

# Structural-Electromagnetic Simulation Coupling and Conformal Antenna Design Tool

By  
© 2018

Pedro Martin Mendoza Strilchuk  
B.S., University of Kansas, 2015

Submitted to the graduate degree program in Aerospace Engineering and the Graduate Faculty  
of the University of Kansas in partial fulfillment of the requirements  
for the degree of Master of Science.

---

Chair: Dr. Emily J. Arnold

---

Dr. Richard D. Hale

---

Dr. Ronald M. Barrett-Gonzales

Date Defended: March 15, 2018

The thesis committee for Pedro Martin Mendoza Strilchuk certifies  
that this is the approved version of the following thesis:

## Structural-Electromagnetic Simulation Coupling and Conformal Antenna Design Tool

---

Chair: Dr. Emily J. Arnold

Date Approved: April 26, 2018

## Abstract

Airborne and spaceborne radar has long been an effective tool for remote sensing, surveillance, and reconnaissance. Most airborne systems utilize antenna arrays that are installed inside the moldline of the aircraft or in radomes that protect the array from in-flight loads. While externally-mounted arrays can offer the advantage of larger apertures, sensor-vehicle interactions often result in performance degradation of both systems. The presence of an externally-mounted array will increase the vehicle's drag and potentially affect the dynamics and control of the vehicle. In addition, in-flight structural loads will deform the array, thus resulting in relative phase errors. While there exist a multitude of physics-based simulation tools to determine the effects of the array on the aircraft performance, existing tools are not sufficient for generating deformed arrays necessary for determining in-flight array performance.

In response to this need, a computer tool for analyzing antennas undergoing structural loads is developed. The Antenna Deformation Tool (ADT) has two primary uses: generating deformed geometry from the output of a structural Finite Element Model (FEM) for use in an Electromagnetic (EM) simulation, and designing conformal antenna arrays. The two commercial software packages ADT is optimized for are MSC NASTRAN and ANSYS HFSS. Specifically, ADT is designed to generate a deformed 3D Computer Aided Design (CAD) model from a NASTRAN structural mesh. The resulting CAD model is compatible with HFSS electromagnetic simulation software for the assessment of the effects of loads on performance. The main purpose for the development of ADT is to facilitate studies of how structural deformations affect airborne antenna arrays performance and to provide the capability to perform studies easily and quickly

using different antennas on the same structural model. ADT capabilities are demonstrated using several representative airborne antenna array structures.

ADT is also demonstrated in the design of conformal antenna arrays. ADT can import CAD geometry and deform it according to a prescribed deformation field. The deformation field can either be determined from structural simulations or be provided by the user. This functionality allows the user to take an existing planar antenna design and conform it to a desired shape. Within the scope of airborne antenna arrays, this would allow an engineer to conform the antenna to the moldline of the aircraft or other support structure.

Currently, ADT can interpret only quad and triangular 2D elements from NASTRAN. In addition, its ability to interpret a surface from a point cloud is limited to surface meshes in which there are exactly four explicit vertices, or surfaces which have a fairly even boundary with no major discontinuities and can be divided into four even segments. ADT is tested on NASTRAN structural models of small to medium complexity, and the geometry generated from simple models is used in HFSS simulations with success (with occasional post processing required). The antenna deformation submodule shows favorable performance with sheet and solid CAD geometry, though post-processing is required in the case of the latter. Results of some deformed antennas simulated with HFSS in the 200 MHz range are presented.

The surface error of the geometry produced by ADT varies with the type of input mesh, with curved meshes and surfaces having higher errors. In terms of average element edge length, the maximum surface error is up to 1% for surfaces with no to small curvatures, and up to 3.6% for highly curved surfaces. This translates to about 0.17% of the mesh diagonal. ADT contains a set of classes and functions which provide ample capabilities for surface generation from meshes, and the process implemented is mostly automatic, requiring minimal user intervention.

Due to ADT defining deformed geometry purely on separate meshes, adjacent surfaces are not associative and continuity between them is not guaranteed, which inherently can result in small intersections. These intersections can cause meshing problems with HFSS; however, these issues can be mitigated by adding a small offset.

While demonstrated applications are still limited, ADT promises to substantially contribute to the design of aircraft-integrated antennas and multifunctional structures. With very limited capabilities for generating and assessing deformed antenna geometry currently existing, ADT represents a unique tool. ADT could be used not only in developing the next-generation of airborne remote sensing technologies, but to characterize in-flight performance of existing systems as well.

## Acknowledgments

Completing my degree, let alone this thesis, would not have been possible without the help of many people, who I would like to thank.

I would like to thank my family for supporting me. My mother and father, for encouraging me to pursue my goals, despite the emotional and material burdens and physical distancing this has caused. My brother and sister, for reminding me of the simple joys of life. Elizabeth, for being my “thesis buddy”, and her family for taking me under their wing in this strange new land. My aunt and grandfather for teaching me life lessons with their wisdom and supporting me throughout my entire life. And to my many other aunts, uncles, cousins and friends, who, to my fortune, are too many to list, and all of whom have helped me surmount obstacles through my life.

Additionally, I would like to thank the University of Kansas and the Institute of International Education for giving me the opportunity to study the field I am passionate about, and providing me with material support. In particular, I would like to thank the department of Aerospace Engineering and my committee members for their commitment.

Finally, I would like to thank my advisor, Dr. Emily Arnold, for being exceptionally supportive, patient, kind, and thoughtful before and throughout the completion of my MS degree. Getting to where I am now would have been all but impossible without her guidance and support.

## Dedication

This thesis is dedicated to the memory of my beloved grandmother Anna, who passed away during the final stages of working on this thesis. Her unbounded kindness, eternal cheer, relentless work ethic, and utter unwillingness to yield in the face of adversity have been and continue to be my main inspiration.

She has taught me many things – among them the importance of remembering one's roots, no matter where in this wide world or in what condition one ends up. Even though life has taken me far from home, thanks to her, I always carry a piece of it in my heart.

*Bucovină, plai cu flori,  
Unde sunt ai tăi feciori?  
Au fost duși în altă țară,  
Dar se-ntorc la primavară.  
Și-napoi când or veni  
Tot pe tine te-or iubi*

Romanian Folk Song

## Table of Contents

Chapter 1: Introduction .....	1
1.1 Motivation .....	2
1.2 Problem Description.....	7
1.3 Thesis Organization.....	9
Chapter 2: Previous Work.....	11
2.1 Estimating Effects of Deformation on Antenna Arrays.....	11
2.2 Multiphysics Software .....	15
2.3 Geometric Conversion Software.....	16
Chapter 3: Program Description .....	19
3.1 The Finite Element Method .....	19
3.1.1 Structural FEA NASTRAN .....	20
3.1.2 Electromagnetic FEA HFSS .....	21
3.2 Software Structure and Requirements .....	23
3.2.1 Core module.....	25
3.2.2 Transitional text file.....	28
3.2.3 CAD export module.....	28
Chapter 4: Geometry Definitions and Methods .....	29
4.1 Geometric Frameworks.....	29
4.1.1 Polygonal meshes.....	29
4.1.2 Bezier curves, B-Splines and NURBS.....	32
4.2 B-spline and NURBS Surfaces .....	34



4.2.1 Bezier curves.....	34
4.2.2 B-spline curves.....	37
4.2.3 NURBS curves.....	38
4.2.4 B-Spline and NURBS surfaces .....	39
Chapter 5: Geometry Reconstruction from Structural Mesh .....	42
5.1 Mesh Importing, Surface Sorting, and Edge and Corner Recognition .....	44
5.2 Mesh Parameterization .....	52
5.2.1 Curve parameterization .....	53
5.2.2 Curve knot vector estimation .....	55
5.2.3 Gridded surface parametrization.....	56
5.2.4 K-harmonic mapping .....	60
5.3 Curve and Surface Interpolation and Approximation.....	63
5.3.1 Global curve interpolation .....	63
5.3.2 Global curve approximation .....	64
5.3.3 Global surface interpolation, approximation and thickening into solids .....	65
Chapter 6: Antenna Deformation Module .....	68
6.1 User Defined Sheet Antenna.....	71
6.2 Deformation of Antenna From CAD File.....	73
6.3 Antenna Deformation Process .....	78
6.2.1 Edge curve deformation .....	79
6.2.2 Plane deformation .....	80
6.2.3 B-spline surface deformation .....	83
6.2.4 Connectivity restoration and solid reconstruction .....	83

Chapter 7: Results .....	86
7.1 Version 1.0 Results and Initial Validation .....	86
7.2 Summary of Version 2.0 Improvements .....	89
7.3 Simulation Setup .....	90
7.3.1 Antenna and Geometry Descriptions .....	90
7.3.2 Structural Simulation Details .....	91
7.3.3 Execution Time .....	95
7.3.4 Geometry, Meshes, and Surface Errors .....	96
7.3.5 EM Simulation Details .....	104
7.4 Test Case 1: Single Element Deformation .....	106
7.5 Effects of Induced Offsets .....	108
7.6 Test Case 2: P3 Array, Fairing Only .....	115
7.7 Test Case 3: MCoRDS Array, Fairing Only .....	119
7.7 Test Case 4: MCoRDS Array, With Wing and Pylons .....	124
7.8 Test Case 5: Large Array With User-Specified Deformation Field .....	129
7.9 Test Case 6: Conformal Deformation of UWB Antenna Element .....	135
7.10 Summary of Results .....	140
Chapter 8: Summary, Conclusions and Future Work .....	142
8.1 Summary and Discussion of Software Requirements .....	142
8.2 Conclusions and Recommendations .....	146
References .....	152
Appendix A: Other Parameterization Approaches .....	A-1
A.1 Least Squares Conformal Mapping .....	A-1

A.2 Gridded Resampling ..... A-4

Appendix B: STEP Files and Supported Entities ..... B-1

Appendix C: Transitional Text File Format..... C-1

Appendix D: Material Properties ..... D-1

Appendix E: Results and Pylon Displacement Simulations .....E-1

Appendix F: B-spline Functions Used for Bowl Deformation .....F-1

## List of Figures

Figure 1: Complete CReSIS sensor suite installed on the NASA P-3. The MCoRDS antenna array can be seen integrated on the wings and under the belly of the fuselage. ....	4
Figure 2: Overview of design process for Airborne Antenna Arrays. ....	6
Figure 3. Original and modified antenna array (Senbokuya & Tsunoda, 2003).....	13
Figure 4. ADT Structure, with optional inputs shown as parallelograms bounded by dashed lines .....	24
Figure 5. Base file class hierarchy .....	26
Figure 6. Polygonal mesh of a face (Peyre, 2009).....	30
Figure 7. Smoothing of mannequin head polygonal model. (a) and (b) show the model using flat and Phong shading, respectively, showing edges, while (c) and (d) show the same without the edges. All use the same mesh. (Peyre, 2009).....	31
Figure 8. Simple NURBS surface with control points shown in red. (Claxton, 2007).....	33
Figure 9. Example of Jaguar F-Type model created with NURBS. Individual NURBS patches are bounded by the dark curves. (Singhvi, 2015) .....	33
Figure 10. Examples of Bezier curves. Control points are shown as blue circles, with the control polyline in red, and the curve itself as black. (Narasimhan, 2014).....	35
Figure 11. Effects of the motion of a single control points on Bezier curve. The control points are shown as blue circles, the control polyline as red lines. Control point a is moved to position b, modifying the shape of curve 1 into curve 2. Moving point b to point c further modifies the curve shape to make curve 3. (Narasimhan, 2014).....	36
Figure 12. The variation of influence of each control point at each parameter value, using a uniformly spaced knot vector. Each control point is symbolized by a different curve in the plot.	

Note that the first and last control points have a value of 1 at the start and end of the curve. This is the effect of clamping the curve, where the start and end values are equal to the curve values.

..... 38

Figure 13. B-spline curve and control points in red. The motion of a single point only affects curve locally..... 38

Figure 14. B-spline surface with control point net and parametric u and v directions. Red points indicate the edges of the surface. .... 40

Figure 15. General process outline ..... 43

Figure 16. Angular adjacency matrix determination. Elements which share the normal vectors of the same color will be grouped together, since they are within tolerance. Sometimes adjacent elements have reversed normal vectors, as between f2 and f3. This is accounted for by counting the tolerance both ways..... 46

Figure 17. Surface recognition algorithm using different angular tolerances: (a) 0 degrees (original mesh), (b) 178 degrees, (c) 175 degrees, (d) 165 degrees. Adjacent elements of the same color are sorted to the same surface..... 47

Figure 18. Surface sorting algorithm for a channel surface, using different angular tolerances: (a) 180 degrees (perfectly strict), (b) 170 degrees, (c) 160 degrees, (d) 135 degrees. Adjacent elements of the same color are sorted to the same surface. .... 48

Figure 19. Mesh example with edge elements identified and shaded as green patches ..... 50

Figure 20. Corner recognition. Corners shown as red circles, and other edge nodes as blue circles. On the left, four corners are detected using a tolerance of  $\pm 30$  degrees. On the right, six corners detected using a very loose tolerance of  $\pm 85$  degrees ..... 51

Figure 21. Corner (red circles) detection on a circular boundary (blue circles). Because the angle between adjacent element edges is small on the boundary, no corners are detected unless the tolerance is very loose, when almost every corner is detected (right). Thus, corners are taken as four evenly spaced nodes (left) automatically by ADT. ....	52
Figure 22. Mesh parameterization, a linear piecewise map from a domain X to a domain U. (Levy, Petitjean, Ray, & Maillot, 2002) .....	53
Figure 23. Chord length parameterization .....	54
Figure 24. Curves obtained though interpolation of different parametrizations of the same data. Data points are shown as blue circles, control points and the control mesh is shown as red points with dotted lines.....	55
Figure 25. Gridded vs. non-gridded mesh parameterization. Blue point parameter values are easily obtained by applying curve parameterization to u and v iso-curves separately in gridded space, but not in non-gridded space .....	57
Figure 26. Uniform and non-uniform mesh generated in PATRAN for same surface.....	57
Figure 27. Two different parameterizations for the same surface, showing iso-curves. ....	58
Figure 28. Surface reconstructions using different parametrizations of the same mesh, showing isolines. Left to right: approximation result using a 10 by 10 control net, interpolation using same bad parametrization, approximation using good parametrization, interpolation using good parameterization.....	60
Figure 29. Barycentric coordinates of point P, t1, t2 and t3 (Weisstein, 2018).....	61
Figure 30. Mesh parameterization and gridding by k-Harmonic mapping. Clockwise from upper left: original mesh in Euclidean space, mesh parameterization in parametric space, uniform grid in parametric space, mapping of uniform grid onto mesh in Euclidean space .....	62

Figure 31. Mesh gridding of a circular object by k-Harmonic mapping .....	63
Figure 32. Gridded mesh fitting. Left to right: original mesh, interpolating the $u$ -directional data, interpolating the control points of the $u$ -direction interpolants in the $v$ direction, Control points $P$ are obtained (Piegl & Tiller, 1997).....	66
Figure 33. Original and thickened surfaces. Thickened surface obtained by moving the control points in the directions indicated by blue arrows, which are local normals. ....	67
Figure 34. Process of transformation of a single surface from (a) a mesh to (b) a deformed mesh then (c) B-spline surface with offset, and (d) solid created from a series of connected surfaces and imported into HFSS.....	67
Figure 35. ADT Antenna deformation model process overview. The dashed line indicates optional input, as the deformation field can also be user defined.....	69
Figure 36. Visualization of the antenna deformation process using a user-specified deformation field for a dipole array.....	70
Figure 37. Original antenna geometry. User must normalize the green points with respect to the defining surface. Normalized green point coordinates are used as input into ADT to build sheet geometry .....	72
Figure 38. Original (above) and normalized or parametric (below) coordinates of antenna conductor surfaces. Blue and red represent two different conductor sheets defined on different surfaces. Red outline is defined by the green defining points from the previous figure. ....	72
Figure 39. Antenna conductor boundaries mapped onto deformed substrate sheets (green), which are on the bottom fairing surface (blue).....	73
Figure 40. B-spline solid used for mapping antenna geometry .....	76
Figure 41. Antenna and bounding box.....	77

Figure 42. Antenna as it is being deformed. Z-scale exaggerated .....	78
Figure 43. Antenna deformation process .....	78
Figure 44. Outline of plane deformation process.....	81
Figure 45. UWB antenna conductor .....	82
Figure 46. Antenna deformation process for component on a plane. (a) Nominal antenna component defined on a plane, with edges and critical points marked. (b) Projection of critical points onto plane coordinates and determination of minimum bounding quadrilateral. (c) Generation of grid on plane (blue points), and mapping of grid into deformed space (green points). (d) Mapping of edges into deformed space and fitting of B-spline surface to deformed point grid.....	82
Figure 47. B-spline deformation process .....	83
Figure 48. Faces of a manifold .....	84
Figure 49. Vertex manifold connectivity matrix.....	84
Figure 50. FEA model of single antenna element and panel with applied loads and boundary conditions.....	87
Figure 51. Simulations and measurements: (a) nominal and deformed geometry in HFSS, (b), measurement setup in anechoic chamber.....	88
Figure 52. Geometry used for structural model creation. All dimensions are in inches.....	92
Figure 53. Regions of the structural model.....	93
Figure 54. FEM model of P3 antenna fairing. Distributed tip load used to generate desire deflections illustrated by the pink arrows. ....	93
Figure 55. Datasets for wing surface of model. (a) Original mesh dataset (generated from Patran FEA mesh). (b) Resampled mesh dataset. (c) Evaluated parameter dataset. (d) Original mesh	



overlaid with resampled mesh. (e) Original mesh overlaid with evaluated mesh at parameter values. Axis represent coordinates.....	98
Figure 56. Four surfaces (red) constructed from larger surface (blue) using parameter values. Above: surfaces constructed using good parameterization. Note that edges are straight. Below: Surface constructed using bad parameterization. Note that surface edges are not straight .....	98
Figure 57. Pylon leading edge curvature, original mesh in black, resampled mesh in blue.....	100
Figure 58. Intersection between pylon trailing edge and fairing trailing edge highlighted in orange.....	102
Figure 59. Deviation of antenna conductor from antenna substrate, blue indicating deviation into the surface, and red indicating away from it. The maximum deviation is 1e-6 inches into the surface.....	103
Figure 60. Deviation of a surface made of a subset of points from underlying surface, blue indicating deviation into the surface, and red indicating away from it. The maximum deviation is 5e-6 inches into the surface.....	103
Figure 61. Fairing model without pylons and wing showing bounding vacuum box, used for Test cases 2 and 3. The wing and pylons were excluded for this test case.....	105
Figure 62. HFSS model with all surrounding geometry including wing, pylons, and fairing....	106
Figure 63. Deformations of a single antenna element, ranging from smallest (green) to largest (gray).....	107
Figure 64. Effect of strain on resonant frequency and bandwidth.....	108
Figure 65. Fairing model showing the locations of the addition of gaps.....	110
Figure 66. Change in resonance parameters without gap (red) and with gap (blue) .....	112
Figure 67. Change in radiation pattern due to presence of gaps.....	114

Figure 68. Return loss parameter S11 for different load cases .....	117
Figure 69. Radiation pattern for P3 array under different load cases .....	119
Figure 70. Change in S11 due to deformation .....	122
Figure 71. Change in radiation pattern due to deformation .....	123
Figure 72. Return loss for each element of array under different load cases.....	125
Figure 73. Radiation patterns for the nominal and deformed cases.....	128
Figure 74 Eight by eight strip dipole antenna array.....	130
Figure 75. B-spline deformation field for 10% wavelength (Claxton, 2007). Control points shown in red.....	130
Figure 76. Array radiation pattern, $\varphi = 0$ degrees, from simulation .....	131
Figure 77. Array radiation pattern, $\varphi = 90$ degrees, from simulation .....	131
Figure 78. Comparison of normalized power between analytical results (blue), simulations with non-deformed elements, (red), and simulations with deformed elements (yellow), when $\varphi=90$ degrees. The order, top to bottom, is the following: nominal array, 10% wavelength deformation, 12.5% deformation, 25% deformation, 50% deformation .....	133
Figure 79. Nominal (above) and conformal (below) UWB antenna configurations .....	136
Figure 80. Port and vias on UWB antenna .....	137
Figure 81. Resistances modelled in UWB antenna.....	137
Figure 82. Return loss of UWB antenna.....	138
Figure 83. UWB antenna radiation patterns .....	139
Figure A.1. Conformal map, where tangents of u and v isocurves are orthogonal and have the same length (Levy, Petitjean, Ray, & Maillot, 2002). .....	A-2
Figure A.2. Vertices i and j, with corresponding angles.....	A-3

Figure A.3. Gridded resampling flowchart ..... A-5

Figure A.4. Spring mesh initialization. (a) initial seeds shown as red circles, original mesh is green, ratio of seeds is determined by length of longest side to shortest side, (b) shortest path between opposing red nodes is computed using Dijkstra (red paths), and intersection nodes are saved (blue circles), (c) initial spring mesh is built by connecting red seed nodes and blue intersecting nodes..... A-6

Figure A.5. Simple spring mesh showing forces applied on different typed of nodes. .... A-7

Figure A.6. Spring mesh node motion flowchart..... A-10

Figure A.7. Spring node motion visualization. Original mesh shown in green, spring mesh in black..... A-10

Figure A.8. Spring mesh subdivision visualization. Original mesh shown in green, spring mesh in black..... A-11

Figure A.9. Mesh resampling algorithm. (a) Initial mesh guess, (b) first refinement step, (c) first subdivision step, (d) second refinement, and (e) ,(f), (g), (h) are subsequent subdivision and refinement steps ..... A-12

Figure A.10. Resampling of irregular mesh using gridded resampling ..... A-13

Figure A.11. Resampling of irregular mesh with circular boundary using gridded resampling .. A-13

Figure B.1. Hierarchy of a solid manifold in STEP AP 203..... B-3

Figure E.1. Set up of the pylon interference simulation .....E-2

Figure E.2. Return loss plots of MCoRDS single element simulations with the pylon at different locations, using an analysis sweep of 50-500 MHz. Only the range from 150-350 MHz is plotted. ....E-3

Figure E.3. Return loss plots of MCoRDS single element simulations with the pylon at different locations, using an analysis sweep of 150-300 MHz.....	E-4
Figure E.4. Return loss plots of MCoRDS single element simulations with the pylon offset 0.35 inches less in the y-direction, using an analysis sweep of 150-250 MHz.....	E-5

## List of Tables

Table 1. Common elements used in NASTRAN (MSC, 2016).....	20
Table 2. Common element properties in NASTRAN (MSC, 2016).....	20
Table 3. Good and bad parameterization parameters.....	59
Table 4. Antennas used in simulations .....	91
Table 5. Structural model parameters .....	94
Table 6. Loads and deformations of structural model .....	94
Table 7. Surface error metrics.....	99
Table 8. Regression coefficients of different fits to data .....	108
Table 9. Gap addition.....	110
Table 10. Effects of induced offset on array resonant frequency and bandwidth.....	111
Table 11. Change in resonant frequency and bandwidth due to induced offset .....	113
Table 12. Antenna strain and change in resonant frequency .....	118
Table 13. Shifts in radiation pattern for several load cases .....	119
Table 14. Resonance and bandwidth frequencies .....	121
Table 15. Pattern and array rotation.....	123
Table 16. Resonant frequencies and bandwidth for different deformation cases for each array element.....	126
Table 17. Radiation pattern parameters for nominal and deformed cases .....	128
Table 18. Changes in UWB resonance characteristics .....	139
Table 19. Changes in UWB radiation pattern due to deformation .....	140
Table D.1. Material properties of materials used in models tested.....	D-1
Table E.1. Clamped sheet deflection simulation results.....	E-1

## Chapter 1: Introduction

Engineering and science have been revolutionized by exponential increases in computational power and advances in numerical methods in recent decades. Methods like the Finite Element Method (FEM) allow simulation of ever more complex models on modern computers, saving time and cost during the design stages of almost any product, from ships to aircraft and spacecraft and all the way down to cell phones and microprocessors. The ability of physics-based software to simulate a design's performance is enormously beneficial. The need for such tools has sprouted an industry of companies, including MSC and ANSYS, solely dedicated to developing computational tools to aid in these simulations. However, differences between modelling approaches and software used in different fields can sometimes make it difficult to perform an analysis of more than one type (a multiphysics analysis) on the same model. This makes it difficult to design objects whose multidisciplinary performance cannot be determined by a single type of simulation. One such example are wing-mounted antennas, which change their performance when deformed by aerodynamic loads during flight.

Multiphysics simulations are a very broad problem. This thesis seeks to address a small portion of this problem by developing a method of generating deformed geometry from structural simulations and to then use this geometry in electromagnetic (EM) analysis. This work presents a computer program called Antenna Deformation Tool (ADT) designed to create a Computer Aided Design (CAD) model from a structural FEM mesh. ADT can interpret simple geometric abstractions, like adding thicknesses to sheets created from 2D structural meshes, and more complex ones, like mapping antenna CAD models onto a user defined footprint or deformation field. Currently, ADT is limited to working with 2D meshes consisting of triangular and quad elements, and the output CAD has only been tested to work in ANSYS HFSS (ANSYS,

2014) and Siemens NX (Siemens, 2016). ADT is a one-way conversion program between different software. It uses B-spline surfaces to interpolate or approximate FEM meshes and recreate the deformed geometry in a CAD format suitable for HFSS simulations.

This rest of this chapter introduces the motivation for ADT, provides a general overview of the problem being solved, and presents an outline of the approach taken.

## **1.1 Motivation**

Airborne and spaceborne remote sensing are critical in modern society. RADARSAT-2, a Canadian remote sensing observation satellite, provides crucial data for marine surveillance, ice and environmental monitoring, resource management, and mapping (Canadian Space Agency, 2017). Data from this program has been used to detect oil spills (Marghany, 2014), estimate soil moisture (Lievens & Verhoest, 2012), monitor crops (Liu, Shang, Vachon, & McNarin, 2013), and for many other purposes. In the military realm, Space Based Radar (SBR) has been conceptualized at least as early as the 1980s in the United States, as a system for battlefield surveillance (Tsandoulas, 1987). Nuclear-powered radar surveillance satellites (known as RORSAT) were used by the Soviet Union up until 1988 (Siddiqi, 1999). More recently, the Defense Advanced Research Projects Agency (DARPA) and the US Air Force Research Laboratory (AFRL), have worked on a number of SBR projects, such as the Innovative Space-Based Radar Antenna Technology (ISAT), which has been successfully tested in a lab environment (Lane, Murphey, & Zatman, 2011).

Airborne radar is also powerful tool for surveillance, reconnaissance, and other remote sensing applications. The Boeing E-A Airborne Early Warning and Control (AWACS) is an aircraft which has long-range radar and other sensors capable of detecting air and surface threats (NATO, 2017). Like the spaceborne platforms, airborne radar can also be used for soil moisture

measurements (Tabatabaenejad, Burgin, Duan, & Moghaddam, 2013), sea ice measurements (Rodriguez-Morales, et al., 2014), and others. However, both airborne and spaceborne radar suffer from a common problem: degradation of antenna performance due to the structural deformation of the platform itself, either due to aerodynamic, thermal, or other loads.

SBR's typically operate at a relatively high frequency. For example, RADARSAT-2's Synthetic Aperture Radar (SAR) operates in the C-band wavelength, at a frequency of 5.4 GHz, with a bandwidth of 100 MHz and uses an antenna which consists of 512 elements (Canadian Space Agency, 2015). At this frequency range, and with such a large and complex antenna array system, even small thermally induced deformations can cause a significant deviation in the main beam and cause pointing errors, causing the SBR to miss its target (Emmons, 1991). Deformations also broaden the mainbeam, reduce gain, and increase sidelobe levels (Wang H. S., 1992). For this reason, many SBRs must either have very stiff structures to maintain array deformation within a tolerable level, or they must have a real-time mechanical deformation correction mechanism (US Patent No. 6,333,712 B1, 2001). Although studies on the effects of array deformations on antenna performance exist (see Chapter 2), these are generally limited to analytical solutions involving specific antenna shapes and types. Thus, it is necessary to have a tool during the design and analysis stage which can allow to quantify the effects that deformations may have on any type of antenna, so that these can be corrected.

Most airborne systems, such as Airborne Early Warning and Control (AWACS), are mounted to the fuselage with its elements fully enclosed in a rotordome (NATO, 2017) to protect the array and prevent large deformations from aerodynamic loads. However, wing integration is required in applications in which large arrays operate at Very High Frequency (VHF) and lower in the cross-track direction (direction perpendicular to that of travel). This type of integration is



of particular interest at the University of Kansas, where the Center for Remote Sensing of Ice Sheets (CReSIS) develops antenna arrays used for ice sheet sounding. While the larger apertures have the advantages of higher resolution and advanced beamforming and signal processing, in-flight wing deflections can be large (as compared to thermal induced deflections in SBR), and they deform the array and surrounding scattering environment. As with SBR, array deformations have been shown to cause pointing errors, broadening of the main beam, reduced gain, increased sidelobe levels, and degradation of beamforming capabilities (Peters, et al., 2007; Arnold, Yan, Hale, Rodriguez-Morales, & Gogineni, 2014; Gustafsson, McCarthy, & Paulra, 1996). Changes in scattering from time-varying wing structures have also been shown to result in degraded electrical performance (Wang C. , Duan, Zhang, & Zhu, 2009).

Figure 1 shows an example of a wing-integrated antenna array developed by the CReSIS at the University of Kansas. The Multichannel Coherent Radar Depth Sounder (MCoRDS) antenna array shown in Figure 1 was flown on the NASA P-3B as part of NASA's Operation IceBridge (OIB).

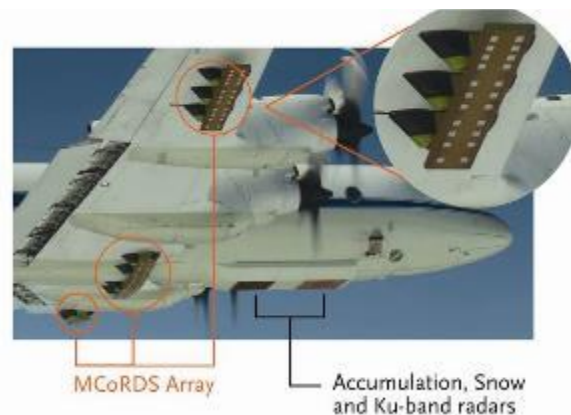


Figure 1: Complete CReSIS sensor suite installed on the NASA P-3. The MCoRDS antenna array can be seen integrated on the wings and under the belly of the fuselage.

In addition to the MCoRDS system, CReSIS has developed several other airborne radar systems that operate over radio and microwave frequencies (Rodriguez-Morales, et al., 2014;

Gogineni, Wong, & Lytle, 2003) to support a variety of snow and ice measurements. One of the primary design requirements for CReSIS ice sounding radars (operating at High Frequencies (HF) and VHF) is the suppression of off-nadir clutter that often masks weak ice-bedrock echoes. While SAR processing is often used in the along-track direction to synthesize a narrow beam (Rodriguez-Morales, et al., 2014; Leuschen, et al., 2014), the only method for suppressing clutter in the cross-track direction is by integrating large aperture antenna arrays onto the wings of the aircraft. These large apertures result in improved directivity and the multiple receivers enable advanced beamforming.

As in the case of SBR, because of airframe-induced array deformation, it is very difficult to accurately characterize and predict the as-installed array performance during the development phase. Thus, a tool that can accurately assess the effect of deformation on antenna performance is necessary. This capability is the main motivation for ADT.

A secondary motivation for the development of ADT is the need to produce conformal antenna arrays based on existing vehicle moldlines and shapes. Conformal antenna arrays are designed to conform to certain shapes, such as aircraft wing skins. This motivation arose from a conformal antenna study being conducted currently at the University of Kansas, which is based on an existing Ultra-Wideband (UWB) antenna design (Yan, et al., 2015). During the development of ADT, it was realized that the program developed could be easily modified to conduct conformal antenna studies. This implies a deformation of the original antenna array dictated by a geometric space constraint, and not structural loads.

The final motivation for ADT comes from the lack of an ability to synchronize different simulation CAD models. Figure 2 gives an overview of the current design process for airborne

antenna arrays. Structural and electromagnetic (EM) simulations occur concurrently, but largely in isolation from each other.

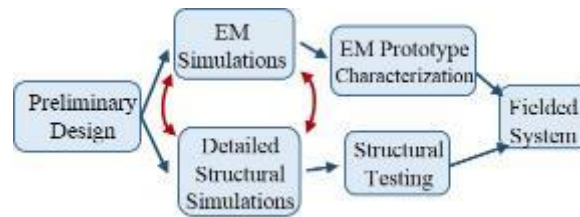


Figure 2: Overview of design process for airborne antenna arrays.

Although these simulations are based on the same CAD model, they are completed in separate analysis packages by different engineers. The detailed designs of the antennas and radomes are dependent on both simulation results, and this iterative process requires updates to the geometry. Not surprisingly, structural and EM simulation models can get out of sync. Having a tool which can easily convert FEM models into geometry would facilitate keeping the models synchronized.

This thesis addresses the creation of ADT to generate deformed geometry associated with antennas and antenna support structure to advance modelling and simulation capabilities for airborne and spaceborne antenna arrays. Deformation fields used by ADT are generated through structural finite element analysis or by a user-defined input in the case of assessing conformal antennas. For the current development process, separate FEM software packages are used for structural and EM simulations. For the applications in this thesis, Patran/NASTRAN (MSC, 2014) is used to assess the structural deformation due to aerodynamic loads, and ANSYS HFSS (ANSYS, 2014) is used to evaluate antenna performance.

## 1.2 Problem Description

The primary topic of this thesis is the creation of a tool to appropriately characterize the effects of in-flight deformations on antenna arrays. The approach chosen to solve this problem consists in using FEM software (NASTRAN and HFSS), to simulate the structural and EM effects separately and bridge the connection between them through a common geometric model. Because the main interest is in the effect of structural deformations on EM characteristics, and not vice versa, only a one-way connection from NASTRAN to HFSS is sought.

NASTRAN uses the Bulk Data File (BDF), which defines the nodes, elements, and element properties comprising the structural FEM, to solve for displacements and stresses. These solutions are stored in a separate file, called the f06 file, which contains all the nodal deformations. Although HFSS can import BDFs directly, this can result in a sluggish model with poor quality surfaces and facets containing no distinction between different regions and surfaces (thus impeding property assignment). Since HFSS uses tetrahedral elements, it requires solid geometry, whereas most elements used in aerospace structural FEMs are 2D, which requires the structural FEM to be modified in most cases to obtain 3D geometry for HFSS. Discarding direct BDF import, ADT was conceptualized as a computer program which could obtain usable deformed geometry from NASTRAN, for HFSS simulations. The initial goals for ADT included:

- 1) ADT must be able to successfully interpret NASTRAN files. This means that 100% of the elements in the NASTRAN file must be identified by ADT and assigned the correct properties and deformations.
- 2) ADT must be able to export the generated geometry into a format which can be read by HFSS.
- 3) ADT must be able to create usable geometry for HFSS. “Usable” is defined as:

- a. The geometry does not significantly impact the graphical performance of HFSS.  
This is a mostly qualitative assessment – the user must not experience sluggishness while using HFSS with ADT geometry. It implies generating smooth continuous surfaces from the thousands of facets making up the BDF.
  - b. The geometry contains the correct type of surfaces and solids, implying the creation of a surface thickening mechanism for thin sheets in which users can control which sheets to thicken.
  - c. The geometry does not cause meshing errors in HFSS.
  - d. The geometry surface error is small enough to be representative of the NASTRAN mesh. Errors cited in literature for B-spline surface (the type of surfaces used in this thesis) fitting vary between 0.05% and 0.2% for mean error and 0.3% to 1.5% for maximum error, where the error is normalized with respect to the diameter of the object being fitted (Eck & Hoppe, 1996). Typical errors (Hausdorff distance) among algorithms fitting discretized meshes to point clouds vary between 0.5 and 50 mm for different datasets, with an average Hausdorff distance of around 5 mm (Berger, Levine, Nonato, Taubin, & Silva, 2013). These errors are discussed in more detail in Chapter 7.
- 4) ADT must be able to correctly deform antenna designs.
  - 5) ADT must be robust, and must be able to correctly interpret as wide a range of structural models as possible.
  - 6) ADT must be easy to use for general research. This can be measured in the amount of commands required to process a model, as well as the ease with which these commands can be accessed and modified by someone new to ADT.

7) Although speed is not a priority during initial development, processing time must still be reasonable. Generally, surface fitting algorithm execution time varies depending on the data and type of fitting performed, as well as the hardware used. Because of this, and since these factors can be subjective, a direct comparison to literature is of little use. Instead, a notional, order of magnitude expectation is set. The models that ADT processes are expected to have between tens of thousands to hundreds of thousands vertices. A processing and surface fitting time on the order of minutes is expected for a model which is in the low range (10,000 vertices), going up to a few (1-4) hours for models with several hundred thousand vertices. In general, processing times which are less than the expected burden of manually creating deformed geometry (which would be on the order of several hours for common models, if the geometry can be manually deformed at all) are deemed acceptable. In terms of antenna model deformation (where no surface fitting occurs), the expectation is of tens of minutes for an antenna consisting of around 2,000 geometric entities, which was the typical value seen for the antennas tested in this thesis. These requirements are revisited at the end of the thesis, to evaluate how well ADT accomplished them.

### **1.3 Thesis Organization**

This document is organized into eight chapters, with six appendices. Chapter 1 is the introduction, which gives a broad overview of the problem and the proposed solution. It defines the types of data which are used, and the program requirements. Chapter 2 presents previous work which has been done in the area of characterizing the effects of deformation on antenna arrays, and provides an overview of some different software which can either do multiphysics analysis or convert geometry from a faceted to a continuous surface model. Chapter 3 describes

the formulation of the problem in more detail and provides an overview of ADT. Chapter 4 goes over the geometry frameworks and definitions which are used by ADT, and how these concepts were applied to this work. Chapter 5 explains the methods which were used to reconstruct geometry from meshes in detail. Chapter 6 describes how the antenna deformation module in ADT works in detail. Chapter 7 presents the results which were obtained with ADT. Chapter 8 finalizes this thesis with a discussion of conclusions and future work for this project.

## Chapter 2: Previous Work

This section reviews past and current research related to modelling the effects of deformation on antenna performance, as well as some existing software capable of surface reconstruction from meshes and point clouds, as well as multiphysics simulation programs. The first section reviews some of the research done in structural and EM coupling, most of which is analytical in nature. The next section reviews some software with multiphysics capability, while the last section reviews software which has the capacity of reconstruct geometry from meshes.

### 2.1 Estimating Effects of Deformation on Antenna Arrays

Array deformation is an area of interest and concern in airborne and space applications; however, there are few tools capable of supporting such analysis. Many researchers focus on analytical approaches to this problem, limiting the types of deformations experienced by the arrays, as well as the shapes and arrangements of these arrays and their elements. Other approaches involve the use of coupling environments to directly connect FEM programs. Some researchers are taking an approach similar to the one described here: they are using existing FEM software and coupling the results through geometric translation. This thesis seeks an approach to obtain the effects of a general deformation on a general type of antenna element or antenna array, using the power and generality of FEM analysis.

Wang and Bao (2008) investigated the effects of array distortion by analyzing the effect of thermal-induced distortion on an Active Phased Array Antenna (APAA), consisting of a rectangular array of elements. Two deformation cases were analyzed: curvature along one axis and curvature about two axes. In both cases the deformation is symmetric about the center. They found that both deformation cases “decrease gain and broaden the main beam” and side lobes exhibit little to no change. While an analytical solution is expedient since a full-wave analysis



was not performed, changes in mutual coupling between the elements, as well as changes in element shape and surrounding structure were not accounted for in the results. In general, ADT allows more versatility in the deformation shape and magnitude, and it assesses the effects of element deformation.

Knott (2010) also took an analytical approach to antenna deformation of conformal arrays. He focuses on discussing methods of compensation for static deformations and modal vibrations and limited his analysis to simple linear shapes and cylinders, since the natural frequencies and mode shapes of these simple shapes are easy to calculate. Compensation methods were applied to the deformed shapes and compared to the original radiation pattern. While the focus of this thesis is not on mode shapes but on static deformation, this could be implemented in the future, as the input program (NASTRAN) is able to compute mode shapes. This could be implemented by extracting mode shape eigenvectors from the f06 file, and using them to apply relative displacements to the mesh. The eigenvectors could be used to sweep through a range of deformations to identify critical excitation levels. Similar to Wang and Bao's analytical method, Knott's method does not take into account coupling between elements and near-field scatters, thus the accuracy of such an analysis is more limited.

In the realm of space applications, Tanaka (2011) used the Ruze formula (1952) to express the relationship between the main lobe changes in antenna gain and errors on the antenna surface to develop an analytical model to predict surface errors of a parabolic cable antenna consisting of 924 cables. He verified this model using numerical simulations and developed a model for correcting these deformations. This analytical model is only applicable to a parabolic antenna, and was not tested for a phased array. The approach presented here allows for more general antenna shapes.

Spline-based analytical approaches for analyzing distorted parabolic reflector performance using divided fitting exist (Wang, Zheng, & Zhang, 2008), but they too are limited to parabolic shapes. Similar papers by these authors include the design of an entire optimization system which couples EM and structural simulations, used for parabolic reflector antenna structure design, (2006), as well as another paper laying out the groundwork on using Coons patches (a type of spline surface) to design and optimize parabolic reflector shape and structure (2004). In these works, Wang et al. design a system which minimizes certain design variables, to maximize the gain of a parabolic reflector and minimize structural weight. Again, this approach is limited to parabolic reflector shapes.

Senbokuya and Tsunoda (2003) worked on a seven-element flexible circular patch array for space applications and used analytical software to determine the effects of the deformation of this antenna on its performance. To properly simulate far field radiation patterns, the original geometry had to be changed from seven circular elements with one circular ground plane to each element having its own, smaller ground plane, as shown in Figure 3.

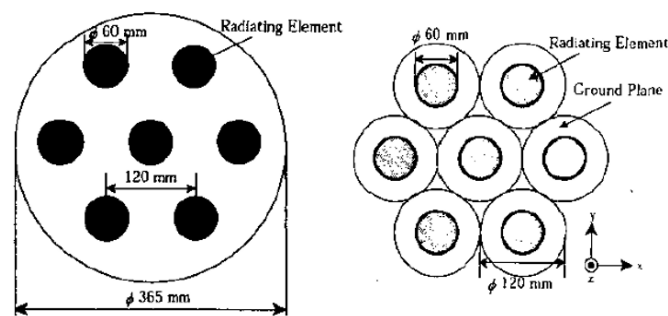


Figure 3. Original and modified antenna array (Senbokuya & Tsunoda, 2003).

This change was made for the software to be able to analyze the antenna model. The use of 3D full-wave FEM packages proposed in this work would avoid this problem, as the geometry

need not be limited to certain shapes or forms, as long as it can be properly interpreted and discretized.

Perhaps the most similar research to the one presented here is the one conducted by Zhou et al. (2015; 2017), who have been working on developing structurally integrated antenna arrays for aircraft. As part of their work, they developed approaches using analytical coupling and deformed geometry from a mechanical FEM software (ANSYS) to reconstruct deformed geometry for HFSS. Specifically, they developed a 64-element microstrip antenna array, with the elements laid out on a single surface in an 8 by 8 array. They used B-splines to interpolate the deformed surface in their earlier work, and Extreme Learning Machine (ELM) in their later work, and then analyzed the deformed model in HFSS. However, they are only capable of generating the deformed functional surface, or antenna surface. The rest of the model, which includes the substrate and other layers, is re-created manually in HFSS. The work presented herein can obtain deformed geometry for all surfaces in a structural model and minimizes user involvement in recreating geometry manually.

Wu, Tan, Zhu, and Zhou (2013) used B-splines as the basis functions for time-domain FEM solvers. This is an interesting development, as it could allow CAD and FEM systems which use the same geometric framework, foregoing the need for geometric conversion between the two. Hughes, Cottrell, and Bazilevs (2005) also demonstrated a similar method, called Isogeometric Analysis (IGA), which can avoid the problems introduced by using different geometric representations for design and analysis. The development of a common CAD and FEM analysis framework is not the focus of this work, but a possible future direction for future research.

## 2.2 Multiphysics Software

Myriad software programs exist in the realm of FEM, CAD, and other computer graphics. It is therefore not surprising that software with similar capabilities to the ones being sought exist, though none quite achieve what is being proposed in this work. Multiphysics simulation software uses the output of one simulation to define the geometry and/or boundary conditions in a subsequent simulation for a different physical interaction. This process is often circular wherein the output of one simulation is continually fed to the next until the solution has converged.

ANSYS Workbench (ANSYS, 2017) is a multiphysics FEM software consisting of several modules for different types of analysis. It allows the user to create the desired geometry, perform a structural analysis using the structural module, and export the deformed shape to be used by the electromagnetic module for analysis. Deformed geometry can also be exported as a CAD file in the Parasolid format for analysis in other software. In addition, Workbench can import BDF files, and export them as Parasolid files. However, during this conversion, thickness is lost, as 2D element surfaces are converted to sheets, which, depending on the model, requires significant manual post processing to recover. Using the Workbench native structural module or a BDF import file with the Workbench Parasolid converter would require including the antennas in the structural simulation - even though they are typically non-load bearing and can add a lot of unnecessary detail. ADT avoids this by allowing the user to simply define a footprint for the antenna in the structural model, and uses an algorithm to interpret and associate antenna geometry with the deformed structure. This allows work on the structural and antenna geometries to proceed separately with a mechanism to seamlessly integrate the two when needed. If the antenna is load-bearing, then it can simply be defined in the structural simulation.

