometry and create the mesh using HyperMesh. This allows for the mesh to be created in such a way that there are nodes in the locations where loads will be applied.

It should be noted that only linear triangular elements are used, as the STL file only supports these elements, however, the EM solver requires a significantly finer mesh than the structural FEA solver does. This mesh size requirement is typically in the order of a tenth of a wavelength (where the

wavelength is calculated at the highest operation frequency of the antenna). The negative impacts of using these elements, lower accuracy for a given mesh size compared to quadratic elements and a resultant constant strain over the element area, should be significantly reduced if not negated entirely. The excessively fine mesh size for the structural analysis may be seen as a problem as it will cause the structural analysis to take longer than necessary. This is not a concern however as in all simulations run thus far the EM analysis took significantly longer to complete than the structural analysis, and thus reducing the computation time for the structural analysis will have little effect on the total runtime. In one example of these simulations, the structural analysis was completed in about 1 minute where as the EM computation time was about 38 minutes, this simulation was run performed on Stellenbosch's HPC (Rhasatsha, 2017). The EM analysis runtime was dominated by the solving the system of linear equations which took about 27 minutes to compute. By coupling the structural and EM meshes a lot of complexity was removed from the exporting/importing processes between the software packages. This also allows for the same mesh to be used for every load case and thus removes differences between the meshes as a variable from the results.

A validation check was performed to compare the difference in results between the linear elements and quadratic elements. Figure 3.6 shows a comparison of the antenna model under that same load case, a gravitational load acting perpendicular to the direction the antenna is pointing. The image on the left is for linear triangular elements and on the right is for quadratic triangular elements. In both images the same element size was used to mesh the surfaces, the only difference being the element types used. In both models the main reflector surface was made up of approximately 45000 elements. It can clearly be seen that the results are very similar and the difference in the maximum deformation is less than 2%. From this, it can be assumed that the error introduced by using the linear elements is not significant.

Another method that was investigated was to interpolate a new surface for the deformed dish surface from the deformed structural mesh. This was found to be unreliable and thus it was decided to take the more robust and simpler, however, more computationally expensive option.

The set-up model step involves; creating a mesh of the geometry that was imported (if a geometry was imported), defining load cases, defining material properties, defining constraints, and setting-up the output requests, everything required to perform the simulation.

In all the simulations run only linear static simulations were considered. The primary reason for using linear statics is that the deformations experienced are expected to be small and thus non-linear deformation is highly unlikely. If a load case would cause the antennas to deform non-linearly then there is either a structural problem with the antenna or the load case is unreasonable. The load cases of concern in this research are the ones that an antenna may experience under normal operating conditions and thus should not cause

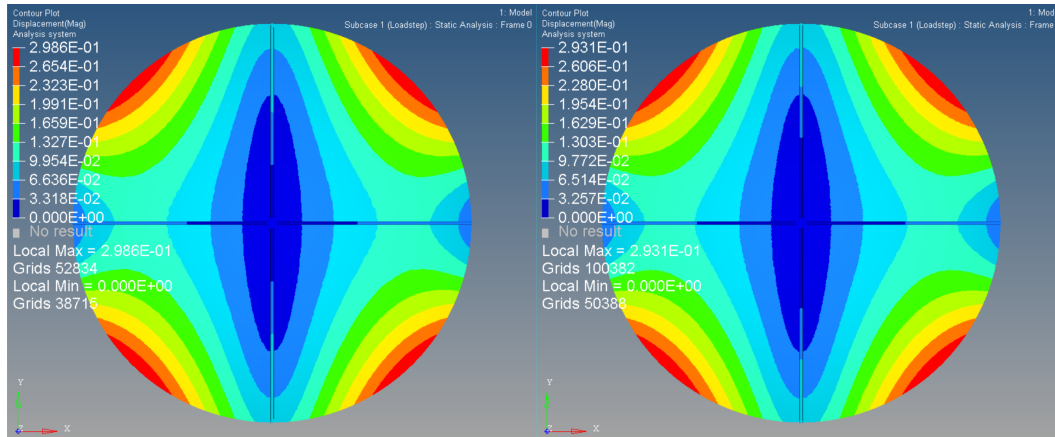


Figure 3.6: Comparison of deformation between linear triangular elements and quadratic triangular elements. It can be seen that there is little difference between the two plots.

significant deformations, for example, load cases such as gravitational loading.

At this stage two bulk data files (BDF), a file containing all of the model's information such as node locations and load definitions, need to be exported. These are for the undeformed and deformed models so that the Python script can determine the nodal deformations of the selected nodes. These nodes need to be on the feed's mesh. The deformed nodal coordinates are also required for determining the rotations of the feed.

The Python script will find the displacements of predefined nodes by finding their nodal coordinates from the two BDFs and then calculating the differences. It also uses the coordinates to determine how much the feed has rotated in each axis. The nodes must be selected as shown in Figure 3.7. The python script requires the identifiers associated with each node. The order in which the nodes are defined is important as this will affect the calculated direction for the rotations. The calculation of these angles will be discussed in the next section. The results are then written to a data file so that EDITFEKO can read them.

The EDITFEKO script performs the following functionality. It starts by importing an "undeformed" feed mesh and a mesh of the deformed dish, and it creates three points which will be used to define the waveguide port. It then reads in the data from the data file created by the Python script and uses that to update the location of the feed mesh and the three points. After that, it creates the waveguide and defines the near field and far field requests. This is the performance output that we want to see. When the script is executed it builds the entire FEKO model and then performs the EM analysis.

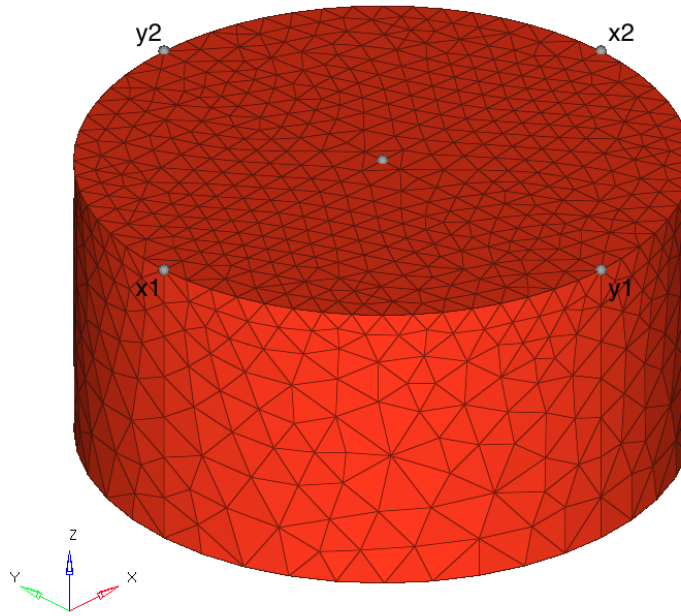


Figure 3.7: Nodes that are used in Python script to calculate the feed rotations.

3.4 Feed transformation calculations

To calculate the feed transformations first the reference node identifiers are recorded so that the correct nodal displacements can be used in the calculations of the feed transformations. Figure 3.7 shows these nodes. Then the BDF of the undeformed antenna is processed for the location of the reference nodes, in x , y , and z coordinates and the same is done for the BDF of the deformed antenna. At this stage node information is simply stored.

Once the information for each node has been stored the displacement of each node is calculated. These displacements are stored in vectors $\mathbf{x1}$, $\mathbf{x2}$, $\mathbf{y1}$, and $\mathbf{y2}$, which correlate to the node they represent. It is important to know where each node is located because the calculations that follow are dependant on the nodes being in the locations shown in the figure.

The feed is known to have its back surface parallel with the xy plane and thus the coordinate displacements can easily be used to calculate the rotations of the feed by using the differences between the reference displacements. Using simple geometry calculations the angle of the feed around each axis can be calculated as follows,

$$\theta_x = \arcsin \left(\frac{y2_z - y1_z}{y2_y - y1_y} \right) \quad (3.3)$$

$$\theta_y = \arcsin \left(\frac{x2_z - x1_z}{x2_x - x1_x} \right) \quad (3.4)$$

$$\theta_z = \arctan\left(\frac{x2_y - x1_y}{x2_x - x1_x}\right) \quad (3.5)$$

To get the effective displacement of the feed in each direction requires some more calculations. First, the rotation matrix is calculated from the feed angles. To simplify the equations the sine and cosine terms for the angles are precomputed.

$$ca_1 = \cos(\theta_x) \quad (3.6)$$

$$cb_1 = \cos(\theta_y) \quad (3.7)$$

$$cc_1 = \cos(\theta_z) \quad (3.8)$$

$$sa_1 = \sin(\theta_x) \quad (3.9)$$

$$sb_1 = \sin(\theta_y) \quad (3.10)$$

$$sc_1 = \sin(\theta_z) \quad (3.11)$$

Auxiliary terms resulting from equating the transformation matrices are then calculated as follows,

$$cc_2 = \frac{cb_1cc_1}{\sqrt{(cb_1cc_1)^2 + (ca_1sc_1 - sa_1sb_1cc_1)^2}} \quad (3.12)$$

$$cb_2 = \frac{cb_1cc_1}{cc_2} \quad (3.13)$$

$$ca_2 = \frac{ca_1cc_2}{cc_1} \quad (3.14)$$

$$sa_2 = cc_2 \frac{sa_1cc_1 - ca_1sb_1sc_1}{cb_1cc_1} \quad (3.15)$$

$$sb_2 = sa_1sc_1 + ca_1sb_1cc_1 \quad (3.16)$$

$$sc_2 = cc_2 \frac{ca_1sc_1 - sa_1sb_1cc_1}{cb_1cc_1} \quad (3.17)$$

Then the Karden angles are calculated which are used by FEKO when rotating geometry components. When rotating the components, FEKO rotates first around z, then around y and then around x axes.

$$a_2 = \arctan\left(\frac{sa_2}{ca_2}\right) \quad (3.18)$$

$$b_2 = \arctan\left(\frac{sb_2}{cb_2}\right) \quad (3.19)$$

$$c_2 = \arctan\left(\frac{sc_2}{cc_2}\right) \quad (3.20)$$

Again the sine and cosine terms are precomputed,

$$s_x = \sin(a_2) \quad (3.21)$$

$$s_y = \sin(b_2) \quad (3.22)$$

$$s_z = \sin(c_2) \quad (3.23)$$

$$c_x = \cos(a_2) \quad (3.24)$$

$$c_y = \cos(b_2) \quad (3.25)$$

$$c_z = \cos(c_2) \quad (3.26)$$

and finally the rotation matrix is calculated as follows.

$$\mathbf{M} = \begin{bmatrix} c_y c_z & -c_y s_z & s_y \\ c_x s_z + s_x s_y c_z & c_x c_z - s_x s_y s_z & -s_x c_y \\ s_x s_z - c_x s_y c_z & s_x c_z + c_x s_y s_z & c_x c_y \end{bmatrix} \quad (3.27)$$

The centre node information is also stored that same way as for the reference nodes. The dot product between the undeformed centre node coordinates, stored in a vector \mathbf{c} , and the rotation matrix is stored as a new vector \mathbf{c}_t ,

$$\mathbf{c}_t = \mathbf{c} \cdot \mathbf{M} \quad (3.28)$$

and then finally the feed displacements in each coordinate, a vector $\boldsymbol{\delta}$, are calculated,

$$\boldsymbol{\delta} = [C_x - C_{tx}, C_y - C_{ty}, C_z - C_{tz}] \quad (3.29)$$

where the subscripts represent the x, y, and z coordinates. The displacements and rotations are then written to a file to be used by FEKO. It was implemented like this as the reference nodes could not easily be placed on a plane that passes through the origin. The offset then requires for the centre node displacements to be corrected such that they effectively represent the origin displacement of the feed model. The Python script that performs this process is presented in Section B.1.

3.5 Verification against analytical approximations

The workflow was compared with the results of existing analytical equations that approximated to the relative gain for axial defocussing and lateral defocussing of the feed. A comparison with results of Ruze's equation for random surface error on the dish surface was also performed. This investigation was performed to ensure that the workflow developed would produce results that were in line with what is expected for these simple aberrations which are well defined.

The approximations for feed defocus rely on the illumination taper of the feed. Illumination taper or edge taper is the level of illumination at the edge of the reflector compared to the illumination at the centre. This value is generally given in dB. Figure 3.8 is a representation of this illumination taper for a normalised beam pattern. In the figure, it can be seen that illumination taper or edge taper, T_e , is -10 dB.

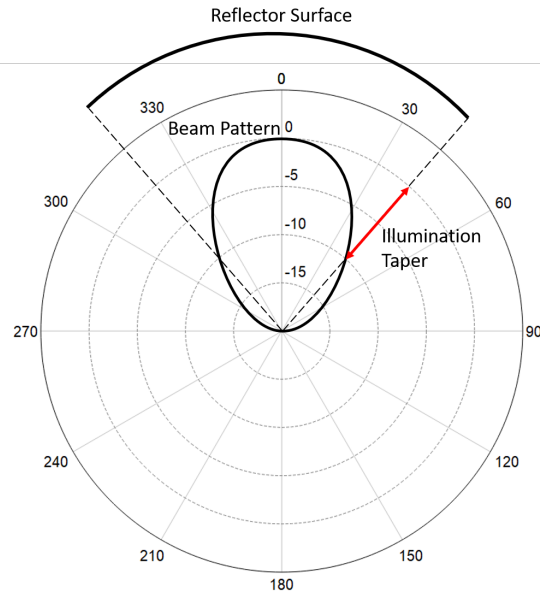


Figure 3.8: A representation of the illumination taper for a normalised beam pattern.

3.5.1 Axial defocus

The results of simulating an axial defocused feed, shifting the feed in the direction of the dish's axis, were compared to the approximations given by Baars (2007). Figure 3.9 shows a representation of axial defocus. In the figure it shows two feed locations, the original and the shifted or defocused feed.

Baars presented an approximation for both full illumination and fully tapered illumination of a parabolic dish and these are as follows. For full illumination the approximation (Baars, 2007)(eq 4.22) is,

$$G(\beta) = 2 \frac{(1 - \cos(\beta))}{\beta^2} \quad (3.30)$$

and for fully tapered illumination the approximation (Baars, 2007)(eq 4.23) is,

$$G(\beta) = \frac{4(2 + \beta^2 - 2 \cos(\beta) - 2\beta \sin(\beta))}{\beta^4} \quad (3.31)$$

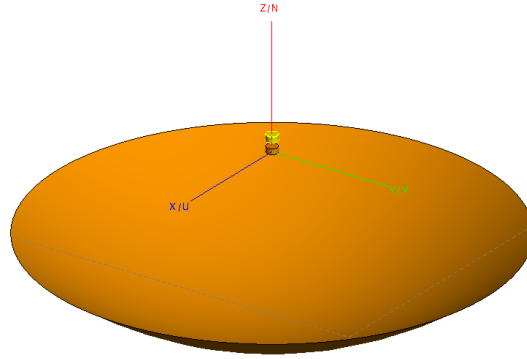


Figure 3.9: A representation of axial defocus, the feed is moved from its original location to the location of the yellow model. This defocusing can also be in the opposite direction.

In both equations 3.30 and 3.31 β is defined as (Baars, 2007)(eq 4.15),

$$\beta(\delta) = \frac{2\pi\delta}{\lambda}(1 - \cos(\psi_0)) \quad (3.32)$$

where ψ_0 , is the open half angle of the paraboloid, and δ is the feed displacement in m. ψ_0 can be calculated using (Baars, 2007)(eq 2.10),

$$\psi_0 = 2 \tan^{-1} \left(\frac{r}{2f} \right) \quad (3.33)$$

where f is the focal length and r is the radius of the parabolic dish.

Figure 3.10 shows the results of the simulations for axial defocusing against the results of equations 3.30, the blue line, and 3.31, the red line.

The uniform illumination and fully tapered illumination can be considered as the bounding conditions and thus it would be expected that the simulations results lie somewhere in between these two curves, as the illumination in the model is neither fully tapered nor fully illuminated. It can be seen that this is, in fact, the case and thus it implies that the simulation results are valid.

3.5.2 Lateral defocus

The results of lateral defocusing, shifting the feed perpendicular to the focal axis, were also compared to the approximations created by Baars (2007). Figure 3.11 shows a representation of axial defocus. In the figure it shows two feed locations, the original and the shifted or defocused feed.

Baars used approximations of the feed patterns for various lateral displacements and then used a max and a first root function to find the relative gain and coma-lobe levels for these displacements. Baars then fitted second order polynomials to the relative gain points and a third order polynomial to the coma-lobe level points. The far field approximation used to determine the field pattern is as follows, (Baars, 2007)(eq 4.33),

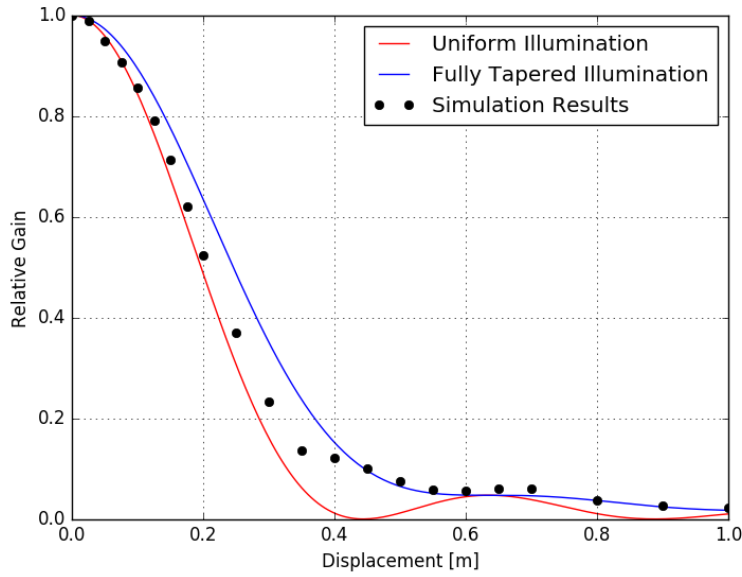


Figure 3.10: Simulation vs. analytical approximation for axial defocus. The simulated results lie between the approximations for uniform and fully tapered illumination. This is what was expected.

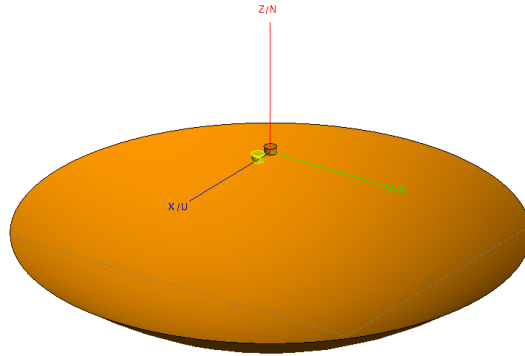


Figure 3.11: A representation of lateral defocus, the feed is moved from its original location to the location of the yellow model.

$$f(u) = 2 \int_0^1 F(r) J_0 \left[r \left(u - \frac{2k\delta}{\Xi} \frac{1}{1 + \left(\frac{r}{\Xi}\right)^2} \right) \right] r dr \quad (3.34)$$

$J_0(a)$ is a zeroth order Bessel function of the first kind, $u = \frac{kD}{2}$, k is the wave number, $k = \frac{2\pi}{\lambda}$, and $\Xi = \frac{4f}{D}$ where D is the diameter of the parabolic dish.

$F(r)$ was approximated to have the *quadratic on a pedestal* distribution and normalised to 1 in the aperture centre, (eq 4.7) which is as follows,

$$F(r) = \tau + (1 - \tau)1 - r^2 \equiv 1 - (1 - \tau)r^2 \quad (3.35)$$

where τ is the edge level which can be determined from the edge taper, T_e , as follows,

$$\tau = 10^{\frac{T_e}{20}} \quad (3.36)$$

Baars (2007) used the method described above to create three approximations for the gain loss of the ALMA antenna which has a slightly different geometry to the antenna that was being used in the simulations. This meant the approximated equations from Baars (2007) could not be used directly to verify the simulation results. The methods Baars used were repeated to get approximations that applied to the antenna in the simulations. Five gain loss functions were created in this way. These were for a uniform illumination, an edge taper of -12dB, -30db (since it closely approximated what Baars (2007) obtained for fully tapered illumination), and two for an edge taper of -14.22dB (the edge taper determined for the simulation). Second order polynomial functions were created for these illumination patterns and for the -14.22dB taper a third order function was also fitted. These functions can be seen in Table 3.1 and they are also plotted against the simulation results in Figure 3.12.

Table 3.1: Gain loss functions for lateral defocus.