Unlike Workbench, MpCCI (Fraunhofer-Gesellschaft, 2017) is a multiphysics coupling environment, which allows coupling of different types of simulations by connecting commercially available FEM software. However, it is designed to work with solvers where structural loads are generated by thermal inputs from electromagnetic components and whose thermal deformation affects the electromagnetics. The primary interest of this work is in large aerodynamic loads, and the effects that the electronics have on antenna support structures are insignificant. Thus, a two-way coupling is not currently required, but might be necessary in the future for spaceborne array analysis. In addition MpCCI also does not currently support ANSYS HFSS.

### **2.3 Geometric Conversion Software**

In addition to multiphysics software, geometric conversion software exist which specialize in converting meshes into parametric surfaces, similarly to the Parasolid conversion module in Workbench. Two of these, Geomagic and Maya, can perform tasks similar to ADT in terms of surface reconstruction.

Geomagic (3D Systems, 2017) is a software which specializes in creating surfaces and gapless models from point cloud data. It is mostly geared towards the creation of CAD models and molds from scanned 3D point data, and not adapted to work with BDF files of FEM meshes. Although point data could be extracted from the original BDF and used in conjunction with this software, it would involve significant user manual work and would not guarantee control over the specific surfaces being generated, which is necessary to apply the desirable properties for each surface.

Maya (Autodesk, 2017) is a software created by Autodesk, mostly geared towards animators and videogame designers. It provides many creative tools to work with polygonal

meshes and other types of geometry. It allows converting polygonal meshes into subdivision surfaces, which are another type of polygonal representation. This representation can in turn be converted to Non-Uniform Rational B-Splines (NURBS), which can then be exported as a standard Parasolid or STEP CAD file. As with Geomagic, the user has little control over the shape and number of surfaces generated by this function, requiring significant mesh pre and post processing to ensure that the right surfaces are created. The work presented here generally builds and groups surfaces by their material properties, and can automatically determine surface shapes by analyzing the BDF before surface fitting begins. In this way, manifold vertices (which are a problem for generating subdivision surfaces, and subsequently NURBS in Maya), can be dealt with accordingly by ADT.

Tehcnosoft (Technosoft, 2018) is a company which specializes in developing engineering software to improve the automation of engineering design and analysis processes. They developed the Adaptive Modeling Language (AML), a powerful object-oriented modeling framework designed for multidisciplinary modeling. In related publications, Schueler, Miller and Hale (2004; 2002) address the development of a tool called Steered Composite Analysis and Design System (SCADS), an integrated system for the design and analysis of composites, based on AML. This system integrates the design and analysis of composites, streamlining the process. It uses Non-Uniform Rational B-Spline (NURBS) geometry to approximate composite fiber paths and offset these paths to create tows. It provides excellent insight into processes which can be used for curve offset, error estimation, and design and analysis integration, since SCADS is capable of interfacing with NASTRAN. This program is very purpose-specific, but incorporates geometric and software concepts which are important to ADT. Many of these concepts may be

implemented in the future, as work continues to integrate ADT into the design and analysis process.

The next chapter describes ADT by outlining the structure and workflow of the program and sets the stage for detailed discussion in the chapters following it.

### **Chapter 3: Program Description**

This chapter discusses and describes some details of the FEM software packages used and outlines ADT. The first section of this chapter discusses the general concepts of Finite Element Analysis (FEA) and some details of NASTRAN and HFSS. The next section gives a general overview of ADT, setting the outline for a more detailed discussion of ADT and its components in subsequent chapters.

#### **3.1 The Finite Element Method**

The FEM or FEA is a method to solve systems of constrained partial differential equations (PDE), representing a mathematical model of a physical phenomenon, over a discretized geometric domain (Surana & Reddy, 2016). The FEM is used widely in all fields of engineering, from structures to electromagnetics. It works by discretizing a geometric domain into units called elements and finding a solution to the pertinent equation systems over each of the elements, based on a set of boundary (or initial value) conditions. The summation of the solution over the discretized space makes up the solution for the entire FEM system. Each element is a geometric and mathematical unit consisting of a set of basis functions defined over its own domain. Basis functions are interpolating functions approximating the behavior of the physics being analyzed. Commonly, basis functions are polynomials of varying degrees. Elements can have as many dimensions as necessary to solve a given problem, depending on the mathematical model used. In structural mechanics, rods are easily modelled as 1D elements, thin surfaces like aircraft skin are best modelled as 2D elements, and thicker structures can be modelled using 3D elements. This section will go over some of the most common elements used in MSC NASTRAN and HFSS, and their properties.



### 3.1.1 Structural FEA NASTRAN

NASTRAN (NASA Structural Analysis) (MSC, 2014) is a FEA code originally developed for NASA and used for structural analysis. It is a solver and does not include the necessary tools to generate geometry or meshes. At the University of Kansas, NASTRAN is primarily coupled with Patran, a software provided by MSC, which provides the necessary user graphical interface for creating FEM models. Two features define NASTRAN elements: their properties (the names of which start with a “P” in the BDF) and their connectivity (the names of which start with a “C” in the BDF) (Siemens, 2014). Some of the elements used by NASTRAN are indicated in Table 1. Table 2 indicates which element properties can be applied to which elements, and whether ADT supports them.

Table 1. Common elements used in NASTRAN (MSC, 2016).

Name	Type	Element Description	Supported by ADT
CSHEAR	2D shear plate	Shear plate defined by 4 nodes	No
CTRIA3	2D shell, linear	Triangular with 3 nodes	Yes
CTRIA6	2D shell, higher order	Triangular with 6 nodes	No
CQUAD4	2D shell, linear	Quadrilateral with 4 nodes	Yes
CQUAD8	2D shell, higher order	Quadrilateral with 8 nodes	No
CBEAM	1D Beam, linear	Beam	No
CBAR	1D Bar, linear	Bar	No

Table 2. Common element properties in NASTRAN (MSC, 2016).

Name	Defining properties	Elements used with	Supported by ADT
PSHEAR	Thickness, material	CSHEAR	No
PSHELL	Thickness, material for membrane, bending and shear	CTRIA3, CTRIA6, CQUAD4, CQUAD8, CTRIAR, CQUADR	Yes
PCOMP	Ply material IDs, thickness and orientation	CTRIA3, CTRIA6, CQUAD4, CQUAD8, CTRIAR, CQUADR	No
PBAR, PBARL	Cross sectional properties, material	CBAR	No
PBEAM, PBEAML	Cross sectional properties, material	CBEAM	No

Because most aerospace structures consist of thin metal sheets, shell elements are the most commonly used, with shear elements used to define reinforced shear panels, where shell elements would be overly stiff. Out of the shell elements, PSHELL are commonly used to define structures made of isotropic materials, like most metals, and PCOMP for composite structures. ADT currently only supports CQUAD4 and CTRIA3 elements using PSHELL properties. Beam and rod elements are also often used when modelling aerospace structures such as stiffeners and other slender objects but are not currently supported by ADT. These elements can be included in the structural model and neglected during the geometry reconstruction process. In the future the ability to interpret these elements will be added so that structural antennas modelled by these can be properly processed. Three dimensional elements are occasionally used for very thick aircraft structures but are not yet implemented in ADT.

Meshing and choice of elements and element properties in NASTRAN is delegated to the user, and quad elements are preferred over triangular elements for 2D element modelling. This contrasts with meshing in HFSS, which is mostly automatic, and employs exclusively triangular and tetrahedral elements.

### **3.1.2 Electromagnetic FEA HFSS**

HFSS (High Frequency Structure Simulator) (ANSYS, 2014) is a high frequency electromagnetic field simulator, which uses the FEM technique, the method of moments, methods of physical optics, and shooting and bouncing rays to solve electromagnetic models. It is frequently used in antenna design and simulation (ANSYS, 2018).

HFSS allows the user to import or generate geometry but allows limited control over mesh generation and refinement. By default, HFSS automatically generates FEM meshes and elements, refining them in subsequent iterations, and allowing the user only to set the initial

mesh settings, if desired. The user can set a convergence criterion used to stop the iterations – a value of 0.02 delta S is used for all simulations run in this thesis, based on common practices. Delta S is the change in return loss parameter, and is used as the default convergence criterion by HFSS. Knowing this, an “acceptable” model or geometry is defined as one which has successfully converged in HFSS using the established convergence criterion. No quantifying metric for model acceptability or goodness can be provided at this point and is left for future work, since the main goal of ADT is to create a geometric model which can run in HFSS. Quantifying metrics for surface error and mesh quality, both of which measure how closely the model generated by ADT resembles the original structural mesh, are provided in the results section.

HFSS uses two types of elements: tetrahedrons and triangles. Tetrahedrons are used to discretize solids, and triangular elements are used to discretize sheet geometry. While some conductors can be modelled as sheets (for lower frequencies), all other parts are usually modelled as solids, including conductors at higher frequencies.

For a HFSS model to run properly, intersections between the different bodies in the geometry (solids and sheets) of the model must be avoided. Ideally, this requires at least positional (G0) continuity between shared geometric boundaries, and preferably G1 and G2 continuity. ADT currently lacks the capacity to guarantee even G0 continuity, which can cause intersections in the geometry. Small intersections can cause meshing errors. Significant intersections will prevent the model from running. Small gaps between surfaces or solids as well as small intersections can cause the model to run indefinitely, possibly because it fails to reach the convergence criterion established. Such features can also cause elements on some surfaces to be refined an order of magnitude more than on other surfaces and make the model fail to

converge. Models with these characteristics are also considered not “acceptable”, though it is difficult to predict which models produced by ADT will have these qualities, since the inner working of HFSS and how the program decides to mesh and solve simulations remains a “black box”.

As mentioned before, and as is discussed further in the results and conclusions sections, many problems with current ADT-generated models not being “acceptable” are caused by the lack of an equivalencing mechanism, or a lack of associative geometry to guarantee at least G0 continuity. The problem is mitigated by using associative geometry in special cases involving sheet antenna geometry, which guarantees positional continuity, but this mechanism is yet to be implemented for the rest of the geometry. Chapter 6 explains these methods in more detail. To completely avoid convergence errors further improvements in robustness are necessary and are left for the future.

The presence of intersections or small gaps requires the not “acceptable” geometry produced by ADT to be post-processed using Boolean or offset operations, which can add significant workload for a complex model. Nevertheless, geometry which may require some manipulation is preferable to the complete lack of ability to produce deformed geometry. In some cases, this involves introducing offsets between touching geometry, which influences the results obtained from the HFSS model.

### **3.2 Software Structure and Requirements**

ADT consists of two modules, as shown in Figure 4: a core module written in MATLAB (MathWorks, 2018) (consisting of a main module, a structural module, and an antenna deformation module), and a CAD export module written in C++. The CAD export module uses

information from the MATLAB module to create a CAD model; it receives this information through a Transitional Text File (TTF), which connects both modules.

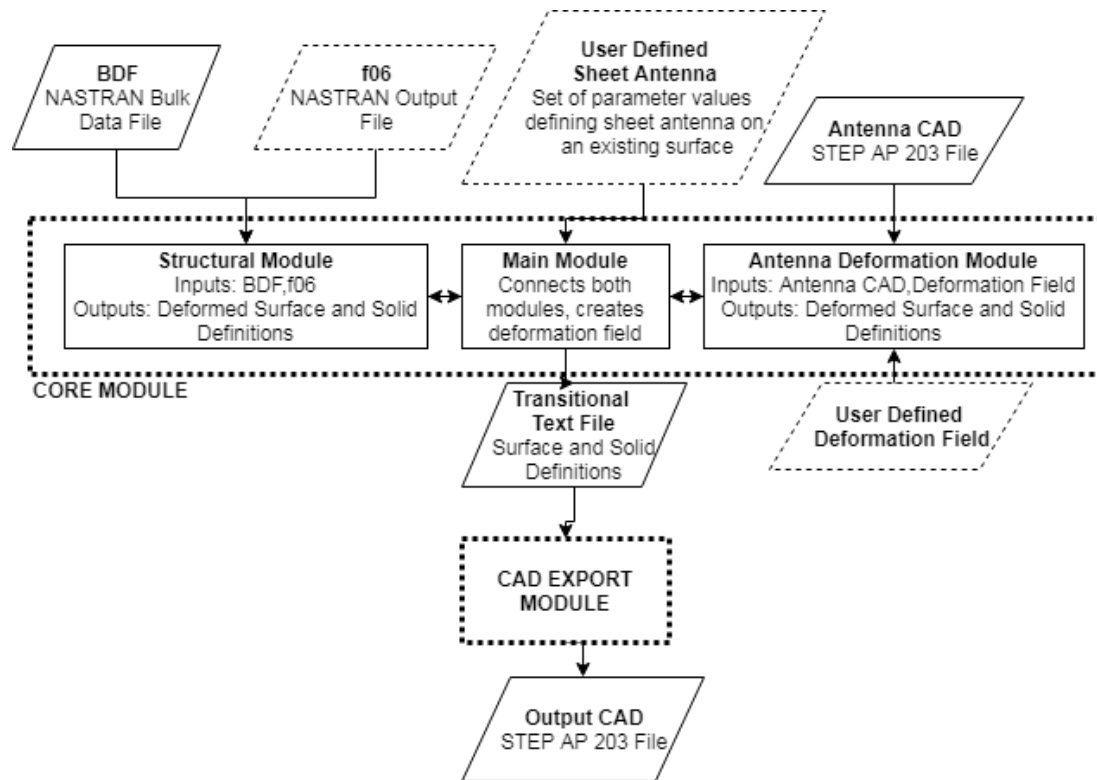


Figure 4. ADT Structure, with optional inputs shown as parallelograms bounded by dashed lines.

The structural submodule processes meshes from NASTRAN, the antenna deformation submodule imports and deforms antenna CAD files, and the main submodule connects them. Both the structural and antenna submodules can operate independently from each other. The product of the core module is the TTF. It contains all the information on surface, edge, and solid definitions of the deformed geometry, information on the regions each surface or solid belongs to, and some connectivity information for valid surface and solid reconstruction.

The CAD export module is an executable file which reads the TTF and generates a STEP CAD model. This CAD model is the final product of ADT; it can either be directly imported into

HFSS for simulations, or it can be post processed in a CAD program, such as NX (Siemens, 2016).

The next sections explain each of the modules and the TTF in more detail.

### **3.2.1 Core module**

The core module constitutes the main portion of ADT and does most of the geometry processing and fitting. It is divided into three submodules: a structural submodule and an antenna submodule which are connected by the main submodule. Since the structural and antenna submodules can operate independently, three main modes of operation are possible: integrated structural and antenna deformation, structural deformation only, or antenna deformation only.

The structural submodule reads the BDF and f06 files, identifies separate surface meshes, fits one surface to each separate surface mesh, and applies a thickness to the surfaces which require it. This process generates an object called a base file, which contains the information on the deformed geometry generated from a BDF and an f06 file and can be used to produce a TTF. The base file contains all the relevant information from the BDF and f06 and follows a structure like that of the BDF. It includes the coordinates and ID numbers of all the nodes in the BDF, connectivity and element information, as well as element property information. The hierarchy of classes and information making up the base file is given in Figure 5.

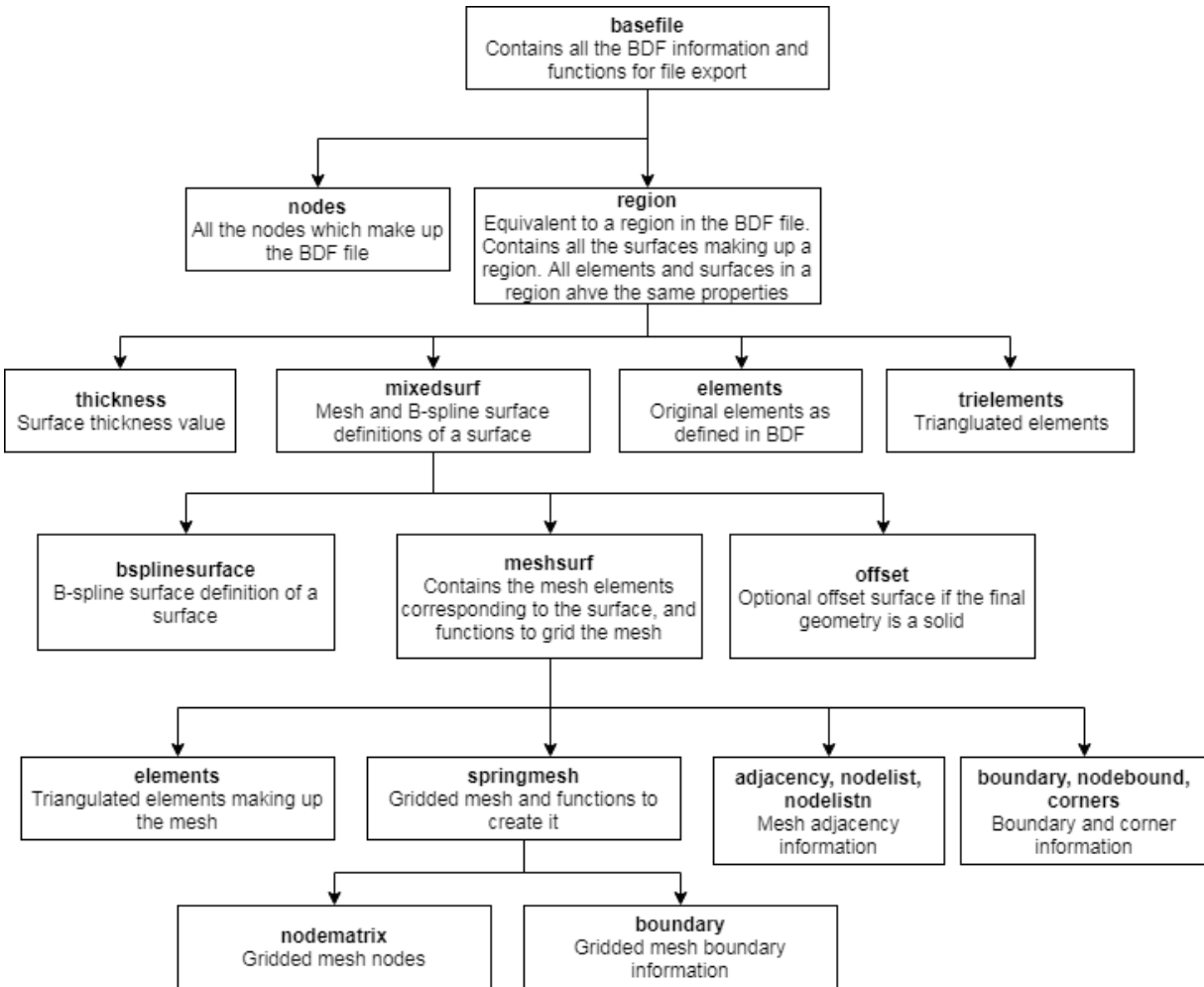


Figure 5. Base file class hierarchy.

The base file consists of a set of regions, where each region consists a set of elements from the FEM which have the same properties. These regions are defined by the user in Patran. Each region can consist of a single or several surfaces (which are the “mixedsurf” type objects in the base file, see Figure 5), where a surface is represented by a mesh (meshsurf), and a B-spline surface fitted to the mesh (bsplinesurface). An optional offset surface is specified the surface is thickened to a solid (explained in Chapter 5). Surfaces are determined based on adjacency and dihedral angle properties, the mechanism for determining them is explained in detail in Chapter 5. The mesh (meshsurf object) has a variety of properties and functions which allow resampling

the original mesh into a gridded quad mesh, which is explained in more detail in Chapter 5 as well.

The currently accepted BDF files must consist of 2-D CTRIA3 or CQUAD4 elements defined by a shell property (PSHELL). This capability will be extended in the future to include elements defined using composite properties (PCOMP). In the current iterations of ADT, averaged material properties were used as shell properties to define elements in NASTRAN, and averaged EM properties were applied to geometry made from dielectric composites. This approach can cause performance differences between EM simulations and EM measurements, due to the use of bulk properties instead of individual layer properties. Layer differentiation within the composite geometry will be implemented in the future, using different properties for each layer of the composite.

The antenna submodule imports geometry from a STEP CAD file and generates a set of objects which contain the geometric definitions from the CAD file. It then transforms these objects using a deformation field (or function) into deformed objects. This deformation field can be either obtained from the structural model or defined by the user. A secondary way of generating purely sheet (2D) antenna geometry is available through the main submodule coupled with the structural submodule. This submodule can be used to define antenna surfaces based on existing structural surfaces. The details of antenna import, deformation, generation, and export are explained in Chapter 6.

The geometry generated is exported as a STEP AP 203 file, which was chosen due to availability of the required code libraries, universality, readability, and convenience. The exported CAD can contain either only the structural or antenna geometry, or both. More details on the STEP file are in Appendix B.



### **3.2.2 Transitional text file**

The TTF is an ASCII text file is created to transfer information from the core module to the CAD export module. A text file format was chosen because it is easy to read, write, and edit. The text file was formatted into a readable format, allowing easy debugging and manual editing. It consists of a series of special markers, which are preceded by the number sign (#) and followed by the data corresponding to the marker. The markers used and TTF structure are explained in Appendix C.

### **3.2.3 CAD export module**

The CAD export module is an executable file, compiled from C++ code which uses STEP libraries provided by STEP Tools (STEP Tools, Inc., 2018) to read the TTF and create a STEP CAD file. Details about the STEP file are in Appendix B.

The next chapter presents the geometric frameworks and definitions used by ADT, which can help understand the material in Appendices B and C better.

## **Chapter 4: Geometry Definitions and Methods**

One of the main focuses of this thesis is geometry and the different ways geometric features can be defined and represented mathematically. While no new geometric definitions are presented, several different concepts were explored with the aim of understanding and connecting the different types of geometric representations employed across FEM, CAD and other computer software. This section summarizes the geometric frameworks used in ADT: B-spline surfaces and polygonal meshes. Much of the BDF file can be thought of as a polygonal mesh, and the end output of ADT are B-spline surfaces (see Section 3.1). Section 4.2 dives more in depth into B-spline surfaces.

### **4.1 Geometric Frameworks**

There are many geometric frameworks used in computer graphics today, with different industries leaning towards certain geometric representations. Many factors influence the decision to use a specific framework, such as its maturity, stability, flexibility, ease of use, computational speed, and the degree of precision in defining surfaces and solids. So, while the computer animation and computer game industry lean heavily on faceted polygon models due to their simplicity and computational speed, splines are preferred in the CAD and CAM industries, driven by the need to precisely define surfaces. The use is sometimes driven by existing standards, since developing and adapting new geometric standards and libraries can be a gargantuan task. This section provides an overview of the geometric frameworks relevant to this thesis and their advantages and common uses.

#### **4.1.1 Polygonal meshes**

Polygonal meshes, also called faceted geometry, are perhaps one of the earliest ways of representing geometry on a computer. They are geometric models consisting of a set of vertices,

edges, and faces. Vertices are points in Cartesian space, and connected vertices define an edge, which is a line segment from one vertex to another. A set of coplanar edges forming a convex closed boundary defines a face. Generally, triangles are used as faces in this type of representation, since they allow the greatest flexibility in approximating different shapes; quadrilateral meshes are also common. Although simple, this type of representation allows for incredible flexibility, and is widely preferred by the computer game and computer animation industries due to its versatility. The main problem is its limitation to precisely represent geometry. This is shown in Figure 6, which is a polygonal model of the face of the bust of queen Nefertiti of Ancient Egypt.

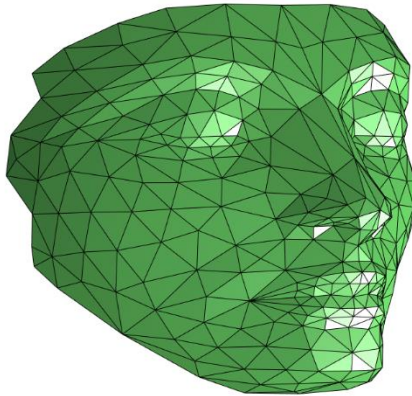


Figure 6. Polygonal mesh of a face (Peyre, 2009).

The features of this face are rough and not detailed. As expected, the precision of a polygonal mesh can be increased by increasing the face count (or density). Coupled with smart shading and smoothing techniques, polygonal meshes can appear very smooth, as shown in Figure 7, which shows the same mesh of a mannequin head under different reflection models. While the faceted nature of the model easily seen when flat shading is used, Phong shading (Bui Thong, 1975) provides a much smoother appearance. Although they appear smooth, these

meshes still do not define a surface precisely, and are not adequate for many engineering applications.

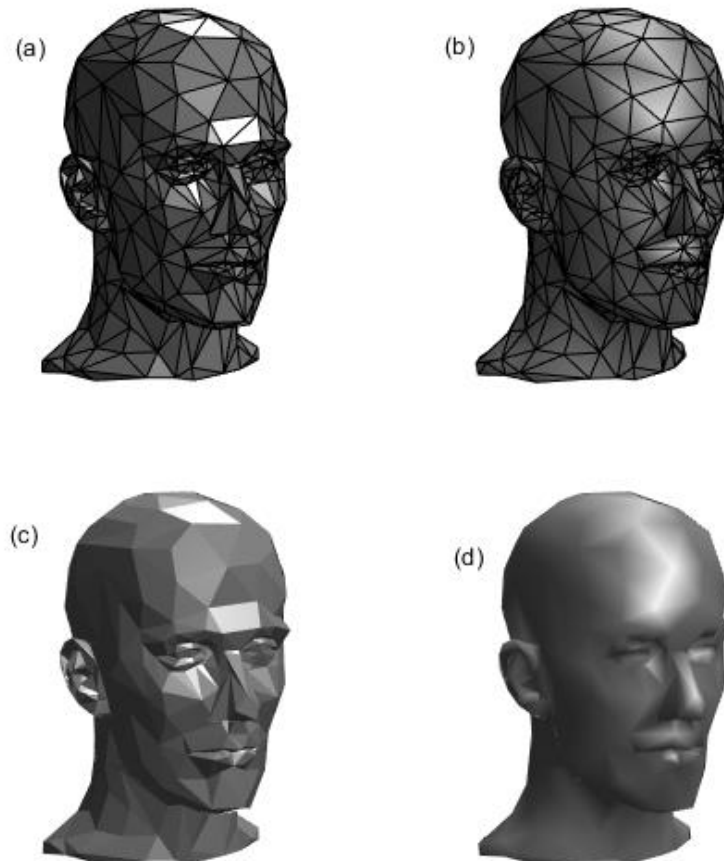


Figure 7. Smoothing of mannequin head polygonal model. (a) and (b) show the model using flat and Phong shading, respectively, showing edges, while (c) and (d) show the same without the edges. All use the same mesh (Peyre, 2009).

An exception is FEM software, which requires discretized meshes to compute solutions. The specific types of faces used vary by software and application, with quads being preferred for most 2D meshes, and tetrahedrons used for the analysis of solids. FEM elements are further discussed in the previous chapter, Section 3.1. Polygonal geometry is also used in several types

of files used for 3D printing, such as the Stereolithography (STL) format and the Wavefront Object (OBJ) format.

For our application, polygonal meshes form the underlying structure of the 2D meshes defined in the BDF, which make up most of the elements used in FEM analysis of aerospace structures. The BDF is imported as a polygonal mesh, and many algorithms which are designed to work with polygonal meshes, such as Dijkstra's shortest path (Dijkstra, 1959), and different types of parameterization, are used. The final goal of the geometric translation module in ADT is to convert these polygonal mesh representations into B-spline surface representations.

#### **4.1.2 Bezier curves, B-Splines and NURBS**

Developed out of the work of Pierre Bezier, a French engineer who worked for Renault in the early 1970s, Bezier curves, B-splines, and Non-Uniform Rational B-Splines (NURBS) are parametric ways to represent curves and surfaces precisely and intuitively. Initially used in the automotive industry to design tools and parts, their use has spread to every branch of engineering and beyond, including font design. They are smooth parametric polynomials, suited to represent most general shapes. Unlike polygonal meshes, they can precisely define an object, and are preferred in engineering applications. The common trait shared by these types of geometry is that they are defined by a set of control points, degree, and (in the case of B-splines and NURBS) a knot vector (Piegl & Tiller, 1997). Figure 8 shows a simple NURBS surface with its control points. For this geometric definition, the shape of the surface is determined by the influence of each control point on each part of the surface. This influence is in turn determined by the B-spline coefficients. One of the main limitations of B-spline surfaces is that they must have a rectangular grid of control points and be topologically equivalent to a disc (Pixar, 2017), which limits their use in modelling more topologically complex shapes and requires the use of several

patches with additional constraints imposed on the boundaries (as shown in Figure 9). This can make the generation of a surface from a random unorganized mesh difficult, as a good parameterization to get a gridded control mesh can be difficult to obtain for non-gridded data. However, since B-spline surfaces are ubiquitous in all CAD programs and standards, they are used for this project. A more detailed definition of these surfaces is given in Section 4.2.

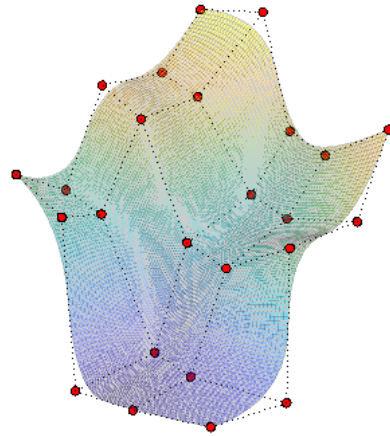


Figure 8. Simple NURBS surface with control points shown in red (Claxton, 2007).

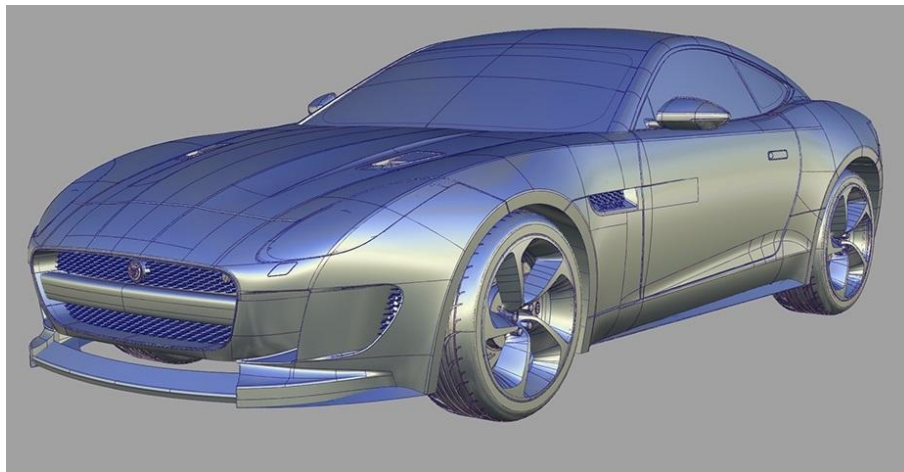


Figure 9. Example of Jaguar F-Type model created with NURBS. Individual NURBS patches are bounded by the dark curves (Singhvi, 2015).

## 4.2 B-spline and NURBS Surfaces

After exploring different types of geometry frameworks, B-splines and NURBS were chosen to develop the geometric framework for the output geometry of this project. This is mostly due to their use in every CAD standard and their versatility in representing surfaces. This section delves into the details of B-spline and NURBS surfaces. It begins by introducing Bezier curves, a simpler form of splines, and continues on to B-spline and NURBS curves and surfaces.

### 4.2.1 Bezier curves

The two most common methods of representing curves and surfaces are by implicit and parametric functions (Piegl & Tiller, 1997). Equations 1 and 2 are both equations for a unit circle centered at the origin. While Equation 1 is implicit, Equation 2 is parametric. In the parametric form, each coordinate is represented as a function of a parameter,  $u$ , making  $C(u)$  a vector valued function. The parametric form is more commonly used in computer geometry, as it is more natural for this application.

$$f(x, y) = x^2 + y^2 - 1 = 0 \quad (1)$$

$$\begin{aligned} C(u) &= (x(u), y(u)) \\ x(u) &= \cos(u) \\ y(u) &= \sin(u) \\ 0 &\leq u \leq \frac{\pi}{2} \end{aligned} \quad (2)$$

Bezier curves (mentioned in the previous section) are polynomial parametric curves defined by a degree and a set of control points. Unlike a power basis polynomial, they are formulated to allow easy interpretation and manipulation. Mathematically, they are defined by Equations 3 and 4 (Piegl & Tiller, 1997).

$$C(u) = \sum_{i=0}^n B_{i,n}(u)P_i \quad (3)$$

$$B_{i,n} = \frac{n!}{i!(n-i)!} u^i (1-u)^{n-i} \quad (4)$$

In Equation 3,  $B$  is the Bezier basis function (also known as Bernstein polynomial),  $P$  are control points,  $n$  is the degree of the curve,  $C$  is the curve being defined, and  $u$  is the parameter value. Thus, every point on curve  $C$  is a function of a parameter value  $u$ . The control points,  $P$ , are points in Cartesian space, and so each  $P$  in Equation 3 is a column vector of  $x$ ,  $y$  or  $x$ ,  $y$  and  $z$  coordinates. The values that  $C(u)$  assumes will also be column vectors, representing the evaluated position at a specific parameter  $u$ . The parameter  $u$  is defined between 0 and 1, where 0 is the start of the curve, and 1 is the end of it. Figure 10 shows examples of Bezier curves, with their respective control points and polylines. Narasimhan's (2014) code is used for MATLAB visualization of Bezier curves.

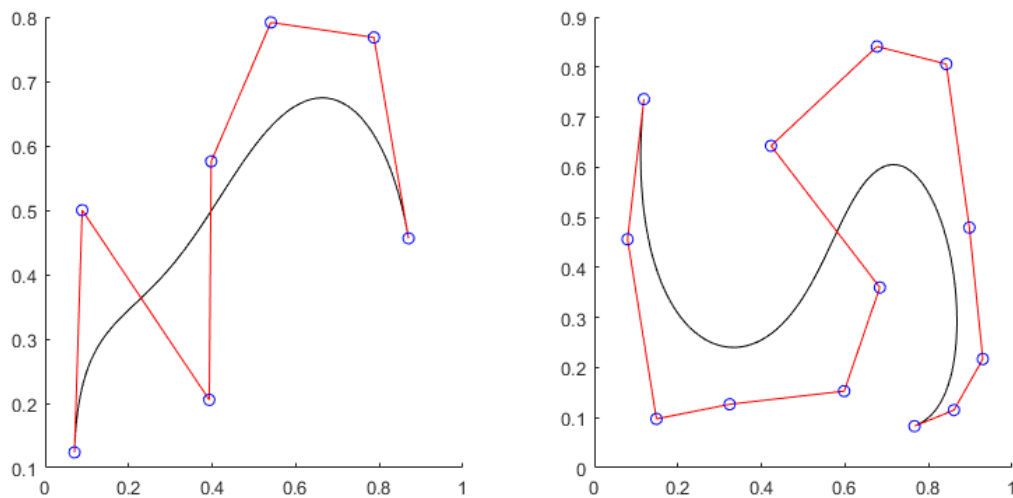


Figure 10. Examples of Bezier curves. Control points are shown as blue circles, with the control polyline in red, and the curve itself as black.



The Bernstein polynomial in Equation 4 represents a weight which is applied to a control point,  $P$ . The summation of these factors in Equation 3 therefore dictates the influence of each control point on the shape of the curve at a specific parameter value. Thus, the shape of the curve is determined by the effect of these polynomials on the control points.

The shape of the curve can be easily modified by moving the control points, as shown in Figure 11. However, moving any control point modifies the entire shape of the curve, i.e. all control points influence all sections (parameters) of the curve. For better shape manipulation, it is desirable to limit the effect of moving a control point to a local parameter span. Additionally, a large polynomial degree is required to fit a Bezier curve to complex shapes, which can cause undesirable wiggles. B-splines, discussed in the next section, solve these problems by using a knot vector.

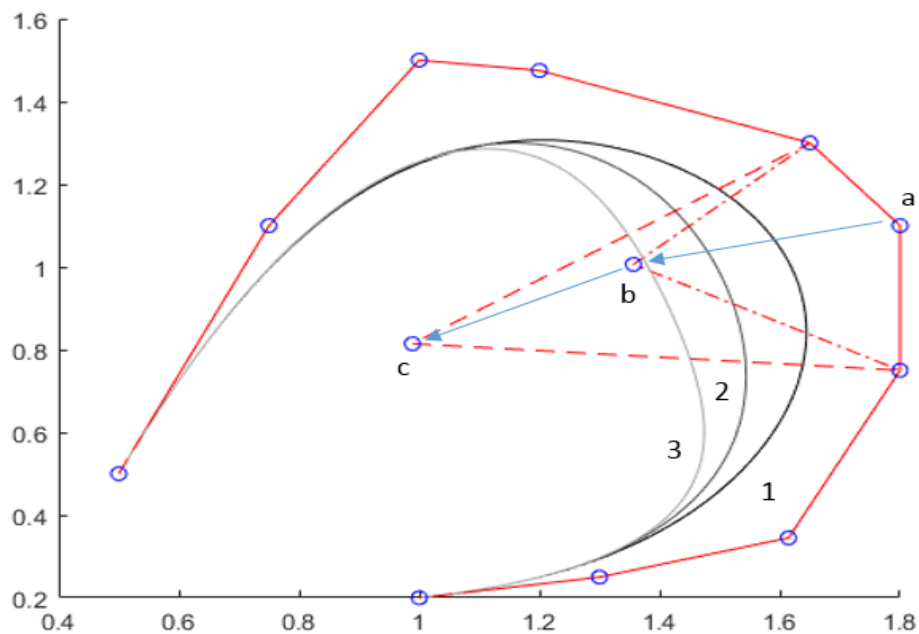


Figure 11. Effects of the motion of a single control points on Bezier curve. The control points are shown as blue circles, the control polyline as red lines. Control point a is moved to position b, modifying the shape of curve 1 into curve 2. Moving point b to point c further modifies the curve shape to make curve 3.

### 4.2.2 B-spline curves

Like Bezier curves, B-splines are parametric polynomial curves defined by a polynomial degree and a set of control points, with an additional feature: a knot vector. This is a vector of parameter values which limits the effect of a control point to a certain span by defining a piecewise polynomial curve and avoiding the problems present with Bezier curves. Equations 5-7 define a B-spline curve (Piegl & Tiller, 1997).

$$C(u) = \sum_{i=0}^n N_{i,p}(u)P_i \quad (5)$$

$$N_{i,0}(u) = \begin{cases} 1 & \text{if } k_i \leq u < k_{i+1} \\ 0 & \text{otherwise} \end{cases} \quad (6)$$

$$N_{i,p}(u) = \frac{u - k_i}{k_{i+p} - k_i} N_{i,p-1}(u) + \frac{k_{i+p+1} - u}{k_{i+p+1} - k_{i+1}} N_{i+1,p-1}(u) \quad (7)$$

In these equations,  $N$  is the B-spline coefficient, and  $k$  is a knot vector of length  $n+p+1$ , where  $n$  is the number of control points and  $p$  is the degree of the curve. By incorporating  $k$  into the equation for  $N$ , the influence of the  $i^{\text{th}}$  control point is limited to a span of  $[k_i, k_{i+p})$ . Knot values can be duplicated in a knot vector, pulling the curve more tightly towards a certain parameter value. This is useful in clamping the curve at its endpoints.

The value of  $N$  changes with degree, and is dependent on the previous values, i.e., it is recursive. This creates a series of varying weights for each control point, which again dictates its influence over a certain parameter span. Figure 12 shows the varying influence of ten control points over the entire span of a curve. This sample curve is plotted in Figure 13, which also shows the local effect of moving a single control point. GNURBS (Claxton, 2007) is used for curve and control point visualization in MATLAB.

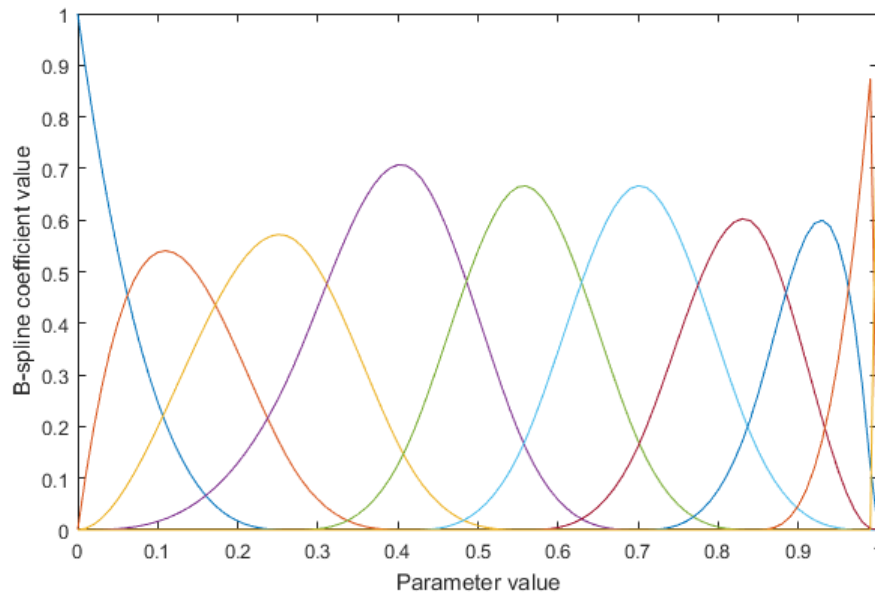


Figure 12. The variation of influence of each control point at each parameter value, using a uniformly spaced knot vector. Each control point is symbolized by a different curve in the plot. Note that the first and last control points have a value of 1 at the start and end of the curve. This is the effect of clamping the curve, where the start and end values are equal to the curve values.

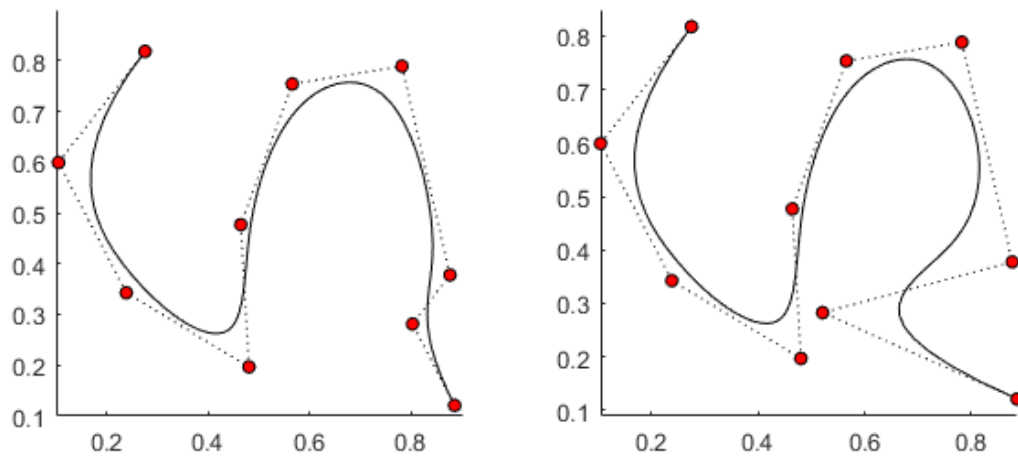


Figure 13. B-spline curve and control points in red. The motion of a single point only affects curve locally.

### 4.2.3 NURBS curves

Although very versatile, B-spline curves cannot represent conics (Piegl & Tiller, 1997).

To represent these shapes, the B-spline framework was generalized to the Non-Uniform Rational

B-Spline, or NURBS. This framework uses homogeneous coordinates instead of regular Cartesian coordinates, adding weight to the three Cartesian coordinates. NURBS are defined by Equations 8-9.

$$C(u) = \sum_{i=0}^n R_{i,p}(u)P_i \quad (8)$$

$$R_{i,p}(u) = \frac{N_{i,p}(u)w_i}{\sum_{j=0}^n N_{j,p}(u)w_j} \quad (9)$$

Here,  $N$  are the already familiar B-spline coefficients presented in Equations 6 and 7,  $P$  is the control point vector, and  $w$  is the B-spline coefficient weight vector, which also forms the fourth coordinate of a control point and turns the Cartesian coordinates into homogeneous coordinates. The weight is a scalar, which as the effect of rationalizing the B-spline basis function, producing  $R$ , known as the rational basis function. Another effect of incorporating the weights is that it allows modification of curve shape by changing the weight, rather than the control point. This makes B-spline curves are a special case of NURBS curves, where the weight  $w$  is 1.

#### 4.2.4 B-Spline and NURBS surfaces

The types of curves discussed above can be extended into surfaces by taking a bidirectional control point grid, the product of the univariate Bezier or B-spline functions, and two sets of knot vectors, if applicable. This means that Bezier surfaces (more commonly known as Bezier patches), B-spline surfaces, and NURBS surfaces can all be generated from the definitions in the previous sections. In fact, NURBS surfaces are a well-established standard employed in most CAD systems (Piegl, 1991). For this reason, they are used as the basis of all

surfaces and solids generated by ADT. Unlike curves, these surfaces are defined in a two-dimensional parametric space, with the parameters  $u$  and  $v$  taking values between 0 and 1 and combinations of these indicating the relative location of a point on a surface. Equation 10 describes a B-spline surface  $S$ . The surface is defined by two sets of degrees ( $p$  and  $q$ ), two sets of knot vectors  $k$ , (of lengths  $m+p+1$  and  $n+q+1$  for the  $u$  and  $v$  directions respectively), and an  $m$  by  $n$  set of control points  $P$ . The B-spline coefficients in the  $v$  direction are defined in the same manner as those in the  $u$  direction, by replacing the  $u$  parameter with the  $v$  parameter in Equation 8 with the corresponding knot vector. Figure 14 shows an example of a B-spline surface with a control point net and parametric directions.

$$S(u, v) = \sum_{i=0}^m \sum_{j=0}^n N_{i,p}(u) N_{j,q}(v) P_{i,j} \quad (10)$$

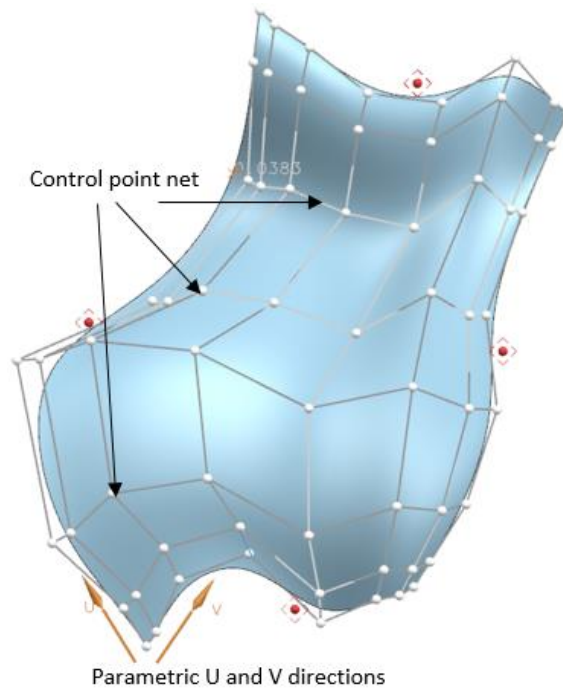


Figure 14. B-spline surface with control point net and parametric  $u$  and  $v$  directions. Red points indicate the edges of the surface.

As in the case of curves, the surface at each parametric point is defined by a combination of factors applied to each control point, which is a point in Euclidean space. These factors are limited to certain parameter spans (regions) by the knot vectors. Thus, a control point will have a decreasing, eventually null influence on an increasingly distant point of the surface. As shown in Figure 14 and discussed in previous sections, the control point net must be rectangular. This must also hold true for a surface which does not physically have four distinct edges, such as a circle, a triangle, or a human face. These edge conditions can be assigned to any Cartesian point, sometimes requiring experimentation to find the best fit. This causes some difficulties for generating surfaces which are not topologically homeomorphic to a rectangle, by either requiring an appropriate approximation or subdivision into several surfaces. While a triangular or circular surface can be generated by collapsing an edge for the former and dividing the edge into equal parts for the latter, more involved surfaces may require trimming.

Another problem is finding a good parameterization for fitting a B-spline surface. When fitting a B-spline surface through a set of data points or a mesh, each data point (node) must be associated with a set of parameters  $u$  and  $v$ . These indicate its relative position on the surface. Obtaining these values is known as parameterization. The parameterization of a mesh is a non-trivial task, especially for meshes which do not have clearly defined edges, corners, or regular internal mesh structures (for example, highly skewed meshes made of different element types). A bad parameterization can also generate a bad knot vector, which will negatively affect surface quality. These problems and their solutions are discussed in Chapter 5.

## Chapter 5: Geometry Reconstruction from Structural Mesh

With the basic geometric concepts used in this thesis discussed in the previous chapter, this chapter delves into the details of the process outlined in Chapter 3, namely, surface reconstruction. Surface reconstruction from meshes is a research topic in itself, and was one of the most challenging technical aspects of this work. Specifically, this chapter addresses the creation of B-spline surfaces from irregular triangular meshes, a task highly dependent on a good parameterization of the data, and crucial to ensure that the obtained geometry works in HFSS. The reader is reminded that the ADT core module consists of two submodules connected by a main submodule: a structural submodule and an antenna deformation submodule. This chapter focuses mainly on the structural submodule. The next chapter focuses on the antenna deformation submodule. Overall, the deformed structural model CAD construction process is outlined in Figure 15, and consists of the following steps:

- 1) Importing BDF and f06 data.
- 2) Creating a deformed mesh using the BDF and f06 node displacement data.
- 3) Sorting and grouping the elements of the BDF into surfaces to which they belong, which creates surface meshes.
- 4) Parameterizing and gridding the surface meshes. This step is necessary to apply the surface fitting algorithms discussed in this section, which require gridded data.
- 5) Reconstructing surfaces by fitting B-spline surfaces to gridded data, using either approximation or interpolation algorithms.
- 6) Optional thickening of faces into solids. This step creates solids from B-spline sheets, if prompted by the user.

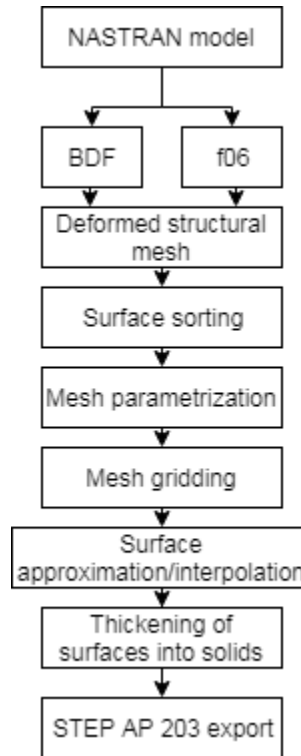


Figure 15. General process outline.

The approach for surface reconstruction presented here uses algorithms developed by Piegl and Tiller (1997) for gridded data. Although algorithms for fitting non-gridded data exist, these are usually more involved, may require smoothing functionals, and risk obtaining a local minimum solution instead of a global one (Krishnamurthy, 2000). Because general meshes resulting from an FEA model can have an arbitrary (non-gridded) structure, the meshes must be gridded first. Several methods for parameterizing and gridding the meshes were investigated, three of which were incorporated into ADT. The main method, k-harmonic mapping (Joshi, Meyer, DeRose, Green, & Sanocki, 2007; Jacobson, Tosun, Sorkine, & Zorin, 2010), is presented in this chapter. The other methods, gridded resampling (Krishnamurthy, 2000), and Least Squares Conformal Mapping (LSCM) (Levy, Petitjean, Ray, & Maillot, 2002) are summarized in Appendix A. The secondary methods serve mainly as a backup if the main method fails. Although k-harmonic is the default, the user is free to choose the desired



parameterization and gridding method for each surface. Generally this would only be done if the k-harmonic method fails to produce a usable mesh. In such a case, LSCM or gridded resampling might prove to be more stable. In the cases tested so far, k-harmonic has never failed to produce a usable mesh, while LSCM has failed for certain curved surfaces. Gridded resampling, as implemented by the author, suffers from slow performance and may require some tuning. It is noted that the original gridded resampling approach as implemented by Krishnamurthy (2000) does not suffer from these problems.

The rest of this chapter details the process in Figure 15, with the next sections discussing surface sorting, edge and corner recognition, mesh parameterization and resampling, and surface interpolation and approximation.

### **5.1 Mesh Importing, Surface Sorting, and Edge and Corner Recognition**

ADT creates an initial mesh by importing a BDF file, which contains all the information regarding elements, nodes, and element properties. This initial mesh can be converted to a displaced mesh by importing nodal displacements from an f06 file. Both the BDF and f06 are ASCII files. They are imported using code written in MATLAB, which uses the structure and markers present in both files to read them and generate sets of elements, each of which consists of a set of nodes. As discussed in Chapter 3, the elements are grouped into “regions” by the user in Patran. Regions are groups of elements which share the same properties (i. e. shell elements with a specific thickness value and material assignment). ADT treats each of these regions as separate units, and produces surfaces for each region. The process by which these surfaces are identified is explained later in this section. All elements which ADT can currently process have to be either triangular or quadrilateral shell elements. A purely triangular mesh is required to apply all the algorithms in this chapter, so any quad elements are split into two triangular

elements. In this context, this process is referred to as “triangulation”, but is completely unrelated to the problem of determining a location.

Geometry reconstruction begins with the organization of triangulated elements and their nodes into the surfaces to which they belong. This initial step is important to correctly identify the number and shape of the surfaces being fitted, which helps parameterize the meshes and ensure a surface is fitted correctly. Not quantifying metric is used for surface “correctness”, which refers to whether the surface grouping obtained by ADT is desirable to the user or not. This is mostly a qualitative assessment, discussed later in this section. Quantifying metrics for surface fitting precision are used, and are discussed in Chapter 7. The algorithm used to group surfaces is based on the dihedral angle between adjacent mesh faces and was developed by Alec Jacobson (2017). This algorithm works as follows:

- 1) A  $n$  by  $n$  dihedral adjacency matrix is created for the  $n$  elements belonging to a region. It contains the angle between each pair of adjacent elements. It is determined by subtracting the angle between the normal vectors of the different elements from 180 degrees -- the angle between two perfectly aligned elements. The normal vector is the normalized cross product of two adjacent edge vectors. Figure 16 shows an example, where red and yellow arrows indicate the normal vectors of elements belonging to different surfaces.

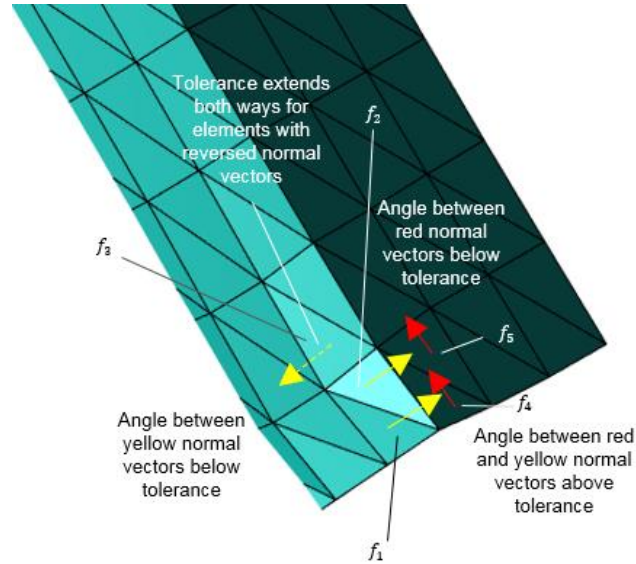


Figure 16. Angular adjacency matrix determination. Elements which share the normal vectors of the same color will be grouped together, since they are within tolerance. Sometimes adjacent elements have reversed normal vectors, as between  $f_2$  and  $f_3$ . This is accounted for by counting the tolerance both ways.

- 2) If the angle is below a user established tolerance, the adjacent elements are tagged as belonging to the same surface, generating an  $n$  by  $n$  matrix  $A$ , where a value of 1 for  $A_{ij}$  means that the elements  $i$  and  $j$  are adjacent and below the angular tolerance, and 0 means that the elements are not adjacent or the angle is not below the tolerance.

The algorithm groups adjacent elements within a user specified angular tolerance into the same surface. Figure 17 and Figure 18 below show some examples of the application of this algorithm using different tolerances on different surfaces.

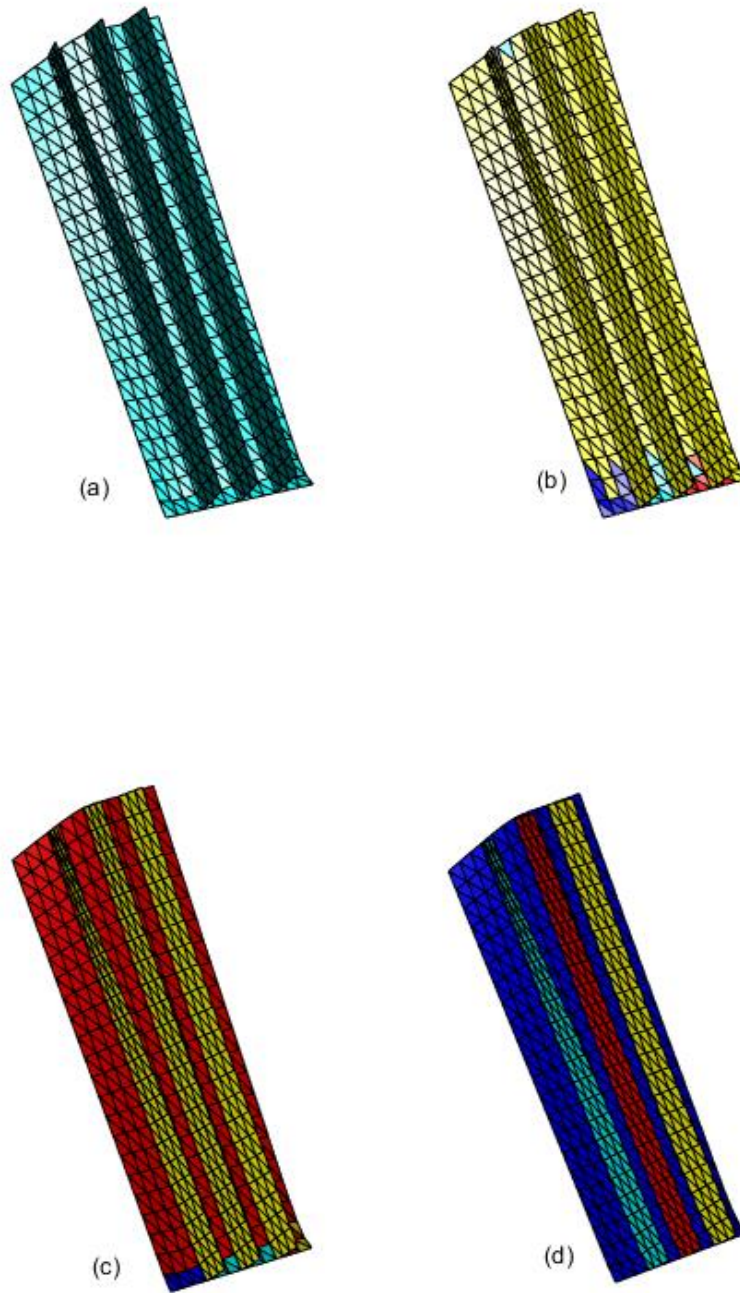


Figure 17. Surface recognition algorithm using different angular tolerances: (a) 0 degrees (original mesh), (b) 178 degrees, (c) 175 degrees, (d) 165 degrees. Adjacent elements of the same color are sorted to the same surface.

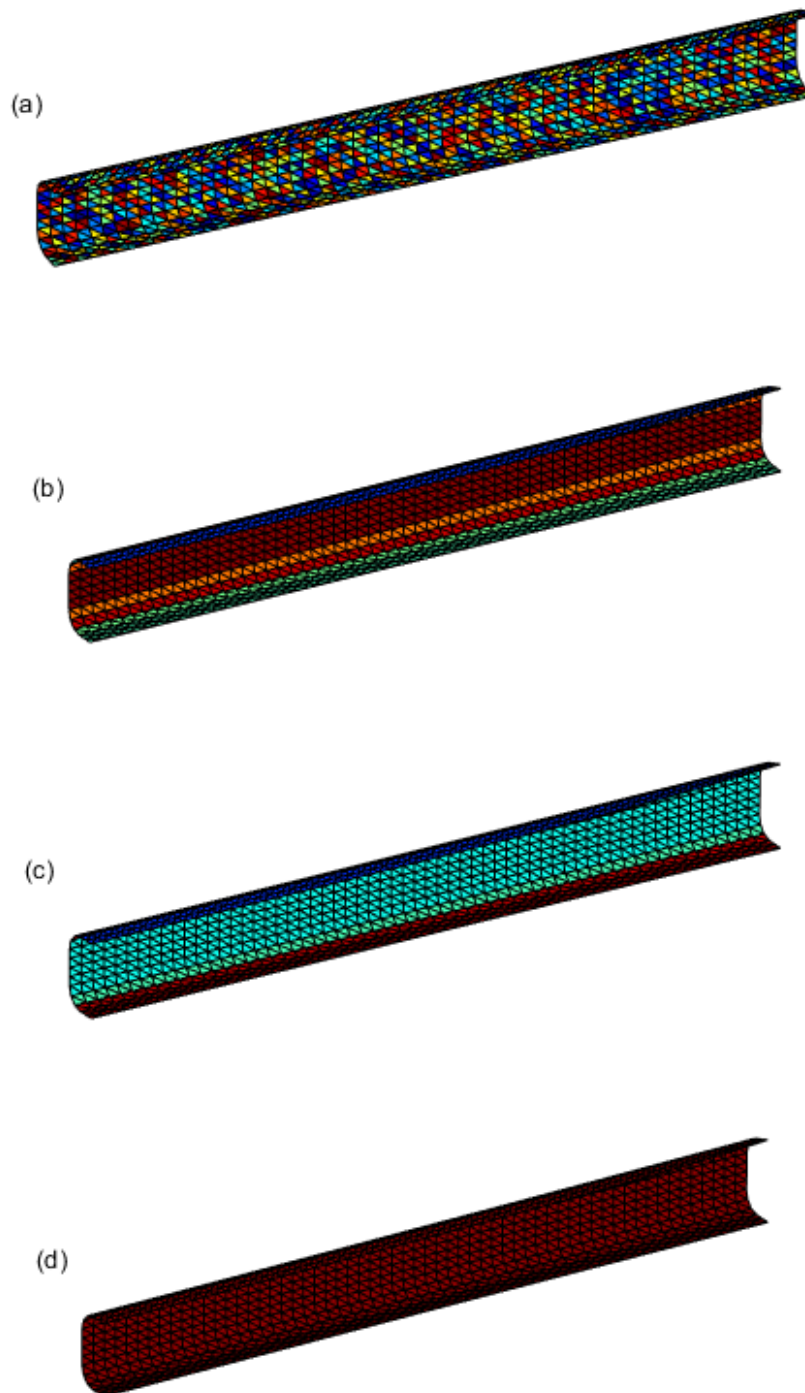


Figure 18. Surface sorting algorithm for a channel surface, using different angular tolerances: (a) 180 degrees (perfectly strict), (b) 170 degrees, (c) 160 degrees, (d) 135 degrees. Adjacent elements of the same color are sorted to the same surface.

As Figure 17 and Figure 18 show, the angular tolerance chosen for sorting the surfaces largely depends on the type of surface being fitted. Lower tolerances are better for surfaces which exhibit more curvature, like channels. If the tolerance is set too low, elements which do not belong in a surface may be grouped together. A tolerance of 135 degrees was found to work well for the surfaces tested – surfaces are grouped correctly, although this assessment is qualitative and depends on the user. A quantitative measure for surface “correctness” could be based on comparing the number of surfaces produced by ADT to the number of surfaces in the original CAD model used to build a FEM, although this type of assessment would not take into account the shapes of the surfaces obtained, nor user preferences. For example, in Figure 18, even though each channel originally consisted of three surfaces in the CAD used to generate the FEM, the user may find it desirable to have each channel modelled as a single surface. This can also occur for surfaces in the CAD model which have been separated to allow easier node equivalencing in NASTRAN, where the user may desire to group these surfaces back into one surface for final model generation. In most cases, obtaining the best surface “correctness” would be a function of grouping all elements belonging to a region into the minimum number of surfaces while maintaining surface error (for the fitted surfaces) below a certain threshold. This threshold can be based on the numbers provided in Section 1.2. In the future, adjustments may be made to individually tailor the dihedral angle tolerances for each surface and allow special features to be modelled as single surfaces (such as L-brackets).

After sorting, surface boundaries and corners are recognized. Surface boundaries are recognized by identifying element edges in a surface group which are only shared by a single element, since two elements share internal edges. Figure 19 shows a mesh with edge elements highlighted.

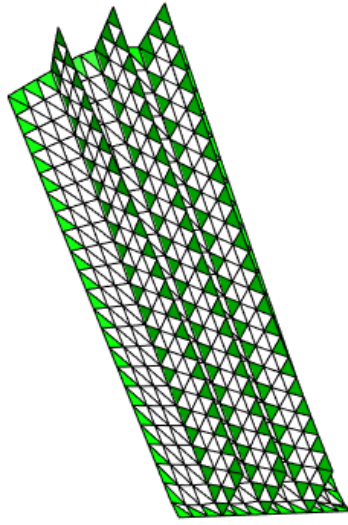


Figure 19. Mesh example with edge elements identified and shaded as green patches.

Corners are recognized by measuring the angle between consecutive surface boundary element edges, which is measured as the angle between the vectors defining consecutive element edges. If the angle between successive edges is in a user-specified range, then the node at which this angle occurs is determined to be a corner. Any surface processed by ADT must have exactly four corners. This is necessary for proper mesh gridding and surface fitting, and stems from the fact that a B-spline surface, as defined in parametric space, is always a square. Thus, to properly implement a fitting algorithm to gridded data, the data must be gridded with respect to four edges in Euclidean space. Surfaces which do not naturally have four corners (such as a circular membrane) must be assigned corners. This is further explained below, and in the next sections. Generally, a tolerance of 60-120 degrees is used (if an angle is 30 degrees to either side of 90 degrees, it is labelled as a corner). Too loose or too tight a tolerance can hinder proper corner recognition, as Figure 20 and Figure 21 show.

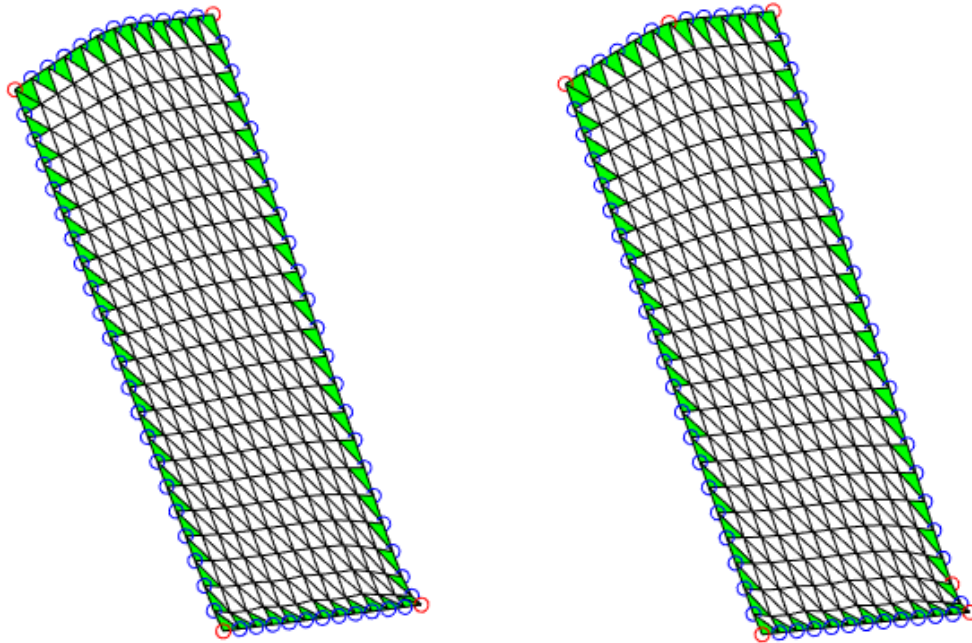


Figure 20. Corner recognition. Corners shown as red circles, and other edge nodes as blue circles. On the left, four corners are detected using a tolerance of  $\pm 30$  degrees. On the right, six corners detected using a very loose tolerance of  $\pm 85$  degrees.

If there are no clear corners (such as in the case of a circular boundary), then the corners are taken to be evenly spaced. Using the regular tolerance value in such a case (60-120 degrees) might yield no corners at all, and the sensitivity is very high. For example, a tolerance of  $\pm 85$  degrees (5-175 degrees) yields no corners at all, but one of  $\pm 90$  (0-180 degrees) yields every node as a corner, as Figure 21 shows. The corner assignment process is automated, and ADT can recognize circular boundaries, and properly assign corners to them.





Figure 21. Corner (red circles) detection on a circular boundary (blue circles). Because the angle between adjacent element edges is small on the boundary, no corners are detected unless the tolerance is very loose, when almost every corner is detected (right). Thus, corners are taken as four evenly spaced nodes (left) automatically by ADT.

As mentioned before, any mesh being processed must have exactly four corners, which includes surfaces that do not naturally have four corners, as the one in Figure 21. This helps ensure a good parameterization and mesh gridding, which is explained in the next section.

## 5.2 Mesh Parameterization

Mesh parametrization is the problem of finding a one-to-one mapping from one domain to another, or generating parameter values for points making up a mesh. It can be thought of as the problem of unravelling, unrolling, or flattening of a mesh, presented visually in Figure 22. It is an important problem in many applications, most notably texture mapping, which is the mapping of images onto meshes to add detail for computer visualization applications, and it is also important in surface fitting. In the specific case studied here, a one-to-one mapping from the mesh in Euclidean space to the parametric domain of a B-spline surface is sought, that is, a mapping to obtain the normalized (parametric)  $u$  and  $v$  coordinates for each data point being fitted. This information is required to be able to approximate or interpolate a B-spline surface, and is used to compute the B-spline coefficients.

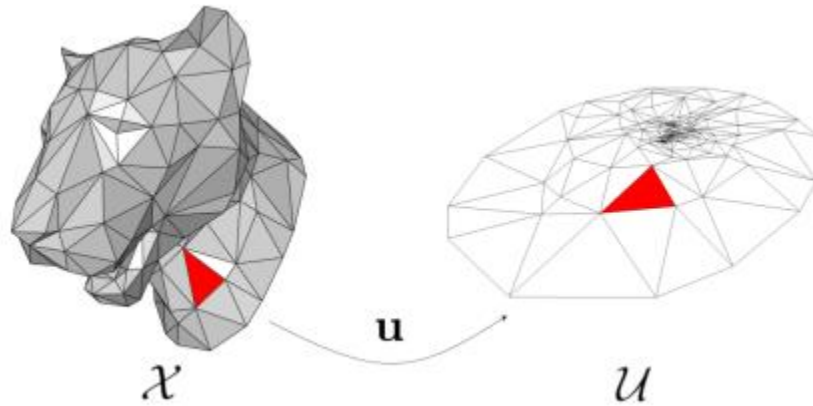


Figure 22. Mesh parameterization, a linear piecewise map from a domain  $X$  to a domain  $U$ . (Levy, Petitjean, Ray, & Maillot, 2002)

ADT incorporates three parameterization approaches: gridded resampling, Least Squares Conformal Mapping (LSCM), and k-harmonic Mapping. K-harmonic mapping is used as the default parameterization method, with LSCM and gridded resampling as back-ups. K-harmonic mapping and LSCM offer the advantage of higher speed and more accurate mesh grids, better suited for mesh interpolation. Gridded resampling is slower (in the author's implementation) but more robust than LSCM. K-harmonic gridding is currently the best performing approach, but the original intent of the code adapted for ADT is for character animation in computer graphics, so it may not be always appropriate. No failing cases have been found for k-harmonic parameterization so far. To introduce the problem of mesh parametrization, the simpler case of a curve parameterization is presented first.

### 5.2.1 Curve parameterization

Curve parametrization is the simpler problem of assigning parameter values to points lying on a curve. If an interpolating curve closely follows the polyline being interpolated, the length of this curve can be approximated by obtaining the chord length of the polyline, as shown in Figure 23. The parameter value of each point can be obtained as the summation of the polyline segments up to that point, divided by the total length, as in Equation 11, where  $u_m$  is the

parameter value of the point  $m$ , and there are  $n$  total points in the curve. Known as the chord length method, this method is fast, reliable, and simple, and is generally used in parameterization problems.

$$u_m = \frac{\sum_{i=1}^m |P_m - P_{m-1}|}{\sum_{i=1}^n |P_n - P_{n-1}|} \quad (11)$$

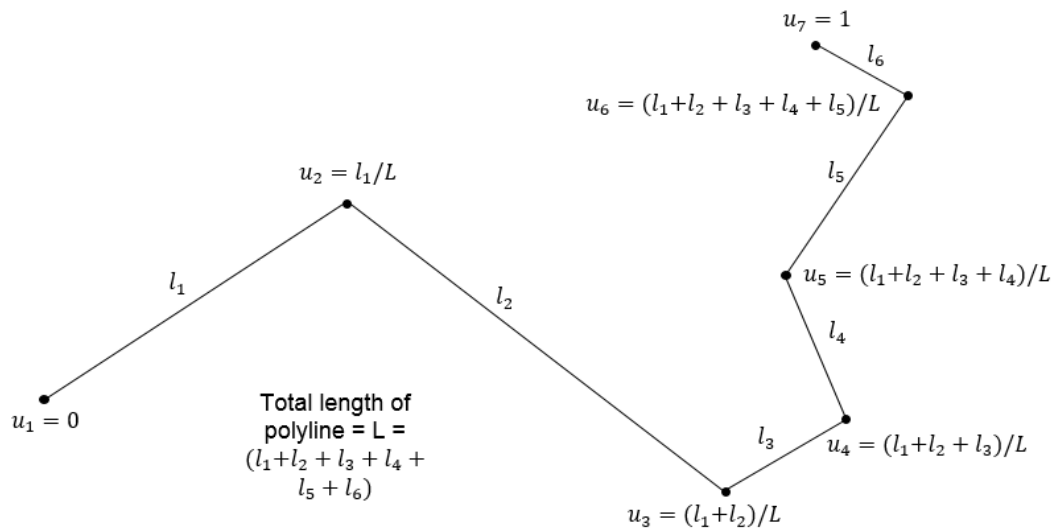


Figure 23. Chord length parameterization.

Figure 24 shows a comparison between different methods used to parametrize the same set of data points (shown as blue points). The uniformly spaced method shows undesirable wiggles in the curve, while the centripetal method provides the smoothest interpolation of all. However, because it is a special case of the centripetal method, and because it generally shows good results without oversmoothing, the chord length method is used in this project.

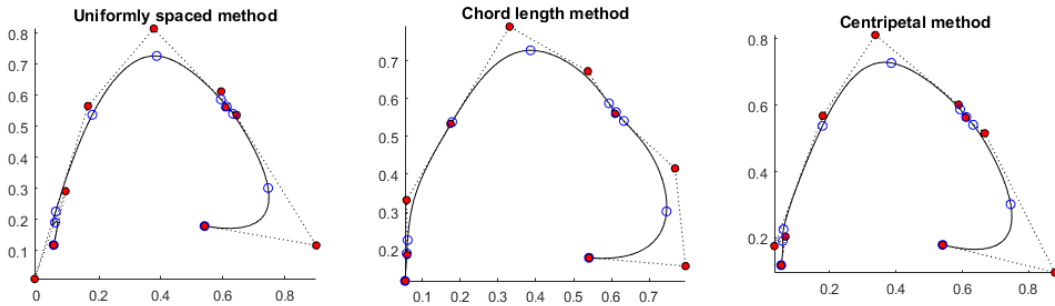


Figure 24. Curves obtained through interpolation of different parametrizations of the same data. Data points are shown as blue circles, control points and the control mesh is shown as red points with dotted lines.

To generate valid B-splines, a knot vector is needed. Knot vector generation is discussed in the next section.

### 5.2.2 Curve knot vector estimation

The knot vector defines the parametric span over which control points have an influence. A commonly used approach to determine a knot vector from parameter values is the averaging knot vector. It is shown in Equations 12 - 14, where intermediate knot values are determined by averaging the knots over a span of parameters of length  $p$ , which is the degree of the curve.

$$k_0 = k_1 = \dots = k_p = 0 \quad (12)$$

$$k_{j+p} = \frac{1}{p} \sum_{i=j}^{j+p-1} u_i, j = 1, 2, \dots, n-p \quad (13)$$

$$k_{m-p} = k_{m-p+1} = \dots = k_m = 1 \quad (14)$$

In Equation 12-14,  $k_i$  is the  $i^{\text{th}}$  knot parameter value,  $p$  is the degree of the curve,  $m$  is the length of the knot vector, and is equal to  $n+p+1$ . To obtain control points from data points using interpolation or approximation, the parameters assigned to the points must belong to certain parameter spans, i.e. there cannot be empty knot spans, or the equation system becomes singular.

Equations 12-14 only guarantee non-empty spans for curve interpolation. A different approach, in Equations 15-18 (Piegl & Tiller, 1997), is used to generate knot vectors for surface approximation, since the number of control points desired becomes an input. In this equation,  $m$  is the number of parameter values,  $n$  is the number of control points of the curve, and  $p$  is the degree of the curve being approximated. This guarantees that every parameter will belong to a knot span.

$$d = \frac{m}{n - p} \quad (15)$$

$$i = \lfloor jd \rfloor, \quad j = 2, \dots, n - p \quad (16)$$

$$a = jd - i \quad (17)$$

$$k_{p+j} = (1 - a)u_{i-1} + au_i \quad (18)$$

### 5.2.3 Gridded surface parametrization

The curve parameterization concepts can be easily extended to a surface if the surface has an equal number of points across the parametric  $u$  and  $v$  directions, meaning that it is gridded. In this case, curve parameterization can be applied once for each iso-curve on which the point lies. Like curve parameterization, this method is quick and reliable, but it requires a gridded mesh as an input. As Figure 25 shows, there can be ambiguity as to which polyline can be considered an iso-curve for a non-gridded mesh, and some points, like the red point, are ambiguous in that they do not appear to lie on any discrete iso-curve.

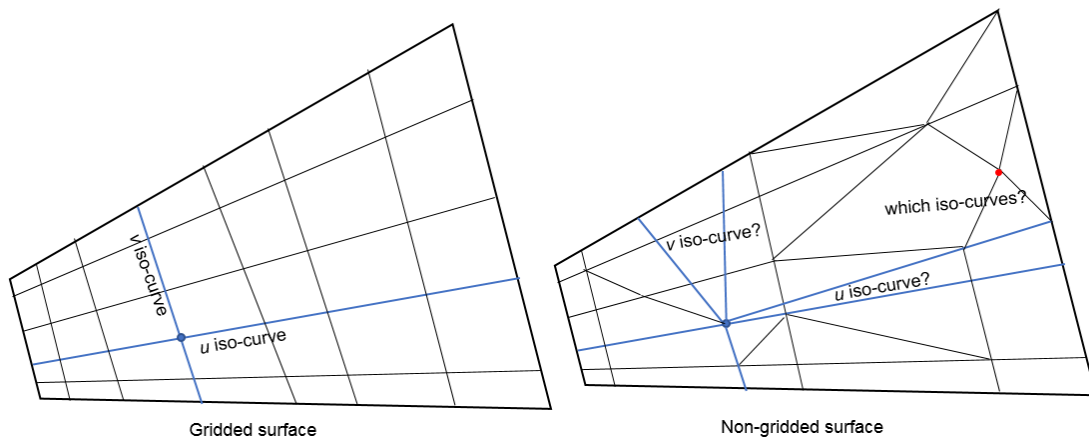


Figure 25. Gridded vs. non-gridded mesh parameterization. Blue point parameter values are easily obtained by applying curve parameterization to  $u$  and  $v$  iso-curves separately in gridded space, but not in non-gridded space.

An averaged knot vector across a set of iso-curves is used for each of the parametric directions since a B-spline surface is defined by only two knot vectors. This further requires a good, preferably gridded, mesh parameterization. Too large of a variation in knot vectors across a single parametric direction could make for a poor surface reconstruction, or even a singular system of equations. To apply this parametrization approach meshes from the BDF must be gridded first as they are not guaranteed to be so, as shown in Figure 26, which shows different meshes generated for the same surface in Patran. Surface reconstruction from a gridded mesh is straightforward and is explained in Section 5.3

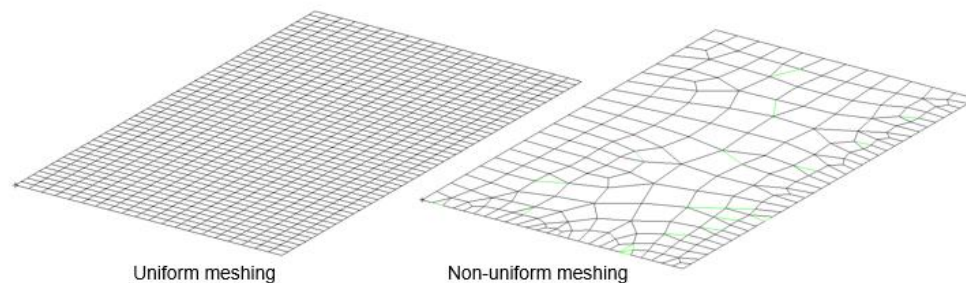


Figure 26. Uniform and non-uniform mesh generated in PATRAN for same surface

A good gridded mesh (suitable for surface reconstruction) should have three main characteristics (Krishnamurthy, 2000):

- 1) Arc length uniformity: the spacing along a particular iso-curve should be uniform
- 2) Aspect ratio uniformity: the spacing along the  $u$  and  $v$  iso-curves should be the same
- 3) Parametric fairness: every  $u$  and  $v$  iso-curve should be of the minimum length while complying with the first two conditions.

Figure 27 shows the iso-curves of a simple surface, as seen from above, both with and without parameterization to the above standards. Iso-curves are curves on the surface where one of the parameter values ( $u$  or  $v$ ) is kept constant, while the other varies.

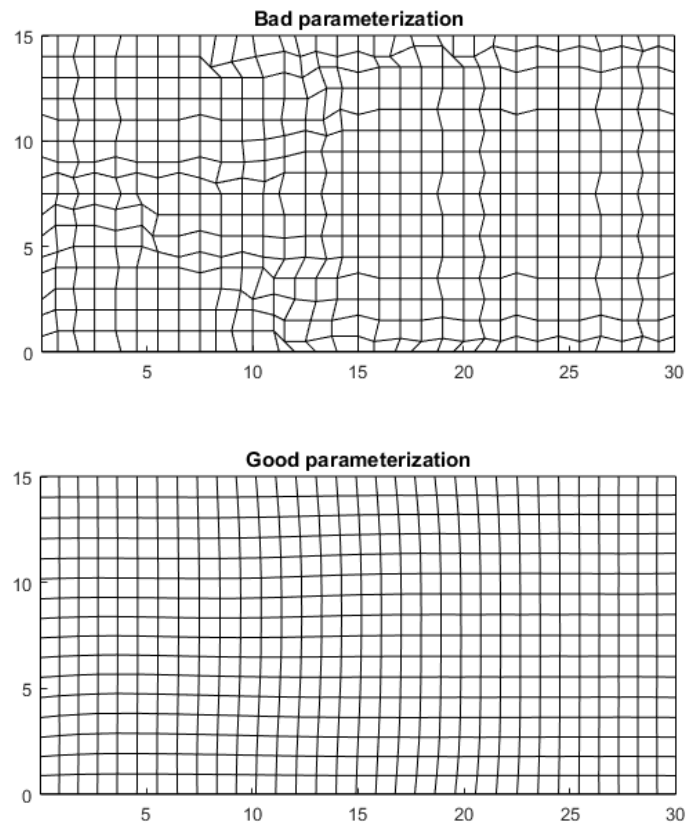


Figure 27. Two different parameterizations for the same surface, showing iso-curves.

A bad parameterization can lead to the construction of an unfair surface (a surface which has undesirable wiggles and bends), as shown in Figure 28. An unfair reconstruction not only misrepresents the true surface shape, but can induce meshing problems and cause non-existing surface intersections, which is detrimental to the HFSS solver. Figure 28 shows the same mesh using a good and bad parameterization, using both approximation and interpolation algorithms to reconstruct it. Table 3 shows values for the good and bad parameterizations used in Figure 28.

Table 3. Good and bad parameterization parameters.

<b>Parameter</b>	<b>Good parameterization</b>	<b>Bad parameterization</b>
Average aspect ratio, $A$ ( $\sim$ )	1.00	1.39
Standard deviation of $A$ ( $\sim$ )	0.00	0.52
Average $u$ iso-line length (in)	120.12	121.71
$u$ iso-line length standard deviation (in)	0.00	2.08
Average $u$ iso-line spacing (in)	0.70	0.5
Average $v$ iso-line length (in)	7.63	7.95
$v$ iso-line length standard deviation (in)	0.00	0.41
Average $v$ iso-line spacing (in)	0.69	0.49

The true length of the isolines in the original surface is 120.12 inches in the  $u$  direction and 7.71 in the  $v$  direction. From Table 3 it is apparent that a good parameterization approaches the true length of the iso-lines more closely (the difference is due to the discretization of the surface) has an element aspect ratio of close to 1 with a small standard deviation, and close to equal spacing along iso-lines with a small standard deviation as well.



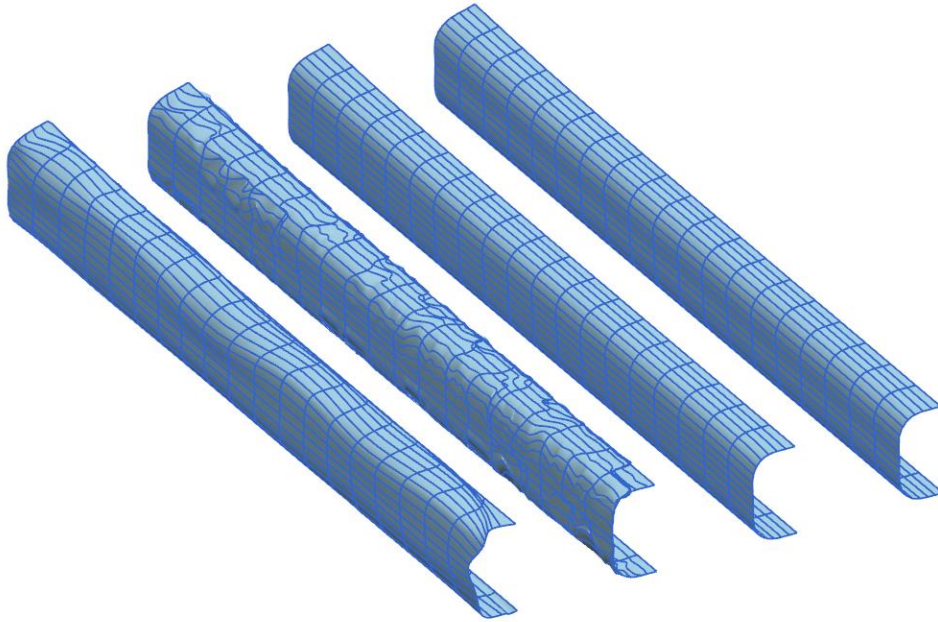


Figure 28. Surface reconstructions using different parametrizations of the same mesh, showing isolines. Left to right: approximation result using a 10 by 10 control net, interpolation using same bad parametrization, approximation using good parametrization, interpolation using good parameterization.

The next section explains the mesh parameterization and gridding approach used in ADT.

#### 5.2.4 K-harmonic mapping

Introduced by Joshi et al (2007), k-harmonic mapping was developed to naturally create and control volume deformations for character articulation (Joshi, Meyer, DeRose, Green, & Sanocki, 2007), and for vibrational surface modelling (Jacobson, Tosun, Sorkine, & Zorin, 2010). It is a harmonic mapping, where the  $k$  is the degree of the Laplacian. ADT uses the default a  $k$  value of 1. This type of mapping solves for weights corresponding to each node on a mesh. It is used in conjunction with a control cage, which encloses a mesh, and can produce deformed meshes by deforming the control cage. The control cage is used to solve for boundary conditions, which are applied to the mesh and are used to solve for the deformed mesh weights.

This mapping can be applied to parametrize a surface, if boundary conditions are not derived from a control cage, but are assigned. Boundary conditions are assigned by identifying each node

on the surface boundary found during the surface sorting step (Section 5.1) with a pair of  $u$  and  $v$  parameter values (see Figure 30). The parameter values themselves are obtained by applying the chord length method (Section 5.2.1) to each surface boundary edge curve, knowing that each of the edge nodes will either lie on the  $u = 0$ ,  $u = 1$ ,  $v = 0$ , or  $v = 1$  iso-curve. The implementation of this method used by ADT is the one developed by Jacobson (2017).

A gridded mesh is obtained by inverse piecewise linear mapping from parametric to Euclidean space. This mapping is obtained by using barycentric coordinates. Barycentric coordinates, shown in Figure 29, are triplets of numbers (or weights)  $t_1$ ,  $t_2$ , and  $t_3$  that are applied at vertices of a triangle and determine the position of a point  $P$  (Weisstein, 2018) with respect to the triangle vertices (see Equation 19).

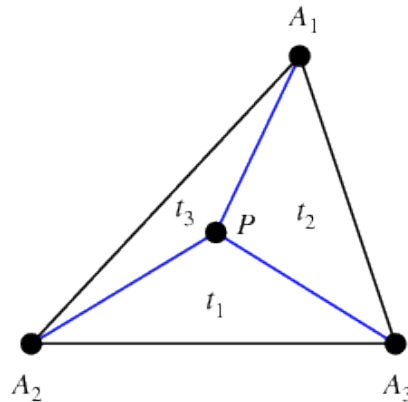


Figure 29. Barycentric coordinates of point  $P$ ,  $t_1$ ,  $t_2$  and  $t_3$  (Weisstein, 2018).

If the conversion from parametric coordinates  $u$ ,  $v$  to barycentric coordinates  $t_1$ ,  $t_2$ , and  $t_3$  is symbolized by  $B$ , then Equation 20 can be used to obtain the Euclidean coordinates of  $P$ , where  $V_1$ ,  $V_2$  and  $V_3$  are Euclidean vertex coordinates of the triangle point  $P$  is inside of.

$$B(u, v) = (t_1, t_2, t_3) \quad (19)$$

$$\sum_{i=1}^3 V_i t_i = P \quad (20)$$

To obtain a gridded mesh, a grid was generated in parametric space and each its points was converted to barycentric coordinates using Equation 19. Equation 20 was used to map the barycentric coordinates into Euclidean space. Figure 30 show examples of the parameterization and gridding approach, and Figure 31 shows this parameterization applied to a circular surface.