Illumination	Gain loss function
Uniform	$1.0051 - 0.0202x - 0.0150x^2$
Taper -12dB	$1.0063 - 0.0241x - 0.0117x^2$
Taper -30dB	$1.0051 - 0.0195x - 0.0095x^2$
Taper -14.22dB	$1.0063 - 0.0240x - 0.0111x^2$
Taper -14.22dB 3rd order	$0.9995 + 0.0070x - 0.0281x^2 + 0.0023x^3$

In the figure, p in HPBW is the number of half-power beamwidths that the feed is shifted off axis. From Figure 3.12 we can see the simulation results fit between the extremes, the red and green lines. The two curves that aim to predict the gain loss for an antenna with the same illumination taper closely approximate the simulation results until about $5p$, which is the limit of what Baars showed. This implies that the simulations are producing valid results.

3.5.3 RMS error

Next, the workflow was tested to see if it could accurately predict gain loss due to an RMS (root mean squared) surface error on a parabolic dish. Ruze (1966) created a formula to approximate this, known as Ruze's equation and the formula is as follows,

$$\eta = \frac{G}{G_0} = e^{-\left(\frac{4\pi\epsilon}{\lambda}\right)^2} \quad (3.37)$$

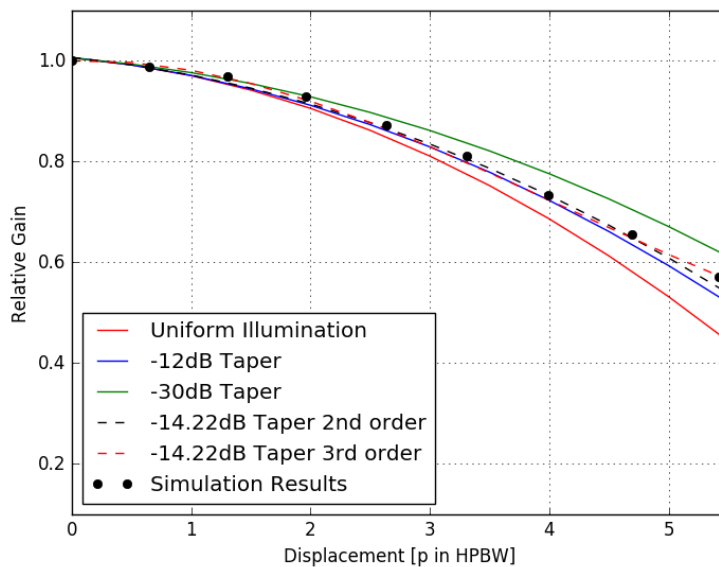


Figure 3.12: Simulation vs. analytical approximation for lateral defocus. Again the simulated fit well with what the analytical equations predict.

where η is the antenna efficiency, G is the gain of the antenna, G_0 is the gain of the same antenna without any surface error and ϵ is the RMS surface error of the dish. This approximation is commonly used by antenna designers.

A Gaussian error was applied to the nodes in the dish of the FEKO model using a Python script and then simulations were run to determine the gain loss. The Python script would add a small random displacement to the nodes that make up the dish's surface in the structural model. The RMS surface error was also calculated at the same time the errors were introduced. It was found that with the model being used that a Gaussian error that would cause an RMS error of more than about 0.5% would result in elements in the FEKO model that exceeded the typical meshing requirement, yielding in inaccurate results. For this reason, the simulations were only run up to an RMS error of less than 0.5%. It was considered that this range would be sufficient for this test. Figure 3.13 shows a comparison between the relative gains determined using Ruze's equation and the FEKO simulations.

From this graph, it can be seen that the FEKO results and the predictions by Ruze's formula show a good correlation. This shows that for random error the simulations have predictions that are similar to those that the commonly used Ruze's equation predicts.

Next, the question was asked as to how well Ruze's equation could predict the effects of a correlated error, such as a deformation. To test this various loads were applied to the dish model and thus the dish deformed. The RMS surface error on the dish was determined in each case using a Python script

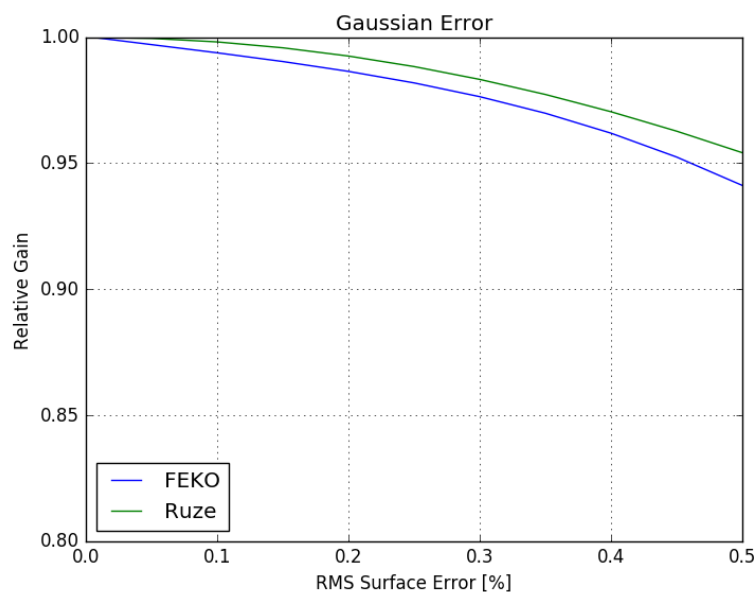


Figure 3.13: Relative Gain vs. RMS Surface Error with a Gaussian Distribution.

and then EM simulations were performed to determine the relative gain due to this deformation. It is expected that since Ruze's equation was designed to predict relative gain for uncorrelated error there will be a noticeable deviation between the FEKO results and the predictions of Ruze's equation. This can be seen in Figure 3.14.

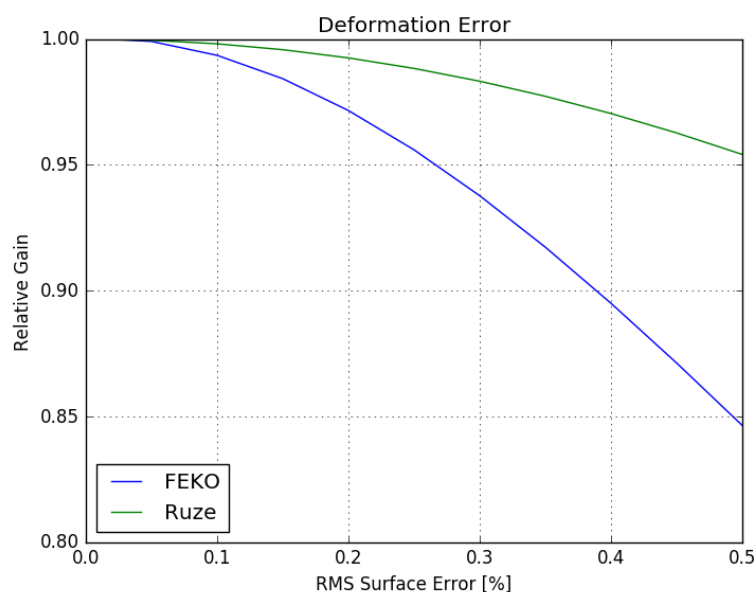


Figure 3.14: Relative Gain vs. RMS Surface Error For a Deformation.

This is more evidence that the model is behaving as expected and demonstrates that Ruze's equation cannot be used to predict the gain loss due to a deformation as the error is correlated.

3.6 Refinement of workflow

Once the workflow had been developed and verified, it was decided to refine the process by automating it as much as possible. It was decided that everything would be automated after the structural model had been fully set up as it is well known what would be expected from this point on. Automating the process before this point would not make much sense as the load case to be analysed is not known, and thus it was decided that the model should be set up completely for the structural analysis. A model that has been built to this point can also be modified easily to test for different load cases. Figure 3.15 shows a block diagram representing how the workflow is automated.

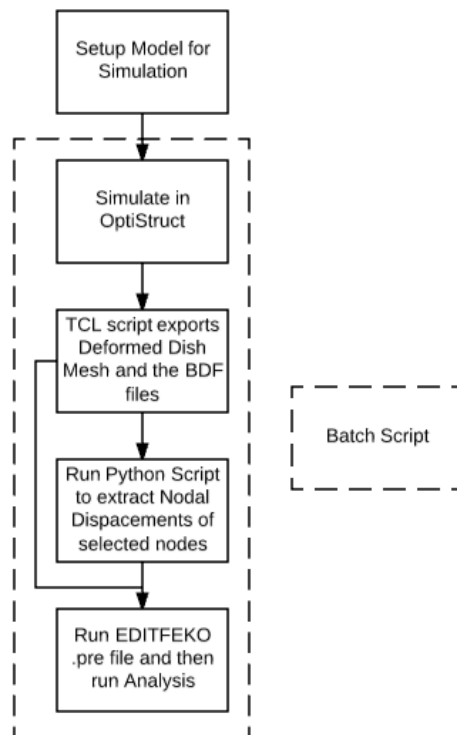


Figure 3.15: A flow diagram representing how the workflow is automated.

A TCL script was written to automate the process of exporting the BDF files needed by the Python script that determines the feed's new location and exporting the deformed dish mesh to be imported into the FEKO model. This script was then modified slightly to allow for the support strut mesh to be

exported along with the dish mesh. This is so the effect of the strut location on the dish can be investigated. This can be changed by modifying a few lines of code if the struts' effects are no longer of interest. This script is shown in Section B.2.

A batch script was written to call each script in the correct order to perform the full analysis. First it takes the .fem input file, which is created by HyperMesh, and runs it in OptiStruct, after that it runs the TCL script, to create all the required files for the steps that follow. The Python script that finds the feed's new location is then run and finally, the .pre file is run in FEKO which builds the FEKO model and performs the EM analysis. With this all the user has to do is create structural model then run the batch file, it should be noted that the .pre file needs to be set up beforehand which will have an existing feed model and read in the feed translations from the Python output file.

3.7 Summary

It was found by using linear triangular elements in the structural mesh for surfaces that will also be modelled in the EM analysis, that these meshes could be exported from the structural FEA software and imported into the EM software. By using the same mesh in both software packages there will be less error introduced compared to what there would be if the deformed mesh from the structural analysis was interpolated into a new surface and then the new surface meshed in the EM software. These meshes can reliably be exported and imported between the HyperMesh and FEKO if the .stl file format is used. This means that objective 1 has been achieved.

A simple primary-fed reflector antenna model was used to aid in the development of the workflow. This workflow utilises the previously mentioned process to integrate the structural analysis and EM analysis software to describe the process for conducting the full structural and electromagnetic analysis. It was determined that the feed should remain in its ideal shape as even small deformations would result in errors in the EM analysis that would prevent it from executing, and thus the feed was made rigid in structural analysis simulations. Objectives 2 and 3 have now been achieved.

The workflow was then used to simulate aberrations that have well defined analytical approximations of their effects on peak gain. It was found that in all cases the simulations predicted results that correlated well with the analytical approximations. From this, it can be concluded that the simulations produce results that are similar to commonly used analytical equations. With this, the simulation process has been verified and thus objective 4 has been achieved.

The effects of a correlated error in a parabolic dish surface due to an applied load on the peak gain of the antenna were compared with predictions from Ruze's equation. It was found that these correlated errors resulted in a greater

loss in antenna gain compared to an equivalent uncorrelated error, in terms of RMSE. This implies that a deformation will have a more significant effect on an antenna's performance and highlights why investigating these effects can be very beneficial.

Chapter 4

Optimisation Study

In this chapter, a numerical optimisation that was applied to a simple reflector antenna model will be explained. VisualDOC 7.2 (VisualDOC, 2013) was used as the GUI to set-up the optimisations that we performed using Design Optimisation Tool (DOT) (DOT, 2013). If one has a parametrised model it is a small step to incorporate an optimisation study. An optimisation study is beneficial as not only can it provide a good candidate model but also provide additional insight into the design space.

A simple model of a dish and feed model was used in the optimisations as this model could quickly be iterated. This means that the optimisation process as a whole will be completed much faster. The simple model is sufficient to prove that the workflow process can be used in this way.

This chapter will cover how the workflow was parametrised to allow for numerical optimisation. The objectives, variables and constraints applied to the optimisations, a description of how the various scripts that were written work, and finally a discussion of the results of the optimisations.

4.1 Initial optimisation investigations

Before any optimisation was performed some initial tests were done to verify that there could be some benefit in performing an optimisation. This process was also performed to aid in the development of the scripts required for the optimisation process to be fully automated. These scripts are described later in this chapter. In these initial investigations, only one variable was changed to see the effect on the performance of the antenna. The selected variable was changed incrementally over some assumed feasible range and the performance output was recorded. In the first investigation the focal length over diameter (FoD) ratio was varied and in the second investigation a backing structure was added to the model and the height of the backing structure was varied.

4.1.1 Effect of focal length over diameter (FoD) ratio

To investigate the effect of the FoD ratio the focal length of the antenna was increased. Figure 4.1 shows two of these models and it can be seen that the feed is further away from the dish in the image on the right and the dish is not as deep. On the figure on the left the diameter, D , and focal length, f , have also been shown. The load applied to the model was a gravitational load that varied in angle for each simulation, thus the elevation angle of the dish was effectively varied.

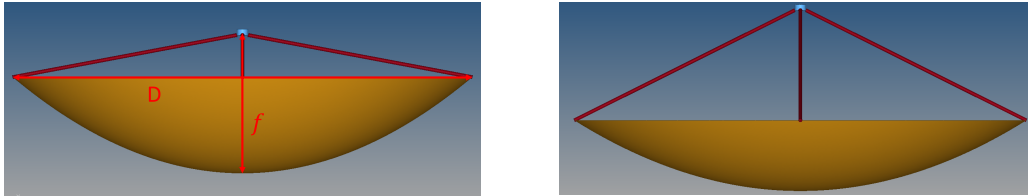


Figure 4.1: Two models with different FoD ratios. The image and the left has the diameter and focal length on the model indicated. In the image on the right the FoD ratio has been increased while keeping the diameter constant.

In these simulations no backing structure was simulated, so the results will show greater performance degradation than is likely to be experienced. This would be due to there being greater deformation to the dish surface without the backing structure. Figure 4.2 shows a graph of the relative gain, which is defined as follows,

$$G_{relative} = \frac{G_{peak}}{G_{ideal}} \quad (4.1)$$

where G_{peak} is the peak gain, G_{ideal} the ideal gain. The latter is defined for the case where there are no mechanical deformations.

Each line represents a different FoD ratio. It can be seen that the FoD ratio has little effect on the relative gain under gravitational load cases. It should be noted that the ideal gain does vary with different FoD ratios, and thus the actual gain achieved at these elevation angles does vary based on FoD ratio. However, these changes in antenna performance are likely the result of the illumination taper varying due to this change in FoD ratio and are not an indication that a specific FoD ratio is worse than another. The FoD ratio was then set to 0.3 for the remain simulations.

In Figure 4.2 it can be seen that the relative gain decreases for greater load angles. At 0° the only deformation the feed is expected to experience is an axial displacement, however as this load angle increases a greater lateral displacement will be introduced as well as a rotation of the feed, this alone should result in a significant loss in performance. However, the dish surface also deforms more as there is an increasing moment applied to the whole structure as well, as the centre of mass for the dish and feed system is no longer above

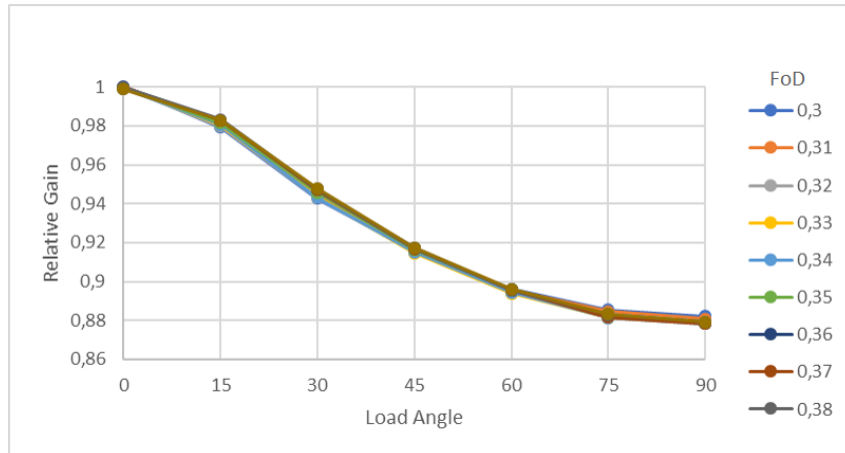


Figure 4.2: Relative gain vs. gravitational load angle for various FoD ratios. It can be seen that the FoD ratio has little effect on the relative gain.

the support. Figure 4.3 is diagrams of a 0° and a 90° load angles, it can be seen that the effective load, the red arrow, acting on the centre of mass will only induce a moment in the second case as the centre of mass is no longer directly above the support.

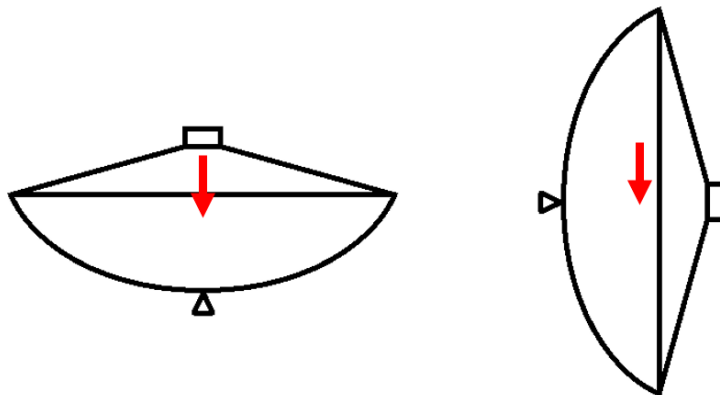


Figure 4.3: This represents two gravitational load cases with the dish model at 0° and 90° . The red arrow represents the equivalent force due to weight of the structure acting on its centre of mass. It can be seen that this will result in an additional moment being introduced for the model on the right (the 90° load case).

4.1.2 Effect of a backing structure

Once the FoD ratio had been investigated it was decided to look at the effect the backing structure could have on the performance. The backing structure

would greatly change the effective stiffness of the antenna and change the shape in which the dish surface would deform. This means that the results could vary quite significantly by changing the backing structure. To perform this investigation a backing structure was added to the model, as shown in Figure 4.4. In the initial investigation only the length of the backing structures "arms", or height of the backing structure, H in Figure 4.4, was varied to see if some optimum could exist. The simulations were conducted under a gravitational load case acting 90° to the direction the antenna is pointing as this was shown in Figure 4.2 to have the greatest effect on the antenna's performance. Figure 4.5 shows two graphs of the results of this investigation.

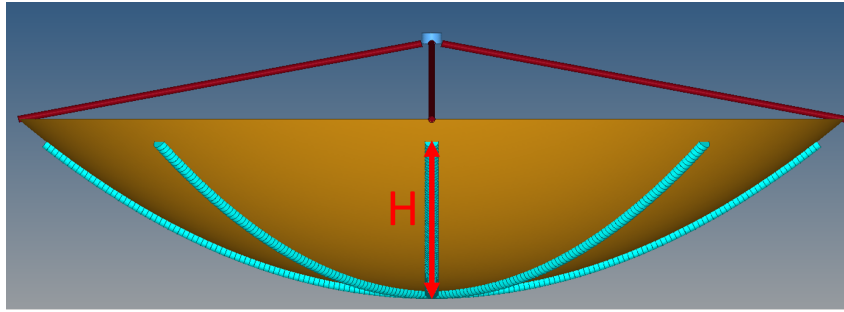


Figure 4.4: The simple dish model.

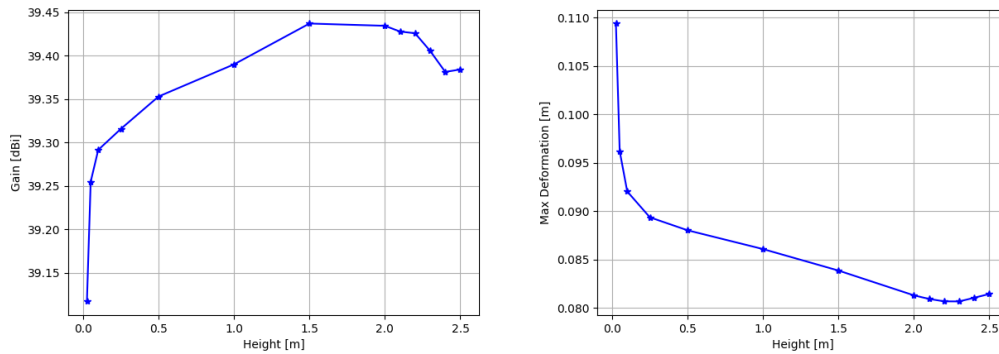


Figure 4.5: Gain vs. Backing Structure Height.

It can be seen that there is an optimum gain (a maximum) and maximum deformation (a minimum) to be achieved from varying the length of the backing structure's arms. However, these optimums are not for the same backing structure height. The same will be true for other responses that may be of interest. The other input variables will also produce similar curves with their own optimums. It is unlikely that the optimum of each of these curves, when found varying just one input variable, will mean you have found the

optimum for that variable. It is expected that each variable will influence to output differently, and the result will still be partially dependant on the other variables.

The optimum found here is also only for the peak gain, the optimum for other output variables, like side-lobe levels or pointing error, may be different. For this reason, an optimiser was used to perform the more complex objective optimisation. In this case, constrained single objective optimisations offered by VisualDOC.

4.2 Parametrisation of workflow

A simple model of a primary-fed parabolic reflector antenna was parametrised, by means of a TCL script and VisualDOC, so that all geometric variables can be changed. Table 4.1 shows these variables.

Table 4.1: Parameterised geometric variables.

Variable	Symbol used
Backing structure offset	offset
Dish diameter	Dm
Focal length over diameter ratio	FoD
Strut base radius	S_x
Height of backing structure	H
thickness of backing structure	t
Difference between backing structure and dish height	t0

This model is the same model used in Section 4.1.2, but in this case, it is completely generated by the TCL script for the geometric variables specified. Figure 4.6 shows an example of this model, the model changes with each iteration in the optimisation process.

These geometric variables are then easily changed using VisualDOC's modify function which will change these values in the TCL script before a new simulation is run. VisualDOC will calculate what these values need to be changed to between each iteration step, and this will be dependant on the optimisation method specified. In this study, a gradient-based method was used.

VisualDOC was chosen to set-up the optimisation as it is user-friendly, with the optimisation process being built through the use of a diagram system. VisualDOC can be set-up to perform the optimisations in many different ways and the exact set-up used in this study will be explained later on in this chapter.

A new model would be built for each iteration and a new mesh generated from the new geometry for each iteration. This was implemented as there could be significant changes to the geometry between iterations and modifying

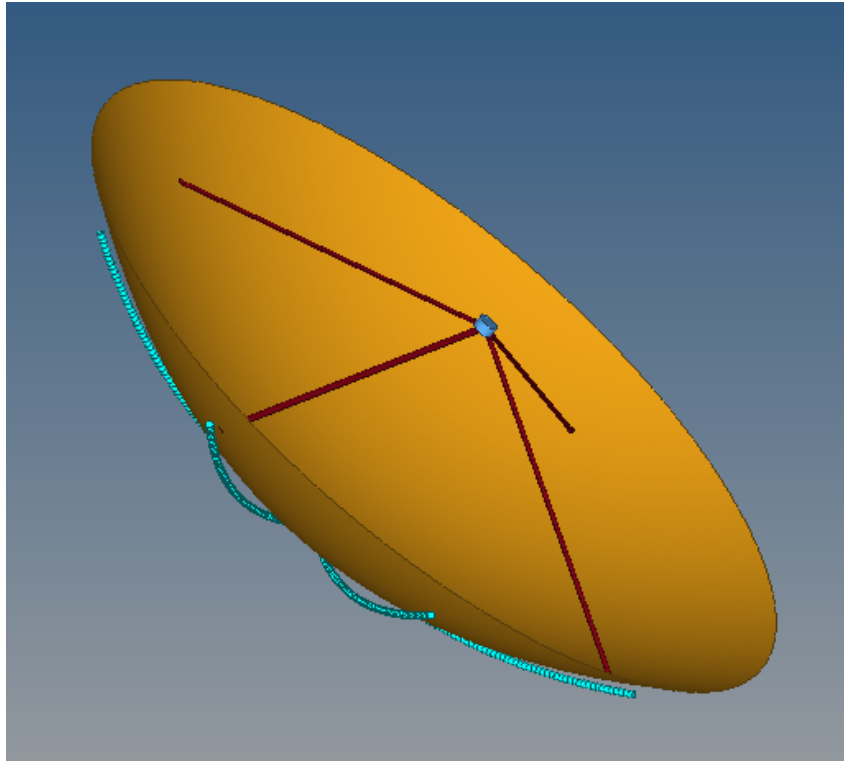


Figure 4.6: Primary-fed reflector antenna generated by the TCL script. This model will change between each iteration of the optimisation.

the existing model to match the new geometry was unreliable. For instance, if the strut location is changed the nodes that it connects to on this dish's surface will change.

VisualDOC needs to be able to read the output values from the simulations in order to use them in the optimisation. VisualDOC, however, had problems reading from the very large files created during to simulations, and thus a few Python scripts were written to retrieve this data and write it to files that VisualDOC could read.

4.3 Optimisation objectives, variables, and constraints

In this section the objectives, variables, and constraints that are used in the optimisation are discussed. The formal definition of optimisation is as follows: Maximise or Minimise

$$F(\mathbf{x}) \tag{4.2}$$

Subject to

$$\begin{aligned} \text{such that } h_i(\mathbf{x}) &= 0 \quad i = 1, n \\ g_j(\mathbf{x}) &< 0 \quad j = 1, m \end{aligned} \quad (4.3)$$

where $F(\mathbf{x})$ is the objective function, $h(\mathbf{x})$ is an equality constraint, and $g(\mathbf{x})$ is an inequality constraint and \mathbf{x} is a vector of input variables. n and m are the number of equality and inequality constraint respectively.

In this case, objectives are the responses that the optimizer will try to maximise or minimise, these are often outputs from the simulations. The variables are the geometry variables or input values that the optimizer will change in order to optimise for the objective and meet the constraints applied to the system. These variables will also have constraints applied to them to prevent the optimizer from choosing unrealistic values for the input values. And finally, constraints are limitations that are applied to either input or output values. These can be either in the form of an equality constraint if a specific value is required, or as an inequality constraint if a value above or below a specific value is desired. How each of the above was applied to the optimisation problems will be explained below.

4.3.1 Objectives

In this optimisation study, there were four responses that were of interest, these being the peak gain, mass, side-lobe level and pointing error. If all these responses were to be considered as objectives for the optimisation they would each need a weighting assigned to them that is indicative of their level of importance. However, it is difficult to determine what these weightings should be. However, if the responses of interest can be within a certain range it can be more desirable to control them via constraint. In this study, it was decided that the objective of the optimisation would be to maximise the peak gain, and the other responses of interest, mass, side-lobe level and pointing error would be controlled by constraints. This yielded a single objective optimisation where no weightings needed to be assigned.

As stated the objective of the optimisation was to maximise the peak gain of the antenna system. Peak gain is the gain of the main lobe in the radiation pattern produced by the antenna, as discussed in Section 2.3. Peak gain is one of the responses that are of great interest in antenna design. The receiving sensitivity of a radio telescope is a function of both the antenna gain, G , and the system noise temperature, T_{sys} . A convenient figure of merit to use is the ratio $\frac{G}{T_{sys}}$ (Kildal, 2015). The higher this ratio, the better the sensitivity of the system to weak signals. It was decided to focus on the gain in this study.

4.3.2 Constraints

As stated in the previous section it was decided to constrain the side-lobe level, mass and pointing error. These responses were constrained to be within what was considered to be an acceptable range.

The minimum side-lobe level, SSL , was constrained to be greater than 20 dB, which translates to a side-lobe gain that is at least 100 times smaller than the peak gain in the far field. As stated in Chapter 2 the side-lobe levels are the difference in gain between the peak gain in the far field, and the gain of the side-lobe, a local maximum in the far field. With the side-lobe gain being limited to being at least 100 times smaller than that of the main beam, the error introduced by the side-lobes should be sufficiently small to be considered negligible. Figure 4.7 shows a portion of a far field pattern and identifies the peak gain or main beam and two of the side-lobes.

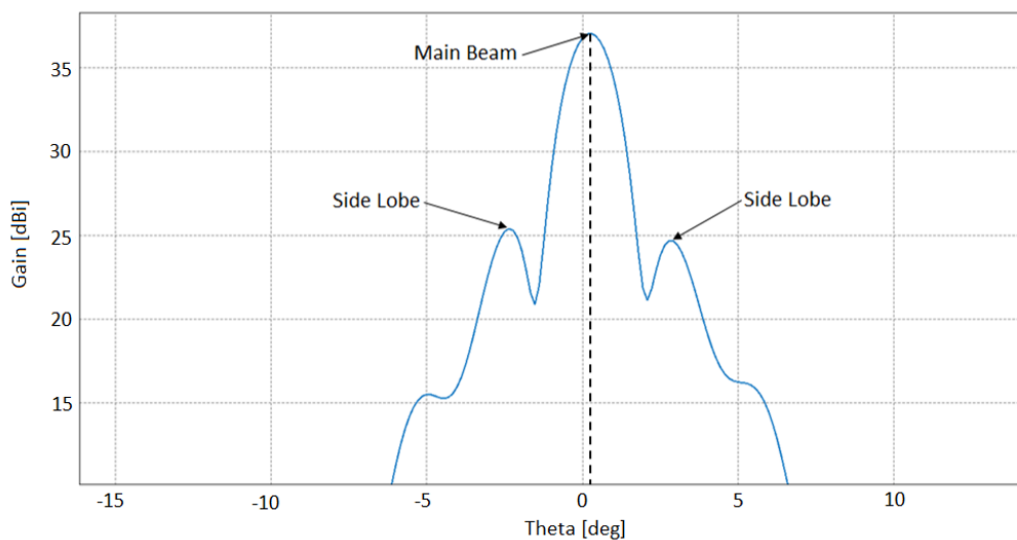


Figure 4.7: A far field pattern with the main beam and two side-lobes indicated. The side-lobe level is the difference between the gain of the main beam and the gain of the side-lobes. It can also be seen that the main beam is not at 0° , this is because there is a pointing error in this far field.

The pointing error is also of interest and was constrained to be less than 10% of the theoretical half power beam width, which is approximated using equation 2.3. The pointing error quantifies the angle at which the antenna's main beam is pointing away from the intended target point. In radio astronomy, this is really important since the targets are so far away that even a small pointing error could lead to the antenna pointing completely off target. This can be compensated for if this error can be approximated accurately, but ideally, this error would be negligible. In Figure 4.7 again we can see that the peak of the main beam is not at 0° , this is pointing error. A pointing error

of less than 10% of the HPBW was chosen as this should result in the target source still being very close to the centre on the main beam of the antenna where it is most sensitive.

Mass was chosen as another response to constrain as it is a simple way to represent the total cost to build the antenna. There is not necessarily a direct relationship between mass and cost, but there is a good correlation. This creates a tradeoff situation that is realistic in real-world antenna design. This is because the more material that is used, and thus the greater the cost, the stiffer the antenna will be, and thus the less it will deform from its ideal shape. It was decided to constrain the mass to be less than 3500 kg as this was to be relatively close to the minimum weight that the model could be within all the geometric constraints that had already been applied to the model. By having this constraint close to the minimum possible weight of the entire model it should push an optimizer to find an efficient design as opposed to simply applying more material until the model is sufficiently stiff.

4.3.3 Variables

In this optimisation study, it was decided to optimise the backing structure and the strut location. First, the backing structure will be discussed, then the strut locations.

The height of the backing structure, the thickness of the arms and the offset of the arms behind the dish surface were varied to find an optimum. Figure 4.8 shows all these variables. The backing structure was set to have eight arms of equal length and thickness that radiated out from the centre, and are spaced evenly around the dish. This does not necessarily represent a structurally optimum design, but this decision is based on having a simple prototype model that allows for various parameters to be changed.

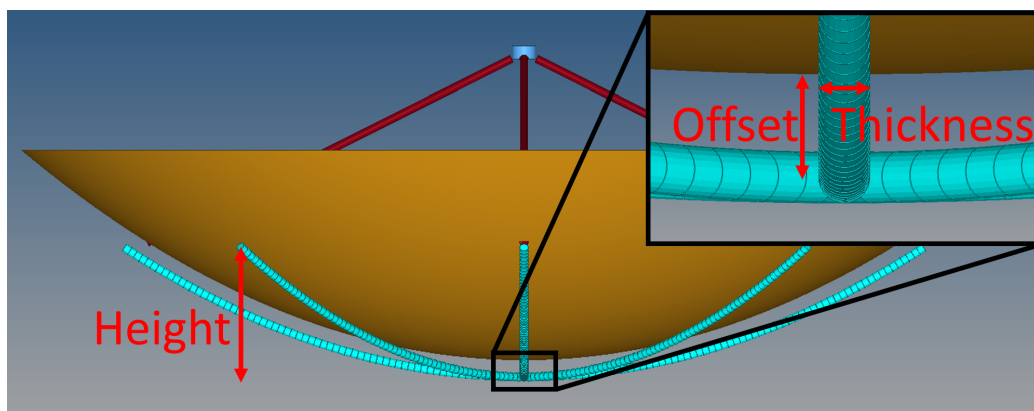


Figure 4.8: Variables of the Backing Structure.

By varying the thickness of the arms they are made to be more or less stiff as there is more or less material respectively. The goal was not to optimise the shape of the arms but to simply represent a stiffer backing structure. For this reason the backing structure was assumed to be made up of cylindrical rods. The thickness was allowed to vary between 0.01 m and 0.1 m. A thickness of zero would mean there is no backing structure and this would never be the case as the dish surface would need to be attached to something, even if its just a small bracket and pin. An upper bound was needed because if there was no upper bound the backing struts would potentially be created unrealistically large, and thus effectively rigid. However the mass constraint, explained later, would also likely prevent this.

The height of the backing structure is allowed to vary between 0.05 m to 2.5 m above the centre/base of the dish. The lengths were not permitted to be zero as it is not possible to build an antenna with no support at all, this case would also have the antenna performance deviate significantly from when there is a backing structure. The upper bound is set to the height/depth of the dish thus the backing structure cannot extend beyond the dish's edge. For the FoD ratio of 0.3 and the dish diameter of 12 m, that means the backing structure cannot be more than 2.5 m heigh. This is calculated as follows,

$$H_{max} = \frac{\left(\frac{D}{2}\right)^2}{4f} \quad (4.4)$$

where D is the diameter of the dish and f the focal length. The longer the arms, the stiffer the whole structure should be, but the greater the load due to weight will be. This added load could result in greater deformation to the dish surface even though the antenna is theoretically stiffer, Figure 4.5 shows this. This suggests that there is likely an optimum length for the backing structure which will result in the least amount of deformation, especially to the dish's centre. Deformations here have a greater effect on antenna performance than deformations at the dish's edges, due to the fact that the dish is illuminated more strongly in the centre.

The offset is the distance that the backing structure is behind the dish surface, as shown in Figure 4.8. This offset is enforced to have the backing structure in a more realistic location relative to the dish surface. The upper bound is set to 0.0 m. The lower bound is needed as the further away the backing structure is the stiffer the structure will be. This is due to the moment of inertia increasing. With this in mind, the lower bound was set to -0.3 m.

As stated previously the strut location was also optimised for. These are the ends that connect to the dish surface and not the ends that connect to the feed, as shown in Figure 4.9. The location of the struts can change the way the antenna deforms by making the dish stiffer in the region that it connects to these struts. The strut locations will also change the effects of aperture blockage on the antenna's performance by varying the 'shadow' that the struts

cast on the dish surface. The distance radially from the centre of the dish to the base of the struts, S_x , was varied, the lower bound was set to 0.1 m and the upper bound was set to 6 m, which is the dish radius in this optimisation. The lower bound was implemented as an antenna will never be designed with the base of the struts so close to the centre of the dish. As stated previously the centre of the dish is where the most power is expected to be reflected, this lower bound is still probably much lower than would be efficient. Figure 4.9 shows a representation of two dishes with the strut bases in different locations.

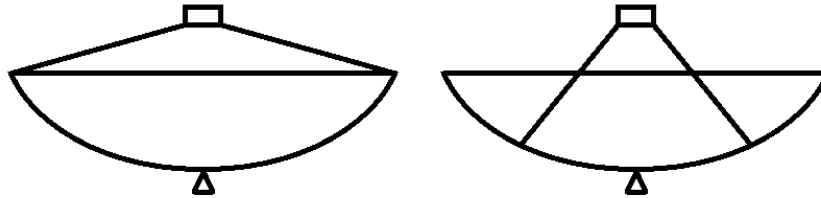


Figure 4.9: Representation of moving strut locations.

4.4 Simulation approach

In this section the scripts used to automate the whole simulation process so that an optimisation study could be performed will be discussed. Then, the actual set-up of the optimiser will be presented.

4.4.1 Scripts

As stated previously there is a TCL script that generates a full model and then runs the structural and EM simulations from there. This script is shown in Section B.3. This utilises the previously developed workflow, Chapter 3, to perform this full simulation as well as a few new scripts to allow the optimizer to get all the information it needs. The TCL script handles calling the automation scripts that follow after the model is built.

This script uses HyperMesh specific TCL commands to build the models. These commands allow for anything you can do using the user interface to be done using scripts, which is really useful in batch processing or optimisation problems. The script first defines all the variables required, some of which are dependant on others, then it creates all the material properties and builds all the geometries; the dish surface, feed, support struts and backing structure, after which it generated meshes for all these components.

Lastly, it creates the boundary conditions and load cases. The dish is fully fixed at its centre and a gravitational load case is applied at 90° to the

direction the dish is pointing, in other words, the dish is pointing towards the horizon. This was found to be the dish orientation that resulted in the greatest reduction in antenna performance for gravitational loads, and thus it is the load case that would benefit the most from the optimisation. If the antenna is optimised to meet specific performance parameters at an angle that has less of a performance reduction than would be present at 90° , it will likely not meet these performance requirements at 90° as more degradation should be expected. The other orientations will be investigated after the optimisations have been completed to ensure that the performance requirements are still satisfied at these angles. When optimising for a specific load case, it is possible that the performance for other load cases can be degraded while the specified load case has an improved performance. For this reason it is important to check that the other load cases still yield performances that are satisfactory.

The TCL script now creates the input file required for the structural simulation and creates a summary file of the model that has been generated, which will contain the masses of each component. The TCL script then executes a Batch script which automates the rest of the process.

The Batch script will run the structural simulation using OptiStruct and then run a different TCL script which exports all the required files from the structural analysis results that will be needed for the EM analysis. The Batch script then runs FEKO to perform the EM analysis before a running a Python script that finds the maximum gain in the results, a .ffe file which contains all the far field data as ASCII, and writes it to a file that VisualDOC then reads for its optimisation. This Python script also finds the two largest side lobe gains and the pointing error and adds them to the file as well. The Python script is needed here as the far field results file is too large for VisualDOC to read, but Python has no issue, thus it finds the results that VisualDOC will require then writes them to a new file which will be much smaller and thus readable by VisualDOC.

VisualDOC uses the summary file and the file created by Python to find the peak gain, side-lobe levels, pointing error, and the total mass of the models to use in the optimisation. VisualDOC uses these results to determine the optimum.

4.4.2 Optimiser set-up

A response surface methodology was used to perform to optimisations. Response surface methodology uses statistical and mathematical techniques which can be useful in the development, improvement, and optimisation of a process (Myers *et al.*, 2009). The response surface methodology makes use of design points to generate an approximated response to the variables in the full design space. This significantly speeds up the optimisation process as the full simulation step is now approximated by a mathematical equation. This equation can be solved almost instantaneously and thus the full optimisation process can be

completed in a matter of seconds as opposed to hours if not days, depending on the complexity of the models.

A design of experiments (DOE) method was used to decide which design points would be used to create this response surface. VisualDOC has a DOE component built in which was used for this step, and this has several DOE methods that can be used. VisualDOC was set to perform a Full Factorial Composite DOE method which in VisualDOC uses the face-centred central composite design method, Figure 4.10 shows a representation of how the design points are selected for this design method for three arbitrary (dimensionless) input variables. The variables have been normalised to lie between the range -1 and +1, with 0 the midpoint in the range. This normalisation is to make the figure easier to understand.

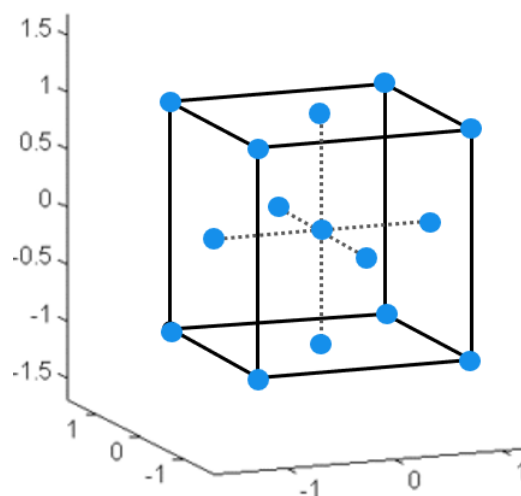


Figure 4.10: Representation of the Face-centred Central Composite Design, design point selection for three input variables. Each dot represents a design point that will be tested.