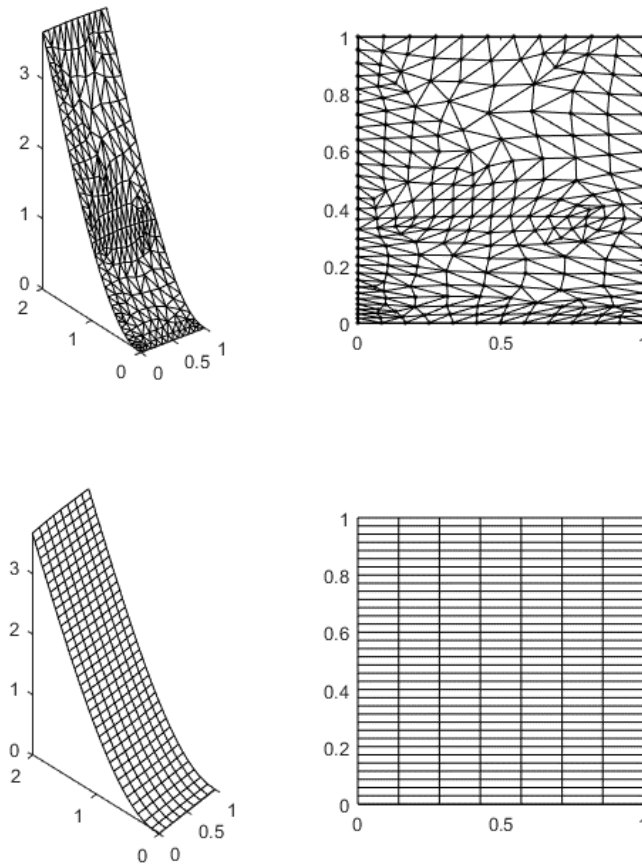


Figure 30. Mesh parameterization and gridding by k-Harmonic mapping. Clockwise from upper left: original mesh in Euclidean space, mesh parameterization in parametric space, uniform grid in parametric space, mapping of uniform grid onto mesh in Euclidean space.

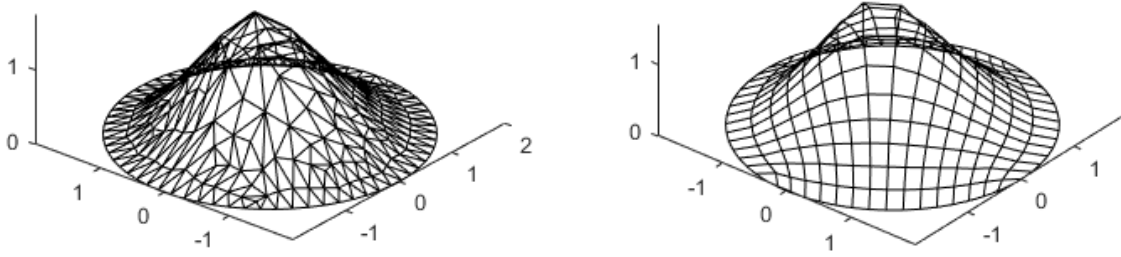


Figure 31. Mesh gridding of a circular object by k-Harmonic mapping.

### 5.3 Curve and Surface Interpolation and Approximation

After a gridded mesh has been obtained, B-spline surfaces can be fitted by applying global curve interpolation or approximation algorithms to each iso-curve. This section explains the process of curve interpolation or approximation, and its extension to surfaces.

#### 5.3.1 Global curve interpolation

Global curve interpolation refers to obtaining a B-spline curve which passes through a set of data points, without using information on local derivatives at these points (which is known as local interpolation). Let  $D$  be a set of  $n$  data points,  $P$  a set of  $n$  control points,  $N$  the B-spline basis functions,  $u_m$  the parameter value of the  $m^{\text{th}}$  data point. Equation 21 defines data point  $D_m$  as the point on curve  $C$  evaluated at parameter  $u_m$ . In it,  $u_m$ ,  $D_m$ ,  $N$  are known, and the control points  $P$  are unknown. The knot vectors used in  $N$  are also known, determined using the methods in Section 5.2.2

$$D_m = C(u_m) = \sum_{i=0}^n N_{i,p}(u_m)P_i, \quad 0 \leq m \leq n \quad (21)$$

The above equation can be set up as a system of linear equations, or matrix multiplication. Three matrices,  $N$ ,  $D$  and  $P$ , are constructed and defined below. The matrix  $N$  contains the B-spline coefficient values for each data points in each row (see Equation 7 in

Section 4.2.2), which are evaluated using the  $u$  parameter values obtained by for each point the curve, by using the methods in the previous section (5.2). Each row in matrix  $D$  is a data point, in Euclidean form, where  $d_{n1}$ ,  $d_{n2}$ , and  $d_{n3}$  are  $x$ ,  $y$  and  $z$  coordinates of the  $n^{\text{th}}$  point. Each row in matrix  $P$  is a control point, also in Euclidean form, where  $p_{n1}$ ,  $p_{n2}$ , and  $p_{n3}$  are  $x$ ,  $y$  and  $z$  coordinates of the  $n^{\text{th}}$  control point.

$$N = \begin{bmatrix} N_{0,p}(u_0) & N_{1,p}(u_0) & \cdots & N_{n,p}(u_0) \\ N_{0,p}(u_1) & N_{1,p}(u_1) & \cdots & N_{n,p}(u_1) \\ \vdots & \vdots & \ddots & \vdots \\ N_{0,p}(u_n) & N_{1,p}(u_n) & \cdots & N_{n,p}(u_n) \end{bmatrix} \quad D = \begin{bmatrix} d_{01} & d_{02} & d_{03} \\ d_{11} & d_{12} & d_{13} \\ \vdots & \vdots & \vdots \\ d_{n1} & d_{n2} & d_{n3} \end{bmatrix} \quad P = \begin{bmatrix} p_{01} & p_{02} & p_{03} \\ p_{11} & p_{12} & p_{13} \\ \vdots & \vdots & \vdots \\ p_{n1} & p_{n2} & p_{n3} \end{bmatrix}$$

With these matrices set up, the matrices  $N$ ,  $D$  and  $P$  can be expressed as Equation 22, and solved for  $P$ , the control points, using Equation 23 (Shene, 2011).

$$D = NP \quad (22)$$

$$P = N^{-1}D \quad (23)$$

### 5.3.2 Global curve approximation

Approximation is a more general process than interpolation, since the number of control points must be specified along with the inputs. The maximum error is controlled by the number of control points used – more control points reduce the error between the mesh and the surface approximation. However, too many control points increase the file size and also may introduce undesirable wiggles into the geometry. For more involved algorithms, a maximum error can be specified, which results in an iterative process, where the number of control points is varied until the desired tolerance is reached. ADT does not incorporate this sort of iteration, which may be added in the future. ADT uses a least squares fitting approach to fit a curve to a specified number of control points without verifying error. This approach is summarized by Equation 24. This is a simplified version of the approximation equation given in Piegl (1997), where point weights and

constraints are not considered. Adding weights and constraints can improve surface quality, but the existing simplified method is deemed sufficient for current work; the task of increasing the complexity of the fitting algorithm is left for the future. The solution to Equation 24 is Equation 25. In these equations,  $P$  and  $D$  have a different number of rows, as the number of control points is set by the user, instead of being equal to the amount of data points.

$$(N^T N)P = N^T D \quad (24)$$

$$P = (N^T N)^{-1} N^T D \quad (25)$$

The algorithm from Piegl (1997) is adapted to implement Equation 25 in ADT, along with the approximation knot vector estimation algorithm from Section 5.2.2, using the data points, parameterization, and number of desired control points as inputs.

### 5.3.3 Global surface interpolation, approximation and thickening into solids

Surface interpolation and approximation of gridded data is simply an extension of the curve fitting algorithms. To approximate or interpolate a surface, the curve-fitting algorithm is applied first to each iso-curve in one direction, for example, the  $u$  direction. After this, the output of the first pass is used as the input for the second pass, across every curve in the other iso-direction, for example,  $v$ . This produces a control point matrix, which can then be used to define the surface being reconstructed. An averaged knot vector across all iso-curves in the  $u$  direction and another across all iso-curves in the  $v$  direction are used as the two knot vectors defining the surface. Figure 32 shows this process.

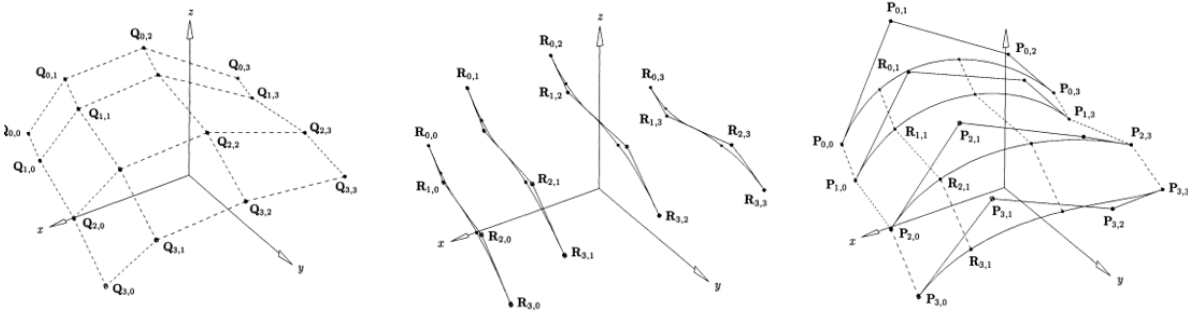


Figure 32. Gridded mesh fitting. Left to right: original mesh, interpolating the  $u$ -directional data, interpolating the control points of the  $u$ -direction interpolants in the  $v$  direction, Control points  $P$  are obtained (Piegl & Tiller, 1997).

Once surfaces are obtained, they can be thickened to become solids for HFSS analysis.

This is important as the thickness of a structure may affect EM performance, but most surfaces in the structural FEM are modelled as thin sheets in two dimensions. To convert the sheets to solids, first an offset surface is created from an interpolated or approximated surface. The thickened surface is created by offsetting the control points of the original surface using local normal vectors; the magnitude of these vectors is given by the region thickness value in the BDF (stored in the base file), but can be overwritten by the user. Each local normal vector is obtained using the averaged cross product of the control point mesh. The direction of the local normals (what side of the original surface they are on) is determined by the user. During CAD export, the original and thickened surface information is exported in the TTF. Figure 33 explains the offset process.

Next, the CAD export module constructs the original and offset surfaces, and connects their corresponding edge curves using connecting surfaces along the offset direction. All surfaces are sewn together in the CAD module into a solid, which is the final product of the CAD export. Figure 34 summarizes this process.

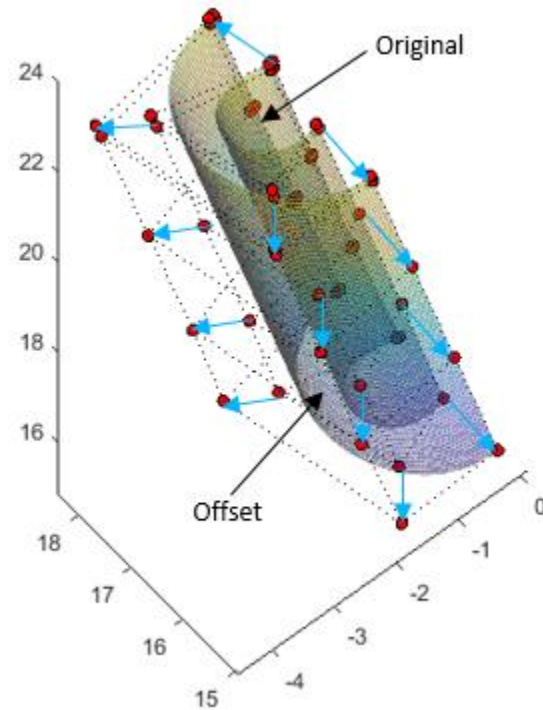


Figure 33. Original and thickened surfaces. Thickened surface obtained by moving the control points in the directions indicated by blue arrows, which are local normals (Claxton, 2007).

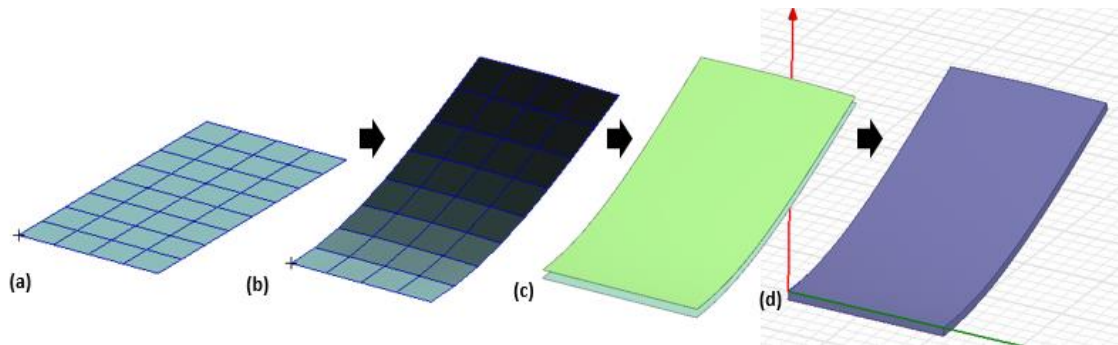


Figure 34. Process of transformation of a single surface from (a) a mesh to (b) a deformed mesh then (c) B-spline surface with offset, and (d) solid created from a series of connected surfaces and imported into HFSS.

Conversion from a surface to a solid is optional, and can be controlled by the user. Once the final CAD geometry is obtained and exported, it can be used in HFSS for simulations. The next chapter explains how antenna geometry is created, imported, deformed, and exported in ADT.



## Chapter 6: Antenna Deformation Module

Several software and analytical approaches for determining the effects of antenna deformation on performance were summarized in Chapter 2 of this thesis. Although many of these approaches and programs can indeed reconstruct geometry from meshes, none of them quite allow for the generality and freedom to voluntarily decouple and combine antenna and structural design the way ADT does. Analytical solutions generally do not include coupling between array elements. In addition, most 3D physics-based software require modelling the antenna as part of the structural simulation, and do not always allow control over the surfaces generated from meshes.

The previous chapter detailed the solution to recreating deformed surfaces from a deformed structural mesh. This chapter explains how ADT defines and interprets, and deforms antenna geometry (which is not limited to basic shapes), and also how ADT allows decoupling structural geometry from the antenna geometry. Concepts from the previous chapter and a good understanding of the structural module are useful to understand the processes in this chapter better.

The reader is reminded that ADT consists of a core module and a CAD export module. The core module is made of a structural and antenna deformation submodules, connected by the main submodule. The structural submodule was covered in the previous chapter. This chapter covers one of the main features of ADT: the antenna deformation submodule. This module can deform an antenna using a deformation field obtained from the structural submodule or from the user, and can accept a large variety of antenna designs. ADT does not require the antenna to be included in the structural FEM, and can treat antenna geometry separately from the structural geometry, combining them in the TTF and CAD export module if requested. The antenna

deformation submodule enables testing the effect of deformation on antenna performance using full-wave analysis, and it also allows designing conformal antennas. The overall process followed by this submodule is depicted in Figure 35 and Figure 36. The types of antennas ADT is designed to work with (and has been tested on) are dipoles, which generally consist of two conductors connected by a port. The width and length of conductor surfaces tested is usually much greater than their thickness, which means that the thickness can be ignored and the conductors can be modelled as 2D sheets in some cases. Although ADT could deform any CAD model made of supported entities (for a list of these, see Appendix C), it has not currently been tested with any other types of antennas. The specific antenna designs tested are presented in Chapter 7.

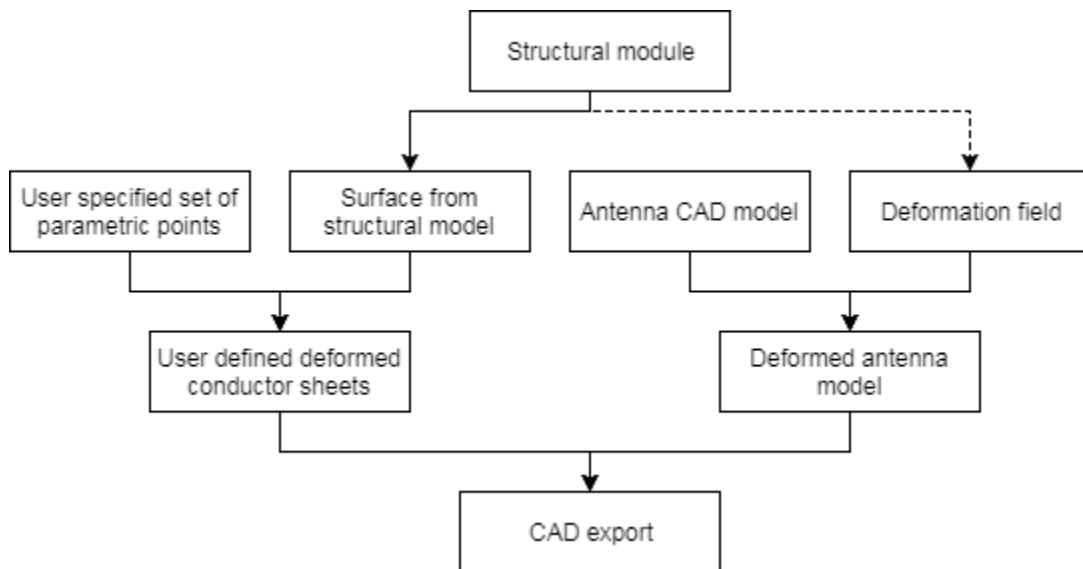


Figure 35. ADT Antenna deformation model process overview. The dashed line indicates optional input, as the deformation field can also be user defined.

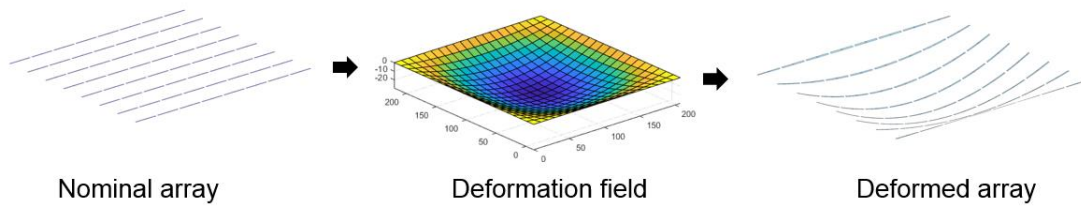


Figure 36. Visualization of the antenna deformation process using a user-specified deformation field for a dipole array.

As Figure 35 shows, ADT can define antennas using two main approaches, which are detailed in the next sections. The first approach uses a set of parametric  $(u, v)$  coordinates to define a sheet antenna from existing structural surfaces. These coordinates define a closed contour on a chosen surface, and are used to trim the surface to generate antenna conductor sheets. For example, a user can define a set of conductor surfaces on a substrate which was modelled in the structural model. When exported, the CAD model will consist of the deformed structural model and the conductor surfaces defined from the substrate surfaces. This approach can only be used to generate sheet geometry (2D) antennas, and it assumes that antenna conductors have the exact same deformation as the surface they are defined on. This approach does not allow the user to define a custom deformation field, but uses a structural surface as the deformation field. This approach is explained in Section 6.1.

A second approach involves importing an antenna CAD model, associating it to a deformation field, and deforming it accordingly. This deformation field can be extrapolated from the structural model, or it can be provided by the user. Unlike in the first approach, this deformation field can be 2D (a surface) or 3D (a solid). This approach is explained in Sections 6.2 and 6.3.

Two approaches were developed to allow for versatility during antenna creation and testing, and to allow modifying antennas within ADT without editing CAD models. The first approach generates associative geometry, where antenna surfaces are guaranteed not to intersect the structural surfaces they lie on. This avoids problems when running an HFSS simulation. The second approach does not generate associative geometry, although it can process much more complex geometry. The next sections explain how each of these approaches works.

### **6.1 User Defined Sheet Antenna**

This is the first of the two approaches use to create antennas in ADT as discussed in the previous section. It does not require the definition or creation of a deformation field, but uses surfaces created by the structural submodule in its place along with to a set of user defined parameter values. Using this approach, an antenna boundary is generated on a selected surface from the structural model, and a new surface is trimmed using this boundary. The surface from the structural submodule selected for defining an antenna conductor is called the defining surface. Initially, a boundary (outline) of a conductor surface is created. This outline is a closed contour of lines, and is defined by the user. The user specifies this contour by entering two columns of data, which represent a set of normalized coordinates defining the contour. Each row represents a vertex of the contour. These coordinates are normalized with respect to the defining surface, and must be manually obtained and entered by the user. Figure 37 shows an antenna CAD model, where the red outline of the conductor surface (yellow) is defined by the green points. These green points are normalized in Figure 38 with respect to the defining surface. The normalized coordinates in the lower half of Figure 38 are then manually input by the user as a set of two columns into ADT.

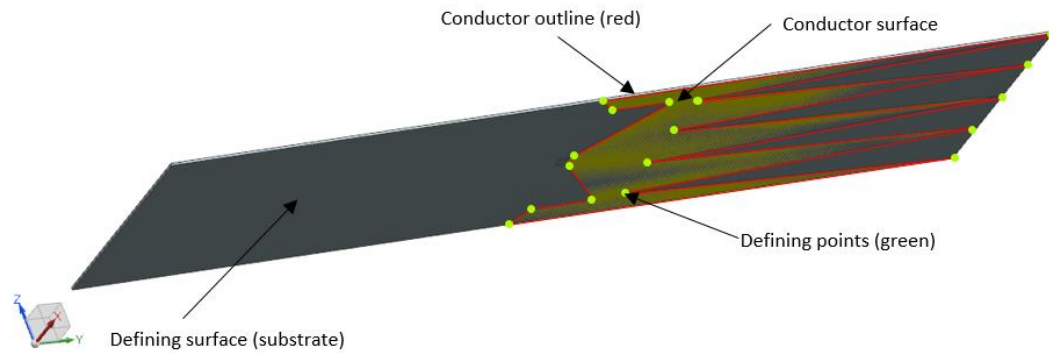


Figure 37. Original antenna geometry. User must normalize the green points with respect to the defining surface. Normalized green point coordinates are used as input into ADT to build sheet geometry.

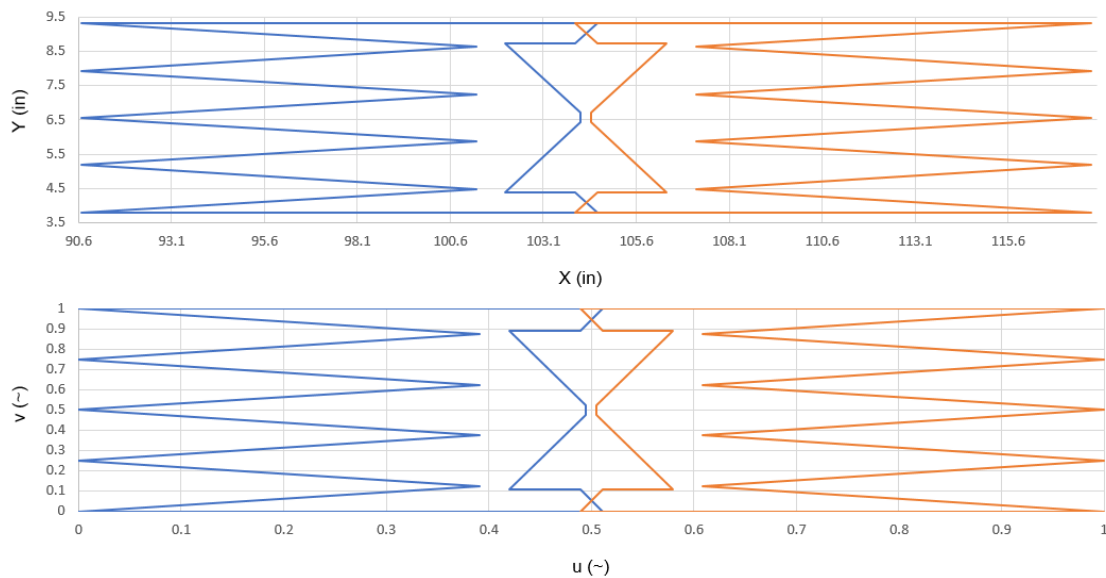


Figure 38. Original (above) and normalized or parametric (below) coordinates of antenna conductor surfaces. Blue and red represent two different conductor sheets defined on different surfaces. Red outline is defined by the green defining points from the previous figure.

ADT uses the normalized coordinates to sample points along the boundary from the deformed version of the defining surface. These points are interpolated to generate bounding curves using methods discussed in Chapter 5. Figure 39 shows a sheet (blue) with four antenna substrates (green) defined on it, taken from the structural model. The normalized coordinates from Figure 38 are evaluated on each substrate (green) surface in Figure 39, producing the

resulting black bounding curves. These bounding curves are used in the CAD export module to trim a copy of the green defining surface and create antenna conductor sheets. This method was used to generate antenna surfaces in the simulations involving the MCoRDS array in the results section (Chapter 7). The substrates are created from the green surfaces, which are part of the structural model, and do not require post-processing. This approach is not recommended for complex geometry, as defining such geometry by using parameter values can be very tedious. It also cannot define 3D geometry, or sheet geometry which cannot be fully defined by parametric lines. The second approach addresses these problems.

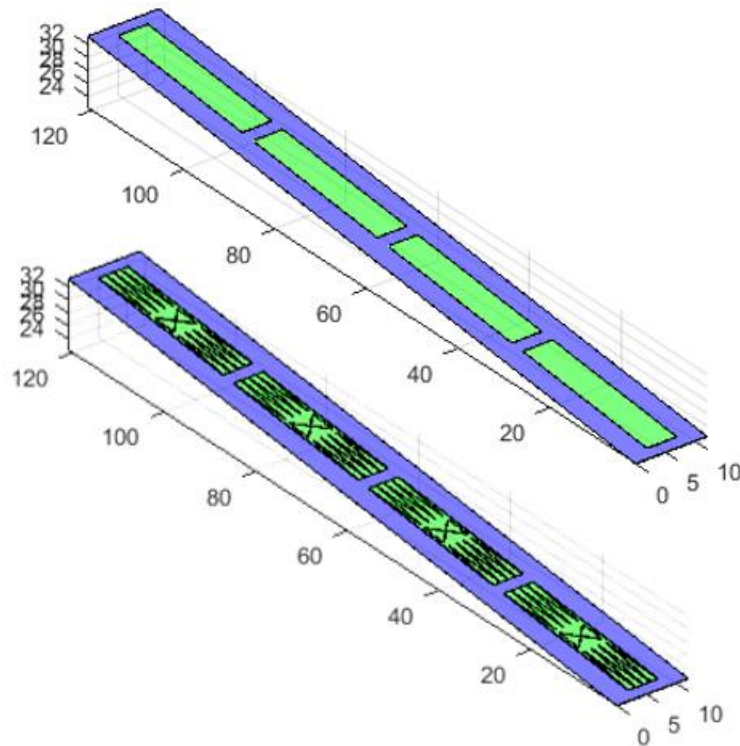


Figure 39. Antenna conductor boundaries mapped onto deformed substrate sheets (green), which are on the bottom fairing surface (blue).

## 6.2 Deformation of Antenna From CAD File

This second approach by which ADT generates antenna geometry involves deforming an antenna defined by a CAD model (as opposed to user-specified parameter values) and uses a

mapping, or deformation field. A code to import STEP CAD files was created, which generates a set of objects in the MATLAB workspace storing the properties and relations of the CAD geometry. Largely, this set of objects follows a structure similar to the STEP file structure, which is defined in Appendix B. Since there is a clear geometric entity hierarchy, the senior entities making up the STEP file (solids and sheets) are deformed by deforming their junior, or defining entities. For example, a flat, rectangular face is typically defined by a plane, a set of four edge curves, and four vertices. It can be deformed into a B-spline surface by deforming the plane and the bounding curves, and combining the new, deformed entities into a new surface definition. The next sections explain the details of this process.

The input antenna CAD contains the geometric features defining the antenna model. This geometry is imported, processed and organized by the ADT antenna deformation submodule. The CAD file is a general set of objects,  $S$ . Each of the objects in this set contains sub-sets of objects, organized hierarchically. Thus, the problem consists in finding the deformed version of every entity making up set  $S$ , and rebuilding the geometric model using the deformed entities.  $S$  is typically a set of solids and faces. The solids are composed from closed shells (closed groups of bounded surfaces), and faces are composed of open shells. To obtain deformed geometry, a transformation function  $T$  acting upon a feature (open or closed shell)  $S$  to obtain a deformed feature  $D$  is needed. The shell (open or closed),  $S$ , is made of a set of faces  $f$ , which are in turn made of a set of one surface definition  $s$ , and a set of boundary loops  $l$ . These loops are made up of a set of bounding curves,  $b$ . These relationships are summarized by Equations 26.

$$\begin{aligned}
T(S) &= D \\
S &= \{f_1, f_2, \dots, f_n\} \\
T(S) &= T(\{f_1, f_2, \dots, f_n\}) \\
f_i &= \{s, l_1, l_2, \dots, l_m\} \\
l_j &= \{b_1, b_2, \dots, b_k\} \\
n, m, k &\in \mathbb{Z}
\end{aligned} \tag{26}$$

The function  $T$  could be any general continuous deformation function. In the case the deformation field is known, the function can be provided by the user (such as a beam deformation function, or the shape of an airfoil), and used to get deformed antenna shapes. This is useful for conformal antenna studies, as it allows to quickly generate geometry of an antenna in various known deformed configurations. In the case that a deformation function is not known, it can be inferred from the structural model obtained in Chapter 5, based on the surface fitting procedures presented in that same chapter. The deformation function obtained this way is based on B-splines. This deformation function is similar to a B-spline surface, and the procedure for mapping is overall somewhat similar to the process explained in Section 6.1. The main difference is that a 3D deformation field is used, and the process is more automated, not requiring the user to manually normalize each defining point of the geometry.

A B-spline can be generalized as a mapping from a domain with an arbitrary number of dimensions to another domain with an arbitrary number of dimensions (Turner & Crawford, 2009). Thus, a mapping can be defined from one three-dimensional space to another three dimensional space, where the mapping, or B-spline solid, represents a deformation field, as in Equation 27.

$$T(u, v, w) = \sum_{i=0}^n \sum_{j=0}^m \sum_{k=0}^l N_{i,p}(u) N_{j,q}(v) N_{k,o}(w) P_{ijk} \tag{27}$$



In this equation, a generic spline transformation  $T$  from a three-dimensional parametric domain to the Euclidean domain is defined by an  $n$  by  $m$  by  $l$  control point array  $P$  (where each  $P$  is a vector) and the corresponding B-spline coefficients. The mapping is a  $p$  degree spline in the  $u$  parametric direction,  $q$  degree spline in the  $v$  parametric direction, and  $o$  degree spline in the  $w$  parametric direction.

If derived from the structural model, a B-spline solid transformation function can be inferred from existing surfaces. This type of B-spline solid is generated from an existing deformed surface (generated during the surface fitting stage) and its thickened surface. The thickness is set by the user (usually, it is the same as the total antenna thickness, including substrate and conductor thickness). The control point nets of the deformed surface and its offset are combined to give the control points of a B-spline solid. In this case, mapping in the  $w$  parametric direction is linear. A larger set of control points in the  $w$  direction can be used to define non-linear mappings, but it would require more information on the deformation in this direction (more stacks of control points in the  $w$  direction). In the cases tested (where the thickness is much smaller than the length and width), a linear mapping in  $w$  was found to be sufficient. Figure 40 shows a B-spline solid used for mapping antenna geometry.

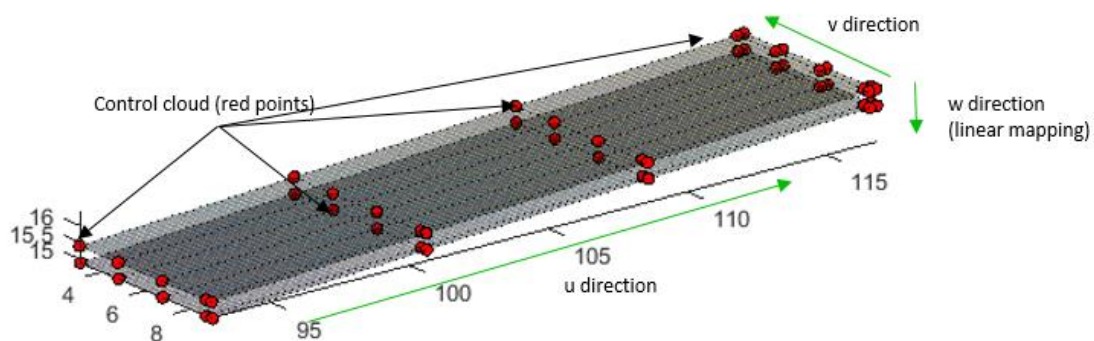


Figure 40. B-spline solid used for mapping antenna geometry.

Using a B-spline solid mapping requires normalizing the antenna geometry with respect to a bounding (defining) box (instead of just a defining surface, as in Section 6.1). This box encompasses the entire antenna geometry. All antenna geometry is normalized with respect to this bounding box, as shown in Figure 41, where all antenna geometry is normalized with respect to the red bounding box.

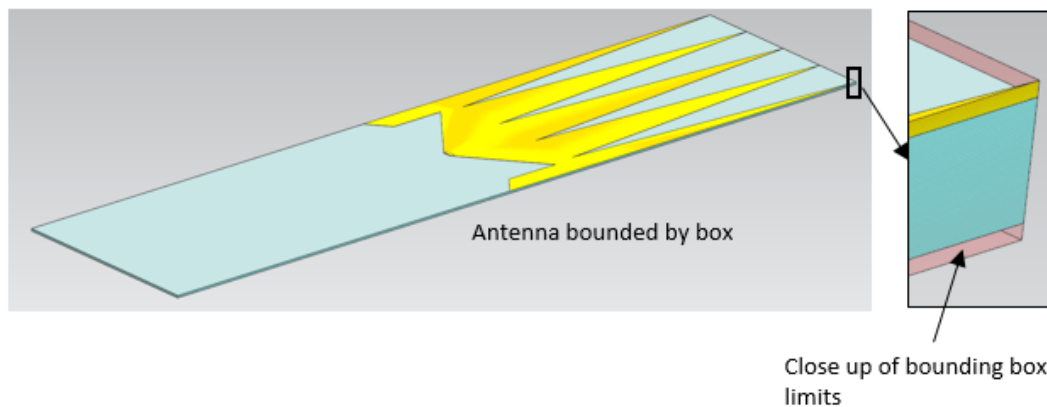


Figure 41. Antenna and bounding box.

The normalized coordinates are essentially a parametrization of the antenna geometry within a bounding box. Similar to the method in Section 6.1, the antenna in this case is considered to be non-load bearing, and follows the shape of a chosen deformed object from the BDF. The normalized antenna features are mapped to a deformed configuration using the transformation function, by evaluating the function using the normalized values. In the case that an explicit custom deformation function is used (not a B-spline solid), normalization of coordinates is not required, and the geometry can be mapped directly using the transformation function. Figure 42 shows an antenna as it is being mapped by a B-spline solid deformation function. The NURBS Toolbox (Splink, 2010) is used for MATLAB NURBS visualization and some computations. The next section details the specific mechanisms used to map antenna CAD geometry.

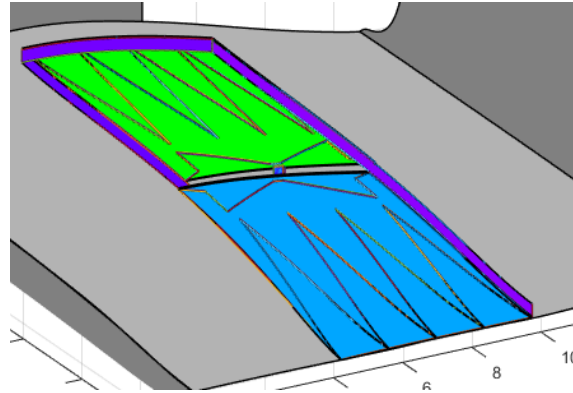


Figure 42. Antenna as it is being deformed. Z-scale exaggerated.

### 6.3 Antenna Deformation Process

Deforming the antenna shape amounts to applying a function to the faces and edges making up the antenna geometry, and restoring their connectivity information to define valid shapes, as summarized by Figure 43.

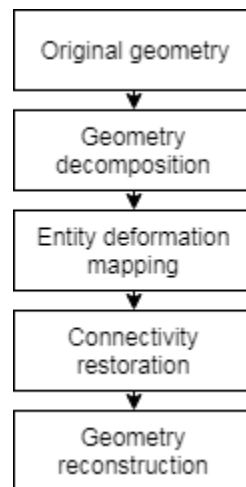


Figure 43. Antenna deformation process.

As explained in the previous section, an input CAD model generally consists of a set of faces, some of which define solids. Most of these faces are either defined as bounded planes (if flat) or bounded B-spline surfaces (if curved). Their boundaries are usually defined by sets of connected curves which are either straight lines or B-spline curves. The deformation functions

convert this geometry into deformed surfaces and curves, which are always defined as B-spline curves or surfaces. This means that there are four main types of deformations that occur:

- 1) B-spline curve to deformed B-spline curve
- 2) Line to deformed B-spline curve
- 3) Plane to deformed B-spline surface
- 4) B-spline surface to deformed B-spline surface

Some cases are not considered, such as the deformation of cylindrical or circular surfaces, which have their own special definition in the STEP file. The code to read, interpret, and deform this type of geometry will be added in the future. The following sub-sections each cover one of the cases described in the above list.

### **6.2.1 Edge curve deformation**

Each edge of a surface boundary being deformed (line or B-spline) is sampled at a dense, uniform interval, and these sampled points are mapped to the deformed configuration. Sampling rate can be adjusted by the user better quality curves. For edge curves which are curves (as opposed to lines) in the original configuration, an algorithm will be added in the future to recognize critical features (such as the change and reversal in curvature) in order to sample all critical defining points.

After a curve is sampled, and the sampled points are mapped, a B-spline curve is fitted through the mapped points to obtain the deformed edge, using the same algorithms presented in Chapter 5. Edge curves belonging to a single surface bounding loop are stored as a set of curves, and their connectivity is restored by matching each end vertex to the next curves start vertex.

### 6.2.2 Plane deformation

Planes in a STEP file are defined by a point and a normal vector; ADT deforms planes into B-spline surfaces. However, while a plane extends infinitely, the transformation functions used generally have a limited domain. Thus, a process was developed to ensure that points sampled from a plane surface would lie within the domain of the transformation function while also capturing all the details of the planar feature being deformed. The outline of this process is given in Figure 44. It consists of finding the minimum bounding quadrilateral which encompasses all critical points of the planar face being transformed, sampling a regular grid of points over this quadrilateral, mapping the grid onto deformed space, and fitting the resulting grid using a B-spline surface. The rest of this section describes this process in detail.

Because a B-spline surface, representing deformed geometry) can only have four edges, and all features of a planar bounded sheet (which can have an arbitrary number of edges) must be encompassed by the transformation function, a minimum bounding quadrilateral is needed. The minimum bounding quadrilateral encompasses all the features of the original surface in the smallest four-edged area possible.

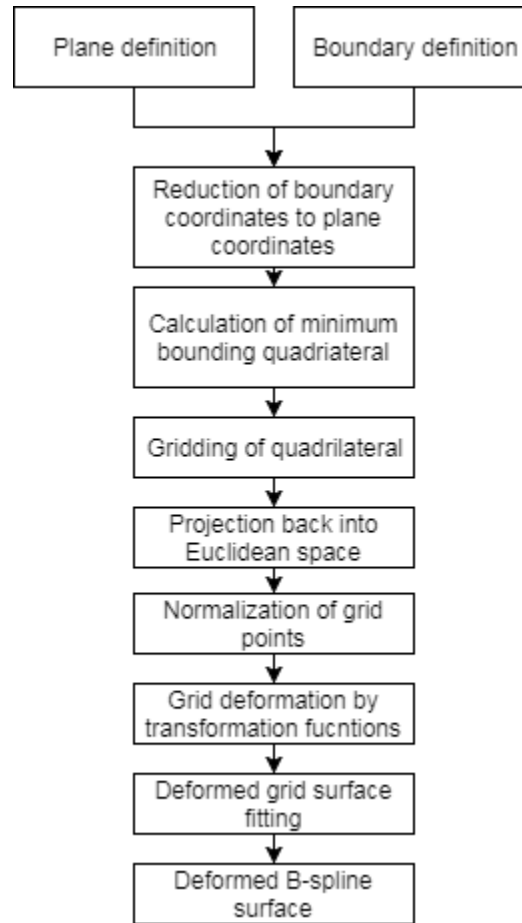


Figure 44. Outline of plane deformation process.

ADT uses a modified version of a code developed by D’Errico (2014) to find a minimum bounding quadrilateral for all critical points defining a boundary. Critical points are points which are vertices of an edge, and if the edge is a curve, a certain number of points along the curve. This procedure finds four points on a plane which form a quadrilateral encompassing all critical points within the minimum area possible. This ensures that all details of a face boundary are captured. A grid is then generated over the minimum bounding quadrilateral, and is mapped from the nominal space to the deformed space. A B-spline surface is fitted through the mapped grid. Figure 45 shows an Ultra-Wide Band (UWB) antenna element (Gogineni, Wong, & Lytle, 2003),

with one conductor surface selected. This surface is deformed using the process described, shown in Figure 46.

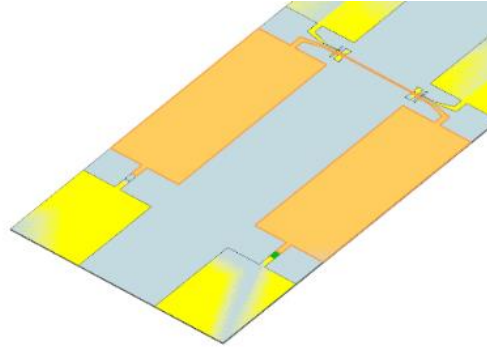


Figure 45. UWB antenna conductor

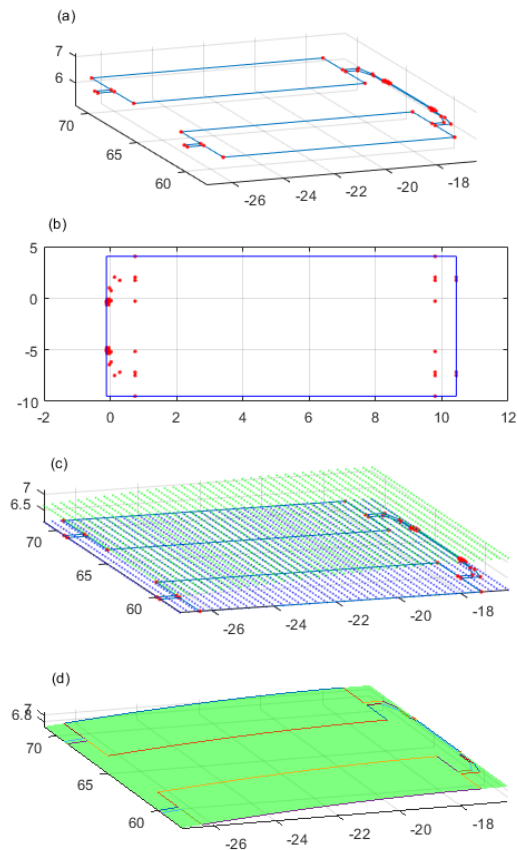


Figure 46. Antenna deformation process for component on a plane. (a) Nominal antenna component defined on a plane, with edges and critical points marked. (b) Projection of critical points onto plane coordinates and determination of minimum bounding quadrilateral. (c) Generation of grid on plane (blue points), and mapping of grid into deformed space (green points). (d) Mapping of edges into deformed space and fitting of B-spline surface to deformed point grid.

### 6.2.3 B-spline surface deformation

To deform a B-spline surface into another B-spline surface, a similar approach to the one presented for planes is used. A grid is obtained from the original B-spline surface, which is mapped onto the deformed domain using the transformation function, as outlined in Figure 47. The use of a minimum bounding quadrilateral is not necessary in this case, as the surface is naturally bounded by its control matrix. The deformed grid is fitted by a deformed surface.

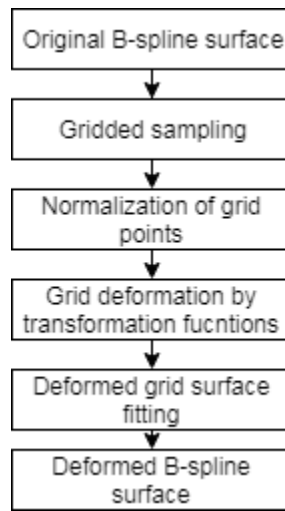


Figure 47. B-spline deformation process.