The design space consists of four input variable, or independent variables, the backing structure height, thickness and offset as well as the strut location. There are also four output variable, or responses, the peak gain, the side-lobe level, the pointing error and the total mass of the structure. Following the face-centred central composite design method, this results in 25 design points which were simulated. As there were a lot of design points that needed to be simulated, it was decided to make use of the High-Performance Cluster (HPC) available at Stellenbosch University to do this (Rhasatsha, 2017). The process of adapting the workflow to work of the HPC is discussed in Appendix A.

After simulating these design points VisualDOC will then use them to calculate the approximate functions that make up the response surface. This is part of the DOE component and is done while the design points are being simulated. VisualDOC was set up to use full quadratic approximate functions

as these yielded the best fit to the design points, as all input variables appeared to have a quadratic response from all the output variables. A full quadratic function for two variables looks as follows,

$$f(x_1, x_2) = a_0 + a_1x_1 + a_2x_2 + a_3x_1x_2 + a_4x_1^2 + a_5x_2^2. \quad (4.5)$$

where a_0 to a_5 are constants calculated by VisualDOC and x_1 and x_2 are the input variables. Each response has an approximate function which defines it. Once VisualDOC had run the full DOE process the approximate functions were extracted from the results and a Python script was written to solve these functions for a given input.

The response surface for the mass is shown in Figure 4.11, this is for varying thicknesses of the backing structure and t_0 , which is the difference in the backing structure and the dish height. The strut locations and backing structure offset are fixed for this surface. It can be seen that the surface closely approximates the trends of the design points, some of these design points are for different offsets and strut locations which in this response surface can have a small effect. This figure helps to demonstrate what is meant by a response surface but also shows that there is an accuracy penalty in using one. That is why the results from a response surface should always be validated as will be discussed in what follows.

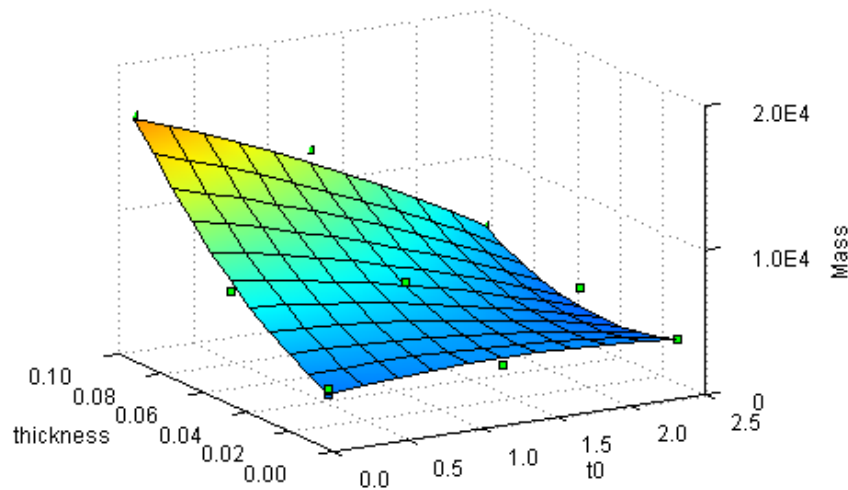


Figure 4.11: Mass Response Surface.

VisualDOC was then set-up to perform an optimisation of the python script. This optimisation was set-up to use the Modified Method of Feasible Direction (MMFD) which is a gradient-based method. It also uses a forward difference for the gradient calculation and the step sizes used and stopping criteria are shown in Table 4.2.

The optimisation of the response surface was run on the local machine as there was little computational requirement, as the python script just needs

Table 4.2: VisualDOC step sizes and stopping criteria.

Component	Value
Relative Finite Difference Step Size	10^{-3}
Absolute Finite Difference Step Size	10^{-4}
Relative Objective Convergence	10^{-3}
Absolute Objective Convergence	10^{-6}
Maximum Number of Iterations	100
Iterations for Convergence	2

to solve simple quadratic equations. The optimum of the response surface is computed very quickly, and so many different objectives and constraint can be applied to the system to get different optimums. Once an optimum is found using the response surface it must be validated, as the response surface is still an approximation. This is done simply by running a full simulation with the input variables that were determined to be the optimum. If the results of this simulation are very similar to the predicted results of the response surface it can be assumed that an optimum has been found. If there is a significant difference, however, a windowing approach can be applied. This approach entails reducing the design space and performing a new DOE and fitting a new response surface to these results.

4.5 Results

This optimisation study has lead to two distinct outcomes. The first is a prediction of a candidate point to meet the constraints and objectives of the optimisation study. The second is the benefit of using the response surface model to gain more insight into the tradeoff between the various responses of interest.

4.5.1 Results of the optimisation

It was found that the response surface model described previously did not capture the design space accurately enough. It had R^2 values of 0.8900, 0.9935, 0.8706, and 0.7671 for its fit to gain, mass, side-lobe level and pointing error. R^2 is the proportion of variation in the dependent variable that is predictable by the independent variables. A higher R^2 generally means there is a better fit. The R^2 value for the pointing error is was considered to be too small, and this is confirmed when comparing the prediction of the response surface model to the simulation results of the candidate point, these results are presented in Table 4.3. It can be seen that there is a noticeable difference between the predicted results and the simulated results, specifically with regards to the pointing error.

Table 4.3: Predicted optimum compared to simulation results for the first response surface optimisation.

	Gain [dBi]	Mass [kg]	SLL [dB]	Pointing error
Prediction	40.45	3523	19.88	0.17°
Simulation	39.62	3442	26.00	0.32°

This led to the decision to reduce the design space. First, it was decided that the offset would be constrained to -0.3 m, thus removing one of the input variables. The offset converged to its lower bound of -0.3 m and it was expected that it would keep converging to this value for all optimisations and thus its value was fixed.

Next, the bounds of the remaining input variables, t_0 , t and S_x , were updated to have a smaller range of allowable values. This reduction was to 61.8%, the golden ratio, of their original ranges and was centred around the previously predicted optimum where possible.

This was done as the initial bounds had been selected with limited knowledge of where the feasible region existed in the design space. Due to this a large portion of the design space would actually yield infeasible results for the constraints applied. By reducing the design space the feasible region will make up a larger portion of that space and a better fit of this reduced design space can be achieved. After performing this first optimisation a better understanding of the feasible region was gained and these bounds could be reduced.

A new DOE was conducted for these new bounds and a new response surface fitted to the results. The Python script was updated with the new response surface and the optimisation was performed again. A better correlation between the predicted and simulated results was found. It was decided, however, to reduce the design space once more. It was also found that there was no feasible region with the constraints that were applied. It was thus decided that the mass constraint would be changed to constrain the mass to be below 3600 kg.

The optimums for the input variables for each iteration of response surface is shown in Figure 4.12. The upper and lower bounds are also represented by the dashed lines. It can be seen that these values are converging on a result with each successive response surface iteration. It was thus decided that no more reductions to the design space were necessary.

Figure 4.13 shows a contour plot of the gain vs. the thickness and t_0 over the initial range. In this plot, the strut location, S_x , is set to the optimum determined by the last response surface model, 4.518 m. The black box represents the design space of the last response surface model and the black line shows predicted optimums of each response surface. The contour plot was generated by fitting a polynomial through all design points generated throughout the optimisation study. The boundary lines between the feasible regions for each constraint are also shown, these are generated from the last response

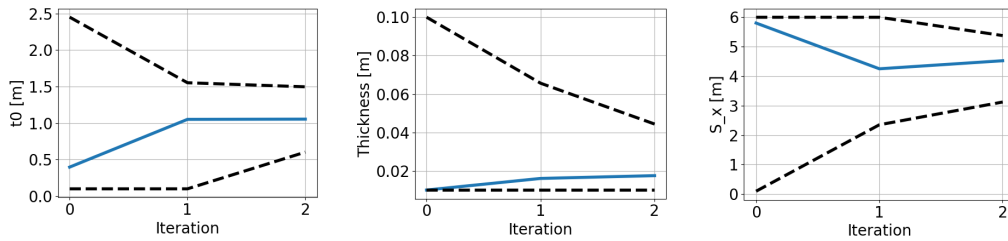


Figure 4.12: Predicted optimums of input variables and the upper and lower bounds compared to the iteration of response surface. It can be seen that the input variables converge on a point.

surface model. The blue dashed line, representing the mass constraint and requires the candidate point to be on its left. The pointing error constraint, the red line, encompasses its feasible region as does the side-lobe level constraint. From this, it can be seen that the feasible region for all the constraints is the line where that blue and red line overlap but are bounded by the green line. It can also be seen that the step between each iteration's candidate point is getting smaller. This is indicative of the result converging.

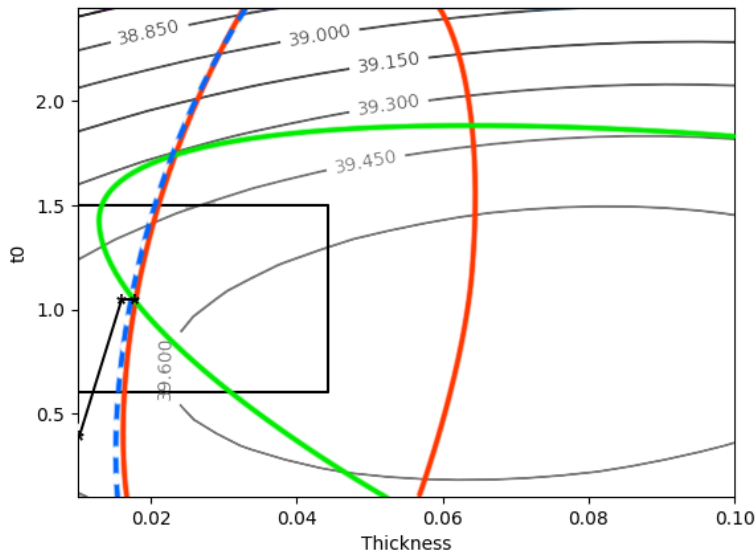


Figure 4.13: A contour plot of the gain vs. thickness and t_0 of the backing structure. The coloured lines represent the boundaries determined by the constraints. In this case, the feasible region is a line where the red and blue lines overlap and is within the bounds of the green line.

This final response surface model had R^2 values of 0.9976, 0.9995, 0.9268 and 0.9921, for gain, mass, side-lobe level and pointing error. This is a big

improvement on the original response surface. The new candidate point predicted in the optimisation of this response surface model was then simulated. Table 4.4 shows a comparison between the predicted and simulated result for this candidate point. It can be seen that the error in the prediction has been reduced. The prediction of the pointing error is still not ideal but this response is very sensitive to the input variables.

Table 4.4: Predicted optimum compared to simulation results for the reduced design space.

	Gain[dBi]	Mass[kg]	SLL[dB]	Pointing error
Prediction	39.57	3602	20.04	0.17°
Simulation	39.56	3579	18.69	0.12°

This was believed to be a sufficient stopping point for this optimisation study as it demonstrates that such a study can be performed using the workflow process. The antenna model being optimised was a simple representative model developed for the purpose of demonstrating how an optimisation could be performed.

The candidate point of the final response surface model was simulated for other gravitational load angles to ensure that the performance had not degraded for these load angles. The angles simulated were 0°, 30° and 60°. It was found that the performance at these load angles was better than at a 90° load angle. This can be seen in Table 4.5. This is what was found in section 4.1.1 and shows that in this case optimising for the worse load case did not cause a degradation of performance for the other load cases.

Table 4.5: In the table, the results of simulation for the candidate model at different load angles are shown. It can be seen that for these load angles the candidate model does not have a degraded performance compared to the load case it was optimised for.

Load Angle	Gain [dBi]	SLL [dB]	Pointing error
0°	39.60	25.02	0.00°
30°	39.59	19.27	0.06°
60°	39.57	18.82	0.10°

4.5.2 Tradeoff investigation

With a response surface model already being described, it is a simple step to investigate the tradeoff between the responses of interest. It was expected that a tradeoff between the mass of the system and the other responses existed as more available mass would mean the structure could be made stiffer. The

last response surface model was then used to find the peak gain for a range of mass constraints, keeping all other constraints the same. The results of this are shown in Figure 4.14. This shows that there is a diminishing return for increasing the mass in terms of peak gain. This demonstrates a tradeoff that will need to be decided upon by the antenna designer. A heavier antenna will have a greater peak gain, but the cost to produce that antenna will be higher. There is also a trend for the side-lobe level to increase along with gain, but it is evident that there is no direct relationship between peak gain and side-lobe level. In other words, an increase in gain will likely see an increase in the side-lobe level but this will not always be the case.

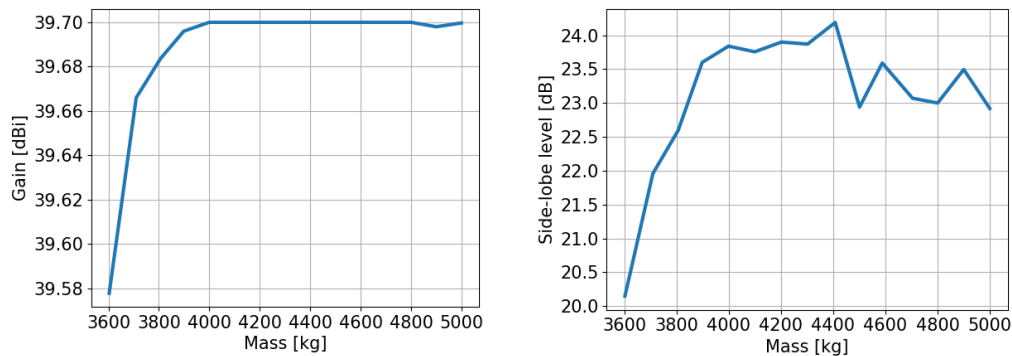


Figure 4.14: The results of the tradeoff investigation show that there is benefit to increasing the allowable mass, but it has a diminishing return.

4.6 Summary

In the initial investigation, it was found that there was a potential benefit to performing an optimisation of the structure. Just varying the height of the backing structure lead to different optimum heights for the backing structure depending on the response to be optimised.

An optimisation study was then set-up using VisualDOC and the optimisation was conducted by DOT. A design of experiments was conducted and the response surfaces fitted to these results. This allowed for an approximate model to be optimised as opposed to performing a full simulation each iteration, which is significantly faster. The first response surface model did not predict the simulation results with sufficient accuracy, so the design space was reduced and a new DOE was conducted and another response surface model was fitted to these results. This windowing process was repeated until the response surface model was considered accurate enough for the purposes of this study. The response surface models are given in Appendix C.

The last response surface model that was generated was used to perform an investigation of the tradeoff between the different responses of interest. It was found that allowing for a heavier system there would be an increase in the peak gain and the side-lobe levels, but with a diminishing return. Pointing error was not considered in this tradeoff investigation as the response surface model did not capture it with sufficient accuracy to inspire confidence in the results.

With the success of this optimisation study, objective 5 has been achieved. In the next chapter, the simulation of a complex antenna model will be discussed and the results presented.

Chapter 5

Case Study: A 32 m Cassegrain Reflector

In this chapter, the full simulation of the Ghana 32 m Communication Telescope will be discussed. This antenna is one of three antennas at the Ghana Intelsat Satellite Earth Station at Kuntunse. This station is located 70 m above sea level with position coordinated of $05^{\circ} 45' 01.5''$ N and $00^{\circ} 18' 18.4''$ N (Asabere *et al.*, 2015). In 2011 a proposal was initiated to convert this antenna from a telecommunication satellite to a fully operational radio telescope by the Ministry of Environment, Science and Technology, Ghana, and the SKA Africa group (Madjitey *et al.*, 2013). The antenna is being converted for use as a single dish observatory and well as for participation in the African Very Long Baseline Interferometry (VLBI) Network (AVN) and the European VLBI Network (Asabere *et al.*, 2015). The initial receiver is planned to operate at 5 GHz and 6 GHz, but later will be fitted with a 1.4 - 1.7 GHz L-band receiver (Asabere *et al.*, 2015).

These simulations of the Ghana antenna were performed to demonstrate that the workflow could be used to perform a full structural and electromagnetic analysis on a complex model. Antennas for radio astronomy have stricter design accuracy requirements than those of communication telescopes, and thus this antenna needs to be analysed to ensure that it can meet the strict accuracy requirements. Figure 5.1 is a photo of the Ghana dish.

It was chosen to perform the full analysis on this antenna as in 2016 Azankpo (2017) created a full Computer Aided Design (CAD) model and a full structural FEA model of this antenna for use in his thesis. It was decided that modifying his already existing models to suit the needs of this research as it would be significantly faster than creating a new model.

In this chapter first the geometry of the antenna will be described, then the changes that had to be made to the numerical model will be discussed, then feed system that was modelled in the EM analysis will also be discussed and finally, the result of the simulations will be presented.



Figure 5.1: A photo of the Ghana 32 m Communication Telescope.

5.1 Antenna geometry

The Ghana antenna is a beam waveguide antenna with Cassegrain reflector surfaces and an altitude over azimuth mounting system. It is mounted on top of the building that houses the control systems and processing units, (Azankpo, 2017). The geometries of interest here are the Cassegrain reflector surfaces. Figure 5.2 shows the variables that are used to define the Cassegrain reflector's geometry, and Table 5.1 shows what these values are. These values are taken from an engineering drawing provided by the SKA office (Venter, 2017).

Table 5.1: Geometry dimensions.

Dimension	Value
D_m	32 m
F_m	10.2792 m
D_s	2.8956 m
$2c$	9.4404 m
$c - a$	0.4929 m
θ_0	8.4983°
ϕ_0	75.7895°

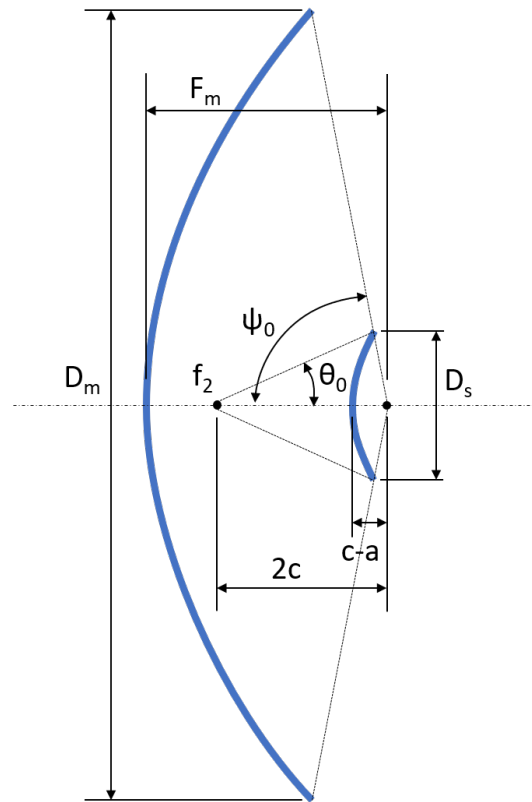


Figure 5.2: Variables used to define the Cassegrain reflector geometry.

5.2 Simulation model

In this section the simulated models will be discussed, it will cover how the structural FEA model was adapted to meet the needs of this study, as well as what feed system was selected for the EM analysis.

5.2.1 Modification of the structural FEA model

In a study by Azankpo (2017) he looked at the main reflector's surface to determine the RMS surface error and the pointing error based on the parabola fitted to the deformed dish surface. The model was design with the focus being on the main reflector surface's deformation and no EM analysis was conducted, as such the model needed some modification to meet the needs of this study.

To perform the full wave simulations the reflector surface meshes need to be in the right locations as dictated by the EM simulation. The meshes need to be on the top surface which differs from what is common practice for structural analysis. The top surface is the surface closest to the direction that the incoming radiation is coming from. In structural analysis it is common to locate the mesh at the mid-surface of a structure when 2D elements are used, Figure 5.3 shows an example of what is meant by a mid-surface mesh.

In the figure, the black box represents the solid that is being approximated by the mid-surface mesh. The mesh location requirement applies to both the main reflector and the sub-reflector. Since the simulations analyse the model at high frequencies the wavelengths are very short, and thus small deviations in the surface locations can have a significant effect on the results. Refer back to Section 3.10 to see how quickly the performance can degrade.

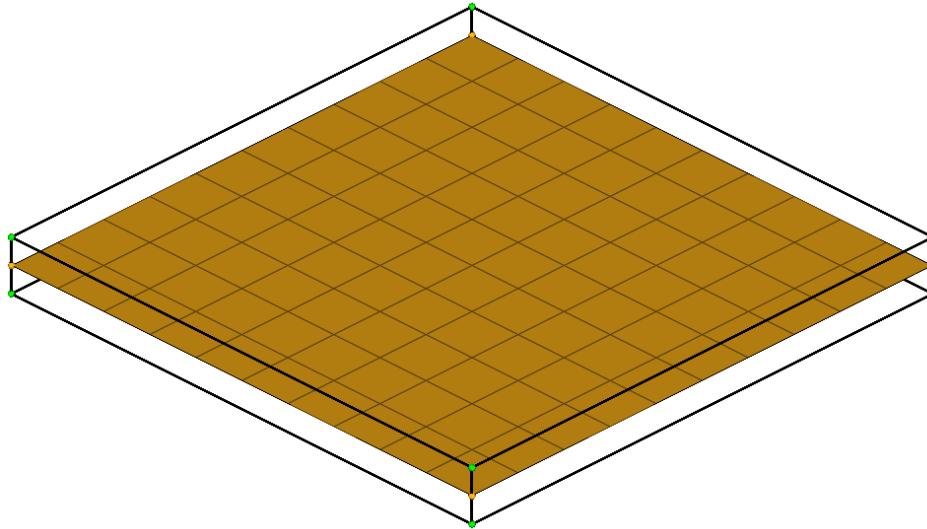


Figure 5.3: This demonstrates what is meant by a mid-surface mesh. This black outline represents the solid geometry that is being approximated by the mid-surface mesh.

The main reflector surface mesh had to be moved to be in the location of the top surface and not at a mid-surface for the component. Using a mid-surface is common practice in structural FEA simulations as it is a simple approximation when using shell elements, and it does not result in significantly more or less theoretical material. It approximately averages out the material overlap that would be created on one side of the surface with spaces created on the other side. However, the top surface needs to be used in the EM simulation as the offset created by using the mid-surface would create a loss in performance, as this means the feed will no longer be at the focal point of the dish surface.

The main reflector mesh was replaced with a new mesh generated from a surface provided by the SKA (Granet, 2017). Before this surface was meshed it was cut into panels that matched the geometries provided in the CAD model that was also developed by Azankpo. The surface needed to be cut into these panels as if it was modelled as a solid surface this would lead to errors in the structural analysis due to a connection between these panels now being modelled.

The mesh size requirements, as stated previously, for EM analysis are often much smaller than they are for structural analysis. Thus the mesh size also needed to be reduced on the surfaces that were going to be used for both the structural analysis and EM analyses. The mesh size will be dependent on the frequency that the antenna will be simulated at, there is a requirement for the mesh size to be several times smaller than the wavelength, which is dependant on frequency. Once the new mesh had been created all the connections between components had to be recreated. To ensure this was possible the new meshes were created to have nodes in all locations where these connections were made.

The sub-reflector was simply modelled as a point mass in Azankpo's structural FEA model. He was only interested in the deformation of the main dish surface, which was used to determine the pointing error, and thus the mass of the sub-reflector was only needed to allow the model to deform accurately. This required the sub-reflector surface to be added to the model and the point mass to be removed. An accurate surface model of the sub-reflector was provided by the SKA (Granet, 2017) which was imported into the structural FEA model. This surface was then meshed according to the EM requirement for mesh size and connected to the support structure mesh through the use of rigid body elements (RBE).

All other components, the entire support structure, was left as they had been originally modelled. It was assumed that these components had been modelled with sufficient accuracy. These components will also not form part of the EM model that will be analysed so they did not need any modification to meet the requirements of such an analysis. This resulted in the structural FEA model that contained approximately 660000 elements, of which approximately 550000 were linear triangular elements describing the main and sub-reflector surfaces. An image of this model can be seen in Figure 5.4

The reflector surfaces were the only geometries/meshes needed in the EM simulations and thus only the meshes for these surfaces were exported to be used in the EM simulations. A number of smaller detail that is not needed or relevant to the EM simulation was removed as they would have little effect on the performance. The EM simulations were also significantly more computationally expensive than the structural analysis simulations, and thus it is important to not waste the resources that are available in these simulations. Modelling every single component in the EM simulations would not likely result in significantly improved results, and thus it was decided to only model the most significant components.

The EM simulations take the reflector surfaces and simulate them along with a feed model to excite the surfaces. The feed model that was used in these simulations will be discussed in the next section. The EM simulations for the Ghana dish, due to the size of the models, were run on Stellenbosch's HPC (Rhasatsha, 2017).

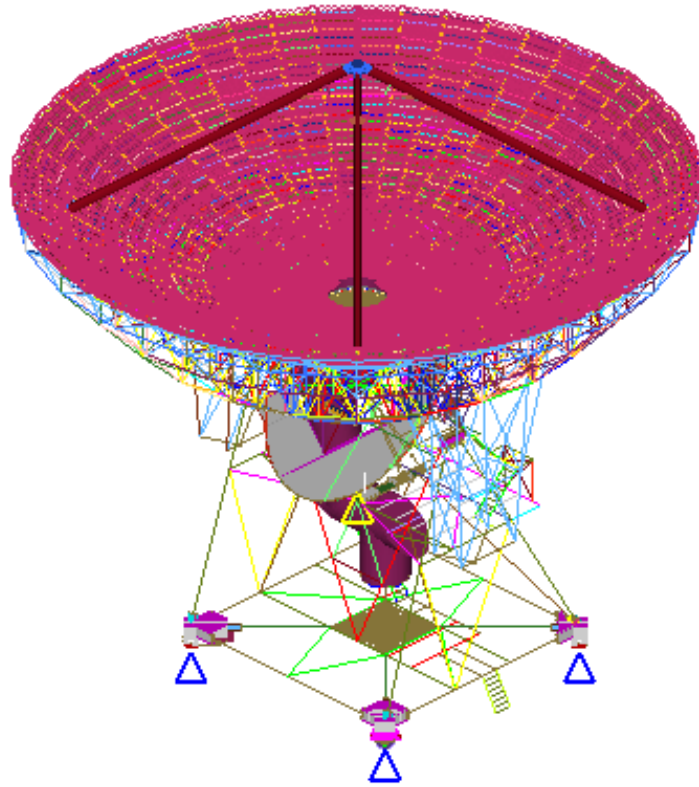


Figure 5.4: The final structural FEA model of the Ghana 32 m antenna. This is the model with an elevation angle of 0° .

5.2.2 Feed structure

To accurately model the antenna electromagnetically a well-matched feed system is required. As noted previously the Ghana antenna is a beam waveguide antenna. The antenna has Cassegrain reflector surfaces and thus it was decided to approximate the antenna as a Cassegrain antenna. The feed system was assumed to radiate from the secondary focus of Cassegrain reflector surfaces. The secondary focus is the point, f_2 , between the main reflector and sub-reflector in Figure 5.2. This simplifies the simulation as the whole waveguide does not need to be simulated. The goal of this case study is to achieve simulation results that correlate to what is expected, and simplifying the feed system will make it easier to predict the simulation results. The main focus is on the effects of the reflector surfaces deforming and the feed structure is assumed to remain ideal.

A spherical mode source of a theoretical feed was used as the feed source. This was modelled as a Gaussian beam with a peak gain of 26 dB and an illumination taper for the sub-reflector of -10 dB. To generate the theoretical feed GRASP 10.4.0 (TICRA, 2015) was used. GRASP is then used to export the spherical wave expansion coefficients, this is a .sph file. The .sph file is then imported into FEKO and used to create the spherical mode source.

The benefit of using a spherical model source is that the feed model does not need to be simulated thus reducing the computational cost. Another reason is that since the antenna modelled has a beam waveguide source there is no physical feed at this secondary focal point.

5.2.3 Verification of the selected feed system

A Cassegrain antenna model, not of the Ghana dish, which had a feed model already defined was used to compare the results of a simulation using a modelled feed and a simulation replacing this feed with a spherical mode source. To create a spherical mode source the spherical mode excitations were extracted from a simulation of just the feed model. A comparison of the far field results of simulating the Cassegrain model with a modelled feed or a spherical mode source is shown in Figure 5.5.

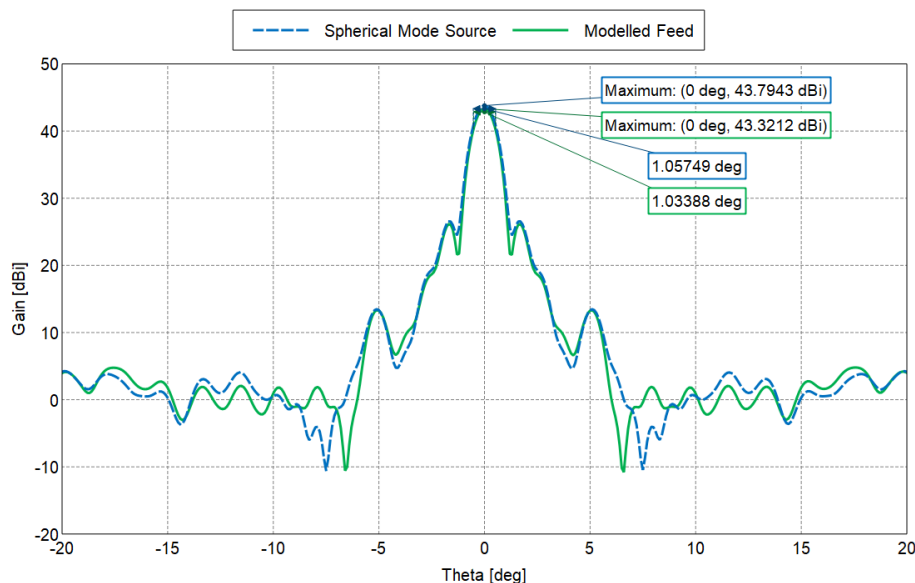


Figure 5.5: Comparison of simulation results of a Cassegrain antenna model with a feed model or a spherical mode source. The spherical mode source results in higher peak gain and a narrower HPBW. This is to be expected.

In the figure, it can be seen that there is an increase in the peak gain and a reduction in the HPBW. These differences in the results are likely due to the

spherical mode source being modelled as a single point and thus there is no coupling between this point source and the reflector surfaces. This spherical mode source will also not cause any blockage. For these reasons, the increase in the performance was expected.

5.3 Simulation results

In this section, the simulations that were run will be discussed and the results presented. First, a verification study was performed on the panelled surfaces that make up the main reflector in the structural FEA model, to determine what kind of effect this will have on the EM results. Then the structural FEA model was modelled under a gravitational load case for various elevation angles and an EM analysis was performed for its deformed shape. All EM simulations were conducted at a frequency of 1 GHz.

5.3.1 Verification of surfaces

The structural FEA model of the 32 m Cassegrain antenna has the main reflector surface made up of large plates that have gaps between them. These gaps could be electrically large depending on the frequency that the EM analysis is being performed at. Thus it was decided to determine the effects of these gaps in the main reflector surface on the far field results for the frequency being used in this study, 1 GHz. First, the main reflector and sub-reflector were modelled using surfaces provided by SKA (Granet, 2017) which are both solid surface with no gaps. Then the main reflector surface was replaced with the panelled surface from the structural FEA model. Both of these cases were simulated and the results and the far field patterns are shown in Figure 5.6.

It can be seen that these gaps in the surface do have an effect on the performance but it is not significant. The peak gain determined in these simulations is ≈ 48.6 dBi and using equation 2.1 this correlates to a total antenna efficiency, η_t , of about 0.65 which is a typical efficiency for a Cassegrain antenna (Bird, 2015). This shows that the model is producing reasonable result even with the panelled surface.

5.3.2 Gravitational loading

The 32 m Ghana antenna was simulated under a gravitational load case for several elevation angles. These angles were 0° , 30° , 90° , -30° and -90° with 0° being the case when the antenna is pointing straight up into the sky. The structural analysis simulations were conducted using Nastran 2014 (Nastran, 2016). This is due to the models originally being developed to run in Nastran and thus there were compatibility issues with using OptiStruct to perform the analysis. The rest of the workflow described in Chapter 3 remains the

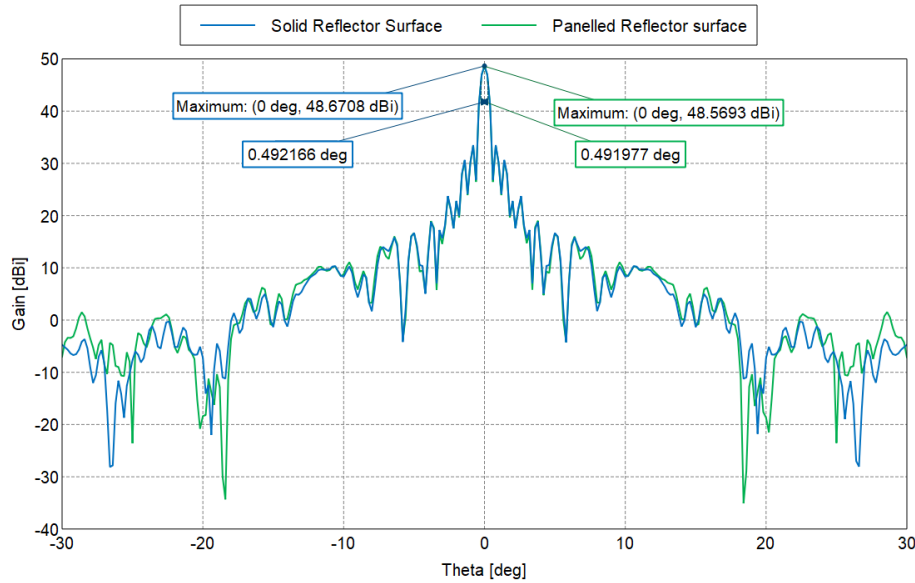


Figure 5.6: Comparison of the far field pattern for a solid main reflector surface and the panelled surface in the structural FEA model. It can be seen the the difference in the pattern is negligible.

same with the only change being to the structural analysis solver used. The deformations predicted by the modified models were compared to the results determined in Azankpo’s simulations for the same load cases. This comparison can be seen in Table 5.2. Appendix D shows the deformation plots of the main reflector for the load cases simulated.

Table 5.2: A comparison of the maximum and minimum deformations experienced by the main reflector for the original and the modified model. It can be seen that these values are very similar.

Angle	Min Deformation [mm]		Max Deformation [mm]	
	Modified	Original	Modified	Original
0°	2.08	2.25	6.94	6.69
30°	2.54	2.65	6.80	6.43
90°	3.30	3.13	5.04	4.52
-30°	2.54	2.65	6.79	6.43
-90°	3.20	3.13	4.87	4.52

It can be seen that in all simulations that the deformations predicted are very similar. These minimum and maximum deformations were experienced in the same region of the model. These differences are small enough that the modified model will be considered to be performing correctly. These small

changes in results are within the range expected due to the changes made to the model.

After the structural analyses had been performed the main reflector and sub-reflector surfaces were imported into an EM model and simulated using the spherical mode source described in subsection 5.2.2. The models were simulated for both the undeformed and deformed case and the results compared. Figure 5.7 shows the far field patterns for each load case as well as for the undeformed case.

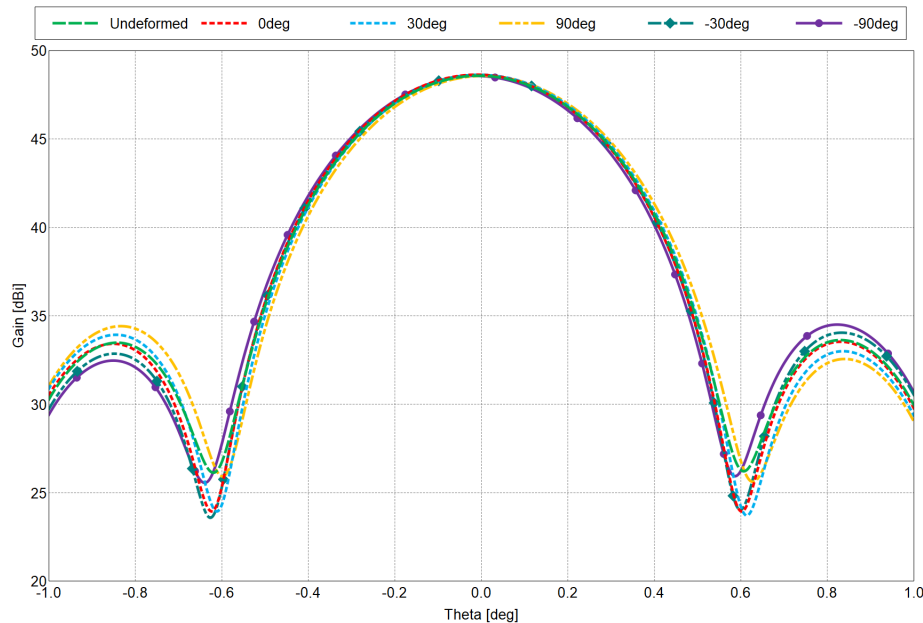


Figure 5.7: The far field pattern for the Ghana dish at the various elevation angles as well as the far field pattern of the undeformed antenna. The most distinct difference between the patterns is the side-lobe levels.

In all the simulated elevation angles there was a negligible change in the peak gain of the antenna, however, in the worst case, the side-lobe level decreased by 0.8842 dB which is very significant. The largest change in the pointing error was about 0.011° . Depending on the application this may not be negligible. For instance, if the antenna is to be used in an array like the MeerKAT antenna, then this pointing error may be too large. 0.011° converts to about 39.6 arcseconds, which is nearly twice the allowable pointing error for the MeerKAT antenna under normal operation. Table 5.3 shows the gain, side-lobe level and pointing error determined in for each of the simulated elevation angles.

Table 5.3: A comparison of the gain, side-lobe level, and pointing error determined in the simulations for the different elevation angle, as well as the undeformed case. It can be seen that the Gain does not change significantly, however, there is a significant drop in the side-lobe levels.

Angle	Gain	SSL	Pointing error
Undeformed	48.5732	14.9461	-0.006
0°	48.6332	15.0997	-0.010
30°	48.6244	14.6981	-0.004
90°	48.5653	14.1515	0.005
-30°	48.6227	14.5777	-0.008
-90°	48.5650	14.0619	-0.016

This demonstrates that just a simple gravitational loading could have a significant effect on a radio telescopes performance. It also shows the benefit of performing a full wave analysis and investigating the effects on more than one response. In this case, there was an insignificant decrease in peak gain however the side-lobe level decreased significantly.

5.4 Summary

An structural FEA model of the Ghana 32 m reflector antenna was modified to meet the requirements for an EM analysis. This required ensuring that the main reflector and sub-reflector meshes were on the top surface of the geometries they represented in the structural FEA model, and that the mesh size for these surfaces was sufficiently fine for the EM analysis. An EM model was then created to perform a full wave analysis on these surfaces, and a feed system had to be modelled. The feed was modelled as a spherical mode source with a Gaussian beam pattern. The structural FEA model was simulated for the antenna at different elevation angles under a gravitational load case to determine the deformations of the reflector surfaces. These deformed surfaces were then simulated in the EM model and the results compared to this model in an undeformed state. It was found that deformations did not have a significant effect on the peak gain of the antenna, but there was a significant decrease in the side-lobe level. The change in pointing error could be considered significant depending on the application of the antenna.

The 32 m Cassegrain reflector antenna was successfully simulated. This demonstrates that the workflow that has been developed is not limited to simple models, but can be used on complex models as well. The simulation results also reveal the benefit of performing the full wave simulation, in that even though an insignificant drop in peak gain was determined the side-lobe levels decreased significantly. This decrease in side-lobe level would result in greater interference from these side-lobes in the received signal.

With this model being successfully simulated this means that objective 6 has been achieved, and thus all objectives for the study have been achieved. In the next chapter the whole study will be concluded and summarised and recommendations for future work will be presented.

Chapter 6

Conclusions and Recommendations

In this chapter the work done will be summarised and conclusions will be drawn. Recommendations on the future work will also be made.

6.1 Summary of work done

Structural deformation to reflector antennas, such as those used in radio astronomy applications, can have a significant effect on the antenna's electromagnetic performance. Being able to determine effects of deformation on a radio telescope's performance will be useful for projects like the SKA where the radio telescopes they are designing have very strict performance specifications.

In this study, a workflow was developed to perform a full structural and electromagnetic analysis of a radio telescope. This workflow was developed to be used with FEKO, HyperMesh and OptiStruct, and Python and TCL scripts were written to automate parts of the process. The workflow was verified against analytical approximations and found to simulate results that had a close correlation to what was predicted by these approximations. It was also found that in terms of RMS error a correlated error due to deformation can have a greater effect on the peak gain of an antenna compared to an uncorrelated random error.

The workflow was then used in an optimisation study to illustrate the manner in which this workflow could be used to optimise a primary-fed reflector antenna. This optimisation process used response surface methodology to develop approximate functions for the performance responses that were of interest. These responses include the mass, pointing accuracy, peak gain, and maximum side-lobe level. This response surface drastically accelerates the process of finding an optimum. The results of optimising the response surface were validated by performing a full simulation at the determined optimum. The first response surface, which was fitted over the entire design space, was not considered accurate enough. The range of the design space analysed

was reduced and a new response surface was created for the reduced space. This was repeated until the accuracy was determined to be acceptable for the purpose of this study. The final response surface was then used to perform a tradeoff investigation. This found that there was a diminishing return on peak gain if the allowable mass was increased. There was also a trend for the side-lobe level to increase with gain, but this was not always true.