### 6.2.4 Connectivity restoration and solid reconstruction

After the individual faces and edges of the solid have been deformed, and if the faces define a closed shell, the connectivity between them must be restored. A valid solid definition in the STEP file must reference the same vertices and edges where faces meet. Connectivity is restored by comparing the coordinates of the vertices of all the edges of the bounds of every face making up a solid. Following a similar principle to the adjacency matrix which was used for surface sorting, a connectivity matrix is assembled, defined by Equation 28.

$$C_{ij} = \begin{cases} 1, & v(i) = v(j) \\ 0, & \text{otherwise} \end{cases} \quad (28)$$

$$i, j = 1, \dots, n$$



In Equation 28,  $v(i)$  represents the coordinates of the  $i^{\text{th}}$  vertex ( $v(j)$  of the  $j^{\text{th}}$  vertex), and there are  $n$  total vertices in the geometric model, before duplicates have been eliminated.  $C_{ij}$  has a value of 1 if vertices  $i$  and  $j$  are the same, and 0 otherwise. An example of this matrix and associated faces are shown in Figure 48 and Figure 49.

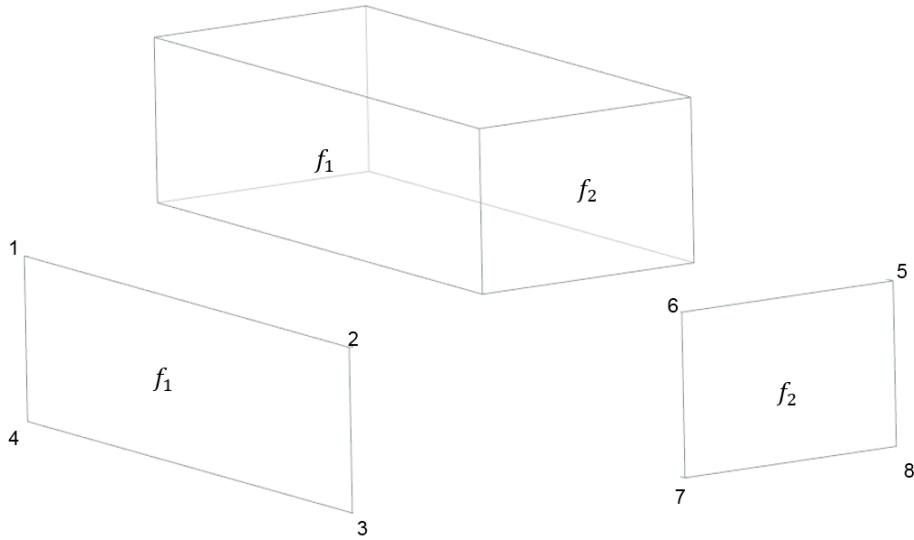


Figure 48. Faces of a manifold.

	$v_1$	$v_2$	$v_3$	$v_4$	$v_5$	$v_6$	$v_7$	$v_8$	$\dots$	$v_n$
$v_1$	1	0	0	0	0	0	0	0	$\dots$	$C_{1n}$
$v_2$	0	1	0	0	0	1	0	0	$\dots$	$C_{2n}$
$v_3$	0	0	1	0	0	0	1	0	$\dots$	$C_{3n}$
$v_4$	0	0	0	1	0	0	0	0	$\dots$	$C_{4n}$
$v_5$	0	0	0	0	1	0	0	0	$\dots$	$C_{5n}$
$v_6$	0	1	0	0	0	1	0	0	$\dots$	$C_{6n}$
$v_7$	0	0	1	0	0	0	1	0	$\dots$	$C_{7n}$
$v_8$	0	0	0	0	0	0	0	1	$\dots$	$C_{8n}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
$v_n$	$C_{n1}$	$C_{n2}$	$C_{n3}$	$C_{n4}$	$C_{n5}$	$C_{n6}$	$C_{n7}$	$C_{n8}$	$\dots$	$C_{nn}$

Figure 49. Vertex manifold connectivity matrix.

Once the connectivity matrix has been assembled, only the lower id equivalent vertex is maintained, and all vertices related to it are referenced to the lower id vertex. Edges which are common to different faces, and thus form the bound of the solid, must be also be uniquely referenced by face definitions. Since it is known from the connectivity matrix which unique vertices form each edge of every bound, unique edges can be determined by comparing their start and end vertices. If different edges have the same start and end vertex, they are taken to be the same edge. Even though it is possible that a different path is taken from the same start to end vertex, this is not expected to be encountered for the geometries used in this project. In the future, this algorithm will be adjusted to account for different edges using the same sets of vertices.

This section concludes the discussion on antenna deformation methods, and also concludes the description of the functionality of ADT. The next chapter presents the results obtained by applying ADT to structural and antenna models, and some of the antennas referenced in this chapter are used.

## Chapter 7: Results

This chapter describes the results obtained using ADT. The focus is on the validity and quality of the generated geometry. In addition, the effects of common deformations on dipole antenna radiation patterns and return loss parameters are discussed. The chapter consists of ten sections, the first three of which discuss preliminary results and validation of an early version of ADT, improvements made in the second version of ADT, the setup of the simulations for the test cases, and the deformed geometry's surface quality. The next seven sections go through each of the six different test cases, which include geometries including advanced antenna designs and wing-integrated arrays undergoing representative in-flight displacement conditions, and conformal antenna designs using existing planar antenna designs and a user-specified deformation field. An additional section is devoted to the study of the effects of induced offsets in the HFSS models, which were necessary to generate valid results. The last section summarizes the results presented, as well as some conclusions drawn from the simulations.

### 7.1 Version 1.0 Results and Initial Validation

Mendoza & Arnold (2017) described an initial version of ADT and illustrated the program's utility by generating the deformation of a single wideband dipole element with a resonance frequency of approximately 200 MHz as well as an array of four elements. The array was attached to a fiberglass panel 120 inches long and 12 inches wide that represented a wing-skin, while the single element was attached to a panel 30 inches long and 12 inches wide. Results produced by the ADT-generated geometry were compared to antenna measurements conducted in the anechoic chamber located at the University of Kansas.

Both panels were modeled in PATRAN/NASTRAN as cantilevered panels, consisting only of quad shell elements, with one short edge clamped and a force applied at the opposite edge. The

single element model is shown in Figure 50. In the NASTRAN model, the maximum deflection for the array panel was 12 inches and 4 inches for the single element panel. The deformed geometry was then brought into HFSS and simulated. Details of simulations setups can be found in Mendoza & Arnold (2017).

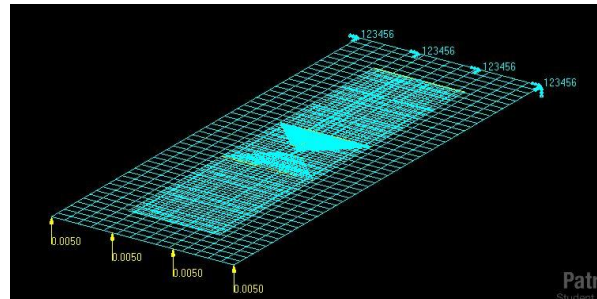


Figure 50. FEA model of single antenna element and panel with applied loads and boundary conditions.

For the measurements, array deformations were produced by attaching a thin fiberglass skin to a frame with the same curvature as that obtained by the model, which is shown in Figure 51. While a direct comparison of the measured and simulated element patterns was not possible because far-field measurement requirements could not be met due to the size of the array, mainbeam rotations due to curvature matched. In general, simulations and measurements for both individual antenna elements and arrays showed a shift in the radiation pattern of about 5-6 degrees. Trends in changes in sidelobe level were also found to be similar between measurements and simulations, but again, direct comparison was not possible due to the nature of the near-field measurement.

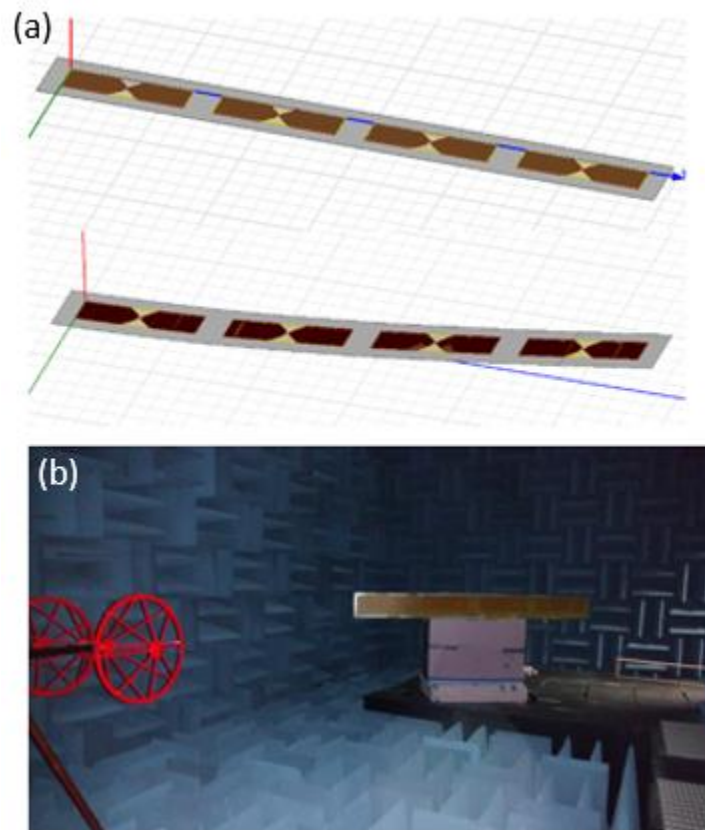


Figure 51. Simulations and measurements: (a) nominal and deformed geometry in HFSS, (b), measurement setup in anechoic chamber.

Although the initial version of ADT produced valid geometry and simulated results were partially validated with measurements, there were serious limitations to this initial version. One of these shortcomings was the requirement for regular quad meshes as an input from the BDF. Antennas were also required to be modelled in the structural model. This added unnecessary complexity and workload, since the structural nodes of the antenna had to be made to match the nodes of the structure beneath it. Generally only a limited number of nodes would match, and multi-point constraints were required; however these constraints caused serious intersections between surfaces, due to the methods used in surface interpolation, and required significant post-processing.

The initial surface sorting algorithm was not very robust, and could not adequately sort complete surfaces, requiring manual separation of surfaces by assigning each to a different region in the BDF. This separation resulted in additional surface and solid intersections, which caused meshing and simulation problems in HFSS. Because of these geometric intersections and meshing issues, return loss parameters were difficult to obtain using this version of ADT.

Since these initial tests, most of the development of ADT focused on improving the overall robustness of the program, implementing and organizing new data structures, and refining the existing surface approximation methods. The results obtained with this second version of ADT form the bulk of results in this chapter and are presented in the following sections.

## **7.2 Summary of Version 2.0 Improvements**

The second and most current version of ADT shows many improvements over its predecessor. Overall robustness has increased (though many improvements in this area remain), as the program has become more capable of interpreting a larger variety of meshes, and additional functionality was implemented. The most critical improvements were achieved in three different areas:

- 1) Mesh interpretation and surface generation. Unlike its first iteration, ADT 2.0 can interpret 2D meshes consisting of any combination of quads and triangles, as long as their properties are defined by a shell in the BDF. Unlike in the initial version, meshes defined on a single surface do not have to have the same number of nodes on opposing sides.
- 2) Antenna geometry import and processing. Perhaps the most significant improvement was in the inclusion of a module to import antenna geometry separately from the

FEM model and attach it to the deformed FEM model for export, or export it separately. Additionally, the allowable geometric definitions of the antenna geometry were significantly expanded, to allow manifold solids containing faces with more than or less than four edges. Previously only faces with exactly four edges could be processed.

- 3) User control. Unlike the previous version, ADT 2.0 can process custom user-defined deformation fields which are not derived from structural FEM; however, these definitions are limited to explicit equations in Euclidean space, or parametric B-spline solid or surface functions.

In the following sections, the simulation models will be described and the results presented. The purpose of these results is to illustrate the utility and capabilities of ADT as well to provide justification that the geometry produced by ADT is adequate and sufficient for simulations of deformed antenna geometry. While antenna performance is discussed, it is done so in this context.

## **7.3 Simulation Setup**




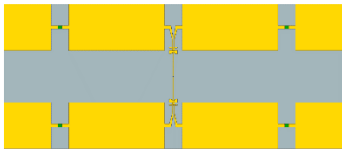
### **7.3.1 Antenna and Geometry Descriptions**

To evaluate the performance of ADT, representative geometry of an existing airborne antenna array was created and used with NASTRAN, where symbolic distributed loads were applied to generate a representative wing tip deflection. The geometry which was created attempts to mimic a simplified model of the MCoRDS (Byers, et al., 2012) array fairing, mounted under the wing of the NASA P-3 aircraft. This section provides a brief description of the antennas, fairing geometry, and structural models used for the simulations presented later in the chapter. Several types of antennas were tested with the current version of ADT using the

same fairing and wing geometry. For each configuration, the deformed geometry was generated from a NASTRAN model at three different load cases.

The various antennas (and their salient attributes) used in the simulations in this chapter are summarized in Table 4. Details regarding each antenna design can be found using the listed reference.

Table 4. Antennas used in simulations.

Name	Picture	Operating Bandwidth	Overall Dimensions	Reference
P3 Element		175-215 MHz	Length: 23.91" Width: 6.02"	Byers, et al. (2012)
MCoRDS Element		175-275 MHz	Length: 27.17" Width: 6.00"	Byers, et al. (2012)
Sheet WB Dipole		194-217 MHz	Length: 26.81" Width: 1.48"	Tuned using methods in Balanis (2005)
UWB Element		150-600 MHz	Length: 31.98" Width: 13.56"	Yan, et al. (2015)

### 7.3.2 Structural Simulation Details

The structural FEM model used in this section is based on the geometry in Figure 52. This geometry was created in NX and consists entirely of sheets. It was imported into, meshed, and processed in Patran, from which a NASTRAN model was generated.



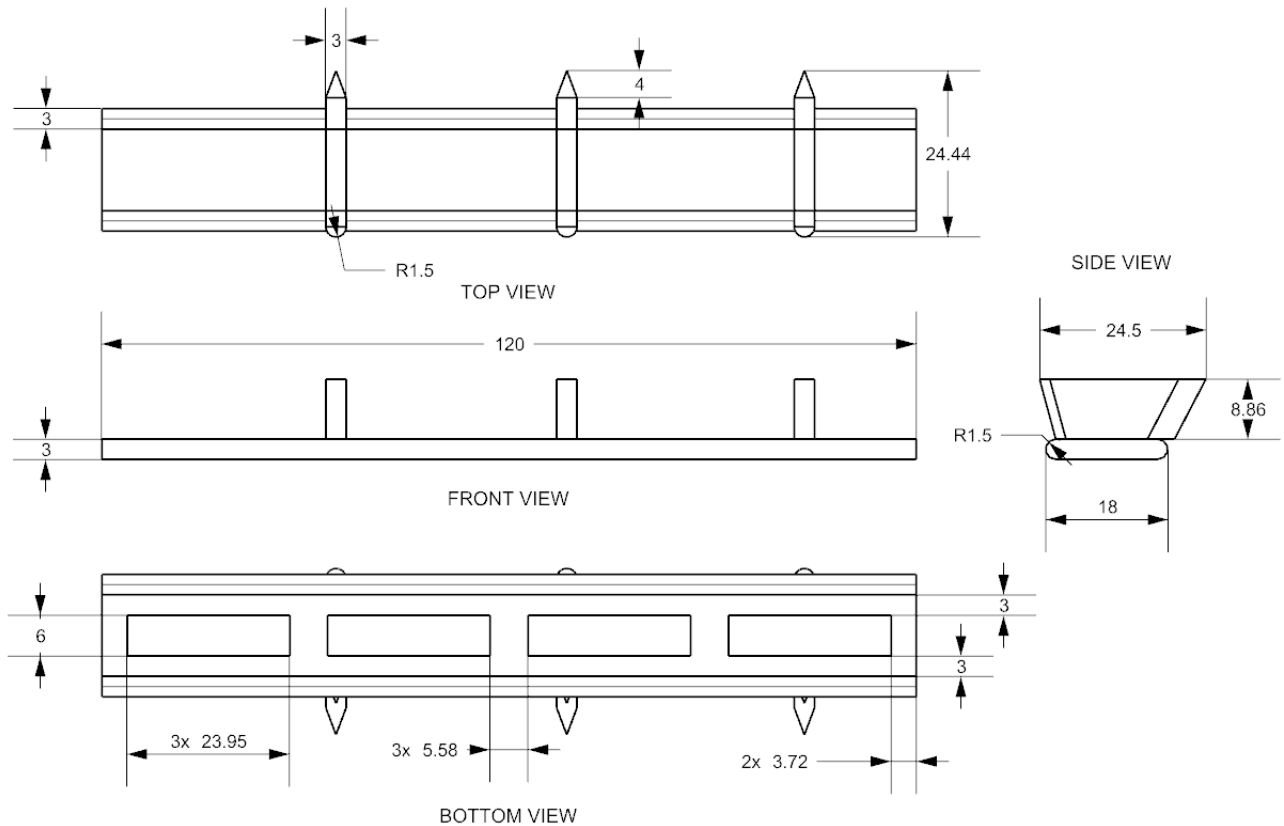


Figure 52. Geometry used for structural model creation. All dimensions are in inches.

The NASTRAN model consists of nine regions, all of which were modelled using CQUAD4 or CTRIA3 shell elements and are shown in Figure 53. Aluminum was used for the wing and pylons, S2 glass was used for the fairing, and FR4 Epoxy for the antenna substrates. The properties used for these materials are summarized in Appendix D. Because ADT currently cannot read composite material properties from the BDF, all materials, including composites, were modeled as isotropic materials. This will be updated in future versions of ADT. In the case of composites, a middle value for Young's modulus and Poisson's ratio from the consulted references was taken, which are in Appendix D.

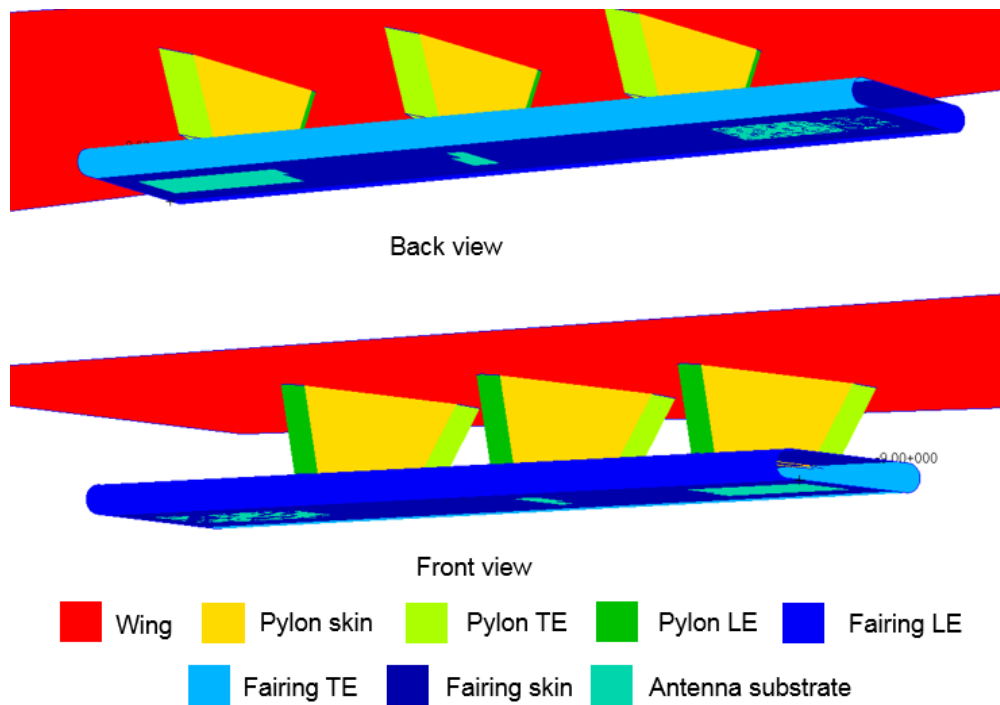


Figure 53. Regions of the structural model.

A general view of the mesh created for the simulation is shown in Figure 54. All meshes which were created, except for the wing mesh, were regular quad meshes. This was not a requirement for ADT, but rather the approach chosen for meshing.

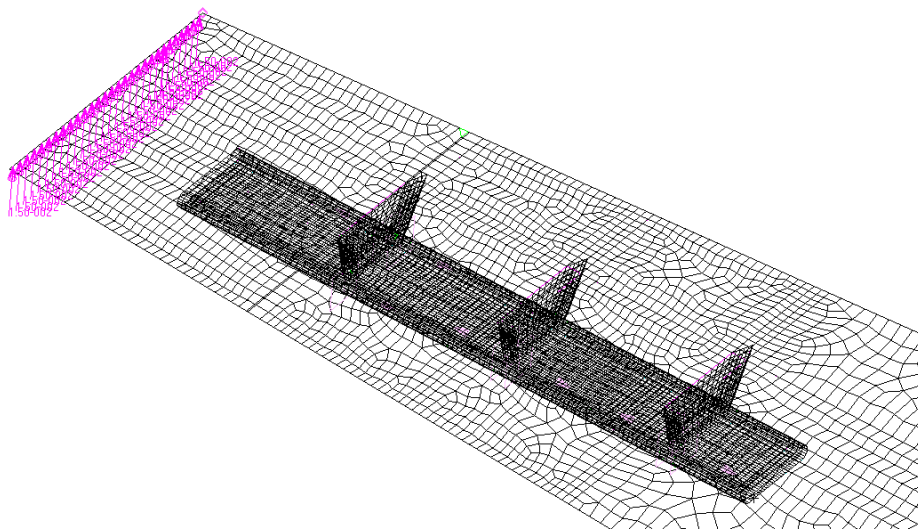


Figure 54. FEM model of P3 antenna fairing. Distributed tip load used to generate desired deflections illustrated by the pink arrows.

Table 5 summarizes the properties applied to each region of the NASTRAN model, and also shows the number of surfaces which was generated by ADT from each region. Because the main objective of this simulation is to test ADT's ability to create deformed geometry from the NASTRAN output, the loads and properties in the structural model are representative, and do not correspond to real load cases. In addition, structural sizing of the wing or array was not conducted.

Table 5. Structural model parameters.

<b>Region</b>	<b>Material</b>	<b>Structural elements</b>	<b>Thickness (in)</b>	<b>Number of surfaces</b>
Wing	2024-T3 Aluminum	Shell, CQUAD4 and CTRIA3	0.25	1
Pylon spars	2024-T3 Aluminum	Shell, CQUAD4	0.25	6
Pylon leading edge	S2 Glass	Shell, CQUAD4	0.02	3
Pylon trailing edge	S2 Glass	Shell, CQUAD4	0.02	6
Fairing leading edge	S2 Glass	Shell, CQUAD4	0.02	1
Fairing trailing edge	S2 Glass	Shell, CQUAD4	0.02	1
Fairing skins	S2 Glass	Shell, CQUAD4	0.02	2
Antenna substrate	FR4 Epoxy	Shell, CQUAD4	0.06	4

A distributed load was applied along the outboard node line of the wing tip to generate a desired deflection, and the wing root was constrained using a clamped boundary condition. In addition, fairing and wing structures were not sized to withstand the applied loads. Table 6 summarizes the load cases applied to the model, and the resulting wingtip deformations.

Table 6. Loads and deformations of structural model.

<b>Load Case</b>	<b>Applied wingtip edge Load (lb/in)</b>	<b>Maximum deformation (in)</b>	<b>Equivalent rotation (degrees)</b>
0	0	0.00	0
1	0.0075	11.7	1.72
2	0.010	17.6	2.59
3	0.015	35.1	5.15

### 7.3.3 Execution Time

As mentioned in Chapter 1, execution time is not a priority during the initial development stage of ADT. However, this parameter will be compared in this section to other surface fitting algorithms, to gauge whether the execution time is acceptable. During the surface fitting stage (structural submodule operation), ADT reads, organizes, and fits surfaces to a mesh. The mesh tested in this scenario consists of 14,788 vertices (nodes), which form quad and triangular faces. All code is written and ran in MATLAB, and the computer used has a 2.1 GHz Intel Xeon CPU E5-2620 v4. The time to process the mesh (which includes surface sorting, boundary and corner recognition, element triangulation, parameterization and mesh gridding), was 290 seconds. The time needed to fit surfaces to the processed mesh was 61 seconds, making the total time 351 seconds. This compares favorably to the times presented in Eck & Hoppe (1996), which are between 250 and 600 minutes for a model with the same number of vertices. However, it is worth noting that their algorithm fits data which does not have connectivity information, and which also automatically generates continuous B-spline patches, and the mentioned tests were ran on a 105 MHz machine in 1996. Krinamurthy's (2000) code was capable of fitting a 25,000 vertex mesh in only 20 seconds on a 250 MHz machine. This indicates that much work remains to be done to improve the speed of ADT, however, the current speed of around 6 minutes for a 15,000 vertex mesh is considered acceptable for initial development. The slowest portion of the code is currently the data sorting and mesh gridding portion. Specifically, the main bottleneck in this process is element triangulation.

The previous estimate does not include antenna deformation (which is a separate process from surface fitting). Antenna deformation speed depends highly on the geometry and method within ADT used. The manual point input method usually requires several minutes (on the order

of 10-20) to manually extract, normalize, and input normalized vertex coordinates into ADT. The evaluation of these coordinates itself is rather fast, on the order of seconds (under 30 for a simple model like the ones shown in this chapter). If using a CAD input model, generally time on the order of minutes is required for antenna geometry import and interpretation (around 1-2 for a simple antenna model like the ones tested in this chapter). The time required to deform the antenna depends highly on the type of deformation field. Deformations using explicit functions for deformation fields are faster, taking between 5-10 minutes for the geometry presented here. Deformations using B-spline solid functions require more time (between 10-20 minutes) for the models presented here. The main bottleneck in this process is the evaluation of the B-spline solids in the  $w$  direction, for which an efficient algorithm has not yet been developed.

When ADT geometry is used in HFSS, the HFSS progress bar can be monitored for simulation progress. Generally, the simulation process can take from tens of minutes to several hours for a medium complexity model as the ones used in this chapter. This is considered an acceptable simulation time. If the simulation stalls, the user is informed via the progress bar.

### 7.3.4 Geometry, Meshes, and Surface Errors

The geometry created by ADT for the antenna simulations in HFSS was generated by interpolating a 3 by 3 degree B-spline surface on the resampled quad mesh, which was obtained using the k-Harmonic parameterization method. Two error metrics are presented here to assess surface quality: Hausdorff distance and maximum surface error.

Hausdorff distance is a way of comparing two sets of data. The one sided Hausdorff distance between sets A and B, consisting of points  $(a_1, \dots, a_n)$  and  $(b_1, \dots, b_n)$ , is defined by Equation 29. The bidirectional distance, which is the distance used to measure error in this section, is defined by Equation 30 (Singhal, 2004).

$$\tilde{\delta}_H(A, B) = \max_{a \in A} \min_{b \in B} \|a - b\| \quad (29)$$

$$\delta_H(A, B) = \max(\tilde{\delta}_H(A, B), \tilde{\delta}_H(B, A)) \quad (30)$$

This distance is the maximum distance of a set to the nearest point in the other set. The lower this distance is, the more similar the two sets generally are. Two datasets, or meshes, are key to the process employed by ADT to generate surfaces: the original mesh dataset and the gridded mesh dataset. The original dataset consists of the points making up the original mesh created in Patran. The gridded dataset consists of a gridded mesh mapped onto the original mesh. A third set is used in this section as a means to compare the accuracy of the parameterization used to grid the mesh. It is created by evaluating the surface fitted to the gridded mesh at the obtained parameter values for each original dataset point. These datasets are shown for the wing surface in Figure 55. One way to compare the accuracy of the mapping generated from the resampled mesh is to compare the third dataset, to the original dataset.

The Hausdorff distance is used to compare the original mesh with the point cloud obtained from evaluating the fitted surface at the parameter values corresponding to each original mesh point. In other words, this second set of points was obtained by sampling points from the surface fitted to the gridded dataset, the sampling being carried out at each of the parameter values given by the k-harmonic parameterization used on the original mesh. This measure is a way of assessing the quality of the parameterization used, and the agreement between the surface obtained from the gridded mesh and the original data. Minimization of this value is particularly important if the tested surface is to be used to generate antenna geometry manually, by inputting parameter values as explained in Section 6.1. A large value can generate unexpected antenna shapes, such as shapes where edges meant to be straight are morphed into wavy edges, as shown

in Figure 56. If the surface is not used to generate antennas, this value is not as significant. The code used to measure Hausdorff distance in this work was developed by Danziger (2013).

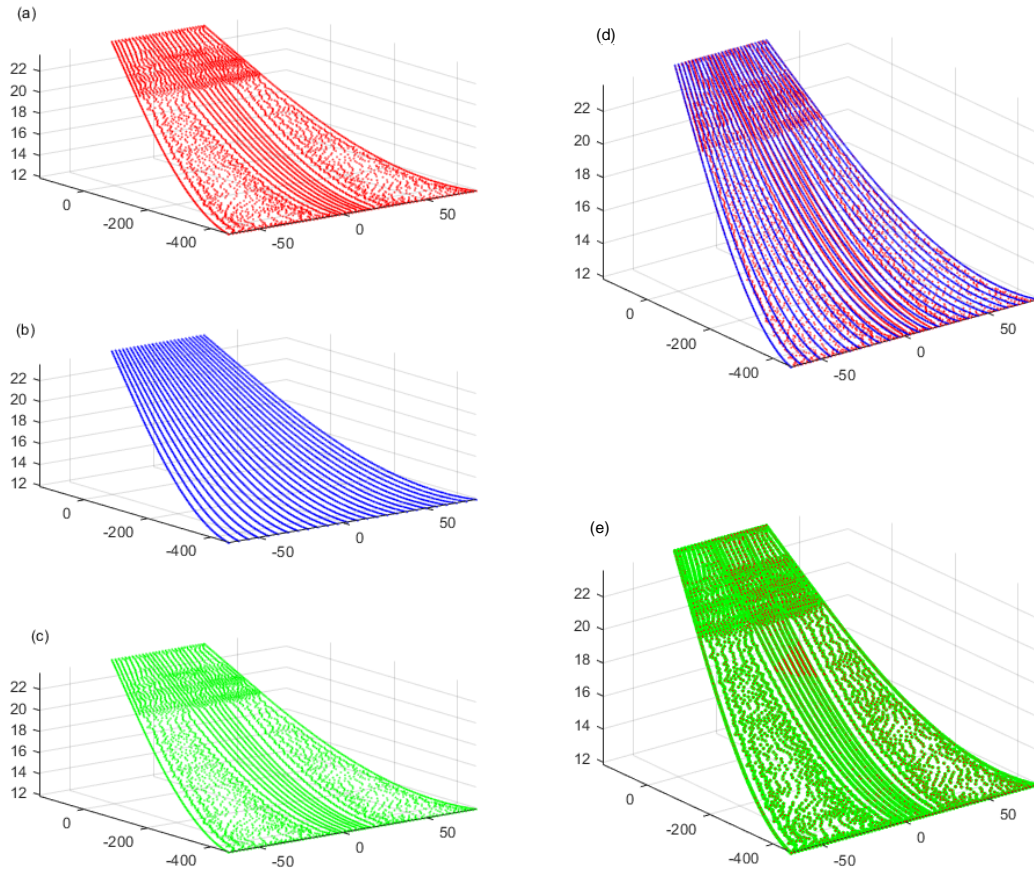


Figure 55. Datasets for wing surface of model. (a) Original mesh dataset (generated from Patran FEA mesh). (b) Resampled mesh dataset. (c) Evaluated parameter dataset. (d) Original mesh overlaid with resampled mesh. (e) Original mesh overlaid with evaluated mesh at parameter values. Axis represent coordinates.

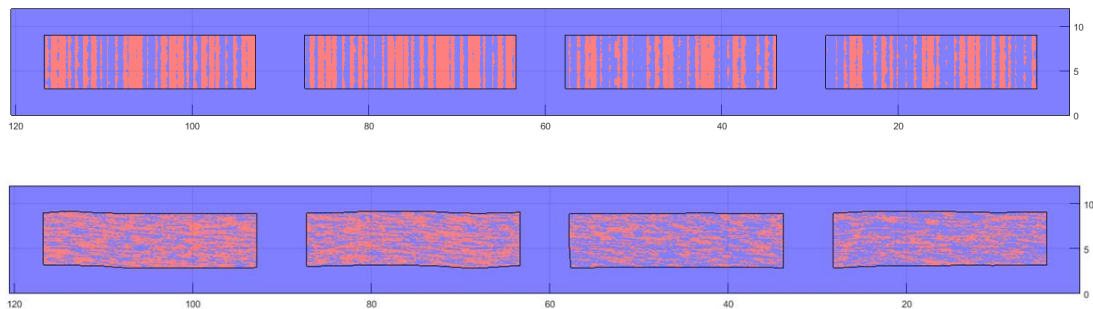


Figure 56. Four surfaces (red) constructed from larger surface (blue) using parameter values. Above: surfaces constructed using good parameterization. Note that edges are straight. Below: Surface constructed using bad parameterization. Note that surface edges are not straight.

Because the minimum error for an original mesh point will not necessarily occur at the surface parameter value corresponding to it, another measure to gauge true maximum surface error is used. The maximum surface error between the original mesh and the interpolated surface is measured by finding the maximum value of all the minimum distances between the nodes in the original mesh and the surface which was fitted from them, as shown in Equation 31 (Piegl & Tiller, 1997), where  $Q_m$  are the node data points, and  $S$  is the fitted surface.

$$\max(\min(Q_m - S(u, v)), 0 \leq u, v \leq 1) \quad (31)$$

All of the metrics presented here were calculated for each of the surfaces created by ADT for the wing and fairing model presented and are summarized in Table 7, which provides a range of metrics for regions consisting of more than one surface.

Table 7. Surface error metrics.

Region name	Hausdorff distance between original mesh and evaluated parameter points		Maximum surface error		
	inches	% of diagonal	inches	% of average edge length	% of diagonal
Wing	7.05E-02	1.14E-02	3.62E-02	1.05E+00	5.85E-03
Pylon LE	1.97E-02	1.91E-01	1.82E-02 to 1.83E-02	2.27E+00	1.77E-01
Pylon TE	2.55E-04 to 3.11E-04	2.03E-03 to 2.48E-03	2.54E-04 to 3.15E-04	2.68E-02 to 2.76E-02	2.03E-03 to 2.52E-03
Pylon spar	3.48E-7 to 2.05E-6	3.29E-6 to 2.09E-5	1.66E-06 to 6.15E-06	1.19E-04 to 6.06E-04	1.57E-05 to 5.82E-05
Pylon skin	1.28E-01 to 1.34E-01	6.09E-01 to 6.37E-01	4.79E-03 to 4.87E-03	3.30E-01	2.30E-02 to 2.33E-02
Antenna substrate	6.15E-07 to 5.88E-06	2.49E-06 to 2.38E-05	6.11E-06 to 1.19E-05	5.84E-04 to 1.36E-03	2.47E-05 to 4.80E-05
Fairing top skin	4.74E-06	3.93E-06	6.00E-05	4.72E-03	4.97E-05
Fairing bottom skin	2.25E-01	1.87E-01	3.14E-02	3.35E+00	2.60E-02
Fairing leading edge	1.59E-01	1.32E-01	3.23E-02	3.54E+00	2.68E-02
Fairing trailing edge	1.55E-01	1.27E-01	3.32E-02	3.62E+00	2.76E-02

The error metrics are measured in mesh units (inches), and in normalized units, (percent of the diagonal of the mesh) for Hausdorff distance, as well as in terms of average element edge



length for surface error. The mesh diagonal is the maximum distance between two points making up the mesh. The Hausdorff distance between the parametrically evaluated point cloud and the original mesh is always less than 0.2 inches, which is around 5 mm, a mean Hausdorff distance value cited in literature (Berger, Levine, Nonato, Taubin, & Silva, 2013), though for a somewhat different surface fitting problem. This indicates that the mesh gridding approach presented here performs to an accuracy similar to that of existing methods. It is smaller (on the order of  $1e-6$  inches) for antenna substrates, which is good since these are used to generate antenna surfaces in this chapter by using the procedure in Section 6.1. Hausdorff values on the order 0.2% of the mesh diagonal show that there is good agreement between the original data and the fitted surface. They also validate the surface fitting approach used. The surfaces which exhibit the largest errors in this category are the ones showing higher levels of curvature, such as the fairing leading and trailing edges, and the pylon leading edges. A shortcoming of the mesh resampling method as currently implemented primarily causes this. Figure 57 shows part of the pylon leading edge, with the original mesh in black, and the resampled mesh overlaid in blue.

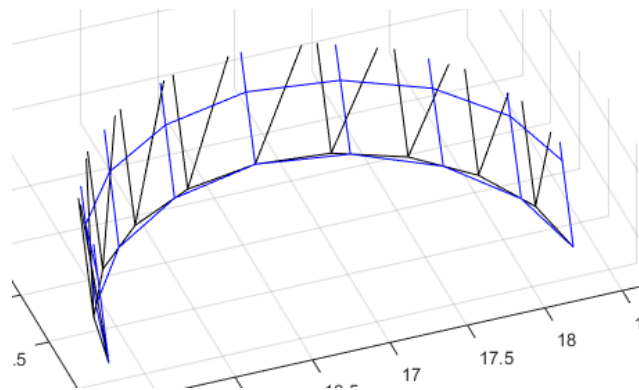


Figure 57. Pylon leading edge curvature, original mesh in black, resampled mesh in blue.

Because the resampling method seeks to obtain elements with an aspect ratio of as close to one as possible, it only takes into consideration the length of the edges and the number of seeds needed to connect opposite edges. This is determined by finding the aspect ratio of the

mesh being resampled, and then seeding a proportional number of seeds on each edge to obtain a total mesh node count equal to or above that of the original mesh. It does not take curvature into consideration. This can cause the resampled points to lie off the original edge, which, in the case in Figure 57, alters the shape of the curve. Although increasing the mesh sampling rate could help capture the discrete curve better, doing so uniformly would capture several points on the same edge, which could reduce the surface quality of the final approximation by forcing the fitted surface to conform too closely to a discretized curve. This problem signals the need for the creation and implementation of an algorithm which can increase mesh resampling rate in areas of higher curvature or ensure that all original points on a curved edge are included in the resampled mesh. This could be done by analyzing first the standard deviation in the directions of the normal vectors of all elements to determine whether curvature exists, or using discrete differential geometry techniques. A different solution to this problem can be implemented during the surface fitting phase, by assigning weights to each of the resampled points. These weights would indicate how closely the fitted surface must match each gridded node, and could be based on the minimum distance between a gridded mesh node and the nodes in the original mesh. An algorithm for implementing this type of approximation is given in Pegl & Tiller (1997).

Similarly, surface error is highest on curved surfaces, both in absolute and relative terms, due to the same reasons stated above. As mentioned, this reinforces the need to a better approach to resample and reconstruct curved geometry. At 0.02 inches and 0.17% of the diagonal, the maximum surface error is larger than desired, but well within the mean surface error of 0.05% to 0.2% of the diagonal found in literature (Eck & Hoppe, 1996). The surface generated is deemed representative of the original mesh geometry. While the error may not change the results of the

HFSS simulation, this can potentially pose larger problems by causing undesirable surface intersections.

Although each individual surface is indeed representative of the mesh it was originated from, there is no associativity or continuity between surfaces implemented. Each deformed geometry element is generated independently from its own point cloud and not associatively based on adjacent elements, and so there is an inherent probability that adjacent elements will intersect. Two types of intersections are the most likely to occur. The first type occurs when using the thickening function to turn sheets into solids. In this case, intersections are often large enough to cause errors in HFSS and prevent a simulation from running, shown in Figure 58.

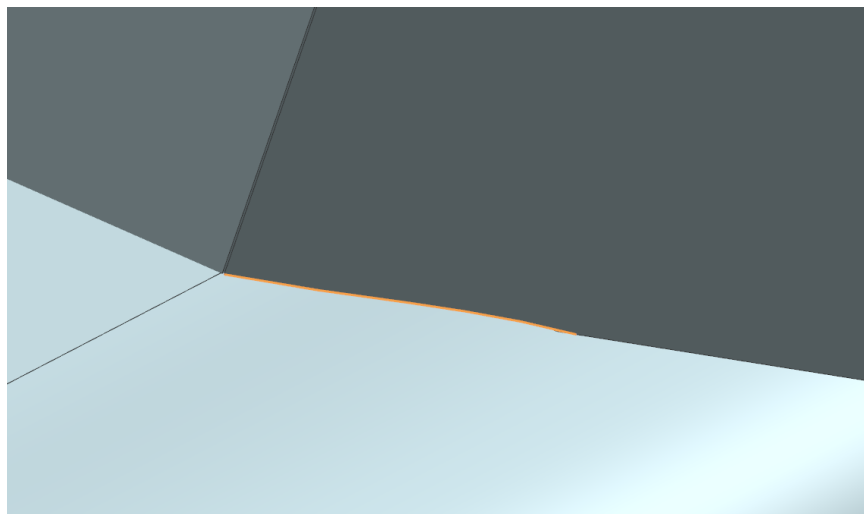


Figure 58. Intersection between pylon trailing edge and fairing trailing edge highlighted in orange.

A second type of intersection which is common is caused by the antenna deformation module of ADT when working with imported CAD files only (i.e. user is not manually defining a sheet antenna on an existing surface). When coplanar surfaces from these CAD files are deformed, as explained in Chapter 6, different grids, or point clouds, are generated and mapped to the deformed space for each surface to be fitted by a B-spline surface. Because the bounds of these coplanar surfaces may be different, the grids which are mapped are also different, causing

slightly different surfaces to be generated (even though the same mapping is used). This causes minuscule intersections and gaps between surfaces which were coplanar in the original configuration. This happens even if a subset of a dataset is used for fitting. Because there is no associativity between surfaces when they are created, surfaces derived from the full set and subset have small inherent differences. Figure 59 shows the difference between surfaces generated from different data sets, and Figure 60 shows the differences between a surface and its child generated from a subset of its own point cloud.

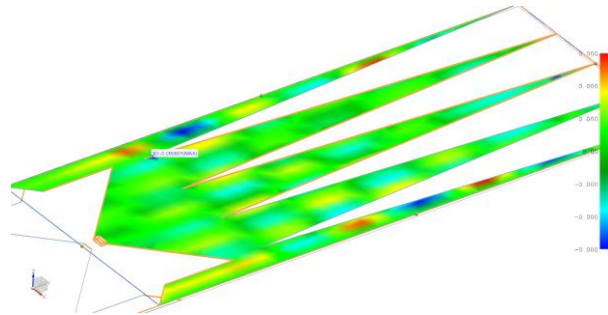


Figure 59. Deviation of antenna conductor from antenna substrate, blue indicating deviation into the surface, and red indicating away from it. The maximum deviation is  $1e-6$  inches into the surface.

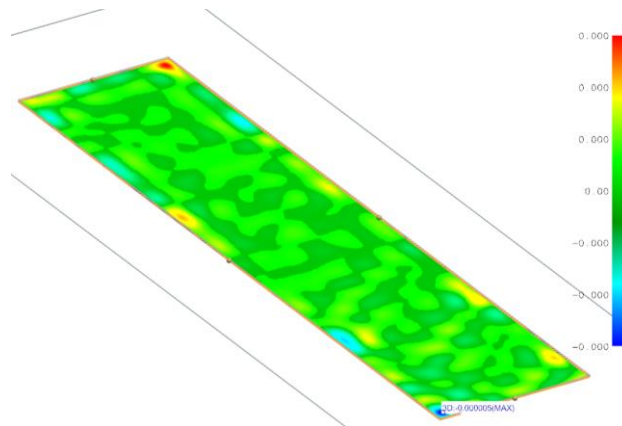


Figure 60. Deviation of a surface made of a subset of points from underlying surface, blue indicating deviation into the surface, and red indicating away from it. The maximum deviation is  $5e-6$  inches into the surface.

Using the deviation gauge tool in NX, the deviation of a surface from the surface under it is measured and color coded. The maximum deviation detected this way on the order of  $1e-6$

inches for both the surfaces from different sets and surfaces from a subset. This is enough to cause problems with meshing in HFSS. This number is for a relatively small deformation but can increase for larger deformations and for meshes of varying density.

As a simple and temporary fix to this problem, it was decided that small offsets should be introduced into all deformed geometry models where intersections are likely, to avoid intersections and allow them to run successfully. Offsets are introduced manually using a CAD program (NX) before importing geometry into HFSS, or after importing the geometry into HFSS using its own geometric processing capabilities. Offsets are introduced between surfaces which are likely to intersect or are intersecting to avoid problems with model meshing or solutions not converging. These are generally introduced by offsetting intersecting faces of intersecting solids in such a way that they do not intersect anymore, with the minimum amount of changes possible. The next section discusses simulation details and the effects of these offsets, and verifies that they do not cause major differences in results.

### **7.3.5 EM Simulation Details**

Several EM models in HFSS were created from the geometry which was deformed by ADT, one for each combination of antenna tested and load case (described in Section 7.3.2). In most cases, the geometry was imported directly as a STEP file, while in some cases, where post-processing was required, the geometry was first imported into NX, processed, and exported as a Parasolid file. The most common post-processing operations performed were trimming for surface intersections, in the case of obvious intersections, and sewing groups of sheets into solids if required.