A case study was performed on a 32 m Cassegrain reflector antenna. This case study was performed to demonstrate that the workflow could be used to analyse a real-life antenna and not just simple models. It was analysed under a gravitational load case with the antenna at different elevation angles. The deformation results were compared to those of Azankpo (2017) and were found to be comparable. In the EM analysis there was not a significant change to the peak antenna gain, however in every load case simulated the side lobe level worsened by a significant margin, up to 0.8842 dB, which means these side-lobes would introduce more interference. There were also predicted increases in the pointing error, of up to 0.011° , which is not insignificant when compared to the accuracy requirements of the MeerKAT antenna. This pointing error is nearly double what is allowable for the MeerKAT under normal conditions.

All of the research objectives stated in Chapter 1 were achieved.

6.2 Recommendation for Future Work

This work could be expanded to perform a more in-depth analysis of a real-life radio telescope. The analysis could include validation of the simulated results to experimental data taken from the radio telescope analysed. This experimental data could include the deformations the real antenna experiences or the stress and strains in key components or locations under a known load case. These measurements could be compared to the structural analysis results. On the EM side, holography measurements could be taken while the antenna is under the same load case. This would allow for a correlation between a real-life deformation and an EM result to be compared to the simulations of that same load case. The 32 m Cassegrain antenna that was analysed in the case study would be a good choice as it is in the process of being re-purposed for use in radio astronomy. The antenna could also be modelled under different load cases, such as wind or thermal loading as well as a combination of these load cases.

A parametric study could be performed on variables that have not already been investigated here to determine their effect on antenna performance. A better understanding of how the mechanical structure of a radio telescope can affect its performance would be very useful in the design of future telescopes. This could include simulating the effects of different types of backing structures, not just the simple model used in this study.

A more comprehensive optimisation study could be performed on a radio telescope, preferably one that is still in development, to optimise its performance under the load cases it is expected to be subjected to during operation. This type of optimisation could include tuning the shape, size, and location of the struts supporting the feed system to find a balance between minimising aperture blockage while still maintain structural stiffness in these struts.

The workflow could be developed into a simple software package to handle the interaction between the structural FEA and EM software packages. This would allow for this full structural and electromagnetic analysis process to be more easily implemented in the antenna design process. The process of extracting the correct information from the structural analysis results to be used in the EM analysis can be time-consuming if done manually and writing scripts to do it can take just as long to create.

These are just some of the potential uses for the work presented here.

List of References

- Asabere, D.B., Gaylard, M.J., Horellou, C., Winkler, H. and Jarrett, T. (2015). Radio astronomy in Africa: the case of Ghana. *Proceedings of SAIP2014, the 59th Annual Conference of the South African Institute of Physics*, pp. 296–301.
- Azankpo, S. (2017 March). *Surface Accuracy and Pointing Error Prediction of a 32 m Diameter Class Radio Astronomy Telescope*. Master's thesis, Stellenbosch University.
- Baars, J.W.M. (2007). *The paraboloidal reflector antenna in radio astronomy and communication : theory and practice*. Springer.
- Balanis, C.A. (1997). *Antenna Theory Analysis and Design*. 2nd edn. John Wiley & Sons, INC.
- Bird, T.S. (2015). *Fundamentals of Aperture Antennas and Arrays: From Theory to Design, Fabrication and Testing*. Wiley.
- Cook, R.D., Malkus, D.S., Plesha, M.E. and Witt, R.J. (2001). *Concepts and applications of finite element analysis*. 4th edn. John Wiley & Sons, INC.
- Davidson, D. (2011). *Computational electromagnetics for RF and microwave engineering*. 2nd edn. Cambridge University Press.
- Dewdney, P., Hall, P., Schilizzi, R. and Lazio, T. (2009 August). The square kilometre array. *Proceedings of the IEEE*, vol. 97, no. 8, pp. 1482–1496.
- DOT (2013). *Design Optimization Tools (DOT) User Manual, version 6*. Vanderplaats Research & Development, 1767 S. 8th Street, Suite 200 Colorado Springs, CO 80905.
- FEKO (2015). Overview of FEKO. Online.
Available at: www.feko.info/product-detail/overview-of-feko
- Figliola, R.S. and Beasley, D.E. (2011). *Theory and Design for Mechanical Measurements*. 5th edn. John Wiley & Sons, INC.
- Garrett, M. (2015). Lecture notes on radio telescopes. Online.
Available at: <https://tinyurl.com/yagrfo38>

- Granet, C. (2017). Approximate geometry of the TIW 32m antenna for SKA South Africa, African VLBI Network. Tech. Rep., Lyrebird Antenna Research, Sydney, Australia.
- HyperMesh (2017). Hypermesh overview. Online.
Available at: <http://www.altairhyperworks.com/product/hypermesh>
- IEEE (1993 March). IEEE standard definitions of terms for antennas. *IEEE Std 145-1993(R2004)*.
- Kellermann, K. (2016). Radio telescope. Online.
Available at: global.britannica.com/science/radio-telescope
- Kildal, P. (2015). *Foundations Of Antennas Engineering: A Unified Approach For Line-Of-Sight And Multipath*. Kildal.
- Kraus, J.D. (1966). *Radio Astronomy*. 1st edn. McGraw-Hill, New York.
- Ludick, D., Venter, M., Davidson, D. and Venter, G. (2015). A multiphysics analysis of dish reflector antennas for radio astronomy applications. *2016 10th European Conference on Antennas and Propagation (EuCAP)*.
- Madjitey, F., Ansah-Narh, T. and Nsor, J. (2013). Analysis of 32-m dual reflector cassegrain antenna feed system at Nkuntunse, Ghana. *Research Journal in Engineering and Applied Sciences*, vol. 2, no. 4, pp. 230–235.
- Minoli, D. (2009). *Satellite systems engineering in an IPv6 environment*. CRC Press.
- Myers, R., Montgomery, D. and Anderson-Cook, C. (2009). *Response Surface Methodology*. 3rd edn. John Wiley & Sons, INC.
- Nastran (2016). *Nastran: 2014 Reference Manual*. MSC Software Corporation, 4675 MacArthur Court, Suite 900, Newport Beach, CA 92660,.
- Optistruct (2017). Optistruct overview. Online.
Available at: <http://www.altairhyperworks.com/product/OptiStruct>
- Python (2016). Python 2.7.13. Online.
Available at: <https://www.python.org>
- Quasar (2017). Definition of 'quasar'.
Available at: <https://www.collinsdictionary.com/dictionary/english/quasar>
- Rhasatsha (2017). University of Stellenbosch's high performance clusters.
Available at: <http://www.sun.ac.za/hpc>
- Ruze, J. (1966 April). Antenna tolerance theory - a review. *Proceedings of the IEEE*, vol. 54, no. 4, pp. 633–640.
- SKA-SA (2012). The South African Square Kilometre Array project. Online.
Available at: www.ska.ac.za

- SKA-SA (2016). South Africa's MeerKAT radio telescope. SKA SA, 3rd floor, The Park, Park Street, Pinelands, 7405.
- SKA-SA (2017). MeerKAT radio telescope.
Available at: <http://www.ska.ac.za/science-engineering/meerkat/about-meerkat/>
- Song, J., Lun, C.-C. and Chew, W.C. (1997 October). Multilevel fast multipole algorithm for electromagnetic scattering by large complex objects. *IEEE Transactions on Antennas and Propagation*, vol. 45, no. 10, pp. 1488–1493.
- TCL/TK (2016). About tcl/tk.
Available at: <https://www.tcl.tk/about/>
- TICRA (2015). Grasp, version 10.4.
Available at: <http://www.ticra.com/products/software/grasp>
- van Tonder, J.J. and Jakobus, U. (2005). Fast multipole solution of metallic and dielectric scattering problems in FEKO. *Wireless Communications and Applied Computational Electromagnetics 2005. IEEE/ACES International Conference*, pp. 511–514.
- Venter, M. (2017 June). AVN ghana station: 32 m antenna optics. Tech. Rep. A0210-0000-064, SKA South Africa.
- VisualDOC (2013). *VisualDOC user's manual, version 7.2*. Vanderplaats Research & Development, 1767 S. 8th Street, Suite 200 Colorado Springs, CO 80905.
- Wolff, C. (2014). Angular resolution.
Available at: <http://www.radartutorial.eu/01.basics/Angular%20Resolution.en.html>

Appendices

Appendix A

Applying the workflow on a high-performance cluster

All simulations and scripts that were presented had all been developed in a Windows environment. It was decided that to increase the number of simulations that could be run in a given time that the High-Performance Cluster (HPC) Rhasatsha (Rhasatsha, 2017) available at Stellenbosch University would be used. This HPC, however, uses a Linux operating system and thus the scripts needed some modification to run properly in this environment.

First, the batch scripts that were written to run on Windows were rewritten as Shell scripts which could run on Linux. Batch scripts can only be run on a Windows operating system and this is why they were replaced. A Shell script is the Linux equivalent of a Batch script, however, the actual commands do differ.

Another problem with running the simulations on the HPC is that it does not use any graphical user interface, and thus in some cases it may be necessary to fake this interface. This was done during the testing of the various scripts run in the optimisation process to ensure they still executed correctly in Linux. However, it was found that this was no longer necessary to do this when VisualDOC was used.

When calling a script from another script in Linux the called script may not have permission to actually execute. This required for every script that would be executed to be given execute permissions before it was called. This was done by having the script which already had execute permissions give any script it was going to call execute permission as well before calling it.

When developing the Python scripts, Python 2.7.13 was used on the Windows PC, thus compatibility with the Python version on the HPC needed to be ensured. The HPC has many different versions of Python installed but it did not have version 2.7.13. It was found that Python 2.7.10 on the HPC was compatible with the scripts that had been written and thus this version of Python was used.

There was a problem when calling HyperMesh directly from VisualDOC

APPENDIX A. APPLYING THE WORKFLOW ON A HIGH-PERFORMANCE CLUSTER **A.2**

that could not be resolved. A solution was found however in using VisualDOC to call a Shell script that then called HyperMesh. It is not clear why this problem exists but the simulation would freeze when VisualDOC tried to call HyperMesh.

When running multiple optimisations on the HPC, each one was given its own directory containing all the files it would need to complete the optimisation. By separating the optimisations into their own directories there is no chance of the results of one simulation overwriting the results of a different one. There is no race condition possible as each optimisation will write to its own result files. Keeping all the other files produced in an optimisation grouped with that optimisation is also convenient if these results are to be queried since all the files used to get those results can easily be found.

Appendix B

Scripts

B.1 Python script to calculate feed transformations

This script reads in two files containing the nodal coordinates for all elements in the model, among other information. It also reads in a file that contains the reference nodes, these nodes are used to determine the feed displacement and rotation using the method described in Section 3.4. It then writes these results to a file to be read by FEKO.

```
1 import numpy as np
2
3 # Rotation function
4 def rotation_matrix(theta_x, theta_y, theta_z):
5     pass
6     # """
7     # Return the rotation matrix used by FEKO.
8     # """
9     # axis = np.asarray(axis)
10    a1 = float(theta_x)
11    b1 = float(theta_y)
12    c1 = float(theta_z)
13
14    # ** Precompute sin() and cos() terms
15    ca1 = np.cos(a1)
16    # print ca1
17    cb1 = np.cos(b1)
18    cc1 = np.cos(c1)
19    sa1 = np.sin(a1)
20    sb1 = np.sin(b1)
21    sc1 = np.sin(c1)
22
```

```

23     # ** Auxiliary terms resulting from equating the
        transformation matrices
24     cc2 = cb1*cc1/(np.sqrt(((cb1*cc1)**2) + ((ca1*sc1-sa1*sb1*
        cc1)**2)))
25     cb2 = cb1*cc1/cc2
26     ca2 = ca1*cc2/cc1
27     sa2 = cc2*(sa1*cc1-ca1*sb1*sc1)/(cb1*cc1)
28     sb2 = sa1*sc1+ca1*sb1*cc1
29     sc2 = cc2*(ca1*sc1-sa1*sb1*cc1)/(cb1*cc1)
30
31     # ** Finally compute the angles which must be used in FEKO
        in the TG card
32     # ** for the rotation order first around z, then around y,
        and then around x
33
34     a2 = np.arctan(sa2/ca2)
35     b2 = np.arctan(sb2/cb2)
36     c2 = np.arctan(sc2/cc2)
37
38     sx = np.sin(a2)
39     sy = np.sin(b2)
40     sz = np.sin(c2)
41     cx = np.cos(a2)
42     cy = np.cos(b2)
43     cz = np.cos(c2)
44
45     return np.array([[cy*cz, -cy*sz, sy],
46                     [cx*sz+sx*sy*cz, cx*cz-sx*sy*sz, -sx*cy],
47                     [sx*sz-cx*sy*cz, sx*cz+cx*sy*sz, cx*cy]])
48
49 def isInt(value):
50     try:
51         int(value)
52         return True
53     except ValueError:
54         return False
55
56 def print_float_8(val, tol=0.0):
57     if abs(val) <= tol:
58         return '0.00000000'
59     if val < 0:
60         return print_float_neg_8(val)
61     else:
62         return print_float_pos_8(val)

```

```
63
64 def print_float_pos_8(val):
65     if val < 0.01 or val > 100000:
66         return print_float_sci_pos_8(val)
67     if val < 0.1:
68         f = '%8.6g' % val
69     else:
70         f = '%8.7g' % val
71     if len(f) > 8:
72         f = f.lstrip('0')
73     return f
74
75 def print_float_sci_pos_8(val):
76     if val < 1.0e-9:
77         f = ('%10.3e' % val).replace('e-', '-')
78     elif val < 1:
79         f = ('%10.4e' % val).replace('e-0', '-')
80     elif val <= 1.0e9:
81         f = ('%10.4e' % val).replace('e+0', '+')
82     else:
83         f = ('%10.3e' % val).replace('e+', '+')
84     return f.strip()
85
86 def print_float_neg_8(val):
87     if val > -0.01 or val < -10000:
88         return print_float_sci_neg_8(val)
89     if val > -0.1:
90         f = '%8.5g' % val
91     else:
92         f = '%8.6g' % val
93     if len(f) > 8:
94         f = f.replace('-0.', '-.')
95     return f
96
97 def print_float_sci_neg_8(val):
98     if val > -1.0e-9:
99         f = ('%-8.2e' % val).replace('e-', '-')
100    elif val > -1:
101        f = ('%-8.3e' % val).replace('e-0', '-')
102    elif val >= -1.0e9:
103        f = ('%-8.3e' % val).replace('e+0', '+')
104    else:
105        f = ('%-8.2e' % val).replace('e+', '+')
106    return f.strip()
```

```

107
108 #####
109 n = 8
110
111 # Read in the required files
112 results = []
113 with open('output.bdf') as inputfile1:
114     for line in inputfile1:
115         results.append(line.strip().split(','))
116 res = np.asarray(results)
117
118 Undeformed = []
119 with open('FEKOmesh.fem') as inputfile2:
120     for line in inputfile2:
121         Undeformed.append([line[i:i+n] for i in range(0, len(line)
122             ), n)])
123 udef = np.asarray(Undeformed)
124
125 #create the output file
126 f = open('Deformed/diff.dat', 'wb')
127
128 token = '␣'
129 nodes = [0, 0, 0, 0, 0]
130 ff = np.zeros((5, 5))
131
132 # Read the reference node file
133 reference = []
134 with open('ref.txt') as ref:
135     for line in ref:
136         reference.append(line.strip().split('␣'))
137 ref_nodes = np.asarray(reference)
138
139 i = 0
140 for nodes1 in res:
141     if nodes1[0] == 'GRID' and nodes1[1] in [ref_nodes[0][0],
142         ref_nodes[0][1], ref_nodes[0][2], ref_nodes[0][3],
143         ref_nodes[0][4]]:
144         for nodes2 in udef:
145             if nodes2[0] == 'GRID␣␣␣␣' and (int(nodes2[1]) == int
146                 (nodes1[1])):
147                 for k in range(len(nodes1)):
148                     if k in [3,4,5]:
149                         cell = nodes1[k]

```

```

147         j = 0
148         if len(cell) > 5 and cell[4] == '-' and
           cell[3] != '␣' and cell[3] != 'e':
149             cell = cell[0:4]+'e'+cell[4:8]
150             nodes1[k] = cell
151         if len(cell) > 5 and cell[5] == '-' and
           cell[4] != '␣' and cell[4] != 'e':
152             cell = cell[0:5]+'e'+cell[5:8]
153             nodes1[k] = cell
154         if len(cell) > 6 and cell[6] == '-' and
           cell[5] != '␣' and cell[5] != 'e':
155             cell = cell[0:6]+'e'+cell[6:8]
156             nodes1[k] = cell
157
158     for k in range(len(nodes2)):
159         if k in [3,4,5]:
160             cell = nodes2[k]
161             j = 0
162             if len(cell) > 5 and cell[4] == '-' and
               cell[3] != '␣' and cell[3] != 'e':
163                 cell = cell[0:4]+'e'+cell[4:8]
164                 nodes2[k] = cell
165             if len(cell) > 5 and cell[5] == '-' and
               cell[4] != '␣' and cell[4] != 'e':
166                 cell = cell[0:5]+'e'+cell[5:8]
167                 nodes2[k] = cell
168             if len(cell) > 5 and cell[6] == '-' and
               cell[5] != '␣' and cell[5] != 'e':
169                 cell = cell[0:6]+'e'+cell[6:8]
170                 nodes2[k] = cell
171
172     nodes[0] = nodes1[1]
173     nodes[1] = str(float(nodes1[3])-float(nodes2[3]))
174     nodes[2] = str(float(nodes1[4])-float(nodes2[4]))
175     nodes[3] = str(float(nodes1[5])-float(nodes2[5]))
176
177     ff[i][0] = int(nodes1[1])
178     ff[i][2] = nodes1[3]
179     ff[i][3] = nodes1[4]
180     ff[i][4] = nodes1[5]
181     i = i + 1
182
183     if nodes1[1] == ref_nodes[0][4]:
184         offset = nodes2[5]

```

```

185         center_undef = nodes2
186         if nodes1[1] == ref_nodes[0][0]:
187             edge_undef = nodes2
188
189 for points in ff:
190     if points[0] == int(ref_nodes[0][0]):
191         x1 = points
192     if points[0] == int(ref_nodes[0][1]):
193         x2 = points
194     if points[0] == int(ref_nodes[0][2]):
195         y1 = points
196     if points[0] == int(ref_nodes[0][3]):
197         y2 = points
198
199 # Write rotations to the output file
200 f.write('Rotations_□[x□y□z]□\r\n')
201
202 # rotation about x
203 dyx = y2[2] - y1[2]
204 dyy = y2[3] - y1[3]
205 dyz = y2[4] - y1[4]
206 theta_x = np.arcsin(dyz/dyy)
207 f.write(str(theta_x)+'□')
208
209 # Rotation about y
210 dxx = x2[2] - x1[2]
211 dxy = x2[3] - x1[3]
212 dxz = x2[4] - x1[4]
213 theta_y = np.arcsin(dxz/dxx)
214 f.write(str(theta_y)+'□')
215
216 # roation about z
217 theta_z = np.arctan(dxy/dxx)
218 f.write(str(theta_z)+'□\r\n')
219
220 # find feed displacement
221
222 # define centre
223 for nodes1 in res:
224     if nodes1[0] == 'GRID' and nodes1[1] == ref_nodes[0][4]:
225         center = nodes1
226
227 M = rotation_matrix(theta_x, theta_y, theta_z)
228 C = np.array([float(center_undef[3]), float(center_undef[4]),

```

```

    float(center_undef[5]))
229 C_t = C.dot(M)
230 delta = ([float(center[3])-C_t[0], float(center[4])-C_t[1],
    float(center[5])-C_t[2]])
231
232 # Write the displacements to the output file
233 f.write('Displacement_□[x□y□z]□\r\n')
234 f.write('%0.14f□' % delta[0] + '%0.14f□' % delta[1] + '%0.14f□' %
    delta[2] + '\r\n')

```

B.2 TCL script to export deformed mesh

This script uses the input file from structural analysis and the results file to generate the two BDF files to be used by the Python which calculates the feed transformations. It also creates the .stl file of the deformed dish mesh. This script was originally created by Nic Minnaar from Altair Stellenbosch but was modified to suit the needs of this study.

```

1 #####
2 # File : export_deformed.tcl
3 # Date : October 5, 2016
4 # Created by : Nic Minnaar
5 # Purpose: : This script imports a fem deck, applies nodal
    displacement
6 # results, and finally exports a new fem deck with the deformed
    mesh
7 # Notes : This script is only compatible with v14.0.130 and
    onwards on Windows
8 #####
9 #log_file myfile.log ;# <<< === append output to a file
10 #####
11 proc ::hm::ExportDeformed {input_deck results_file output_file}
    {
12     # Purpose: 1)Import .fem deck 2)Load results file 3) Apply
        displacements
13     # 4)Export deformed me
14     # Args: input_deck - The .fem file generated by HM for
        Optistruct
15     # results_file - the .res file generated by optsitruact
16     # output_file - The desired path and filename of the
        outputfile
17     # Notes: The arguments need to be prepended with the full
        path names to the files

```

```
18 puts "start"
19 #1)
20 *feinputpreserveincludefiles
21 *createstringarray 8 "Optsitruct " " " "ANSA " "PATRAN " "
    ASSIGNPROP_BYHMCMENTS" \
22 "LOADCOLS_DISPLAY_SKIP " "SYSTCOLS_DISPLAY_SKIP " "
    CONTACTSURF_DISPLAY_SKIP "
23 *feinputwithdata2 "\#optistruct\\optistruct" $input_deck 0 0
    0 0 0 1 8 1 0
24
25 #Undeformed Dish
26 *allsuppressoutput 1
27 # set component, dish, to export
28 *entitysuppressoutput comps 1 0
29 #*entitysuppressoutput comps 3 0
30 if {[file exists Ideal] == 1} {
31     *feoutputwithdata "/apps/altair/hyperworks/14.0/altair/
        templates/feoutput/misc/triasto.stl" "Ideal/dish.stl" 0
        0 2 1 1
32 }
33 #2)
34 *analysisfileset $results_file
35
36 #3)
37 *createmark nodes 1 "all"
38 *applyresults nodes 1 1 "total disp"
39
40 #4)
41 # Need to make sure that only elements and gridpoints are
    exported
42 *createmark comps 1 all
43 *createstringarray 2 "elements_on" "geometry_on"
44 *isolateonlyentitybymark 1 1 2
45 *createstringarray 3 "HM_NODEELEMS_SET_COMPRESS_SKIP" "
    HM_FREE_FORMAT_EXPORT_OPTION" "HMCMENTS_SKIP"
46 hm_answernext yes
47 *feoutputwithdata "/apps/altair/hyperworks/14.0/altair/
    templates/feoutput/optistruct/optistruct" $output_file 0
    0 0 1 3
48 # set all components to not exported
49 *allsuppressoutput 1
50 # set component, dish, to export
51 *entitysuppressoutput comps 1 0
52 *entitysuppressoutput comps 3 0
```



```
53     *createstringarray 1 "CONNECTORS_SKIP"
54     *feoutputwithdata "/apps/altair/hyperworks/14.0/altair/
      templates/feoutput/misc/triasto.stl" "Deformed/dish.stl"
      0 0 2 1 1
55
56 }
57
58 proc ::hm::ReadFileContents {file_name} {
59
60     set file [open $file_name r]
61     while { ![eof $file]} {
62         gets $file line
63         set argus [split $line ,]
64     }
65     close $file
66     return $argus
67 }
68
69 ##Read in arguments necessary for function call from text file
      created by batch script
70 #set files [::hm::ReadFileContents arguments.txt]
71 #set wdir [::hm::ReadFileContents wdir.txt]
72
73 #Prepand file names with the path directory
74 #set files [lreplace $files 0 0 "[lindex $wdir 0]\\\"[lindex
      $files 0]"]
75 #set files [lreplace $files 1 1 "[lindex $wdir 0]\\\"[lindex
      $files 1]"]
76 #set files [lreplace $files 2 2 "[lindex $wdir 0]\\\"[lindex
      $files 2]"]
77
78 #Call the function previously created with all relevant
      arguments
79 #::hm::ExportDeformed [lindex $files 0] [lindex $files 1] [
      lindex $files 2]
80 ::hm::ExportDeformed FEK0mesh1.fem FEK0mesh1.res output.bdf
81
82 #set input_deck FEK0mesh1.fem
83 #set results_file FEK0mesh1.res
84 #set output_file output
```

B.3 TCL script for VisualDOC

This script generates the primary-fed reflector antenna model that is optimised in Chapter 4. This script defines the material properties, builds the geometries and creates the meshes of them, and creates the load case. It then calls the batch script that automates the rest of the simulation process.

```
1 set t0 1.443
2 set back 0.01704
3
4 *templatefileset [file join [hm_info -appinfo ALTAIR_HOME]
   templates feoutput optistruct optistruct]
5
6 proc Gen {FoD name gx gy gz back t0 Ideal} {
7   cd "../"
8   *renumberall 1 1 0 0
9   hm_answernext yes;
10  *deletemodel
11
12  set offset -0.3
13  set sx 4.263
14
15  set pi 3.14159265358979323846
16  set eps0 8.85418781761e-12
17  set mu0 [expr {$pi*4e-7}]
18
19  set c0 [expr {1/sqrt($eps0*$mu0)}]
20  set Dm 12
21
22  set F [expr {$FoD*$Dm}]
23
24  set freq 1000000000
25
26  set wavelength_centre [expr {$c0/$freq}]
27
28  set guide_radius_lam 0.5
29  set guide_radius [expr {$guide_radius_lam*$wavelength_centre}]
30  set guide_length $guide_radius
31
32  set MRmeshFact 0.25
33  set MRmesh [expr {$MRmeshFact*$c0/$freq}]
34
35  set PCshift -0.0055583
36
```

```
37 set portMesh [expr {$guide_radius/10}]
38
39 set Wl 0.1
40 set H [expr ($Dm/2)**2/(4*$F)]
41
42 set name "$name-back$back-offset$offset-strut$sx"
43 if {[file exists "Test/$name"]} == 1} {
44     set name "$name-1"}
45
46 # Create materials & properties
47 # Dish
48 *createentity mats name=Steel1 cardimage=MAT1
49 *setvalue mats id=1 STATUS=1 1=200e9
50 *setvalue mats id=1 STATUS=1 2=86e9
51 *setvalue mats id=1 STATUS=1 3=0.3
52 *setvalue mats id=1 STATUS=1 4=8000
53
54 *createentity props name=Steel1 cardimage=PSHELL
55 *setvalue props id=1 materialid={mats 1}
56 *setvalue props id=1 STATUS=1 95=0.003
57
58 #feed
59 *createentity materials name=Steel2 cardimage=MAT1
60 *setvalue mats id=2 STATUS=1 1=200e24
61 *setvalue mats id=2 STATUS=1 2=86e24
62 *setvalue mats id=2 STATUS=1 3=0.3
63 *setvalue mats id=2 STATUS=1 4=8000
64
65 *createentity props name=Steel2 cardimage=PSHELL
66 *setvalue props id=2 materialid={mats 2}
67 *setvalue props id=2 STATUS=1 95=0.003
68 #struts
69 *createentity materials name=Steel3 cardimage=MAT1
70 *setvalue mats id=3 STATUS=1 1=200e9
71 *setvalue mats id=3 STATUS=1 2=86e9
72 *setvalue mats id=3 STATUS=1 3=0.3
73 *setvalue mats id=3 STATUS=1 4=8000
74
75 *createentity props name=Steel3 cardimage=PSHELL
76 *setvalue props id=3 materialid={mats 3}
77 *setvalue props id=3 STATUS=1 95=0.003
78
79 # Create Dish
80 *collectorcreate component Dish Steel1 50
```

```
81 *setvalue comps id=1 propertyid={props 1}
82 *setvalue comps id=1 materialid={mats 1}
83
84 set H1 [expr {- $F + ($Dm/2*$Dm/2) / (4 * $F)}]
85 set H2 [expr {- $F - ($Dm/2*$Dm/2) / (4 * $F)}]
86
87 *linecreateconic [expr {$Dm/2}] 0 $H1 [expr {- $Dm/2}] 0 $H1 0
    0 $H2 0.5
88 *createnode 0 0 -4.2 0 0 0
89 *linesplitatpoint 1 1
90 *createplane 1 0 0 1 0 0 0
91
92 *createmark lines 1 2
93 *surfacecreatespinlinewithoffsetangle 1 1 0 45 0
94 *createmark lines 1 5
95 *surfacecreatespinlinewithoffsetangle 1 1 0 45 0
96 *createmark lines 1 10
97 *surfacecreatespinlinewithoffsetangle 1 1 0 45 0
98 *createmark lines 1 16
99 *surfacecreatespinlinewithoffsetangle 1 1 0 45 0
100
101 # Create Feed
102 *collectorcreate component Feed Steel2 30
103 *setvalue comps id=2 propertyid={props 2}
104 *setvalue comps id=2 materialid={mats 2}
105
106 *createnode [expr {- $guide_radius}] 0 [expr {$guide_length+
    $PCshift}] 0 0 0
107 *createnode [expr {$guide_radius}] 0 [expr {$guide_length+
    $PCshift}] 0 0 0
108 *createnode 0 [expr {$guide_radius}] [expr {$guide_length+
    $PCshift}] 0 0 0
109 *createlist nodes 1 2 3 4
110
111 *createcirclefrompoints 1 1
112 *surfacemode 4
113 *createmark lines 2 25
114 *createlist lines 1 25
115 *splinesurface lines 2 1 0 0 0
116 #*splinesurface entities mark plane_flag plane options
117 *createnode [expr {- $guide_radius}] 0 [expr {$PCshift}] 0 0 0
118 *createnode [expr {$guide_radius}] 0 [expr {$PCshift}] 0 0 0
119 *createnode 0 [expr {$guide_radius}] [expr {$PCshift}] 0 0 0
120 *createlist nodes 2 5 6 7
```

```
121
122 *linecreatestraight 0 0 [expr {$PCshift}] 0 0 [expr {
      $guide_length+$PCshift}]
123 *createlist lines 2 27
124
125 *surfacemode 4
126 *createplane 2 0 0 -1 0 0 0
127
128 *linedraglinetoformsurface 1 1 2 2 2 1
129
130 # Create Struts
131 *collectorcreate component Struts Steel3 20
132 *setvalue comps id=3 propertyid={props 3}
133 *setvalue comps id=3 materialid={mats 3}
134
135 set dx [expr {$sx-$guide_radius}]
136 set dz [expr {(-$F+($sx**2.0)/(4.0*$F))-$PCshift}]
137 set theta [expr {atan($dz/$dx)}]
138
139 set x [expr {sin($theta)*($Wl/2)}]
140 set z [expr {cos($theta)*($Wl/2)}]
141
142 # Strut 1
143 *createnode [expr {$sx}] [expr {$Wl/2}] [expr -$F+($sx**2.0)
      /(4.0*$F)] 0 0 0
144 *createnode [expr {$sx}] [expr {-Wl/2}] [expr -$F+($sx**2.0)
      /(4.0*$F)] 0 0 0
145 *createnode [expr {$sx-$x}] 0 [expr {[expr -$F+($sx**2.0)
      /(4.0*$F)]+$z}] 0 0 0
146 *createlist nodes 1 8 9 10
147
148 *createcirclefrompoints 1 1
149
150 *createplane 2 [expr {$guide_radius}] 0 [expr {$PCshift}] [
      expr {$sx}] 0 [expr -$F+($sx**2.0)/(4.0*$F)]
151 *linecreatestraight [expr {$sx}] 0 [expr -$F+($sx**2.0)/(4.0*
      $F)] [expr {$guide_radius}] 0 [expr {$PCshift}]
152
153 *createlist lines 1 31
154 *createlist lines 2 32
155
156 *linedraglinetoformsurface 1 1 2 2 2 1
157
158 # Strut 2
```

```

159 *createnode [expr {- $s_x$ }] [expr { $W_1/2$ }] [expr - $F+(\mathit{s_x}^{**2.0})$ 
      /(4.0* $F$ )] 0 0 0
160 *createnode [expr {- $s_x$ }] [expr {- $W_1/2$ }] [expr - $F+(\mathit{s_x}^{**2.0})$ 
      /(4.0* $F$ )] 0 0 0
161 *createnode [expr {- $s_x+\mathit{x}$ }] 0 [expr {[expr - $F+(\mathit{s_x}^{**2.0})$ 
      /(4.0* $F$ )]+ $z$ }]0 0 0
162 *createlist nodes 1 11 12 13
163
164 *createcirclefrompoints 1 1
165
166 *createplane 2 [expr {- $g_{\text{guide\_radius}}$ }] 0 [expr { $PC_{\text{shift}}$ }] [
      expr {- $s_x$ }] 0 [expr - $F+(\mathit{s_x}^{**2.0})/(4.0*F)$ ]
167 *linecreatestraight [expr {- $s_x$ }] 0 [expr - $F+(\mathit{s_x}^{**2.0})/(4.0*
      F)$ ] [expr {- $g_{\text{guide\_radius}}$ }] 0 [expr { $PC_{\text{shift}}$ }]
168
169 *createlist lines 1 36
170 *createlist lines 2 37
171
172 *linedraglinetoformsurface 1 1 2 2 2 1
173
174 # Strut 3
175 *createnode [expr { $W_1/2$ }] [expr {- $s_x$ }] [expr - $F+(\mathit{s_x}^{**2.0})$ 
      /(4.0* $F$ )] 0 0 0
176 *createnode [expr {- $W_1/2$ }] [expr {- $s_x$ }] [expr - $F+(\mathit{s_x}^{**2.0})$ 
      /(4.0* $F$ )] 0 0 0
177 *createnode 0 [expr {- $s_x+\mathit{x}$ }] [expr {[expr - $F+(\mathit{s_x}^{**2.0})$ 
      /(4.0* $F$ )]+ $z$ }] 0 0 0
178 *createlist nodes 1 14 15 16
179
180 *createcirclefrompoints 1 1
181
182 *createplane 2 0 [expr { $g_{\text{guide\_radius}}$ }] [expr { $PC_{\text{shift}}$ }] 0 [
      expr { $s_x$ }] [expr - $F+(\mathit{s_x}^{**2.0})/(4.0*F)$ ]
183 *linecreatestraight 0 [expr {- $s_x$ }] [expr - $F+(\mathit{s_x}^{**2.0})/(4.0*
      F)$ ] 0 [expr {- $g_{\text{guide\_radius}}$ }] [expr { $PC_{\text{shift}}$ }]
184
185 *createlist lines 1 41
186 *createlist lines 2 42
187
188 *linedraglinetoformsurface 1 1 2 2 2 1
189
190 # Strut 4
191 *createnode [expr { $W_1/2$ }] [expr { $s_x$ }] [expr - $F+(\mathit{s_x}^{**2.0})$ 
      /(4.0* $F$ )] 0 0 0

```

```
192 *createnode [expr {-Wl/2}] [expr {$sx}] [expr -$F+($sx**2.0)
    /(4.0*$F)] 0 0 0
193 *createnode 0 [expr {+$sx-$x}] [expr {[expr -$F+($sx**2.0)
    /(4.0*$F)]+$z}] 0 0 0
194 *createlist nodes 1 17 18 19
195
196 *createcirclefrompoints 1 1
197
198 *createplane 2 0 [expr {-guide_radius}] [expr {PCshift}] 0 [
    expr {-sx}] [expr -$F+($sx**2.0)/(4.0*$F)]
199 *linecreatestraight 0 [expr {$sx}] [expr -$F+($sx**2.0)/(4.0*
    $F)] 0 [expr {guide_radius}] [expr {PCshift}]
200
201 *createlist lines 1 46
202 *createlist lines 2 47
203
204 *linedraglinetoformsurface 1 1 2 2 2 1
205 # set mesh 0.5
206 *setoption node_tolerance=0.01
207 # Mesh Feed
208 *currentcollector component Feed
209
210 *createvector 1 0.0000 0.0000 1.0000
211
212 *linecreatestraight [expr {-guide_radius-0.5}] 0 [expr {
    guide_length+PCshift}] [expr {guide_radius+0.5}] 0 [expr
    {guide_length+PCshift}]
213 *createmark surfaces 1 5 6
214 *surfacemarksplitwithline 1 51 1 1 0
215
216 *linecreatestraight 0 [expr {-guide_radius-0.5}] [expr {
    guide_length+PCshift}] 0 [expr {guide_radius+0.5}] [expr
    {guide_length+PCshift}]
217 *createmark surfaces 1 5 6 11 12
218 *surfacemarksplitwithline 1 59 1 1 0
219
220 *createmark surfaces 1 5 6 11 12 13 14 15 16
221 *defaultmeshsurf 1 $portMesh 0 0 1 0 0 0 1 0.01 0 0 0
222
223 # Mesh Strut
224 *currentcollector component Struts
225 *createmark surfaces 1 7 8 9 10
226 *defaultmeshsurf 1 $portMesh 0 0 1 0 0 0 1 0.01 0 0 0
227
```

```
228 *createarray 21 123456 123456 123456 123456 123456 123456
      123456 123456 123456 123456 123456 123456 123456 123456
      123456 123456 123456 123456 123456 123456 123456
229 *createdoublearray 21 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0
      1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0
230 #clear all temp nodes
231 *nodecleartempmark
232
233 hm_createmark nodes 2 "by lines" 38
234 *rbe3 2 1 21 1 21 0 123456 1
235 hm_createmark nodes 2 "by lines" 39
236 *rbe3 2 1 21 1 21 0 123456 1
237
238 hm_createmark nodes 2 "by lines" 48
239 *rbe3 2 1 21 1 21 0 123456 1
240 hm_createmark nodes 2 "by lines" 49
241 *rbe3 2 1 21 1 21 0 123456 1
242
243 hm_createmark nodes 2 "by lines" 43
244 *rbe3 2 1 21 1 21 0 123456 1
245 hm_createmark nodes 2 "by lines" 44
246 *rbe3 2 1 21 1 21 0 123456 1
247
248 hm_createmark nodes 2 "by lines" 33
249 *rbe3 2 1 21 1 21 0 123456 1
250 hm_createmark nodes 2 "by lines" 34
251 *rbe3 2 1 21 1 21 0 123456 1
252
253 *currentcollector component Dish
254
255 # Mesh Dish
256 *createmark lines 1 23
257 *surfacecreatespinlinewithoffsetangle 1 1 0 45 0
258 *createmark lines 1 82
259 *surfacecreatespinlinewithoffsetangle 1 1 0 45 0
260 *createmark lines 1 91
261 *surfacecreatespinlinewithoffsetangle 1 1 0 45 0
262 *createmark lines 1 101
263 *surfacecreatespinlinewithoffsetangle 1 1 0 45 0
264 #*linecreatestraight $Dm $Dm 0 -$Dm -$Dm 0
265
266 *createnode 0 [expr {$sx}] [expr -$F+($sx**2.0)/(4.0*$F)] 0 0
      1
267 *createnode 0 [expr {- $sx}] [expr -$F+($sx**2.0)/(4.0*$F)] 0 0
```



```
1
268 *createnode [expr {$sx}] 0 [expr -$F+($sx**2.0)/(4.0*$F)] 0 0
1
269 *createnode [expr {-sx}] 0 [expr -$F+($sx**2.0)/(4.0*$F)] 0 0
1
270 #*createmark 1 surfaces "by component name" "Dish"
271 #*createmark 2 nodes "by component name" "Dish"
272 #*surfacemarkaddnodesfixed 1 2 0.0001 0
273 *hideall
274 *createmark surfaces 1 1-4 17-20
275 *createmark nodes 2 "displayed"
276 *surfacemarkaddnodesfixed 1 2 0.01 0
277 *createmark surfaces 1 "by component id" "1"
278 *defaultmeshsurf 1 $MRmesh 0 0 1 0 0 0 1 0.01 0 0 0
279
280 # Equivalence nodes
281 *createmark comps 1 all
282 *equivalence comps 1 0.001 1 0 0
283 *showall
284
285 # Create gravitational load
286 *createentity loadcols name=Gravity
287 *setvalue loadcols id=1 cardimage="GRAV"
288 *setvalue loadcols id=1 STATUS=2 2899=9.81
289 *setvalue loadcols id=1 STATUS=2 2900=$gx
290 *setvalue loadcols id=1 STATUS=2 2901=$gy
291 *setvalue loadcols id=1 STATUS=2 2902=$gz
292
293 # constraint
294 hm_createmark nodes 1 "by cylinder" "0 0 [expr -$F-0.01] 0 0 1
0.001 0.02 inside 1 1 0"
295 *loadcreate 1 3 1 0.0 0.0 0.0 0.0 0.0 0.0
296 *createmark loads 1 1
297
298 *createmark loadcols 1
299 *loadstepscreate "Loadstep" 1
300
301 *setvalue loadsteps id=1 STATUS=2 3240=1
302
303 *setvalue loadsteps id=1 ids={loadcols 1}
304
305 # Create folder and move over all scripts
306 file mkdir "Test/$name/"
307 set filename "Test/$name/ref.txt"
```

```
308 # open the filename for writing
309 set fileId [open $filename "w"]
310
311 hm_createmark nodes 1 "by cylinder" "[expr -$guide_radius] 0 [
    expr $guide_length+$PCshift] 0 0 1 0.01 0.02 inside 1 1 0"
312 hm_highlightmark nodes 1 "h"
313 set data "[hm_getmark nodes 1]"
314 puts -nonewline $fileId $data
315 puts -nonewline $fileId " "
316
317 hm_createmark nodes 1 "by cylinder" "[expr $guide_radius] 0 [
    expr $guide_length+$PCshift] 0 0 1 0.01 0.02 inside 1 1 0"
318 hm_highlightmark nodes 1 "h"
319 set data "[hm_getmark nodes 1]"
320 puts -nonewline $fileId $data
321 puts -nonewline $fileId " "
322
323 hm_createmark nodes 1 "by cylinder" "0 [expr -$guide_radius] [
    expr $guide_length+$PCshift] 0 0 1 0.01 0.02 inside 1 1 0"
324 hm_highlightmark nodes 1 "h"
325 set data "[hm_getmark nodes 1]"
326 puts -nonewline $fileId $data
327 puts -nonewline $fileId " "
328
329 hm_createmark nodes 1 "by cylinder" "0 [expr $guide_radius] [
    expr $guide_length+$PCshift] 0 0 1 0.01 0.02 inside 1 1 0"
330 hm_highlightmark nodes 1 "h"
331 set data "[hm_getmark nodes 1]"
332 puts -nonewline $fileId $data
333 puts -nonewline $fileId " "
334
335 hm_createmark nodes 1 "by cylinder" "0 0 [expr $guide_length+
    $PCshift] 0 0 1 0.01 0.02 inside 1 1 0"
336 hm_highlightmark nodes 1 "h"
337 set data "[hm_getmark nodes 1]"
338 puts -nonewline $fileId $data
339 puts -nonewline $fileId " "
340
341 close $fileId
342
343 *view "iso1"
344
345 *collectorcreate component Back Steel3 50
346 # Create Beam section
```

```

347 *beamsectioncreatestandardsolver 11 1 "Rod" 0
348 *createdoublearray 3 $back 10 10
349 *beamsectionsetdatastandard 1 3 1 11 0 "Rod"
350 # Create Bar prop
351 *createentity props cardimage=PBEAM name=BAR
352 *setvalue props id=4 materialid={mats 3}
353 *setvalue props id=4 STATUS=2 3186={beamsects 1}
354 #*setvalue props id=4 STATUS=2 3179={beamsects 1}
355
356 *setvalue comps id=4 propertyid={props 4}
357 *setvalue comps id=4 materialid={mats 3}
358
359 hm_createmark nodes 1 "by cylinder" "[expr {$Dm/2}] 0 $H1 0 0
    1 0.01 0.02 inside 1 1 0"
360 hm_createmark nodes 2 "by cylinder" "0 0 [expr -$F] 0 0 1 0.01
    0.02 inside 1 1 0"
361 *nodelistbypath "[hm_getmark nodes 1]" "[hm_getmark nodes 2]"
    1
362 *linemesh_preparenodeslist1 1
363 *linemesh_savedata_bar1 nodes 1 60 0 1 0 0 0 0 0 0 1 0
364
365 hm_createmark nodes 1 "by cylinder" "[expr {-Dm/2}] 0 $H1 0 0
    1 0.01 0.02 inside 1 1 0"
366 *nodelistbypath "[hm_getmark nodes 1]" "[hm_getmark nodes 2]"
    1
367 *linemesh_preparenodeslist1 1
368 *linemesh_savedata_bar1 nodes 1 60 0 1 0 0 0 0 0 0 1 0
369
370 hm_createmark nodes 1 "by cylinder" "0 [expr {-Dm/2}] $H1 0 0
    1 0.01 0.02 inside 1 1 0"
371 *nodelistbypath "[hm_getmark nodes 1]" "[hm_getmark nodes 2]"
    1
372 *linemesh_preparenodeslist1 1
373 *linemesh_savedata_bar1 nodes 1 60 0 1 0 0 0 0 0 0 1 0
374
375 hm_createmark nodes 1 "by cylinder" "0 [expr {Dm/2}] $H1 0 0
    1 0.01 0.02 inside 1 1 0"
376 *nodelistbypath "[hm_getmark nodes 1]" "[hm_getmark nodes 2]"
    1
377 *linemesh_preparenodeslist1 1
378 *linemesh_savedata_bar1 nodes 1 60 0 1 0 0 0 0 0 0 1 0
379
380 hm_createmark nodes 1 "by cylinder" "[expr {($Dm/2)*cos($pi/4)}]
    [expr {($Dm/2)*cos($pi/4)}] $H1 0 0 1 0.01 0.02 inside 1