For all models, the solution space was defined by a vacuum box, which was sized to be at least one quarter of a wavelength of the resonance frequency away from any radiating

component. This is a standard widely accepted by other antenna modelers. Radiation boundary conditions were used on the faces of the box. All conductive sheets were modelled as perfect electric conductors, and all excitations were modeled as lumped ports at an impedance of 50 ohm. FR4 Epoxy was used for the antenna substrate materials in all models, S2 Glass was used for the fairing skins, leading, and trailing edges, and aluminum was used for the pylon spars. Material properties can be found in Appendix D. The solution frequencies for the model were defined to be the center of the bandwidth frequencies shown in Table 4. Two different set ups were used for the simulations involving the wing and fairing model. One of the set ups only included the fairing geometry without the wing and pylons, and the other set up included all geometry, with wing and pylons. This was done to test the performance of ADT in increments and examine the effects of metallic surroundings on antenna performance. Figure 61 and Figure 62 shows screenshots of the EM model set up. However, before any arrays with any of these set ups were simulated, a single element was simulated with varying levels of deformation, to gauge the effects of deformation on a single element and establish baseline expectations.

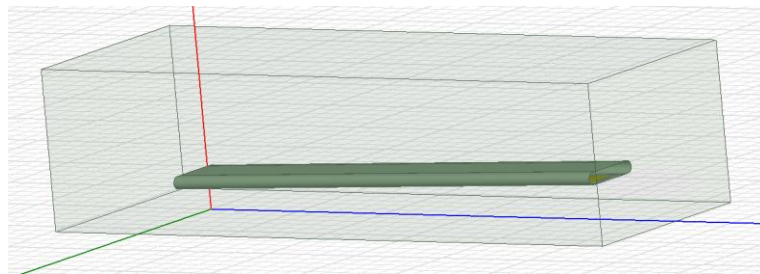


Figure 61. Fairing model without pylons and wing showing bounding vacuum box, used for Test cases 2 and 3. The wing and pylons were excluded for this test case.

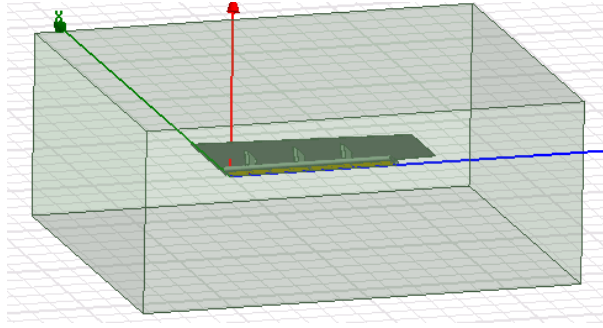


Figure 62. HFSS model with all surrounding geometry including wing, pylons, and fairing.

#### 7.4 Test Case 1: Single Element Deformation

This test case involves the simulation of a single P3 antenna element subjected to a clamped condition at one end, and a distributed load condition on the opposite edge. This was modelled by a single panel in NASTRAN, with one edge clamped, and a distributed load applied on the opposite edge. Progressive deformations, shown in Figure 63, were simulated to determine the effect of them on the resonance frequency and bandwidth of a single antenna element. Some extreme cases were simulated, where the strain was about 0.6, which would never be encountered in practical applications since the material would fail before this point is reached. The purpose of this simulation is to demonstrate the effects of deformation on a single element in the absence of other elements and metallic surroundings, and to determine the relationship between frequency and antenna length and strain. The large strains/deformation, make the changes in antenna performance easier to identify. This simulation is used to establish baseline expectations, and contrast these to more complex model simulations involving antenna arrays with surroundings present.

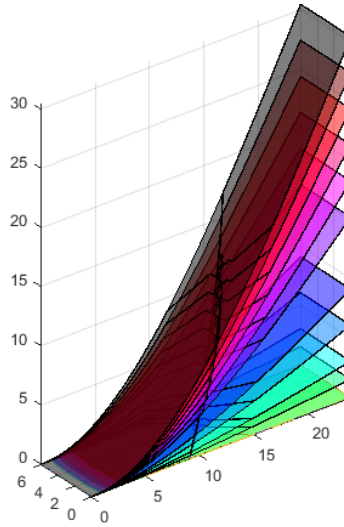


Figure 63. Deformations of a single antenna element, ranging from smallest (green) to largest (gray).

The detailed results of these simulations are in Appendix E. Table 8 summarizes the results of different data fitting functions to the relationship between strain and resonance frequency and bandwidth, which is visually presented in Figure 64. Resonant frequency and bandwidth are the parameters most dependent on antenna element strain, because the resonant frequency is inversely proportional to length of the antenna. Dipoles are designed to be slightly shorter than the half wavelength of the desired resonant frequency. It is therefore expected that with an increase in dipole length due to strain, the corresponding antenna resonant wavelength will also increase, thereby decreasing frequency. This relationship is expected to be linear in the absence of any interference, due to the inversely proportional relationship between frequency and antenna length.

As Table 8 and Figure 64 show, linear regression fits the data well (though not perfectly). Bandwidth also follows an approximately linear trend, decreasing with larger deformations. Since a dipole's bandwidth is related to its width, the decrease in bandwidth is likely due to the decrease in width relative to resonant frequency. These resonance results help put in context the results in the next sections involving arrays and more complex models and will help justify that



the geometry produced and subsequently analyzed by HFSS is adequate and produces realistic results. A resolution of 1 MHz was used for antenna resonance simulations, which resulted in a dense point cloud near the small deformation cases in the plots in Figure 64.

Table 8. Regression coefficients of different fits to data.

Relationship	Linear $R^2$	Exponential $R^2$	Second order Polynomial $R^2$	Logarithmic $R^2$
Length/Resonant Frequency	0.99	1.00	1.00	1.00
Strain/Resonant Frequency	0.99	1.00	1.00	N/A
Length/Bandwidth	0.98	0.99	0.99	0.99
Strain/Bandwidth	0.98	0.99	0.99	N/A

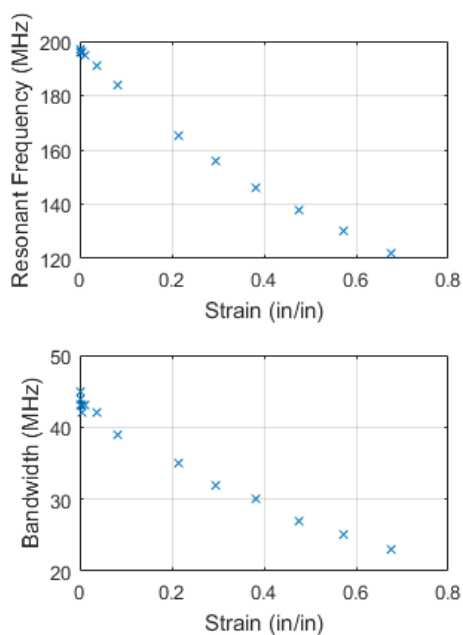


Figure 64. Effect of strain on resonant frequency and bandwidth.

Although the maximum strains simulated are not realistic, the plots presented all demonstrated the utility of ADT. Clearly, antenna element strains and deformations affect the performance of the antenna. With the development of this program, engineers will now have the capability to better characterize the effects of integrating antennas on flexible platforms.

## 7.5 Effects of Induced Offsets

As discussed in Section 7.3.3, during the process of building and running simulations, it was discovered that very small intersections or gaps between otherwise adjacent solids and surfaces could conduce to errors in the simulations, in particular meshing errors where the mesh would either not be generated, or where a mesh would be refined indefinitely by the meshing algorithm in HFSS. To avoid this problem, small offsets between intersecting or potentially intersecting surfaces had to be created in several of the models. This section assesses the effects of adding these offsets. Two cases were tested, one involving the effects of the offsets of a dielectric skin on a single element, where no other geometry is present, and another involving these effects on a four-element array which has fairing geometry present. Only the results of the latter are presented in this section. Offsets were introduced in the nominal model, which was then simulated, and its results were compared to the original nominal model results. In both the individual and array models, the original and offset models were analyzed at 200 MHz, with a frequency sweep from 150 to 250 MHz. Antenna return loss and radiation patterns simulation data was produced and compared.

The model used in the offset test simulations only contains the fairing geometry and excludes other geometry, such as a ground plane and metallic pylons. This was done mainly to characterize the effect of a shift in thickness of the dielectric fairing skin the antennas are on, without any other interference. The relatively small changes in the other surrounding geometry (up to 0.1%  $\lambda$ ) are not expected to cause much of a change. The offset distance for surrounding geometry was based on trial and error. Only one offset case was tested (chosen as a result of the single element offset tests), 0.001 inches (3.33E-3%  $\lambda$  for the given model), since it was found to allow (in the cases tested in this chapter) a successful simulation run, thus deeming the geometry as “acceptable”. The reader is reminded that no quantifying metric for “acceptability” is provided

at this stage of development, rather, a model is “acceptable” if it runs and converges. A 0.001 inch offset was chosen through trial and error; it was the smallest offset distance which did not cause problems with the model not converging. Smaller offsets, although desirable to maintain the original shape, caused problems with simulation convergence. This offset was used in all subsequent simulations involving deformed geometry, for consistency and to produce acceptable models. Figure 65 shows and labels the solids making up the simulated array, and Table 9 shows the offsets that were added.

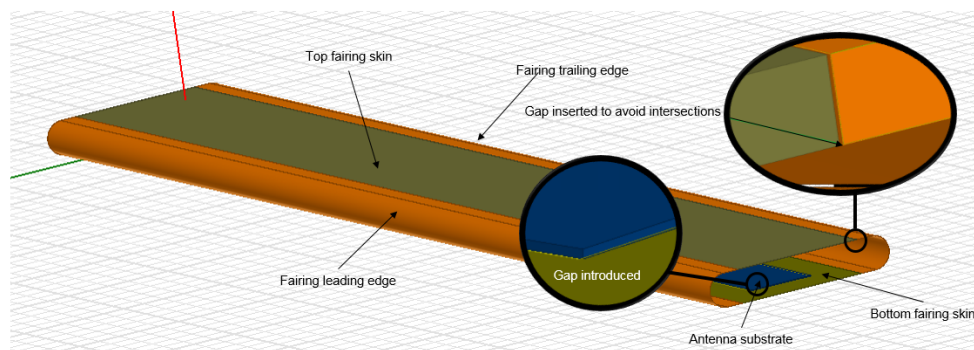


Figure 65. Fairing model showing the locations of the addition of gaps.

Table 9. Gap addition.

Element pair	Added offset (in)	Resulting thickness (in)
Bottom fairing panel, fairing leading edge	0.001	unchanged
Bottom fairing panel, fairing trailing edge	0.001	unchanged
Top fairing panel, fairing leading edge	0.001	unchanged
Top fairing panel, fairing trailing edge	0.001	unchanged
Bottom fairing panel, antenna substrates	0.001	Bottom fairing panel: 0.019 (thinner)

Figure 66 shows and compares the return loss parameters for all array elements for both the nominal and the offset EM models. Table 10 and Table 11 summarize the results obtained from both simulations, and the differences observed due to the inclusion of the offset. In terms of MHz, the changes in resonance due to gaps are significant, up to 2 MHz. In terms of percentage,

this is just 1% or less, and there is always an upward shift in frequency while the depth of the null increases. This result is expected due to dielectric loading, which occurs when an antenna is placed on a substrate. The dielectric loading tends to lower the resonant frequency of an antenna while also inducing a slight impedance mismatch (causing decreased null depth). Thus, when the antenna is no longer in contact with the dielectric skin of the fairing, the resonant frequency increases and the return loss deepens.

Table 10. Effects of induced offset on array resonant frequency and bandwidth.

		<b>Resonant Frequency (MHz)</b>	<b>Lower BW limit (MHz)</b>	<b>Higher BW limit (MHz)</b>	<b>Bandwidth (MHz)</b>
S11	Nominal	189.7	173.3	212.4	39.1
	Offset	191.6	174.8	215.2	40.4
S22	Nominal	191.3	176.3	212	35.7
	Offset	192.5	177	213.8	36.8
S33	Nominal	191.1	176	211.7	35.7
	Offset	192.8	177.3	214.5	37.2
S44	Nominal	190.2	173.7	212.3	38.6
	Offset	191.3	174.5	214.6	40.1

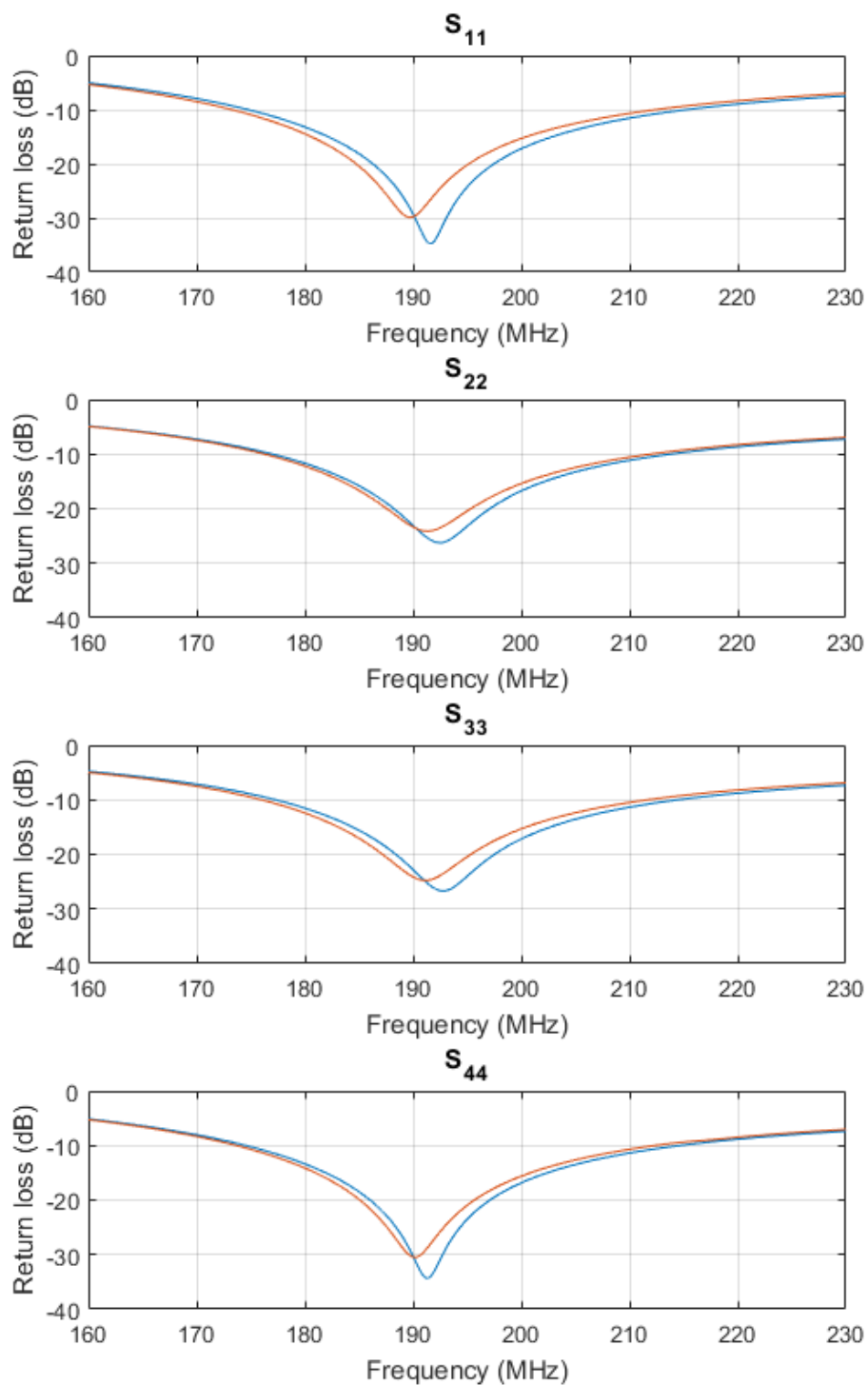


Figure 66. Change in resonance parameters without gap (red) and with gap (blue).

Table 11. Change in resonant frequency and bandwidth due to induced offset.

	Resonant Frequency (MHz)		Lower BW limit (MHz)		Higher BW limit (MHz)		Bandwidth (MHz)	
	MHz	%	MHz	%	MHz	%	MHz	%
S11	1.9	1.0	1.5	0.87	2.8	1.3	1.3	3.3
S22	1.2	0.63	0.7	0.4	1.8	0.85	1.1	3.1
S33	1.7	0.89	1.3	0.74	2.8	1.3	1.5	4.2
S44	1.1	0.58	0.8	0.5	2.3	1.1	1.5	3.9

The bandwidth of all array elements increases, and both the lower and higher bandwidth limits increase. The overall change is of 1.5 MHz or less, and hovers between 3 and 4.5%. Although in terms of MHz it is similar to the resonance frequency change, the relative (percentage) change is higher. This indicates that bandwidth is more sensitive to the offsets introduced.

These results seem plausible, given that only relatively small changes are expected. While it is true that the effect of the gap size will be dependent on the operational frequencies of the antenna (higher frequency antennas will be affected more by the same size offset than lower frequency antennas), inclusion of the model offsets was deemed acceptable for several reasons. While recognizing that the effects of the gaps are frequency dependent, ADT is being developed with a focus on the development of wing-integrated antenna arrays, and it is expected that most of the arrays examined will operate at low-UHF or lower frequencies. It is expected that resulting resonance errors will be less than 5%. Clearly this effect needs to be further characterized for higher frequencies, and users of ADT should be aware of this limitation in the program

Figure 67 shows a comparison of the simulated radiation patterns both with and without gaps. The patterns illustrate some differences, with the largest difference being a 3.5 dB depth difference near the pattern nulls. The shape of the sidelobes is also slightly different, especially

around -80 degrees, having a wider sidelobe by approximately 2 degrees, which is a widening of 2%. The half power beamwidth (HPBW) is the same for both models, at 24 degrees. This indicates that the radiation pattern is much less sensitive to the inclusion of the 0.01” gap than the return loss, and small changes can be expected at the most extreme nulls. As with the resonant frequency, it is expected that the effect will be larger at higher frequencies.

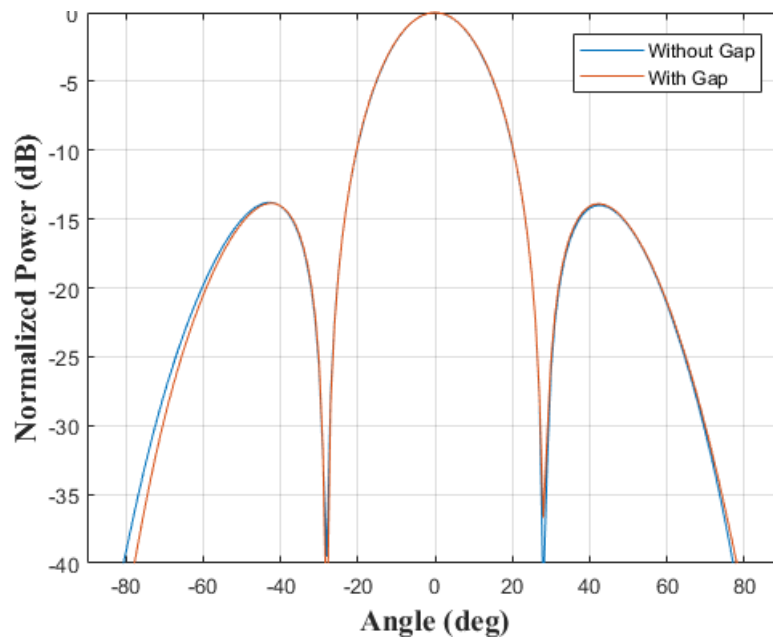


Figure 67. Change in radiation pattern due to presence of gaps.

While the inclusion of the offsets did influence simulation results, the errors caused by these gaps are considered acceptable given that the alternative is no ability to simulate deformed arrays. However, subsequent users should be aware of this induced error. For any given application, it is recommended that nominal arrays both with and without the offset be simulated so that there is a better comparison to the deformed array. In addition, it is also recommended that the effects of this gap be fully characterized across a wide range of frequencies in future work. Most likely minimum offset size is a function of solution frequency, rather than geometry,

but this hypothesis must be tested with additional antenna simulations operating at different solution frequencies.

Having established that gaps cause differences of up to 1% in resonance frequency and 4% in bandwidth for the model analyzed, and to further determine the performance of ADT, the rest of the chapter proceeds with several test cases involving the antennas mentioned at the beginning of the chapter. All of the models used in subsequent sections involving the antenna fairing model have the same induced offsets, or gaps, as tested in this section. First, the array simulated in this section (P3 element array) will be studied, followed by an array of MCoRDS elements without surrounding geometry, and then a more complex model involving the MCoRDS array with added wing and pylon geometry. Additional cases involving other antennas are tested in the section following those.

### **7.6 Test Case 2: P3 Array, Fairing Only**

An array of four P3 elements was simulated, which is the same array which was modeled and simulated using the first version of ADT. This simulation does not include wings or pylons, and only includes the fairing geometry. The elements are numbered one through four, starting from the element closest to the wing root, and going outboard, towards the wingtip. All load case models have induced offsets, as discussed in Section 7.5. Figure 68 show the changes in return loss with increasing wingtip deformation. The reader is referred back to Section 7.3.2 for the description of each load case.

The resonant frequencies of all the elements are slightly different and show slightly different trends. It is difficult to establish a definitive trend for the resonant frequencies of the antenna elements, but they are consistently lower for the highest deformation case, which corresponds to the highest strain case, like what was seen in Section 7.4. The change in length,



strain, and change in frequency are listed in Table 12. A consistent pattern which can be drawn from this data is that the bandwidth of the antenna decreases as strain increases. This matches the single element simulation case, although unlike in that case, this rate does not seem to be linear due to element coupling.

Because the deformations experienced are small, any changes in resonant frequency or bandwidth are also expected to be small. As shown in Figure 68 and Table 12, small changes are observed in the resonant frequency, which varies by up to 1%. Bandwidth varies by a larger amount, up to 8.33%. This value is not too far from the value seen in the individual element case for a similar strain level, which hovered around 6-7%. It should be noted that this variation, however, is comparable to the variation due to the inclusion of the offsets in Section 7.5. This again indicates the need to perform a detailed study of the effects of offsets on any individual model, as well as the necessity for the development of a mechanism which guarantees G0 continuity between surfaces at the very least, which would avoid the need for induced offsets. A comparison of the results of this model to the individual element deformation indicates that the differences in resonant frequencies with deformation and the absence of a linear trend is most likely due to element coupling and interaction with the surroundings, and not due to artifacts present in the geometry generated by ADT. In addition, the small changes for small deformations seen in the results are an indication that the geometry produced by ADT is adequate and sufficient for this particular simulation. Particular attention should be given in future studies and uses of ADT to determine whether the results produced are valid and when ADT is best not used.

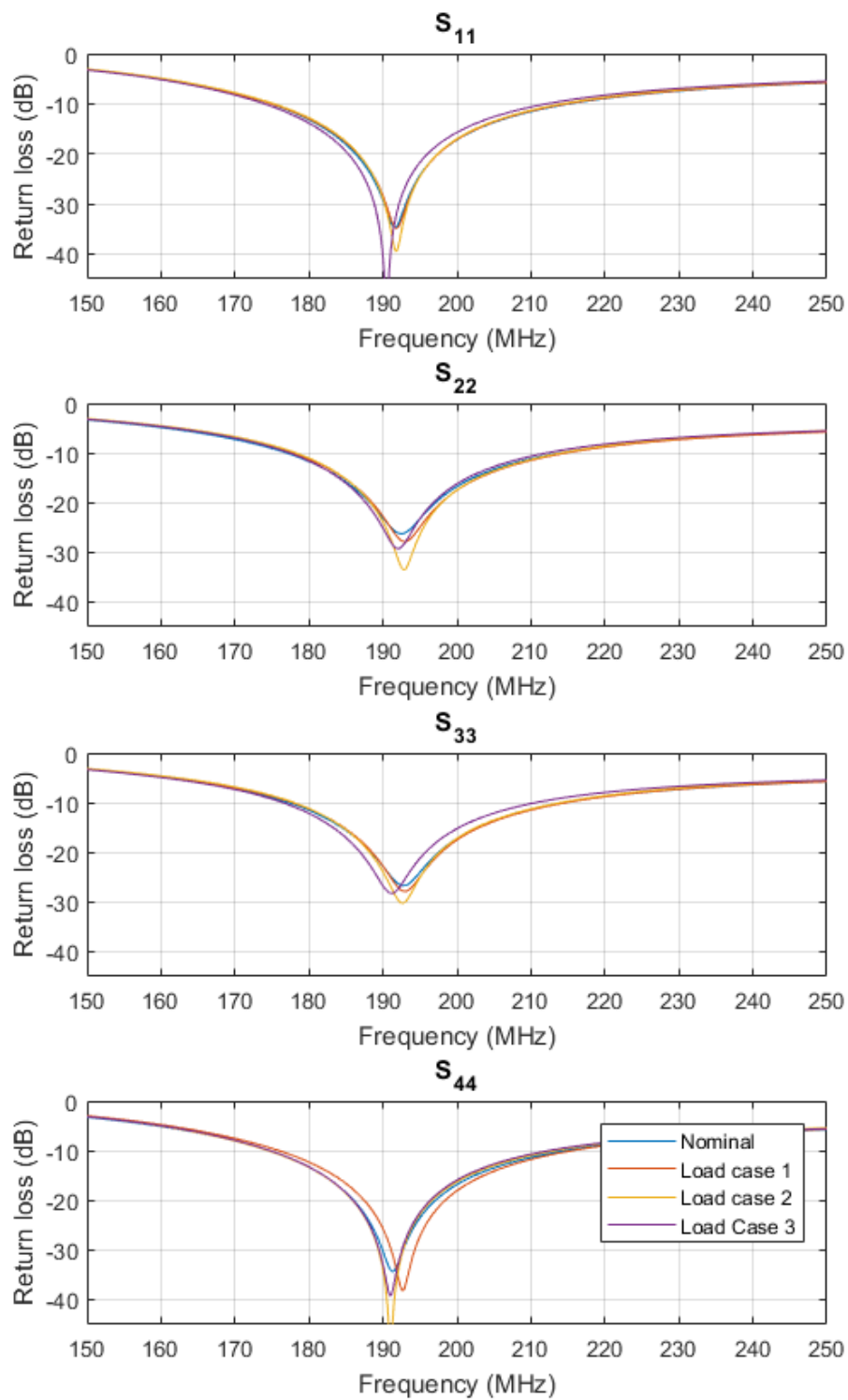
Figure 68. Return loss parameter  $S_{11}$  for different load cases.

Table 12. Antenna strain and change in resonant frequency.

Element	Load Case	Antenna length (in)	Change in antenna length (in)	Strain (in/in)	Resonant frequency (MHz)	Lower BW limit (MHz)	Higher BW limit (MHz)	Bandwidth (MHz)
S11	Nominal	23.9500	0	0.00E+00	191.6	174.8	215.2	40.4
	Case 1	23.9609	1.09E-2	4.53E-04	191.8	174.9	214.9	40
	Case 2	23.9744	2.44E-2	1.02E-03	191.8	175.2	214.3	39.1
	Case 3	24.0475	9.75E-2	4.07E-03	190.5	173.9	212.2	38.3
S22	Nominal	23.9500	0	0.00E+00	192.5	177	213.8	36.8
	Case 1	23.9608	1.08E-2	4.53E-04	192.9	177.8	214.7	36.9
	Case 2	23.9744	2.44E-2	1.02E-03	192.8	178.2	214.1	35.9
	Case 3	24.0475	9.75E-2	4.07E-03	192	177.4	212.1	34.7
S33	Nominal	23.9500	0	0.00E+00	192.8	177.3	214.5	37.2
	Case 1	23.9609	1.09E-2	4.53E-04	194	179.1	215.9	36.8
	Case 2	23.9744	2.44E-2	1.02E-03	192.6	178	213.9	35.9
	Case 3	24.0475	9.75E-2	4.07E-03	191.2	176.5	210.6	34.1
S44	Nominal	23.9500	0	0.00E+00	191.3	174.5	214.6	40.1
	Case 1	23.9609	1.09E-2	4.53E-04	192.7	175.9	215.6	39.7
	Case 2	23.9744	2.44E-2	1.02E-03	191	174.5	213.2	38.7
	Case 3	24.0475	9.75E-2	4.07E-03	191	174.8	212.4	37.6

Most of the deformation experienced for the load cases tested can be visualized as a rotation, since there is little to no change in the curvature of the fairing skin that the antennas are on, and only a small change in strain. This mostly rotational motion should cause a constant shift in the radiation pattern. To verify this, the rotation angle was measured with respect to the horizontal axis. The changes in the radiation pattern are shown in Figure 69. The radiation pattern shows a constant shift to the left with each load case, which was measured as the change in the direction of the mainbeam. The measured values are shown in Table 13.

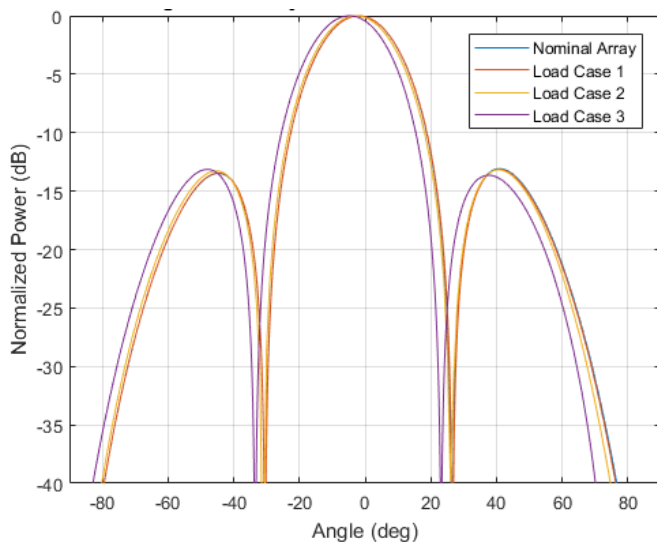


Figure 69. Radiation pattern for P3 array under different load cases.

Table 13. Shifts in radiation pattern for several load cases.

Load Case	Shift in radiation pattern (degrees)	Equivalent rotation (degrees)	Percent difference
Nominal	0	0	0
1	-1.9	-1.7	10
2	-3.0	-2.6	13
3	-4.7	-5.2	9

The shift in the radiation pattern and the linear approximation of the rotation angle fit within a 13% margin of error, which translates to less than 0.5 degrees in the case studied. This indicates that change in radiation pattern mostly agrees with the expectation – a rotation. This is also supported by the fact that the radiation pattern shapes are almost identical, meaning that the geometry does not undergo some unknown deformation.

### 7.7 Test Case 3: MCoRDS Array, Fairing Only

Using the same fairing geometry from the previous section, a MCoRDS antenna array was simulated. Even though they have different dimensions than the P3 antenna, the MCoRDS antennas were deformed directly in ADT without the creation of an additional structural FEM to

accommodate their different size. This type of resizing can be done whenever a smaller surface fits entirely within a larger one. In this case, because the antenna was defined parametrically with respect to the substrate, the substrate had to be resized first. The smaller surface (antenna substrate) can be defined in terms of the larger one (fairing skin), producing new geometry (similarly to the way antennas are defined in Section 6.1). Although it is a valid approach for generating geometry, this can have implications on the veracity of the structural model. The scale of these changes would depend on the mechanical properties of the material making up the antenna substrate, as well as the physical properties of the substrate itself (size). In the case tested here, the antenna substrate stiffness is negligible compared to the fairing stiffness, so derived surfaces are deemed acceptable.

After substrate resizing, antenna designs were swapped onto existing geometry models (derived from structural models), using the built-in ability to generate sheet antenna geometry from the structural geometry, explained in Section 6.1. Like in the previous simulation, return loss parameters and radiation pattern results were obtained. The results of these simulations are presented in Table 14, and Figure 70.

Table 14. Resonance and bandwidth frequencies.

		<b>Resonant Frequency (MHz)</b>	<b>Lower BW limit</b>	<b>Higher BW limit</b>	<b>Bandwidth (MHz)</b>
S <sub>11</sub>	Nominal	205	173	281	108
	Load case 1	206.1	172.7	278.6	105.9
	Load case 2	208	176	283.1	107.1
S <sub>22</sub>	Nominal	209	174.5	273	98.5
	Load case 1	205.8	174.7	282.2	107.5
	Load case 2	239.9	174.8	279.9	105.1
S <sub>33</sub>	Nominal	215.5	175	273	98
	Load case 1	210.7	175.1	280.9	105.8
	Load case 2	202.7	173.1	282.2	109.1
S <sub>44</sub>	Nominal	210	173.5	274.5	101
	Load case 1	233.2	173.7	278.5	104.8
	Load case 2	241.2	176.6	281.9	105.3

As in the previous case, it is difficult to establish a clear pattern between load case (strain) and resonance frequency and bandwidth, but again, this is not unexpected due to the non-linearity of coupling. The bandwidth of each antenna element in general is larger than the bandwidth of each element in the previous antenna model (P3). The resonant frequencies for the outer array elements (S<sub>22</sub>, S<sub>33</sub>) increase, by up to 15%, while the maximum decrease is 6%. This is significantly larger than the changes seen in the P3 antenna model simulations, which did not exceed 1%. In general, however, the bandwidth and resonance plot shape match measurements well, especially regarding the bandwidth and resonance frequency, which were generally found to be between 175-275 MHz and around 200 – 225 MHz respectively (Byers, et al., 2012). These results are expected, and comparable to measurements, however, more experimental evidence would be necessary to conclusively validate the ADT model.

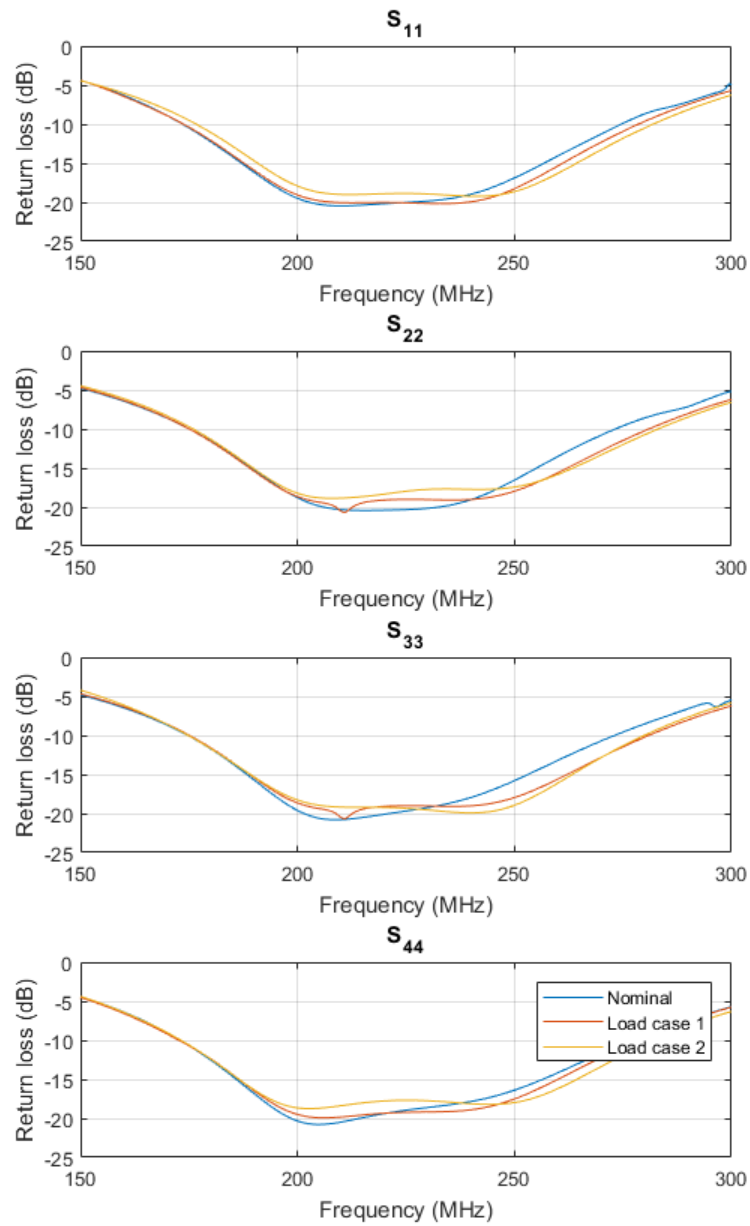


Figure 70. Change in  $S_{11}$  due to deformation.

The radiation pattern, in Figure 71, shows a shift which is proportional to the wingtip deformation, similarly to the shift in the previous array tested. This shift is quantified in Table 15. The difference between the angle of this deformation and the pattern shift is off by 0.5

degrees, which is similar to what was seen in the simulation of the P3 array. The shapes of the patterns are similar and are consistent with what would be expected in a rotation.

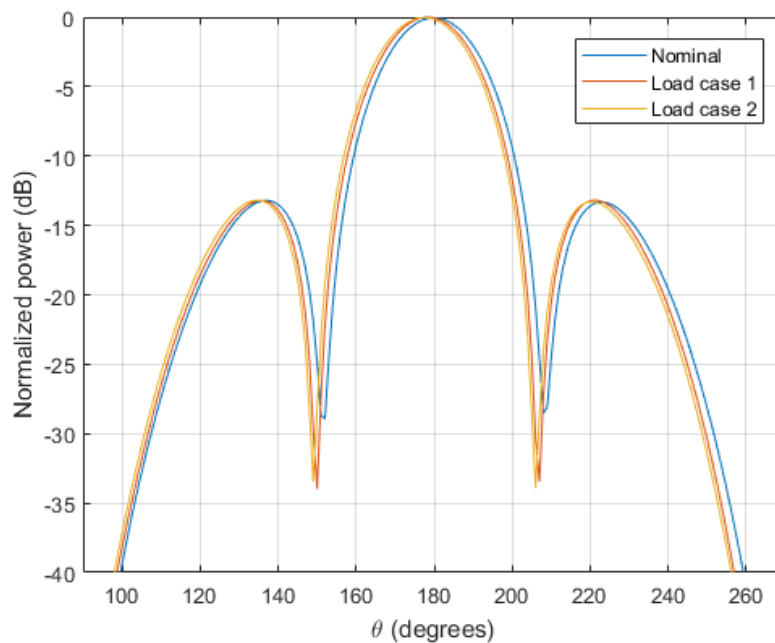


Figure 71. Change in radiation pattern due to deformation.

Table 15. Pattern and array rotation

Load Case	Radiation pattern shift (degrees)	Equivalent rotation (degrees)	Percent difference
1	-2.0	-1.7	14
2	-3.0	-2.6	14

The next section focuses on testing this same array with surrounding geometry included, to see the effects of the surrounding on antenna performance and validate ADT for more complex models.



### **7.7 Test Case 4: MCoRDS Array, With Wing and Pylons**

Building upon the previous two test cases, and to test the capability of ADT with more complex models, this test case includes the full structural model (fairing, pylons, and wings) presented at the beginning of this chapter, using the MCoRDS antenna array in the previous section. The intention is to demonstrate and validate the use of ADT for more complex models, where the environment can have effects on the antenna performance. As in the previous simulations, antenna return loss parameters and radiation patterns are analyzed to determine their change due to deformation, and whether these changes are realistic.

Figure 72 shows the return loss parameter values for the simulations at different deformation levels, the results for which are also summarized in Table 16. The shapes of these plots differ significantly from those of just the array and fairing presented earlier, where models did not include the surroundings. It is therefore reasonable to suppose that these differences are due to the interaction of the antennas with the surrounding metal structures: pylons and wings. The precise nature of these interactions is hard to predict due to the high nonlinearity of the system, and thus demonstrate the need to be able to simulate the full structure. Only the first two load cases and the nominal model from Section 7.3.2 were simulated since the last load case caused meshing problems in HFSS.

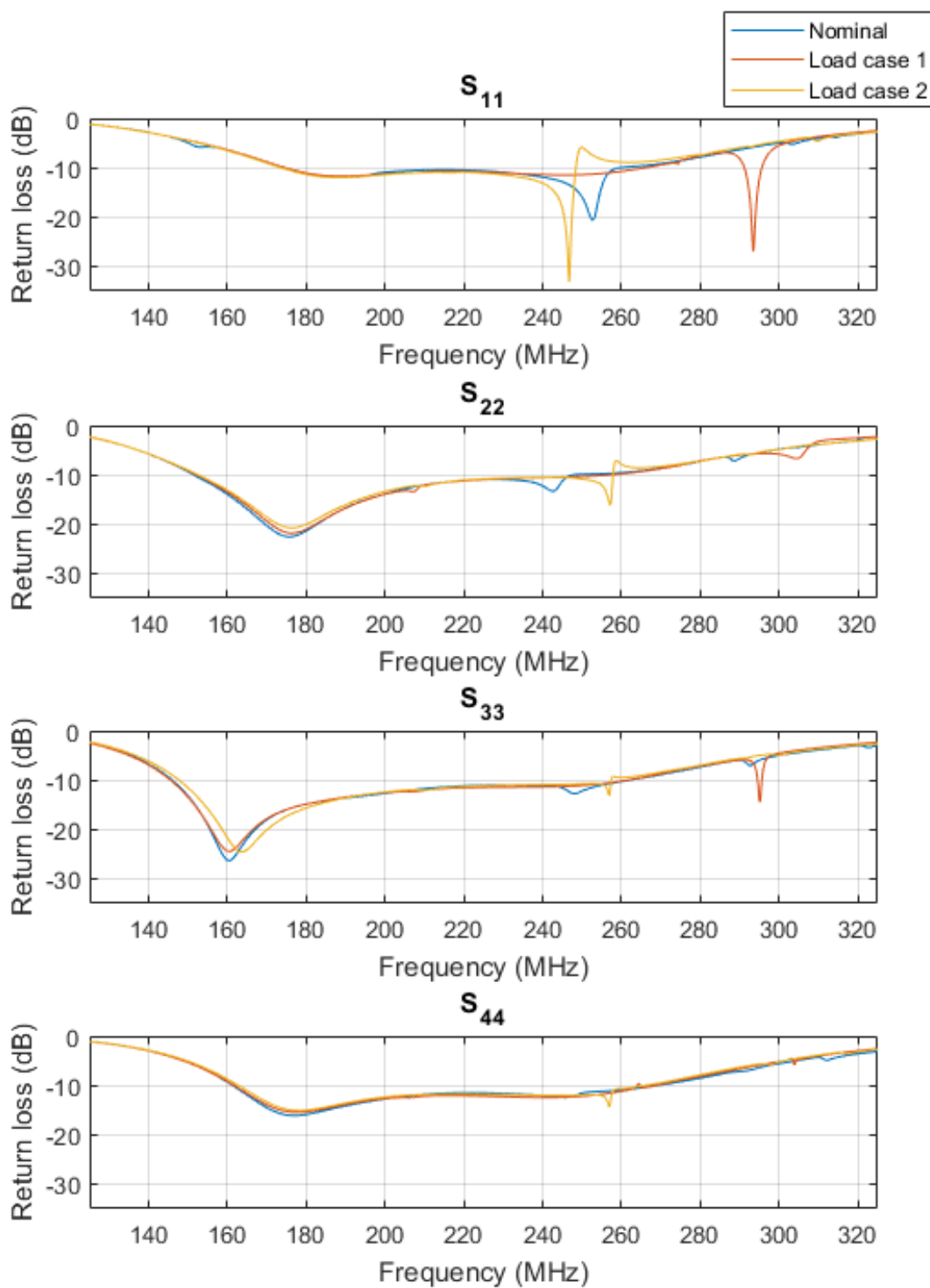


Figure 72. Return loss for each element of array under different load cases.

Table 16. Resonant frequencies and bandwidth for different deformation cases for each array element.

		<b>Resonant Frequency (MHz)</b>	<b>Lower BW limit (MHz)</b>	<b>Higher BW limit (MHz)</b>	<b>Bandwidth (MHz)</b>
S <sub>11</sub>	Nominal	252.7	173.8	259.6	85.8
	Load case 1	293.5	174.7	295.6	120.9
	Load case 2	246.7	173.6	248.4	74.8
S <sub>22</sub>	Nominal	175.7	151.4	247.8	96.4
	Load case 1	176.3	152.1	257.1	105
	Load case 2	176.3	152.6	257.9	105.3
S <sub>33</sub>	Nominal	160.4	145.9	262.1	116.2
	Load case 1	160.4	145.6	261.6	116
	Load case 2	163.6	147.7	257.4	109.7
S <sub>44</sub>	Nominal	177	161.3	269.3	108
	Load case 1	177.6	161.6	268.6	107
	Load case 2	178.2	162.7	266.2	103.5

The resonant frequencies for all deformation cases are much below the resonant frequency for an element without metallic surroundings, which tends to be around 200 MHz, from the previous test case. The apparent resonances in return loss plots (for example at 250 MHz in the S<sub>11</sub> plot for Load Case 2) are caused by the presence of large metallic pylons; in the case of the first element, one pylon is located directly above the middle of the element. To further test that pylon placement along the element length causes the change in the shape of the return plot (and is not some artifact of the ADT geometry), additional simulations were performed, and are detailed in Appendix E. It was found that the return loss was very sensitive to pylon placement, in a nominal (no deformation) case. Most likely, the electric field generated by the antenna creates currents in the pylon, which generate their own radiation that interferes with the antenna element radiation. In particular, the first element (S<sub>11</sub>) seems to be extremely sensitive to the location of the surroundings, and when including element coupling, produces very unpredictable shifts in resonance frequency and bandwidth, which change by up to 16% and

40%, respectively. As stated, these large changes are attributed to the interactions with the surrounding metallic structures and are not caused by faulty geometry generated by ADT. This holds true for the rest of the antenna elements ( $S_{22}$ ,  $S_{33}$ ,  $S_{44}$ ), although the resonance of these elements does not seem to be as sensitive to deformation as the  $S_{11}$ . For these elements, the nulls are more pronounced, and are closer to the expected and there is no consistent decrease in bandwidth with increasing strain, like in the individual case. This is an indication that for arrays with complex geometry (presence of other elements and near-field scatters) changes in array performance are largely driven by changes in coupling between array elements, rather than the changes to individual elements. This observation could only be made with full-wave simulations of deformed geometry, like those enabled by ADT.