```

```

1 0"
381 *nodelistbypath "[hm_getmark nodes 1]" "[hm_getmark nodes 2]"
1
382 #hm_highlightlist nodes 1 "h"
383 *linemesh_preparenodeslist1 1
384 *linemesh_savedata_bar1 nodes 1 60 0 1 0 0 0 0 0 0 1 0
385
386 hm_createmark nodes 1 "by cylinder" "[expr {-($Dm/2)*cos($pi
/4)}] [expr {($Dm/2)*cos($pi/4)}] $H1 0 0 1 0.01 0.02
inside 1 1 0"
387 *nodelistbypath "[hm_getmark nodes 1]" "[hm_getmark nodes 2]"
1
388 *linemesh_preparenodeslist1 1
389 *linemesh_savedata_bar1 nodes 1 60 0 1 0 0 0 0 0 0 1 0
390
391 hm_createmark nodes 1 "by cylinder" "[expr {($Dm/2)*cos($pi/4)
}] [expr {-($Dm/2)*cos($pi/4)}] $H1 0 0 1 0.01 0.02 inside
1 1 0"
392 *nodelistbypath "[hm_getmark nodes 1]" "[hm_getmark nodes 2]"
1
393 *linemesh_preparenodeslist1 1
394 *linemesh_savedata_bar1 nodes 1 60 0 1 0 0 0 0 0 0 1 0
395
396 hm_createmark nodes 1 "by cylinder" "[expr {-($Dm/2)*cos($pi
/4)}] [expr {-($Dm/2)*cos($pi/4)}] $H1 0 0 1 0.01 0.02
inside 1 1 0"
397 *nodelistbypath "[hm_getmark nodes 1]" "[hm_getmark nodes 2]"
1
398 *linemesh_preparenodeslist1 1
399 *linemesh_savedata_bar1 nodes 1 60 0 1 0 0 0 0 0 0 1 0
400
401 *isolateonlyentity comps "by id" 4
402 hm_createmark elems 1 "by cylinder" "0 0 [expr $H1-$t0] 0 0 1
[expr $Dm/2] $t0 inside 1 0 0"
403 *deletemark elems 1
404
405 *createmark elements 1 "by component name" Back
406 *createvector 1 0 0 1
407 *barelementupdatewithoffsets 1 0 0 1 0 0 0 0 "" 0 0 0 1 2 0 0
$offset 1 2 0 0 $offset
408 #output files and start next sims
409 file copy -force "Batch_Files/Deformed" "Test/$name"
410 if {$Ideal == 1} {
411 file copy -force "Batch_Files/Ideal" "Test/$name"

```

```

412 }
413 file copy -force "Batch_Files/arguments.txt" "Test/$name"
414 file copy -force "Batch_Files/Auto2.py" "Test/$name"
415 file copy -force "Batch_Files/automate.sh" "Test/$name"
416 file copy -force "Batch_Files/Batch.sh" "Test/$name"
417 file copy -force "Batch_Files/export_deformed.tcl" "Test/$name"
    "
418 file copy -force "Batch_Files/node_finder.py" "Test/$name"
419 file copy -force "Batch_Files/RMStest.py" "Test/$name"
420 #cp "Batch Files/*" "Test/FoD$name"
421 hm_answernext yes;
422 *feoutputwithdata "/apps/altair/hyperworks/14.0/altair/
    templates/feoutput/optistruct/optistruct" "Test/$name/
    FEKOmesh.fem" 0 0 1 1 0
423 hm_answernext yes;
424 *summary "/apps/altair/hyperworks/14.0/altair/templates/
    summary/nastran/components" "VisualDoc/summary.txt" 0 1
425 cd "Test/$name"
426 file attributes "Batch.sh" -permissions +x
427 exec "Batch.sh"
428 }
429
430 proc Simulation {angle back t0 Ideal} {
431     set pi 3.14159265358979323846
432     for {set i 0} {$i < 1} {incr i 1} {
433         set x [expr {$i+0.0}]
434         set gx [expr sin(($angle)*$pi/180)]
435         set gy 0
436         set gz [expr -cos(($angle)*$pi/180)]
437         Gen [expr 0.3+$x/100.0] "opt_$t0-Grav$angle-FoD[expr 0.3+$x
            /100.0]" $gx $gy $gz $back $t0 $Ideal
438     }
439 }
440
441 # define gravitational load angle
442 set angle 90
443
444 Simulation $angle $back $t0 $Ideal
445
446 *detailedelements_beamvis 1

```

Appendix C

Response surface definitions

The approximate functions that make up the response surface models are presented here. x_1 is $t0$, x_2 is the backing structure thickness, x_3 is the offset and x_4 is the strut location. For the first response surface there are four input variables, thus the quadratic function is,

$$\begin{aligned}
 f(x_1, x_2, x_3, x_4) = & a_1 + a_2x_1 + a_3x_2 + a_4x_3 + a_5x_4 + a_6x_1x_2 \\
 & + a_7x_1x_3 + a_8x_1x_4 + a_9x_2x_3 + a_{10}x_2x_4 \\
 & + a_{11}x_3x_4 + a_{12}x_1^2 + a_{13}x_2^2 + a_{14}x_3^2 + a_{15}x_4^2
 \end{aligned} \tag{C.1}$$

Table C.1: First response surface model coefficients

$f(\mathbf{x})$	Gain	Mass	Side-lobe level	Pointing error
a_1	24.98	2403	-4.779	3.754
a_2	4.344	1650	7.353	-1.244
a_3	108.9	57016	192.3	-40.16
a_4	-31.79	-1299	-39.32	11.24
a_5	1.060	71.42	0.7750	-0.3588
a_6	-5.220	-52852	-43.31	-0.4925
a_7	2.150	-6.571x10 ⁻¹²	9.027	-1.034
a_8	-0.03826	-0.3728	-0.1190	0.04265
a_9	322.4	-1.836x10 ⁻¹²	79.57	-73.30
a_{10}	0.3475	106.8	-0.4599	1.114
a_{11}	0.1148	-3.252x10 ⁻¹²	2.760	-0.3341
a_{12}	-1.741	-415.5	-2.275	0.5053
a_{13}	19.70	8.514x10 ⁵	-309.3	104.4
a_{14}	17.39	-4330	10.57	6.312
a_{15}	-0.09855	-9.946	0.2447	0.02438

For the second and third response surface model, the offset had been removed as an input so the quadratic function becomes,

$$f(x_1, x_2, x_4) = a_1 + a_2x_1 + a_3x_2 + a_4x_4 + a_5x_1x_2 + a_6x_1x_4 + a_7x_2x_4 + a_8x_1^2 + a_9x_2^2 + a_{10}x_4^2 \quad (\text{C.2})$$

Table C.2: First response surface model coefficients

$f(\mathbf{x})$	Gain	Mass	Side-lobe level	Pointing error
a_1	35.37	2979	6.753	0.05162
a_2	-001431	412.5	1.898	-0.005937
a_3	11.16	21454	111.2	-4.331
a_4	1.569	12.04	1.680	0.04881
a_5	0.4922	-27385	-44.09	-0.8049
a_6	0.01936	-1.132	0.8277	0.002838
a_7	-0.6624	98.51	10.21	-0.7162
a_8	0.01608	-69.03	-1.635	0.02319
a_9	-87.67	1.058x10 ⁶	-645.3	67.52
a_{10}	-0.1469	0.5010	0.01525	6.709x10 ⁻⁴

Table C.3: Third response surface model coefficients

$f(\mathbf{x})$	Gain	Mass	Side-lobe level	Pointing error
a_1	33.86	2969	9.017	0.09848
a_2	-0.2657	309.7	7.437	0.001207
a_3	24.85	20388	169.89	-8.014
a_4	2.197	19.04	-2.342	0.03133
a_5	0.9656	-19982	-100.8	-1.622
a_6	0.02627	-0.4854	1.653	0
a_7	-2.187	23.80	17.61	-1.293
a_8	0.04982	-58.41	-4.871	0.03312
a_9	-233.9	9.876x10 ⁵	-490.6	192.0
a_{10}	-0.2120	-0.2313	0.4284	0.005267

Appendix D

Deformation plots of the Ghana 32 m antenna's main reflector

The deformation plots of the main reflector of the Ghana 32 m antenna under gravitational load cases for the different elevation angles of the dish are shown in Figures D.1 to D.5.

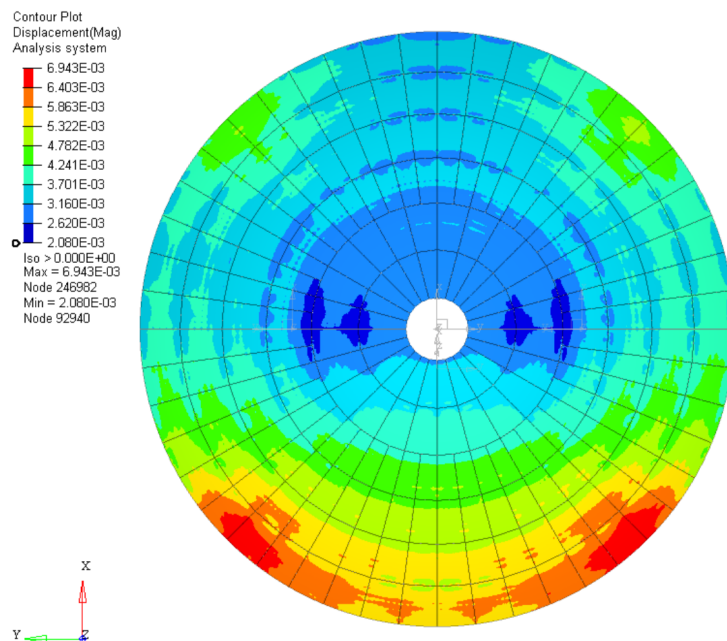


Figure D.1: Deformation plot of the main reflector at 0°

APPENDIX D. DEFORMATION PLOTS OF THE GHANA 32 M ANTENNA'S MAIN REFLECTOR D.2

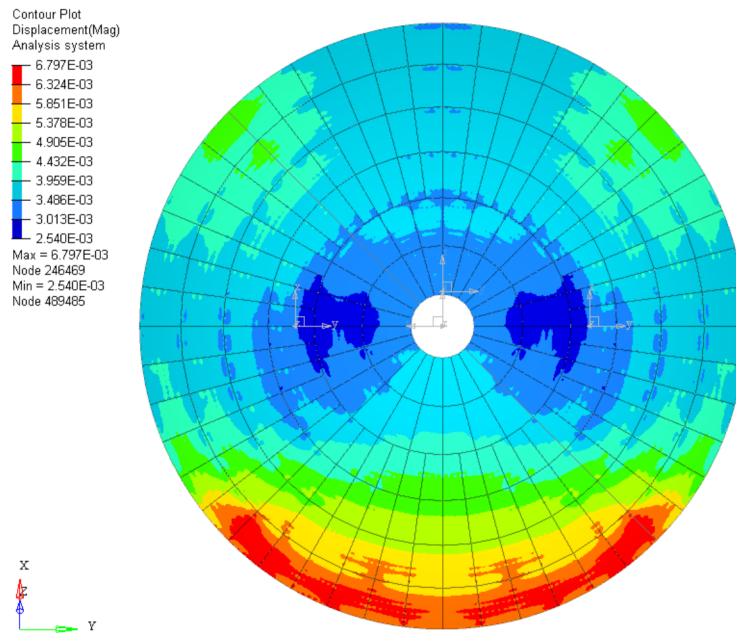


Figure D.2: Deformation plot of the main reflector at 30°

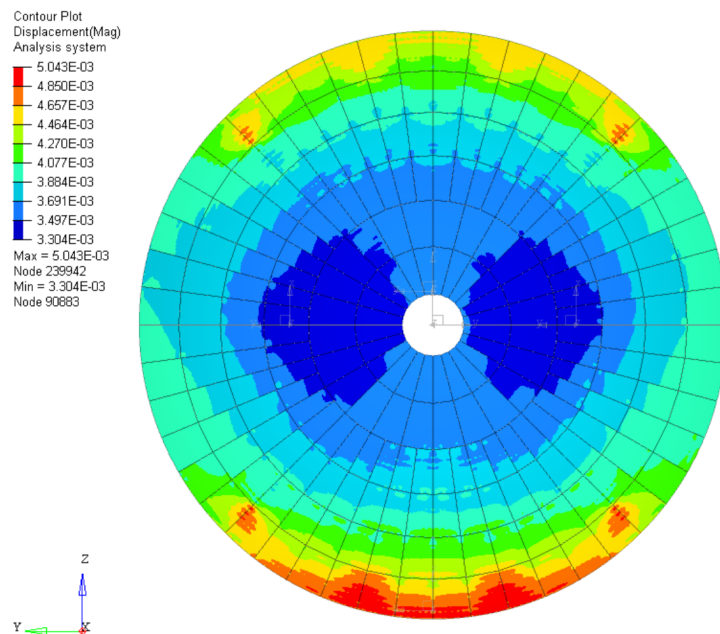


Figure D.3: Deformation plot of the main reflector at 90°

APPENDIX D. DEFORMATION PLOTS OF THE GHANA 32 M ANTENNA'S
MAIN REFLECTOR D.3

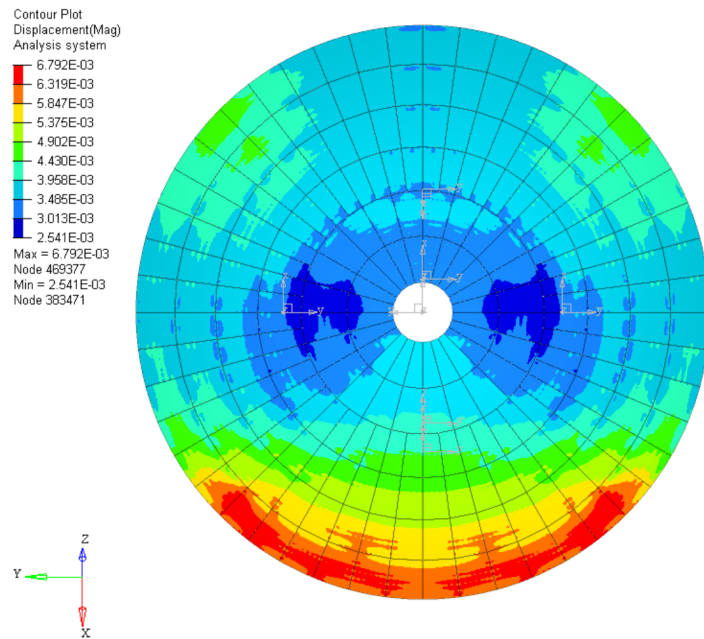


Figure D.4: Deformation plot of the main reflector at -30°

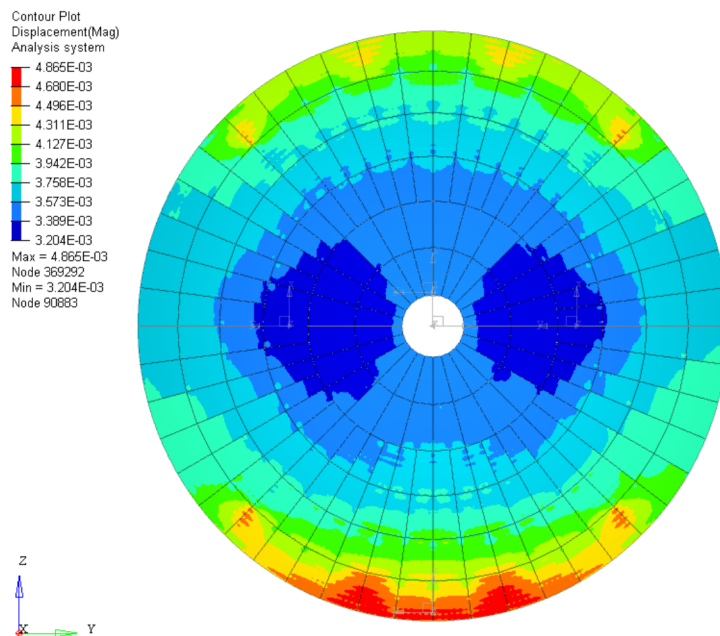


Figure D.5: Deformation plot of the main reflector at -90°