In general, ADT is incredibly useful for establishing the sensitivity of element resonance to structural deformation and as well as discovering antenna performance characteristics that may not be easily determined otherwise. While more testing and measurements are encouraged to validate the response generated by the simulations and determine similarity between simulations and actual performance, it is noted that these measurements would be incredibly difficult, if not impractical, to conduct given the large size of the array.

As in the previous test cases, the radiation patterns in Figure 73 for the deformations mostly shift left with increasing deformation, since the array mostly experiences a rotation. The shape of the front and side lobes does not change significantly.

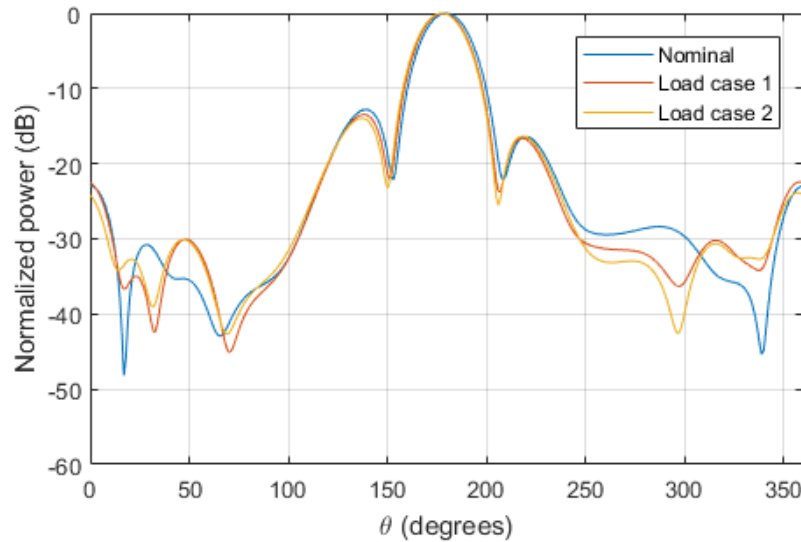


Figure 73. Radiation patterns for the nominal and deformed cases.

The mainbeam rotation is the same as in the model without the wings and pylons, which provides confidence that the results produced by the ADT geometry are accurate.

The largest change in the pattern is observed in the back lobe, most likely due to the relative locations of the pylons (to which the model is highly sensitive, see Appendix E). Just as in the case with just the fairing, the rotation of the radiation pattern differs by about half a degree from the apparent rotation of the array. Table 17 summarizes this information.

Table 17. Radiation pattern parameters for nominal and deformed cases.

	<b>Nominal</b>	<b>Load case 1</b>	<b>Load case 2</b>
Mainlobe direction	180	178	177
HPBW (degrees)	23	23	23
Normalized right null power (dB)	-22.1	-23.8	-25.5
Right null direction (degrees)	208	207	206
Normalized left null power (dB)	-22.2	-22.1	-23.2
Left null direction (degrees)	153	151	150

This case ends the study of the model presented in Section 7.3. The different cases tested progressed through geometry of increasing complexity. These test cases demonstrated that ADT can be used with models of varying complexity, and antennas of different shapes. These test cases also proved that ADT is useful to establish relationships between deformation and antenna performance, particularly resonance, which may be difficult to obtain otherwise. Because a full-wave analysis is performed, these relationships involve complex non-linear systems, and may be difficult to estimate otherwise. Thus, ADT has a great potential to be used for structural and EM coupling studies. This potential is somewhat hampered by the problematic geometric intersections and the need to manually remove them, but this problem can be solved and is left for future work.

Concluding the test cases involving deformation obtained from a structural FEA, the next two test cases involve antennas and arrays and deformed using user defined deformation fields.

### **7.8 Test Case 5: Large Array With User-Specified Deformation Field**

A user-defined, custom deformation field using a B-spline surface definition was created and used to test several deformations of an eight by eight dipole array. This array uses the strip dipole antenna in Table 4, and elements are spaced a half wavelength in both axes. The deformation is described by a symmetric 3<sup>rd</sup> degree bowl shape B-spline surface, the function for which is provided in Appendix F. The maximum depth of the bowl was set at 10%  $\lambda$ , 12.5%  $\lambda$ , 25%  $\lambda$  and 50%  $\lambda$  in the center, approximately. The purpose of testing this array is to highlight the ability to use custom deformation fields, and to validate ADT by comparing the results to the analytical solution of the array factor. This analytical solution does not consider coupling between elements, but is a common approach to assessing effects of array deformation in

literature (Wang, Wang, & Bao, 2008; Braaten, et al., 2013; Peters, et al., 2007) The nominal array used is shown in Figure 74, and one of the 10% wavelength deformation field is shown in Figure 75.

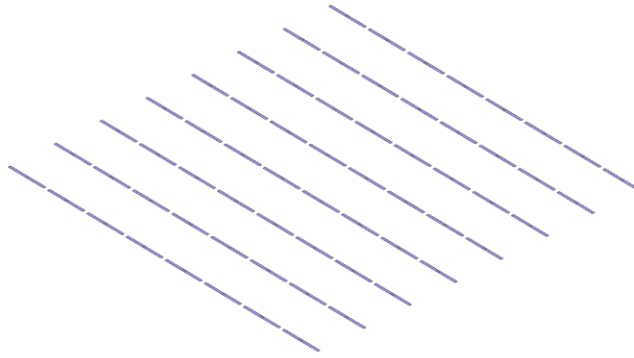


Figure 74 Eight by eight strip dipole antenna array.

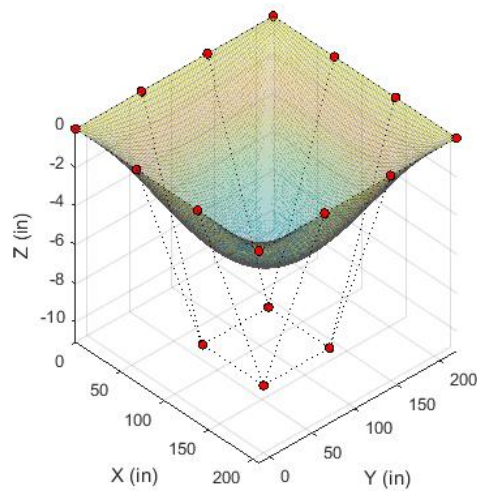


Figure 75. B-spline deformation field for 10% wavelength (Claxton, 2007). Control points shown in red.

Figure 76 and Figure 77 show radiation pattern plots for the simulations in two orthogonal planes, when  $\varphi=0$  and  $\varphi=90$  (aligned along the element nulls), respectively. The trends observed in the plots agree well with what had been published in literature regarding the effects of array deformations on array radiation patterns, namely an increase in sidelobe level, and a decrease in directivity (Wang, Wang, & Bao, 2008; Peters, et al., 2007; Wang H. S., 1992).

Also of note, the changes in the patterns in the two orthogonal planes are relatively the same, which is expected because the deformation is symmetric in the two planes. Differences in the patterns are largely due to the element factor. Because of the shapes of the patterns and the shapes of the elements are not symmetric about both of its axes, the patterns at  $\varphi=0$  and  $\varphi=90$  degrees are slightly different

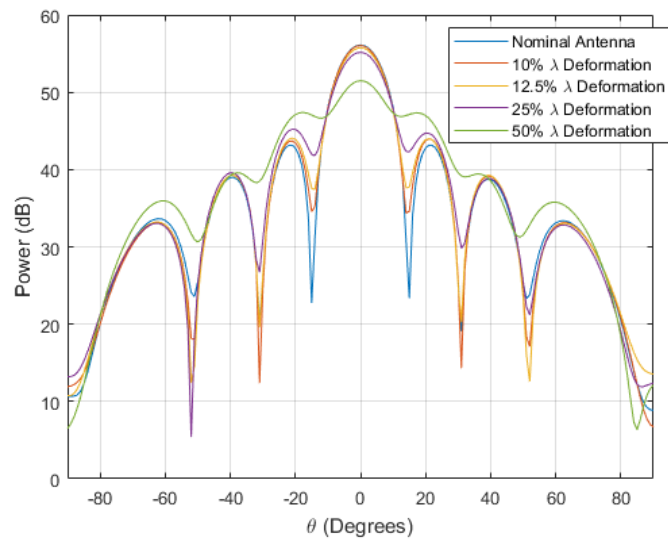


Figure 76. Array radiation pattern,  $\varphi = 0$  degrees, from simulation.

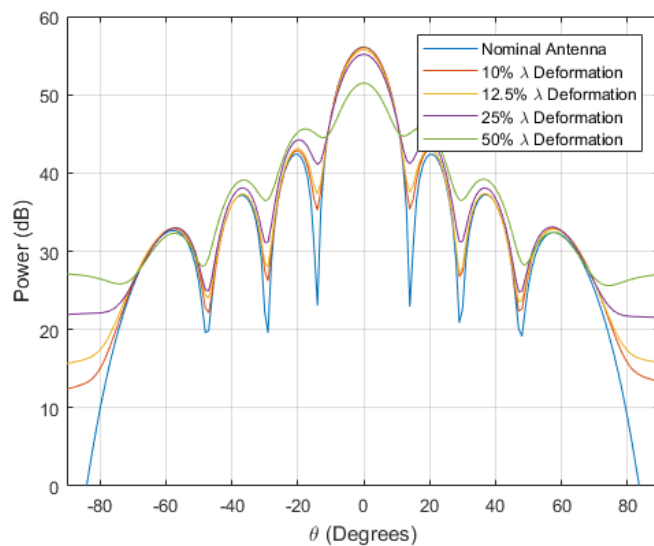


Figure 77. Array radiation pattern,  $\varphi = 90$  degrees, from simulation.



Directivity changes by very little for a 10% and 12.5% wavelength deformation, losing only 0.06 dB, but this loss increases drastically with higher deformations, up to 4.61 dB at 50% wavelength deformation.

Figure 78 compares the normalized array patterns obtained in three different ways: from analytical results, from the simulation using the output of ADT (in which the array elements themselves are deformed), and from a simulation with the array element centers at their deformed locations, but with the elements themselves in their original shape. As Figure 78 shows, patterns agree within 3 dB between the analytical results and nominal element results, but there are significant differences between analytical results and deformed element simulation results.

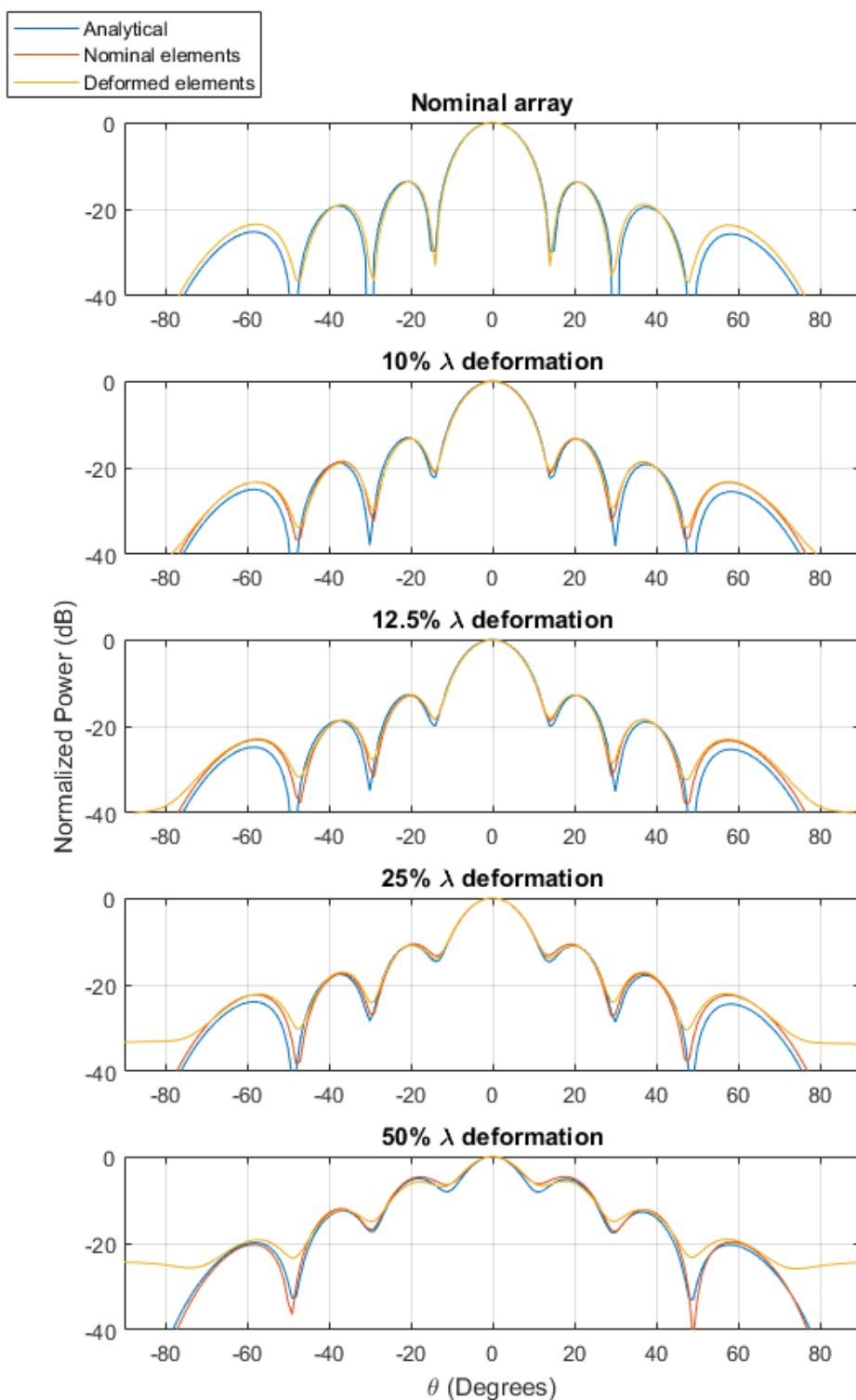


Figure 78. Comparison of normalized power between analytical results (blue), simulations with non-deformed elements, (red), and simulations with deformed elements (yellow), when  $\varphi=90$  degrees. The order, top to bottom, is the following: nominal array, 10% wavelength deformation, 12.5% deformation, 25% deformation, 50% deformation.

Overall, the analytical results show smaller sidelobes than either of the simulations, and the HPBW is one degree larger in the simulations (13 degrees). Across deformation cases the sidelobe level increase is the same for the analytical results and simulations for angles less than 40°. The differences in the simulated and analytical nominal patterns is due to the coupling in the elements (this is not taken into account in the analytical solutions). While the shapes of the simulated radiation pattern agree well for the first deformation case, they begin to diverge for the subsequent deformation cases. This indicates that element shape is influential in the overall pattern shape and not just the positional phase center error. Since the individual element patterns contribute to the total pattern, changes in these radiation pattern is the explanation between the two simulations. This is something which would have been hard to discover without ADT, either by using analytical calculations or just nominal elements at different positions. The most dramatic effect occurs at the edges of the pattern (-90 to -70 degrees and 70 to 90 degrees). At these locations, the difference between analytical results and simulations can reach up to 15 dB, although these results have not been validated experimentally.

These results are comparable to analytical and simulation results. They also capture the utility of ADT, by demonstrating that element deformation, an effect not always taken into account during structural-electromagnetic coupling, can have a significant impact on radiation patterns. This model simulation also revealed the need for the automatization of port and antenna property assignment, since assigning 64 ports for each of the five simulation cases is a tedious manual task. Additional experimental tests would be necessary to fully validate ADT, but these may be difficult to perform on a large array due to its size.

### 7.9 Test Case 6: Conformal Deformation of UWB Antenna Element

As described in the previous test case, ADT can deform antennas to a prescribed deformation field, which is useful in the study of conformal antenna design. This ability can be used to conform the antenna shape to, for example, an airfoil. For this test case, the UWB element from Table 4 was deformed using the equation of a simply supported beam, (shown in Equation 32) and the simulation results for a nominal and deformed case were compared. A ground plane was used in the simulations, but an antenna substrate was not included, the reasons for which are detailed later in this section.

$$\begin{aligned}
 z &= \alpha x(l^3 - 2lx^2 + x^3) \\
 \alpha &= 3.0516 \times 10^{-6} \\
 l &= 32
 \end{aligned}
 \tag{32}$$

In Equation 32,  $\alpha$  is a constant representing load and cross section,  $l$  is the long side of the antenna (measured in inches),  $x$  is the position along the x axis, and  $z$  is the resulting displacement in the z axis. Values in the y axis are left unchanged. This test case is slightly different from the previous one in that the function used is not based on B-splines, and so it does not require being normalized with respect to a certain surface or volume. This is an important addition to ADT because it allows modelling distortions with custom explicit equations, where the domain need not be necessarily limited. For some applications this may be advantageous because a B-spline is always limited by its control grid. The nominal and deformed antenna element configurations are shown in Figure 79. To obtain this configuration, a deformation field in Equations 32 was used.

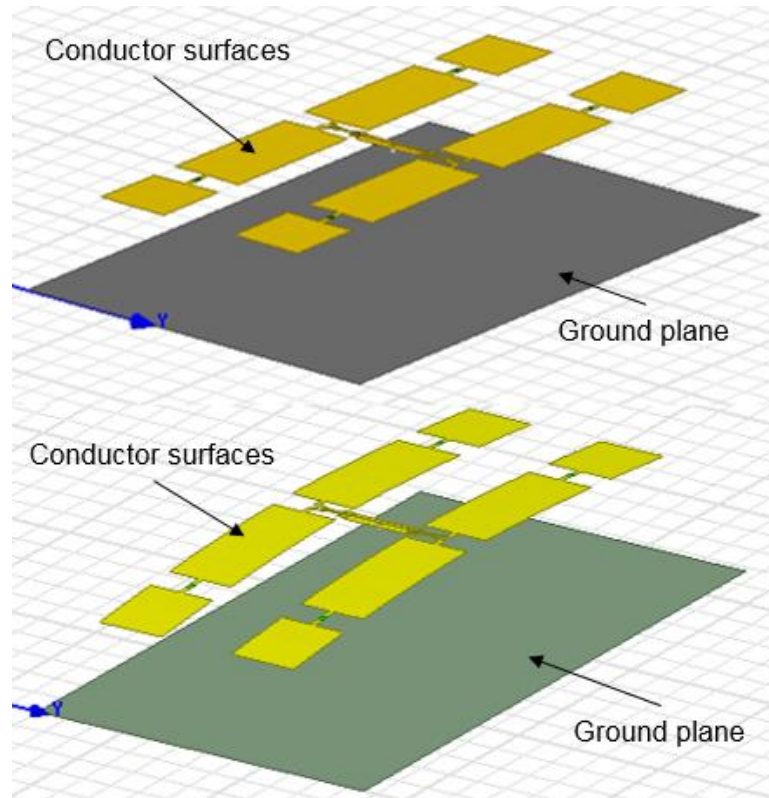


Figure 79. Nominal (above) and conformal (below) UWB antenna configurations.

Although the original antenna CAD model included an antenna substrate, through which cylindrical vias and a cylindrical port pass (shown in Figure 80), this substrate was not modelled. The substrate of the antenna shown in Figure 79 had to be omitted due to a suspected intersection between the substrate and the conductors. However, the nominal antenna was modelled with and without the substrate, and it was found that the omission of the substrate did not significantly affect the bandwidth or radiation pattern. In addition the vias and ports were modeled in post-processing because ADT currently lacks the ability to interpret and deform cylindrical (and other conical) shapes and surfaces. This ability will be added in the future.

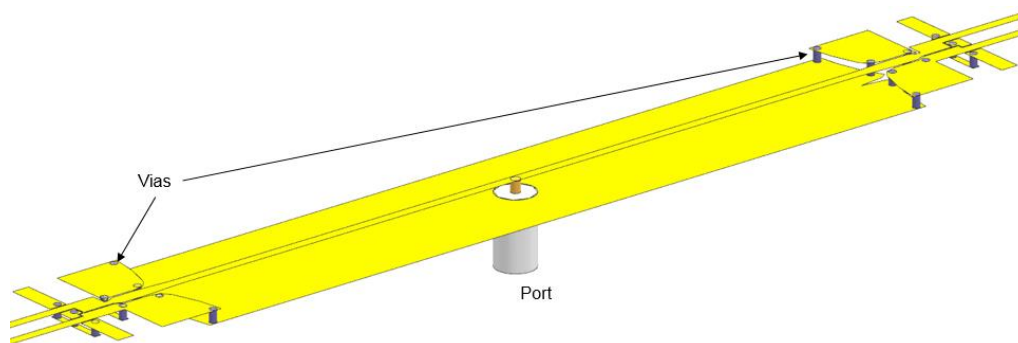


Figure 80. Port and vias on UWB antenna.

As in the P3 array simulations, radiating boundary conditions were used with the bounding box consisting of a vacuum extending at least a quarter wavelength away from any radiating surfaces. All conductors and vias were modelled with finite conductivity using the standard HFSS library properties of copper. Four resistances of 200 ohms each, modelled as lumped resistances, were used to connect the conductors at the points shown in Figure 81. The port was modelled as a lumped port, with a resistance of 50 ohm.

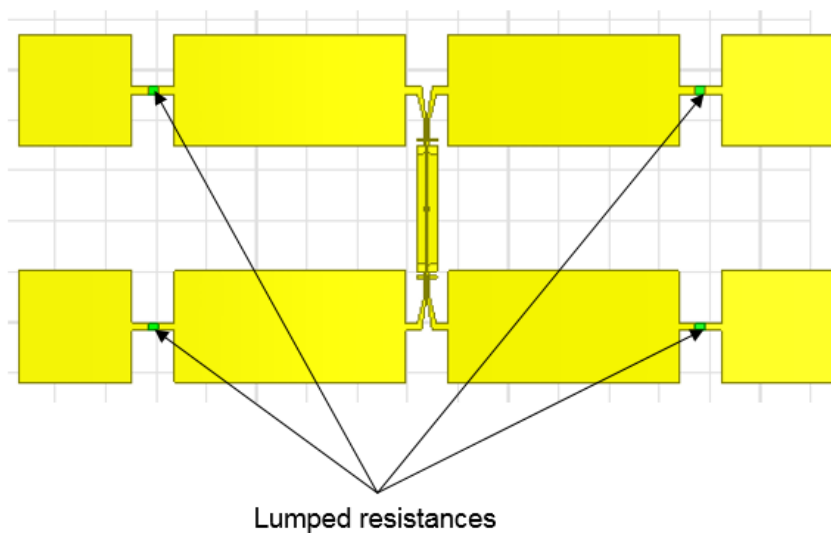


Figure 81. Resistances modelled in UWB antenna.

As with previous simulations, the antenna return loss ( $S_{11}$ ) and radiation pattern were examined to assess the validity of the simulation, and the utility of ADT. Because the deformation is symmetric about the midline of the antenna, only minor broadening of the main lobe is expected. In addition, small changes in the antenna resonating frequency are expected due to the strain applied on the antenna element.

Figure 82 shows the return loss plot of the UWB antenna. Although there is a change in the overall shape of the return loss plot with deformation, the bandwidth and center frequency remain the same for both the nominal and conformal antenna. Nevertheless, the change in the plot shape demonstrates that it is still desirable to simulate the conformal case, as different deformation may have different effects which cannot be readily predicted without the use of ADT. Table 18 summarizes these changes.

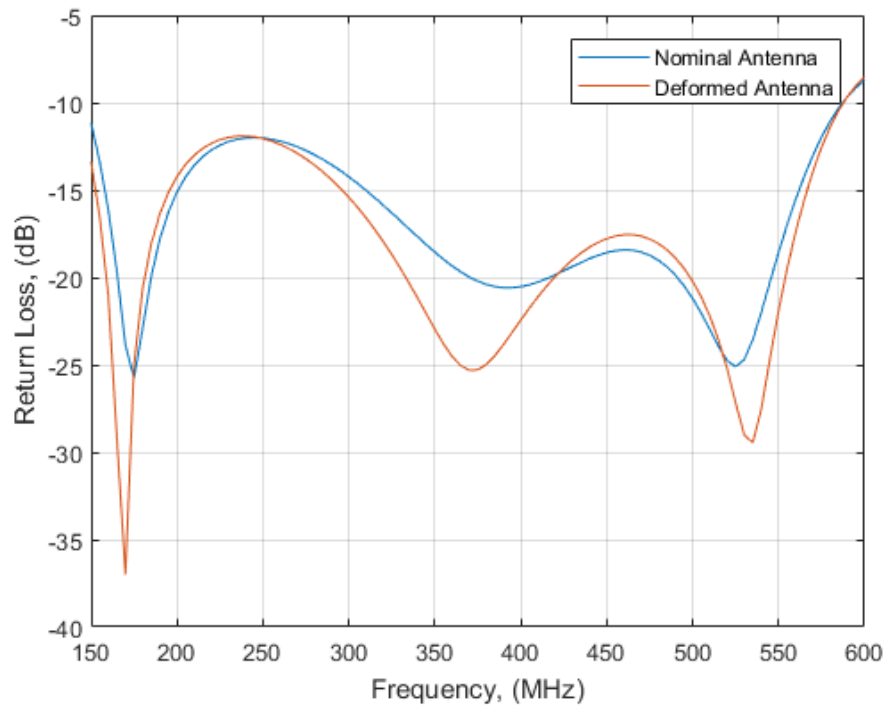


Figure 82. Return loss of UWB antenna.

Table 18. Changes in UWB resonance characteristics.

Characteristic	Nominal	Conformal
Length (in)	31.98	32.06
Change in length (in)	0	0.08
Strain (in/in)	0	0.003
Lower BW limit (MHz)	150	150
Higher BW limit (MHz)	590	590
Bandwidth (MHz)	440	440
Center frequency (MHz)	370	370

Figure 83 and Table 19 show the antenna radiation pattern, which does not change significantly due to the broad element pattern and symmetric deformation. The HPBW of the deformed antenna does widen by five degrees, similarly, the back lobe narrows slightly. These changes in the pattern shape reflect the changes in the antenna geometry and are expected.

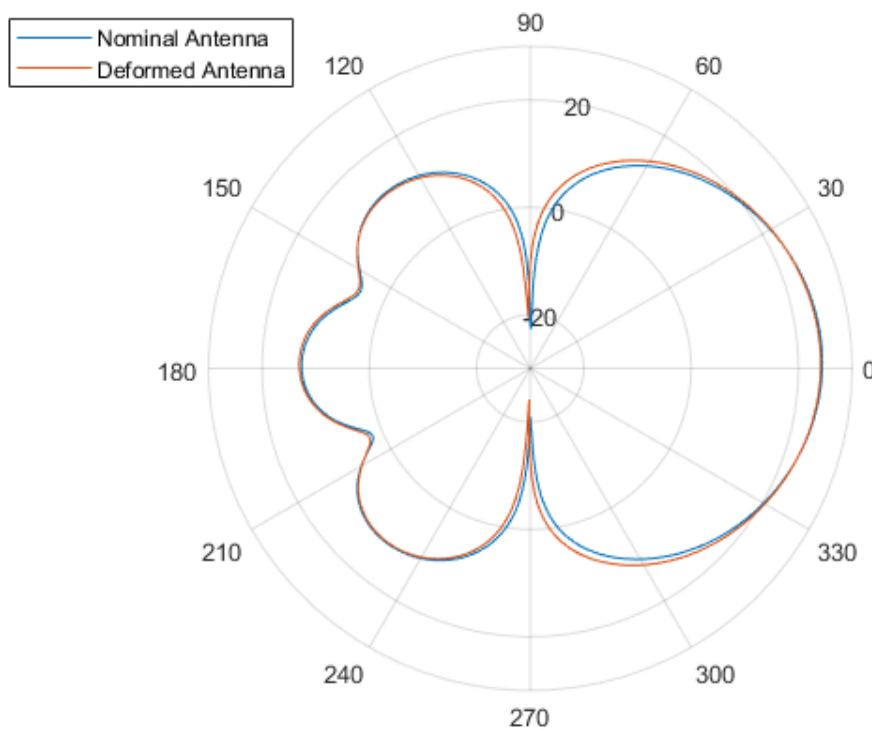


Figure 83. UWB antenna radiation patterns.



Table 19. Changes in UWB radiation pattern due to deformation.

Characteristic	Nominal	Conformal	Change
Mainlobe power (dB)	24.45	24.16	-0.27
Mainlobe direction (degrees)	2	4	2
HPBW (degrees)	55	60	5
Right null power (dB)	-21.04	-24.18	-3.14
Right null direction (degrees)	-90	-92	-2
Left null power (dB)	-22.68	-20.66	2.02
Left null direction (degrees)	89	92	3

From these trends observed in the results, there can be confidence that ADT produces usable geometry for simulating conformal antennas whose geometry is produced from a nominal antenna and deformation field. Additional work is required to ensure that intersections between geometry do not occur, which is left for the future.

## 7.10 Summary of Results

This chapter has demonstrated the utility of ADT using several different antenna models and configurations. In particular, six different cases were tested, involving individual antenna elements, relatively small antenna arrays with surroundings, and large arrays. Four of the tested cases used a geometric model derived from a structural FEM, with different antennas, and two of the test cases used antenna models created in CAD and deformed through a user-specified deformation field. The results of the obtained simulations were compared to literature, similar simulations, and measurements (when available).

In particular, it was demonstrated that ADT is useful to establish the sensitivity of antenna resonance to structural deformation. Some antenna array models were shown to be

particularly sensitive to small deformations, and it would have been difficult to discover this relationship without ADT. It was also demonstrated that ADT can reliably produce deformed geometry. ADT's abilities are limited to geometry that is relatively easily interpreted and parameterized by the software. Occasionally geometry may require some post-processing to eliminate intersections which may cause errors in HFSS. It was found that for surfaces where small intersections or gaps may exist, the insertion of induced offsets is necessary to generate a valid model capable of producing results. The effects of this offset should be characterized for any model with which ADT is used.

In addition, ADT has been shown to be useful in demonstrating the effect of deformation of an antenna element itself on array performance, which is not usually taken into consideration by analytical solutions and which can have a considerable effect at higher deformation levels. This may prove to be particularly useful in the design of conformal antenna arrays.

Finally, the results in this section have revealed the need for three major improvements: a better surface fitting algorithm to reduce surface error, a reliable and smart algorithm to avoid surface intersections and implement surface continuity, and the need for further automatization of certain features, such as material and port assignment. Correcting for intersections and setting up simulations take a considerable amount of time, which is currently a bottleneck in the process.

## Chapter 8: Summary, Conclusions and Future Work

### 8.1 Summary and Discussion of Software Requirements

The Antenna Deformation Tool (ADT) software program was developed to address the need for advanced design tools for airborne antenna arrays. ADT is capable of generating deformed CAD geometry from the output of a structural FEA model. The resulting geometry was required to be of sufficient quality for use in subsequent HFSS EM simulations, and several examples were presented to illustrate the satisfactory performance of the program. The following is an in-depth discussion of the software requirements that were identified by the author prior to software development and the performance of the software with respect to those requirements. The list of requirements is repeated here for convenience, and commented.

1. *ADT must be able to successfully interpret NASTRAN files*

ADT has demonstrated the ability to read and interpret NASTRAN files, however its capabilities are currently limited to reading and interpreting quad and tri shell elements. ADT must be improved in the future to read and interpret other types of elements and element properties.

2. *ADT must be able to export the generated geometry into a format which can be read by HFSS*

ADT uses the STEP AP 203 standard to export CAD geometry. This geometry can be imported by HFSS, although HFSS's incorporated geometry healing function is sometimes necessary to ensure that all entities are indeed valid. Generally, the geometry generated from structural deformation is imported into HFSS more easily than the geometry generated from the deformation of antenna CAD models, especially when this geometry involves solids. This point is expanded on in the next item.

3. *ADT must be able to create usable geometry for HFSS. “Usable” is defined as:*

a. *The geometry does not significantly impact the graphical performance of HFSS*

The geometry which was created by ADT does not negatively impact the graphical performance of HFSS. Unlike the BDF and STL files initially tested (which use hundreds to thousands of faceted surfaces to represent objects), the continuous solids and sheets produced by ADT do not impose unnecessary meshing requirements.

b. *The geometry contains the correct type of surfaces and solids, regardless of whether the NASTRAN input is a surface or a solid, which implies the creation of a surface thickening mechanism for thin sheets, with user control over which sheets to thicken.*

A surface thickening algorithm has been included in the ADT process flow. This algorithm is rather simple and creates valid solids. However, user control over the direction of the offset has been difficult to implement, due to the need to clearly visualize this direction. Better visualization, and graphical user interaction would enormously facilitate this task. It would also allow equivalencing and implementing G0 continuity and avoiding intersections but this is left as a task for future improvement of the software. This is also discussed in the next point.

c. *The geometry does not cause processing or meshing errors in HFSS*

While several successful implementations of ADT were presented, further improvement regarding geometry refinement is required. The ADT-generated geometry often requires post-processing to remove all possible intersections, which causes errors and prevents the simulations from running. This is due to the

lack of an equivalencing mechanism to ensure G0 continuity and geometry associativity. The lack of these qualities causes the biggest problem with the geometry generated by ADT – this geometry is not guaranteed to not produce errors. This will be further discussed in the conclusions and recommendations section.

- d. *The geometry surface error is small enough to be representative of the NASTRAN mesh.*

Surface assessment has shown that maximum surface error varies significantly from model to model, and errors depend on the quality and fineness of the surface mesh. In general, surface error has been shown to be comparable to what has been published by other researchers for similar types of surface fitting. However, large errors are present in curved surfaces due to the gridding mechanism used. This is discussed further in the conclusions.

4. *ADT must be able to correctly deform antenna designs*

This is one of the main features of ADT, and it was successfully demonstrated for several antenna designs; however, as in item 3 above, the exported geometry must be post-processed to remedy intersections if the input geometry contains solids.

5. *ADT must be robust, and must be able to correctly interpret as wide a range of structural models as possible.*

While ADT has been demonstrated with rather basic models that only include CTRIA3 and CQUAD4 elements, code has yet to be developed to support all types of elements supported by most FEA programs. In addition, ADT has limited capabilities when meshing and rebuilding surfaces that lack four easily identifiable edges. However, results

in Chapter 7 illustrate ADT's ability to process geometry common to aircraft and antenna fairing structures. More improvements to robustness must be implemented in the future.

6. *Since it is intended for general research use, ADT must be easy to use*

In its current form, ADT requires improvement. Although general overarching commands exist to load, process, and export a model (which can be done in just three command lines) more detailed work and setting of user parameters requires source code manipulation.

7. *Although speed is not a priority during initial development, processing speed must still be reasonable*

Processing times are, as expected, slow (around six minutes for 15,000 vertices), and increase with model complexity, but they are reasonable for an initial version of ADT.

Careful optimization and memory allocation, and smarter algorithms have the potential of greatly improving performance. Additionally, ADT can be exported as an executable using MATLAB's app packaging capacity, which would also increase performance speed.

Overall, ADT fully meets three of the seven main design requirements set forth by the author as intended (speed, correctly deform antennas, export geometry), and partially meets two (usable geometry, interpret NASTRAN files). Two requirements that were that were not sufficiently demonstrated include robustness and simplicity. Though the program has successfully generated conceptual airborne array geometry, there are clearly improvements that could be made to the program to increase the robustness of the program as well as to make it more user friendly. These improvements will be discussed in the next section in the context of the conclusions and recommendations.

## 8.2 Conclusions and Recommendations

The following conclusions and recommendations are made about ADT:

1. ADT has been designed and optimized for use with the NASTRAN structural FEA solver and HFSS EM software. While ADT has not been demonstrated with any other software packages, due to its use of common file structures, it is expected that the ADT would be compatible with any structural software packages that output BDF and .f06 files and any 3D EM software packages that can read in STEP files.
2. ADT still lacks robustness to interpret more complex structural models. Most notable is the lack of support for elements other than CTRIA3 and CQUAD4, and properties other than shell elements. This limitation can easily be addressed by integrating code to interpret these other types of elements and properties. Additionally, ADT has limited capabilities for meshing and rebuilding surfaces which cannot be easily molded into four sided patches for parameterization. Although corners can be specified manually to alleviate this problem, in practice this is a tedious task, especially since ADT lacks a graphical interactive interface. Failure to establish corners correctly is one of the major causes of failed surface generation. Smarter edge recognition and corner detection algorithms are necessary to fix this problem. Additionally, ADT cannot yet distinguish internal and external edges for meshes with holes, and a reliable parameterization and trimming of this type of mesh has yet to be implemented. Free boundary parameterization can help solve this problem.
3. Mesh regularization ended up being one of the most difficult challenges to address in generating deformed geometry. In the end three different regularization approaches are implemented in ADT. In general, the k-harmonic is the default, preferred approach, since

it has shown stable performance and fast running speeds. The other two, LSCM and mesh gridded resampling, are backups. Although no failing case has been yet found for k-harmonic parameterization, in such a case, LSCM is used first to attempt parameterization. In the event LSCM is not successful, mesh gridded resampling is attempted. Mesh gridded resampling is the last option due to its slow implementation. A failed parameterization can be detected by either a runtime error, or extremely skewed elements.

4. While ADT produces smooth and continuous solids and surfaces from point clouds, there is an inherent probability of intersecting surfaces due to the non-associative nature of the individual geometric element definitions. In addition, even though in some cases HFSS geometry validation does not identify an intersection, small gaps (on the order of 0.005 inches for a solution frequency of 200 MHz) between surfaces seem to cause meshing errors.

To mitigate processing and meshing errors, small offsets were introduced to the geometry, when these are found to be needed based on HFSS geometry validation and visual inspection. For an antenna operating at 200 MHz effects of gaps of up to 0.001 inches were found to have insignificant effects on radiation pattern, but they did cause a 1.5% shift in the resonance frequency, and a 3-4% shift in bandwidth. While this is a relatively small change, users should be aware of this limitation in the software. For any given application, it is recommended that nominal arrays both with and without the offset be simulated so that there is a better comparison to the deformed array. In addition, it is also recommended that the effects of this gap be fully characterized across a wide range



of frequencies in future work. This would be helpful in determining when it is best not to use ADT.

An implementation of an algorithm for checking and removing intersections and small gaps is necessary to fix this, though there is yet no clear direction as to how to implement this. Most likely this would involve an approach similar to the connectivity matrix in Chapter 6, where surfaces would be checked for intersecting curves, rather than edges be checked for intersecting vertices. The incorporation of a graphical user interface might also help alleviate this problem.

5. Surface quality for meshes which have no curvature is good, and in in general consistent with values found in literature (Eck & Hoppe, 1996; Berger, Levine, Nonato, Taubin, & Silva, 2013). Surface errors of flat surfaces was found to be less than 0.5% of the average edge length of the mesh (or on the order of  $1e-3$  model units) for the models tested. However error for curved surfaces are significant, -- around 3.5% of average edge length (or about 0.003% of the mesh diagonal) This is caused by the mesh gridding approach used, explained in Section 7.3.3. These errors add to the possibility of intersecting adjacent surfaces that prevent the simulation from running. Surface quality could be improved by implementing more advanced surface fitting algorithms, which make use of point weighting, knot insertion and removal, and variation of control points, to set the surface to within a user specified tolerance; however, this will be left for future work. Implementing associative geometry for structural meshes would solve this problem completely, and is also left for future work.
6. With regards to antenna deformation validity, although comparison to measured results has been limited, the initial tests suggest that the results of the antenna deformation

simulations are within the expected range, and are comparable to analytical results.

Additional simulations and experimental antenna measurements must be completed to more accurately assess this. Similar issues related to surface intersection were also present in the antenna deformation process. These intersections arise from the lack of associativity in the geometry defined, which causes small intersections when coplanar surfaces are deformed and fitted using slightly different data sets. These intersections have been found to be on the order of  $1e-6$  inches deep. Fixing these intersections would imply detecting and keeping track of coplanar surfaces, and ensuring that they have associative definitions when deformed.

7. Although suitable for work with simpler models, a more involved user interface should be developed to allow greater user control over model generation, which is intended to be one of the main advantages of ADT over other software with similar capabilities. In general, it has been found that the balance between ease of use and amount of control is a very delicate one. The focus of ADT for this thesis has been on automating most procedures and reducing user involvement. Future development should focus on allowing easier user control in areas like corner detection tolerance and selection, mesh dihedral adjacency tolerance setting, thickening and offset direction and control, and visualization. It would also allow to more easily set continuity conditions between surfaces, to ensure that G0, G1 and G2 continuity is met. This might require the implementation of an interactive graphical user interface, as well as dedicated configuration files with a user's presets.

8. ADT can be integrated into the design process at the detailed design stage. It is recommended that deformed antenna trade studies be performed concurrently with the detailed structural and EM analyses.

In addition to the functional improvements which have been suggested above, future work could leverage ADT for the development of a custom FEA program specifically designed for the coupling of structural and EM simulations. The current version of ADT could serve as a basis for importing, exporting and processing geometry, and the libraries and classes used to develop ADT might be useful to develop a stand-alone FEM tool capable of performing basic FEM simulations, which could be developed further into a general Multidisciplinary Optimization (MDO) platform, such as those developed by Wang, Duan & Shao (2006). Development of a custom program may also allow the integration of CAD and FEM to perform IGA analysis, which would not require geometry conversion between CAD and FEM analysis. Having a custom software structural/EM coupling software using IGA could greatly simplify the task of performing multiphysics analysis by foregoing the problems introduced by using different geometric representations, namely by using meshes and gridding them, which could decrease simulation time and increase accuracy.

In conclusion, the utility of ADT has been demonstrated for coupled structural and EM simulations, as well as for the design and simulation of conformal antennas. A major advantage of ADT over existing methods is that does not require the antenna model to be included in the structural FEM model, which considerably eases workload, and is a unique feature of the program. The separation and seamless integration of a structural and antenna deformation submodules allows to easily generate several CAD models corresponding to different load cases in a short timeframe, and to easily swap different antenna designs onto existing FEA models.

This feature considerably expedites the process of testing various antenna configurations designed for a similar role, or to be integrated on the same platform. Although improvements remain to be made, ADT is a first step into integrating structural and EM simulations, reducing engineering hours, improving existing simulation capabilities, and designing better remote sensing systems.

## References

- 3D Systems. (2017). Geomagic. Rock Hill, South Carolina: 3D Systems.
- Aerospace Specification Metals, Inc. (2018). *Aluminum 2024-T3*. Retrieved from Aerospace specification metals, Inc.:  
<http://asm.matweb.com/search/SpecificMaterial.asp?bassnum=ma2024t3>
- ANSYS. (2014). ANSYS HFSS 2014.0.7. Cannonsburg, Pennsylvania: ANSYS.
- ANSYS. (2017). ANSYS Workbench. Cannonsburg, Pennsylvania: ANSYS.
- ANSYS. (2018). *ANSYS HFSS capabilities*. Retrieved from ANSYS: <http://www.ansys.com/it-it/products/electronics/ansys-hfss/hfss-capabilities>
- Arnold, E., Yan, J., Hale, R., Rodriguez-Morales, F., & Gogineni, P. (2014). Identifying and compensating for phase center errors in wing-mounted phased arrays for ice sheet sounding. *IEEE Transaction on Antennas and Propagation*, 60(6), 3416 - 3421.
- Autodesk. (2017). Maya. San Rafael, California: Autodesk.
- AZO Materials. (2018). *S-glass fibre*. Retrieved from AZO materials:  
<https://www.azom.com/properties.aspx?ArticleID=769>
- AZO Materials. (2018). *Stainless steel - grade 304 (UNS S30400)*. Retrieved from AZO materials: <https://www.azom.com/properties.aspx?ArticleID=965>
- Balanis, C. A. (2005). *Antenna theory, analysis and design, 3rd ed.* Hoboken, NJ: Wiley.
- Berger, M., Levine, J. A., Nonato, L. G., Taubin, G., & Silva, C. T. (2013). A benchmark for surface reconstruction. *ACM Transactions on Graphics*, 20:1-20:17.
- Braaten, B. D., Roy, S., Nariyal, S., Al Aziz, M., Chambelain, N. F., Irfanullah, I., . . . Anagnostou, D. F. (2013, February). A self-adapting flexible (SELFLEX) antenna array

- for changing conformal surface applications. *IEEE Transactions on Antennas and Propagation*, 655-665.
- Bui Thong, P. (1975). Illumination for computer generated pictures. *Communications of the ACM*, 311-317.
- Byers, K., Harish, A. R., Seguin, S., Rodriguez-Morales, F., Paden, J., Arnold, E., & Hale, R. (2012). A modified wideband dipole antenna for an airborne VHF ice penetrating radar. *IEEE Transactions on Instrumentation and Measurement*, 1435 - 1444.
- Canadian Space Agency. (2015, October 7). *Satellite characteristics*. Retrieved from Canadian space agency: <http://www.asc-csa.gc.ca/eng/satellites/radarsat/radarsat-tableau.asp>
- Canadian Space Agency. (2017, December 14). *RADARSAT-2*. Retrieved from Canadian space agency: <http://www.asc-csa.gc.ca/eng/satellites/radarsat2/>
- Claxton, D. (2007, November 20). *GNURBS*. Retrieved from MathWorks.
- Danziger, Z. (2013, April 03). *Hausdorff Distance*. Retrieved from MathWorks: <https://www.mathworks.com/matlabcentral/fileexchange/26738-hausdorff-distance>
- D'Errico, J. (2014, May 23). *A suite of minimal bounding objects*. Retrieved from MathWorks: <https://www.mathworks.com/matlabcentral/fileexchange/34767-a-suite-of-minimal-bounding-objects?focused=3820664&tab=function>
- Dielectric Corporation. (2018). *G-10 / FR-4 / G-11 (glass-epoxy) general material properties*. Retrieved from Dielectric corporation: <http://www.dielectriccorp.com/downloads/thermosets/glass-epoxy.pdf>
- Dijkstra, E. W. (1959). A note on two problems in connexion with graphs. *Numerische Mathematik*, 269-271.

- Duan, B. Y., Wang, C. S., & Zhang, F. S. (2004). On multidisciplinary optimization of antenna structures. *10th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*. Albany, NY.
- Eck, M., & Hoppe, H. (1996). Automatic reconstruction of B-spline surfaces of arbitrary topological type. *ACM SIGGRAPH* (pp. 325-334). New Orleans: ACM.
- Emmons, G. H. (1991). *Compensating for groundplane deformations of a space-based radar to improve clutter cancellation performance*. Lexington, MA: Lincoln Laboratory, Massachusetts Institute of Technology.
- Fraunhofer-Gesellschaft. (2017). MpCCI. Munich, Germany: Fraunhofer-Gesellschaft.
- Gogineni, S., Wong, K. K., & Lytle, V. (2003). An ultra-wideband radar for measurements of snow thickness over sea ice. *Proceedings of the IEEE Geoscience and Remote Sensing Symposium*. Toulouse: IEEE.
- Gustafsson, K., McCarthy, F., & Paulra, A. (1996). Mitigation of wing flexure induced errors for airborne direction-finding applications. *IEEE Transactions on Signal Processing*, 44(2), 296 - 304.
- Haugse, E. D., Ridgway, R. I., Hightower, C. H., & Warren, R. C. (2001). *US Patent No. 6,333,712 B1*.
- Hughes, T. J., Cottrell, J. A., & Bazilevs, Y. (2005). Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement. *Computer Methods in Applied Mechanics and Engineering*, 4135–4195.
- Jacobson, A. (2017, December 18). gptoolbox.

- Jacobson, A., Tosun, E., Sorkine, O., & Zorin, D. (2010). Mixed finite elements for variational surface modeling. *Eurographics Symposium on Geometry Processing*. Lyon: Eurographics .
- Joshi, P., Meyer, M., DeRose, T., Green, B., & Sanocki, T. (2007). Harmonic coordinates for character articulation. *Proceedings of ACM SIGGRAPH* (pp. 71-1-71-9). San Diego: ACM.
- Kirk, J. (2015, March 13). Dijkstra's minimum cost path algorithm. MathWorks.
- Knott, P. (2010). Deformation and vibration of conformal antenna arrays and compensation techniques.
- Krishnamurthy, V. (2000). *Fitting smooth surfaces to dense polygon meshes*. Stanford, CA: Stanford Univeristy.
- Lane, S. A., Murphey, T. W., & Zatman, M. (2011). Overview of the innovative space-based radar antenna. *Journal of Spacecraft and Rockets*, 135-145.
- Leuschen, C., Hale, R., Keshmiri, S., Yan, J.-B., Rodriguez-Morales, F., Mahmood, A., & Gogineni, S. (2014). UAS-based radar sounding of the polar ice sheets. *Geoscience and Remote Sensing Magazine*, 8-17.
- Levy, B., Petitjean, S., Ray, N., & Maillot, J. (2002). Least squares conformal maps for automatic texture atlas generation. *SIGGRAPH* (pp. 362-371). San Antonio: ACM.
- Lievens, H., & Verhoest, N. E. (2012). Spatial and temporal soil moisture estimation from RADARSAT-2 imagery over Flevoland, The Netherlands. *Journal of Hydrology*, 44-56.
- Liu, C., Shang, J., Vachon, P. W., & McNarin, H. (2013). Multiyear crop monitoring using polarimetric RADARSAT-2 data. *IEEE Transactions on Geoscience and Remote Sensing*, 2227-2240.



- Marghany, M. (2014). Utilization of a genetic algorithm for the automatic detection of oil spill from RADARSAT-2 SAR satellite data. *Marine Pollution Bulletin*, 20-29.
- Mendoza Strilchuk, P. M., & Arnold, E. J. (2017). Structural-electromagnetic simulation coupling tool for airborne antenna arrays. *IEEE Aerospace Conference*. Big Sky, MT: IEEE.
- MSC. (2014). Patran/NASTRAN version 2014a. Santa Ana, California: MSC Software.
- MSC. (2016). *NASTRAN Quick reference guide*. MSC Software.
- Mullen, P., Tong, Y., Alliez, P., & Desbrun, M. (2008). Spectral conformal parameterization. *Eurographics Symposium on Geometry Processing*. Copenhagen: Blackwell Publishing.
- Narasimhan, L. (2014, May 14). *Generalised bezier curve matlab code*. Retrieved from MathWorks: <https://www.mathworks.com/matlabcentral/fileexchange/33828-generalised-bezier-curve-matlab-code>
- NATO. (2017, September 28). *AWACS: NATO's eyes in the sky*. Retrieved from North Atlantic Treaty Organization: [https://www.nato.int/cps/en/natohq/topics\\_48904.htm?selectedLocale=uk](https://www.nato.int/cps/en/natohq/topics_48904.htm?selectedLocale=uk)
- Peters, E., Blankenship, D. D., Carter, S. P., Kempf, S. D., Young, D. A., & Holt, J. W. (2007). Along-track focusing of airborne radar sounding data from west Antarctica for improving basal reflection analysis and layer detection. *IEEE Transactions on Geoscience and Remote Sensing*, 2725-2736.
- Peyre, G. (2009, July 19). *Toolbox Graph*. Retrieved from MathWorks: <https://www.mathworks.com/matlabcentral/fileexchange/5355-toolbox-graph>
- Piegl, L. (1991). On NURBS: A Survey. *IEEE Computer Graphics & Applications*, 55-71.
- Piegl, L., & Tiller, W. (1997). *The NURBS Book*. Berlin: Springer Verlag.

- Pixar. (2017, August 01). *Subdivision Surfaces*. Retrieved from OpenSubdiv:  
[https://graphics.pixar.com/opensubdiv/docs/subdivision\\_surfaces.html](https://graphics.pixar.com/opensubdiv/docs/subdivision_surfaces.html)
- Rodriguez-Morales, F., Gogineni, S., Leuschen, C., Paden, J., Li, J., Lewis, C., . . . Panton, C. (2014, May). Advanced multi-frequency radar instrumentation for polar research. *IEEE Transactions on Geoscience and Remote Sensing*, 2824-2842.
- Ruze, J. (1952). The effect of aperture errors on the antenna radiation pattern. *Il Nuovo Cimento Suppl.*, 364-380.
- Schueler, K., & Hale, R. (2002). Object-oriented implementation of an integrated design and analysis tool for fiber placed structures. *43rd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*. Denver: AIAA.
- Schueler, K., Miller, J., & Hale, R. (2004). Approximate geometric methods in application to the modeling of fiber placed composite structures. *Journal of Computing and Information Science in Engineering*, 251-256.
- Senbokuya, Y., & Tsunoda, H. (2003). Study on radiation characteristics of a flexible patch antenna. *Antennas and Propagation Society International Symposium* (pp. 397-400). Columbus, OH: IEEE.
- Shene, C. K. (2011, May 4). *Curve global interpolation*. Retrieved from CS3621 Introduction to Computing with Geometry Notes:  
<https://pages.mtu.edu/~shene/COURSES/cs3621/NOTES/INT-APP/CURVE-INT-global.html>
- Siddiqi, A. (1999). Staring at the sea: The Soviet RORSAT and EORSAT programmes. *Journal of the British Interplanetary Society*, 397-416.
- Siemens. (2014). *Element library reference*. Siemens.

Siemens. (2016, July 29). NX 11.0. Siemens.

Singhal, N. (2004, April 22). CS273: Algorithms for structure and motion in biology. Stanford University.

Singhvi, J. (2015, November 30). Jaguar F-type. Pune, India.

Splink, D. M. (2010, January 15). *NURBS toolbox*. Retrieved from MathWorks:

<https://www.mathworks.com/matlabcentral/fileexchange/26390-nurbs-toolbox-by-d-m-spink>

STEP Tools, Inc. (2018). *STEP tools, Inc. - Digital thread, STEP and IFC solutions*. Retrieved from STEP tools, Inc.: <https://www.steptools.com/>

STEP Tools, Inc. (2018, April 25). *The STEP standard - ISO 10303*. Retrieved from STEP tools, Inc.: [https://www.steptools.com/stds/step/step\\_1.html](https://www.steptools.com/stds/step/step_1.html)

Surana, K. S., & Reddy, J. N. (2016). *The finite element method for boundary value problems: Mathematics and computations*. Lawrence, KS: CRC Press.

Tabatabaenejad, A., Burgin, M., Duan, X., & Moghaddam, M. (2013). Airborne microwave observatory of subcanopy and subsurface radar retrieval of root zone soil moisture: Preliminary results. *RadarCon*. Ottawa: IEEE.

Tanaka, H. (2011). Surface error estimation and correction of a space antenna based on antenna gain analyses. *Acta Astronautica*, 1062-1069.

Technosoft. (2018, April 21). *Tehnosoft adaptive modeling*. Retrieved from Tehnosoft adaptive modeling: <http://www.technosoft.com/>

Tinevez, J.-I. (2017, February 15). Tree data structure as a MATLAB class. MathWorks.

Tsandoulas, G. N. (1987). Space-based radar. *Science*, 257-262.

- Turner, C. J., & Crawford, R. H. (2009). N-dimensional nonuniform rational B-splines for metamodeling. *Journal of Computing and Information Science in Engineering*.
- Wang, C. S., Duan, B. Y., & Shao, X. D. (2006). On multidisciplinary visual optimization analysis system for large parabolic reflector antenna structures. *11th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*. Portsmouth, VA.
- Wang, C. S., Wang, W., & Bao, H. (2008). On coupled structural-electromagnetic modeling and analysis of rectangle active phased array antennas. *IEEE/ASME International Conference on Advanced Intelligent Mechatronics*. Xian: Institute of Electrical and Electronics Engineers.
- Wang, C., Duan, B., Zhang, F., & Zhu, M. (2009). Analysis of performance of active phased array antennas with distorted plane errors. *International Journal of Electronics*, 549-559.
- Wang, C.-s., Zheng, F., & Zhang, F. S. (2008). On divided-fitting method of large distorted reflector antennas based Coons surface. *IEEE Radar Conference*. Rome: IEEE.
- Wang, H. S. (1992). Performance of phased-array antennas with mechanical errors. *IEEE Transactions on Aerospace and Electronic Systems*, 28(2), 535-545.
- Weisstein, E. W. (2018, January 26). *Barycentric coordinates*. Retrieved from MathWorld--A Wolfram web resource: <http://mathworld.wolfram.com/BarycentricCoordinates.html>
- Wu, X., Tan, Y., Zhu, B., & Zhou, L. (2013). Study of time domain FEM for 3-D electromagnetic radiation problem based on several temporal basis functions. *Chinese Journal of Electronics*, 410-414.
- Yan, J. B., Hale, R. D., Mahmood, A., Rodriguez-Morales, F., J., L. C., & Gogineni, S. (2015). A polarization reconfigurable low-profile ultrawideband VHF/UHF airborne array for

fine-resolution sounding of polar ice sheets. *IEEE Transactions on Antennas and Propagation*, 4334-4341.

Zhou, J., Huang, J., He, Q., Tang, B., & Song, L. (2017). Development and coupling analysis of active skin antenna. *Smart Materials and Structures*, 1-17.

Zhou, J., Song, L., Huang, J., & Congsi, W. (2015). Performance of structurally integrated antennas subjected to dynamical loads. *International Journal of Applied Electromagnetics and Mechanics*, 409-422.

## **Appendix A: Other Parameterization Approaches**

This appendix summarizes other parameterization approaches which were implemented in earlier versions of ADT, but which are not currently used, unless specifically directed by the user.

### **A.1 Least Squares Conformal Mapping**

LSCM is a technique developed for texture mapping by Levy et al (Levy, Petitjean, Ray, & Maillot, 2002), which minimizes angle distortions when mapping from one domain to another. Texture mapping is the task of mapping textures, which are 2D images, to 3D models, which are frequently defined as polygonal meshes, in order to add detail and realism for computer visualization applications. This method is of interest in this project as it allows to obtain a mesh parametrization without gridding it first, as in the Krishnamurthy method. Because the surface fitting algorithms used are adapted to be used with gridded data, the LSCM can be used to grid the mesh by mapping from parametric space back to 3D Euclidean space using triangular barycentric coordinates. In this method, barycentric coordinates are obtained for each point on a regular grid, and these can be easily translated to the 3D mesh if the vertex positions and mesh connectivity are known. This method has the advantage of being faster (and more robust in some cases than the gridded resampling method) while providing meshes at the same or higher quality level. If used carefully, it may also be integrated with free boundary parametrization techniques, which can allow surface reconstruction of meshes with holes in them and boundaries that are not best parametrically fitted by a square. These are limitations which the gridded resampling method cannot easily overcome.

To introduce LSCM, the concept of Dirichlet energy must be discussed. Dirichlet energy is defined as the  $L^2$  norm of the gradient of a map  $\mathbf{u}$ , from a differential surface patch  $X$  to its image in  $U$ , shown in Equation A.1 (Mullen, Tong, Alliez, & Desbrun, 2008).

$$E_D = \frac{1}{2} \int_X |\nabla \mathbf{u}|^2 dA \quad (\text{A.1})$$

It can be shown that harmonic maps minimize the Dirichlet energy ( $\Delta \mathbf{u} = 0$ ), and if the minimal Dirichlet energy is attained, the map is said to be conformal, meaning that there is no angular distortion caused by the map. Mathematically, conformality is defined by Equation A.2, and visualized in Figure A.1, which defines a map that maintains orthogonality between iso-curves. In this equation,  $X$  is a mapping of a parametric  $(u, v)$  domain, and  $N(u, v)$  is the unit normal to the surface.

$$N(u, v) \times \frac{\partial X}{\partial u}(u, v) = \frac{\partial X}{\partial v}(u, v) \quad (\text{A.2})$$

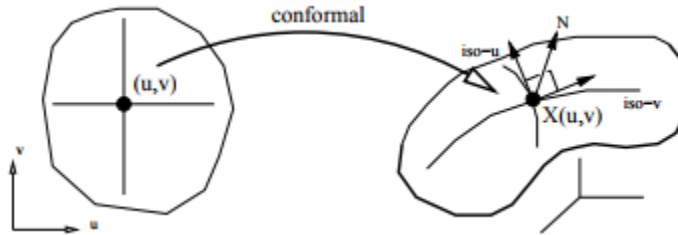


Figure A.1. Conformal map, where tangents of  $u$  and  $v$  isocurves are orthogonal and have the same length (Levy, Petitjean, Ray, & Maillot, 2002).

Conformal energy is defined as Equation A.3:

$$E_c(\mathbf{u}) = E_D(\mathbf{u}) - A(\mathbf{u}) \quad (\text{A.3})$$

Where  $A(\mathbf{u})$  is the area of the parametrization. The map  $\mathbf{u}$  is conformal if this energy is zero. Equation A.3 can be expressed in a piecewise linear form, applied to a triangular mesh, as Equation A.4, where  $E_D$  is the Dirichlet energy,  $\theta_{ij}$  and  $\theta_{ji}$  are the two angles opposite to the edge

linking the nodes  $x_i$  and  $x_j$ , and  $u_i$  and  $u_j$  are the parametric coordinates of nodes  $x_i$  and  $x_j$ , and  $e_{ij}$  is the edge between  $u_i$  and  $u_j$ . The angles and vertices are shown in Figure A.2.

$$E_D(\mathbf{u}) = \sum_{e_{ij}} \frac{1}{4} (\cot(\theta_{ij}) + \cot(\theta_{ji})) (\mathbf{u}_i - \mathbf{u}_j)^2 \quad (\text{A.4})$$

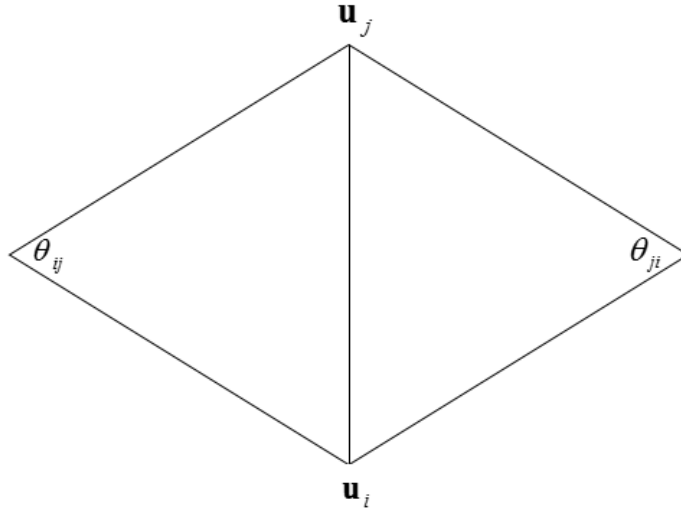


Figure A.2. Vertices i and j, with corresponding angles.

The area of the mesh parametrization is found using the coordinates of the boundary vertices, and is expressed in Equation A.5.

$$A(\mathbf{u}) = \sum_{e_{ij} \in \partial U} \frac{1}{2} (u_i v_j - u_j v_i) \quad (\text{A.5})$$

If certain boundary conditions are set (such as square boundary conditions, useful for parametrizing a mesh for B-spline fitting), then it is impossible to obtain a truly conformal map because the conformal energy cannot be zero. However, LSCM minimizes the difference between this energy and the area of the parametrization, now termed the LSCM energy, expressed in Equation A.6.

$$E_{LSCM}(\mathbf{u}) = E_D(\mathbf{u}) - A(\mathbf{u}) \quad (\text{A.6})$$



When this energy is minimized, a LSCM mapping  $\mathbf{u}$  is obtained. The code used to implement this type of parametrization was developed by Alec Jacobsson (Jacobson, 2017), and it requires the input of a mesh and boundary constraints.

LSCM is faster than the author's implementation of mesh gridding presented in the previous section, and provides better quality meshes, but can sometimes produce undesirable parametrizations.

## A.2 Gridded Resampling

Krishnamurthy (2000) proposed a method that involves resampling an irregular triangular mesh into a regular quad mesh. The mesh is resampled into a grid by generating and moving nodes on the mesh surface until a regular distribution has been achieved. It is an iterative process which generates a new mesh, and consists of two steps: relaxation and refinement. The resampling aims to achieve a mesh with three main characteristics:

- 4) Arc length uniformity: the spacing along a particular iso-curve should be uniform
- 5) Aspect ratio uniformity: the spacing along the  $u$  and  $v$  iso-curves should be the same
- 6) Parametric fairness: every  $u$  and  $v$  iso-curve should be of the minimum length while complying with the first two conditions.

The resampled mesh which is created from the original mesh is called the spring mesh, since it aims to minimize the spring energy between adjacent spring mesh nodes, and the overall process to obtain it is shown in Figure A.3.

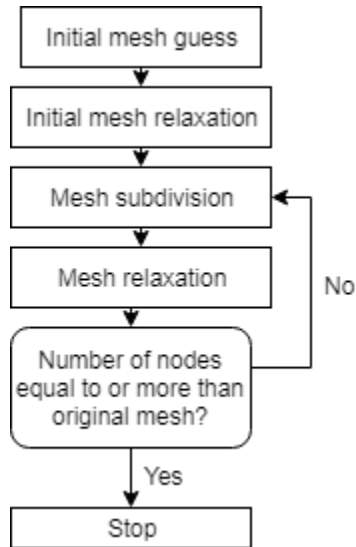


Figure A.3. Gridded resampling flowchart.

To create the initial spring mesh, a small number of nodes is sampled from the original mesh. The initial nodes are sampled regularly along the surface edges, where the aspect ratio uniformity criterion is enforced by seeding a number of nodes proportional to the length of the edge on which they are seeded. In other words, the aspect ratio of the surface is determined first, as the ratio of the longest to the shortest edge length. The number of seeds used can be set by the user, but smaller numbers typically give better results. Each of these nodes is connected to its opposite node on the other edge of the surface using the Dijkstra shortest path algorithm (Dijkstra, 1959), (Kirk, 2015). The initial internal nodes are then obtained as intersections of these shortest paths. Figure A.4 summarizes the spring mesh initialization procedure.

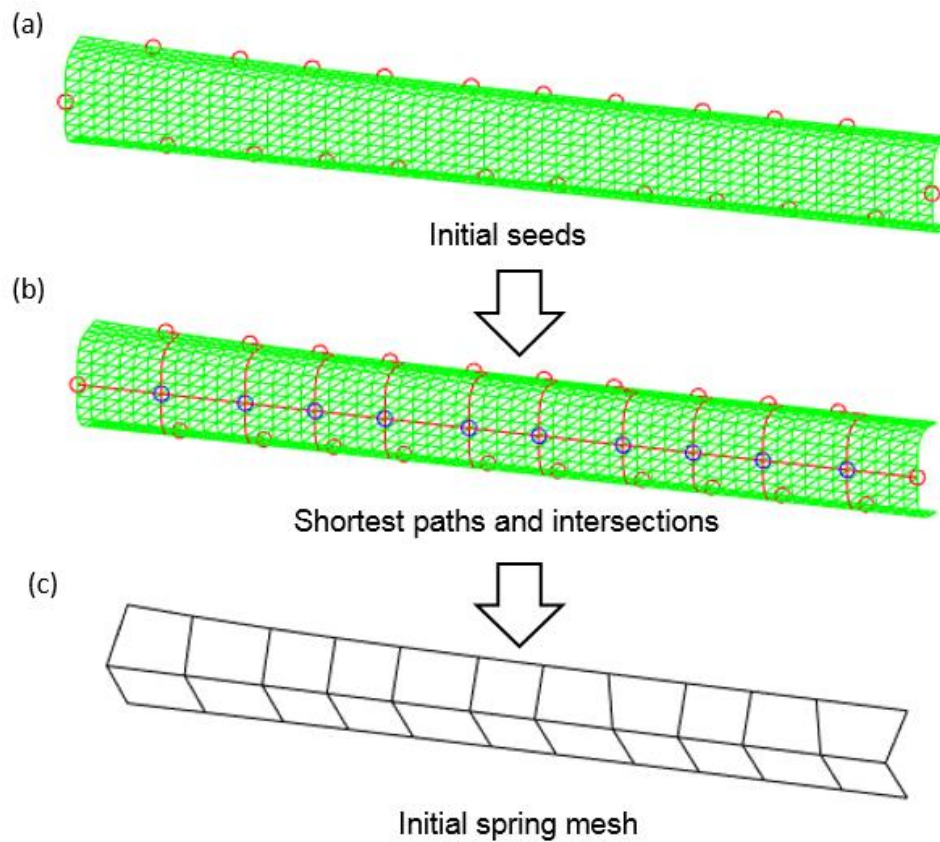


Figure A.4. Spring mesh initialization. (a) initial seeds shown as red circles, original mesh is green, ratio of seeds is determined by length of longest side to shortest side, (b) shortest path between opposing red nodes is computed using Dijkstra (red paths), and intersection nodes are saved (blue circles), (c) initial spring mesh is built by connecting red seed nodes and blue intersecting nodes.

The first relaxation step occurs next, which involves moving mesh nodes along the surface by applying forces to each node which indicate the distance and direction each point should move along the mesh surface. The objective of moving these nodes is to obtain a fair mesh by spacing the nodes out evenly and as far away as possible from each other along a single iso-curve. There are two forces which are applied:

- 1) **Arc force**: this force is the force which moves the node closer to the neighbor it is most distant from. It is responsible of generating equal spacing between the nodes.

- 2) Fairness force: this is the vector sum of the forces acting on a node, and has the effect of smoothing the mesh and reducing the skewness of the elements. It tightens the iso-curves that nodes lie on.

These forces are computed based on the Euclidean distance between the spring mesh node being moved,  $P$ , and its four neighbors (two in the case of edge nodes, and zero in the case of corner nodes). These are graphically explained in Figure A.5, where  $P$  represents the position of the node forces are being applied to, and  $P_{up}$ ,  $P_{down}$ ,  $P_{left}$  and  $P_{right}$  are the positions of the neighboring nodes. The arrows show what type of forces affect each node. Internal nodes are affected by all four forces, while edge nodes only by either up and down or left and right, depending on location. Corner points are not affected by any forces, and are never moved.

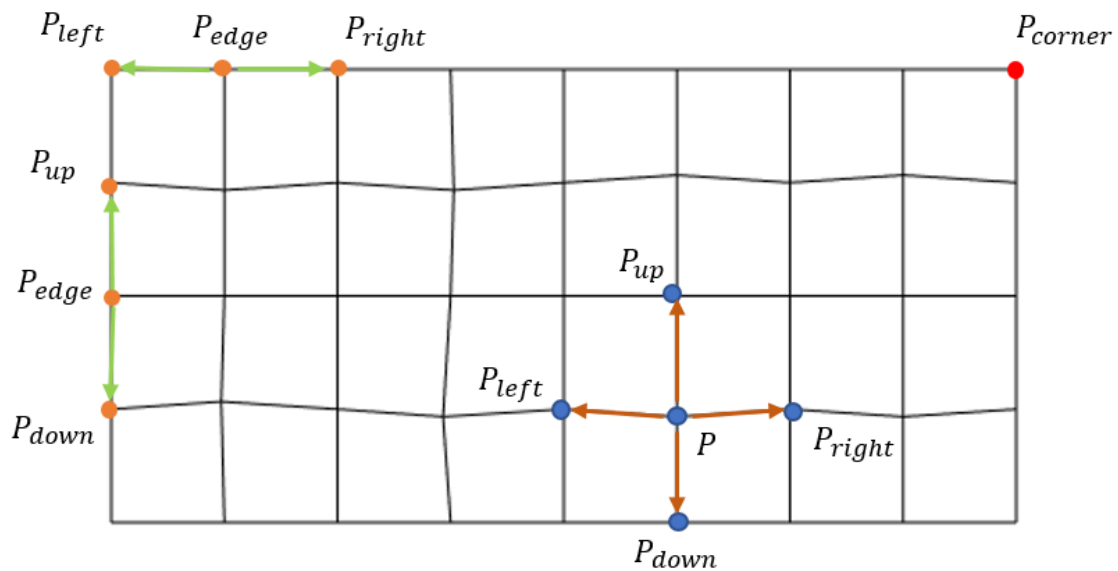


Figure A.5. Simple spring mesh showing forces applied on different types of nodes.

Edge node (excluding corner nodes) motion is contradictory to Krishnamurthy's original work, which suggests not moving any edge nodes. It was found that moving these nodes had a favorable impact on mesh quality. The definitions in Figure A.5 are used in Equations A.7-9 to

define the different forces used to regularize the mesh. In these equations,  $P$  is a position vector in Euclidean  $(x, y, z)$  space, not in parametric  $(u, v)$  space.

$$F_{arc-ud}(P) = (|P_{up} - P| - |P_{down} - P|) \frac{t_{max}}{\|t_{max}\|} \quad (A.7)$$

$$F_{arc-lr}(P) = (|P_{left} - P| - |P_{right} - P|) \frac{t_{max}}{\|t_{max}\|} \quad (A.8)$$

$$t_{max} = \max((P_{up} - P), (P_{down} - P)) \quad t_{max} = \max((P_{left} - P), (P_{right} - P)) \quad (A.9)$$

Equations A.7-9 define the up-down and left-right arc forces, and Equation A.10 defines the total arc force, which is the actual quantity used for spring node motion.

$$F_{arc} = F_{arc-ud} + F_{arc-lr} \quad (A.10)$$

The fairness forces are also divided into up-down and left-right pairs and are defined by Equations A.11-12 where, once again, the sum of both forces in Equation A.13 defines the total force which is applied in node motion calculations.

$$F_{fair-ud} = P_{up} + P_{down} \quad (A.11)$$

$$F_{fair-lr} = P_{left} + P_{right} \quad (A.12)$$

$$F_{fair} = F_{fair-ud} + F_{fair-lr} \quad (A.13)$$

The total force by which the node is moved is a weighted vector sum, given in Equation A.14, where  $F_i$  is the total force for iteration  $i$ , and  $a_i$  and  $b_i$  are weights obtained by Equations A.15-16. Krishnamurthy's work gives generic guidelines as to the weights used ( $a$  for the arc force and  $b$  for the fairness force) suggesting reducing  $a$  from 1 to 0.5, while increasing  $b$  from 0 to 0.5 as the refinement iterations progress (Krishnamurthy, 2000). In this work, it was found that the best results are obtained by setting  $a_{max}$  and  $b_{max}$  to 0.5, and using Equations A.15-16 to get the exact weight for each iteration. However, the user can adjust  $a_{max}$  and  $b_{max}$  to any values

if the mesh results are not satisfactory. By slowly varying the weights, the computation of the gridded mesh is more stable.

$$F_i = a_i F_{arc} + b_i F_{fair} \quad (\text{A.14})$$

$$a_i = a_{\max} \left( \frac{n-i+1}{n} \right) \quad (\text{A.15})$$

$$b_i = b_{\max} \left( \frac{i-1}{n} \right) \quad (\text{A.16})$$

In Equation A.15-16,  $n$  is the total number of iterations performed, and  $i$  is the current iteration number. After a user set number of iterations of the relaxation step (usually 10-20), mesh refinement occurs. This is the step in which the mesh is subdivided, by inserting a spring node in the middle of each pair of spring nodes, at the midpoint of the shortest path between the nodes on the original mesh nearest to the spring nodes being subdivided. Refinement is needed to capture all the details of the original mesh.

The mechanism of point motion is defined in Figure A.6 and Figure A.7. Spring mesh node motion occurs by first projecting the resultant force obtained by Equation A.14 on the element nearest to point P in the direction of the resultant force  $F_r$ .

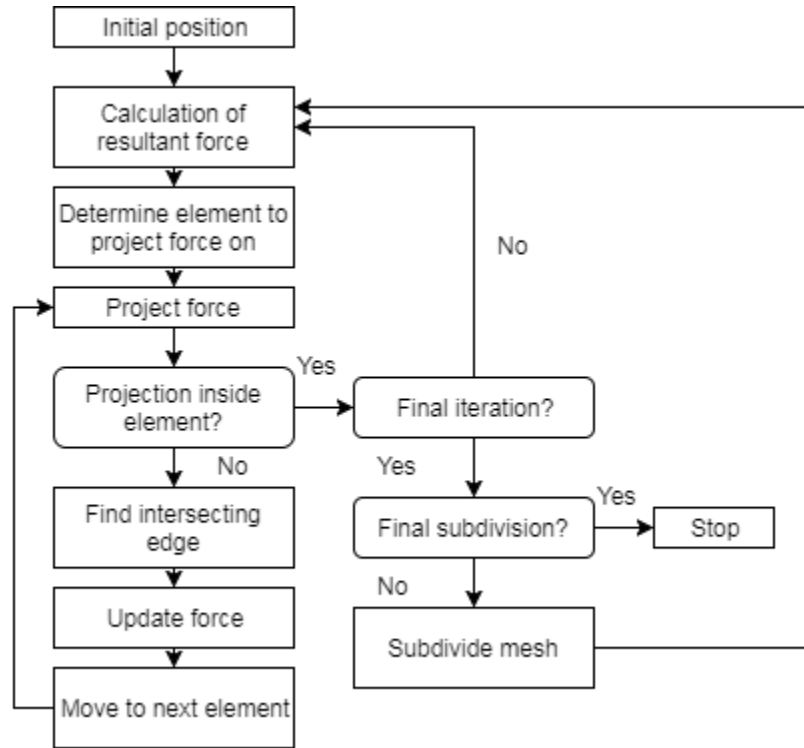


Figure A.6. Spring mesh node motion flowchart.

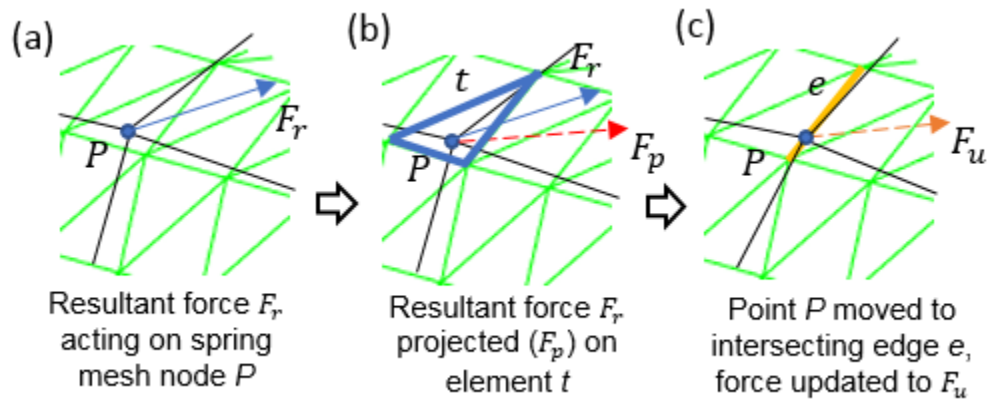


Figure A.7. Spring node motion visualization. Original mesh shown in green, spring mesh in black.

If the projection,  $F_p$ , ends in the same element, motion is stopped until the next relaxation step iteration. If not, the intersecting edge is found, and the point is moved to it, and  $F_p$  is updated to account for the motion, resulting in  $F_u$ . Then the process is repeated until the motion

ends inside an element on the final relaxation iteration. The whole relaxation routine is repeated a set number of times per each mesh refinement step, and the mesh is refined until the number of nodes making up the spring mesh is equal to or larger than the number of nodes in the original mesh.

The mesh refinement process is explained in Figure A.8. During refinement, a spring mesh node is inserted between each pair of existing spring mesh nodes, and between any unconnected points between existing spring mesh nodes. This means that each existing face of the spring mesh is divided into four faces. During refinement, the nearest nodes on the original mesh to the spring nodes being subdivided are found, and the shortest path between them is computed using the Dijkstra algorithm. Then, the midpoint of this path,  $M$ , is set as the new spring mesh node between the existing spring mesh nodes. This node may or may not correspond to a node on the original mesh. Figure A.9 shows several consecutive relaxation and refinement steps.

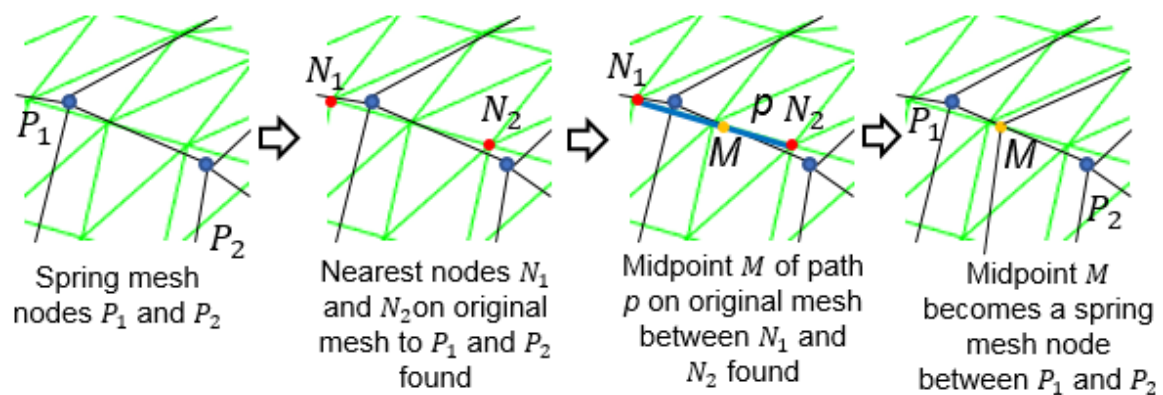


Figure A.8. Spring mesh subdivision visualization. Original mesh shown in green, spring mesh in black.



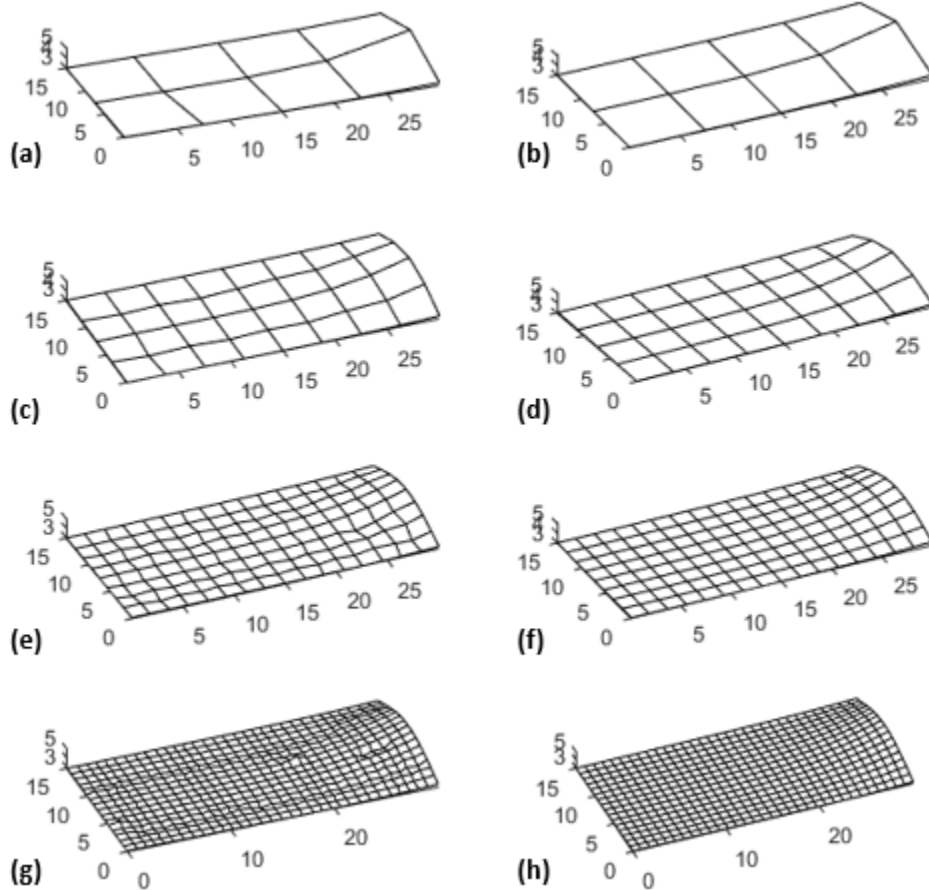


Figure A.9. Mesh resampling algorithm. (a) Initial mesh guess, (b) first refinement step, (c) first subdivision step, (d) second refinement, and (e), (f), (g), (h) are subsequent subdivision and refinement steps.

Figure A.10 and Figure A.11 show the results of running the mesh resampling algorithm on two different irregular meshes. Qualitatively, the results are satisfying. A quantitative comparison between the several methods used is presented at the end of the discussion on meshing methods.

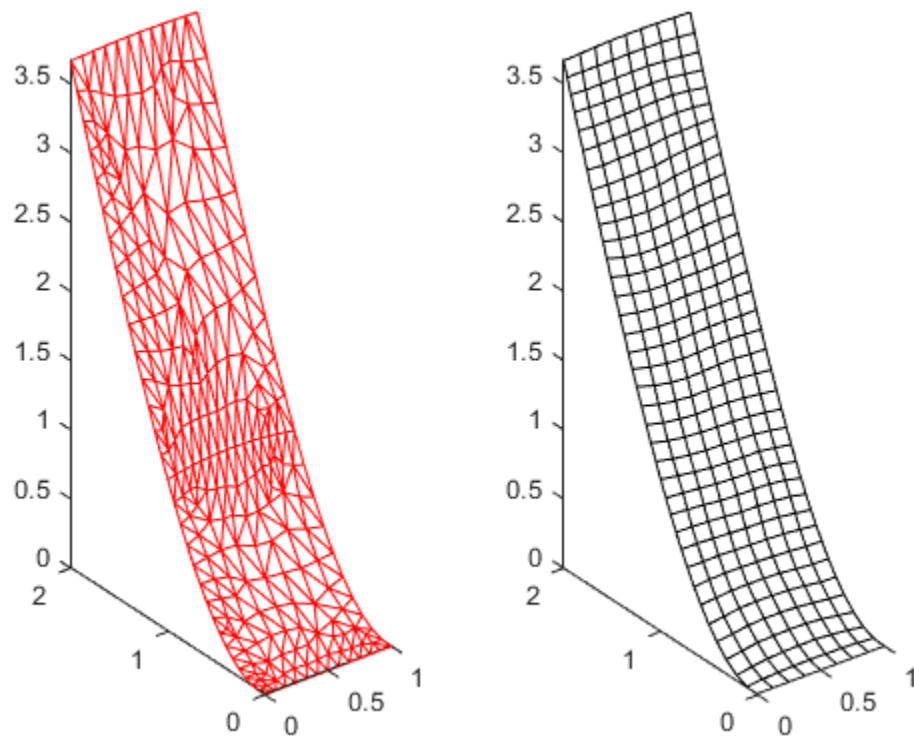


Figure A.10. Resampling of irregular mesh using gridded resampling.

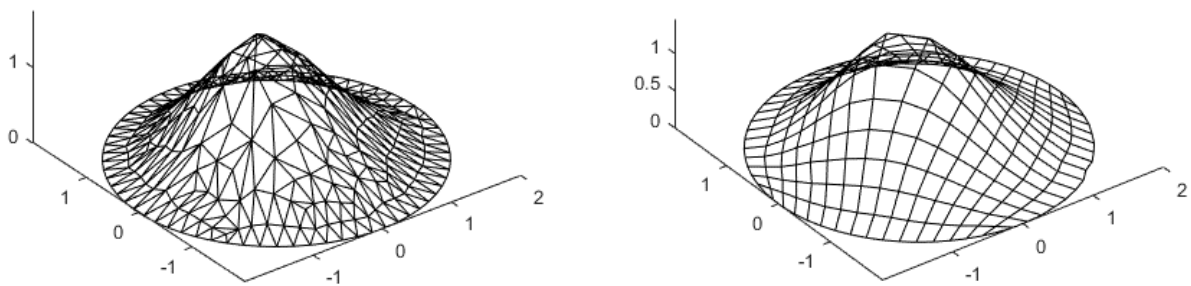


Figure A.11. Resampling of irregular mesh with circular boundary using gridded resampling.

## **Appendix B: STEP Files and Supported Entities**

The STEP standard, known officially as ISO 10303, is an ISO standard which is used to describe and represent digital product information (STEP Tools, Inc., 2018). This includes information such as the geometric description of the product, manufacturing information, and other product information. The STEP Part 21 file (ISO 10303-21) is an easily readable ASCII file which specifies an exchange format in the EXPRESS language, which is a data modelling language for product data. The Part 21 is the file read by a CAD import program and which is written by the program developed.

In addition, the STEP has different Application Protocols (APs), or schemas defined in the EXPRESS language. These are used to write or read the Part 21 files. These protocols have different scopes for different applications, but are all built on the same basic definitions using the same basic information. They are numbered, with more advanced APs with more complex features usually having a higher number. Some examples are:

- AP 201: Explicit Draughting – used for 2D designs
- AP 203: Configuration Controlled Design – used for 3D designs
- AP 209: Multidisciplinary Analysis and Design
- AP 214: Automotive Design – Similar to 203
- AP 242: Managed Model Based 3D Engineering – combined version based on 203 and

In this project, STEP files are used for all CAD input and output operations. Out of the available APs, 203 was chosen. The reasons for using STEP files and the specific AP are listed below:

- 1) AP 203 was the most compatible (out of the STEP file APs) among the software used at KU Engineering, although recent software updates may allow using a different AP.
- 2) AP 203 is the most commonly used format in the USA, and is mainly supported by the Aerospace and Defense Industries.
- 3) STEP Part 21 files are easily human-readable, which is not the case with some other CAD exchange formats, such as Parasolid. This makes for easier debugging and development of a custom STEP file reader.
- 4) Basic versions of the STEP schema were more easily and cheaply accessible for research purposes through STEP Tools Inc.

In this work, STEP, STEP AP 203, AP 203, all refer to Part 21 files written using the ISO 10303-203 standard.

A typical STEP file will contain many geometric objects, which are called entities. These entities often build upon each other, where more complicated entities, (such as solids), will be defined upon simpler entities (such as connected faces), which in turn are defined by other entities. The whole geometric model is placed in a geometric context, where a coordinate system, units used, level of uncertainty, among other quantities, are defined. This means that for any entity in a STEP file, a hierarchy exists, which can be easily determined by reading the STEP file, as each entity contains references to its defining entities. Figure B.1 shows an example of how a solid manifold (MANIFOLD\_SOLID\_BREP) entity is constructed, with some simplifications made.

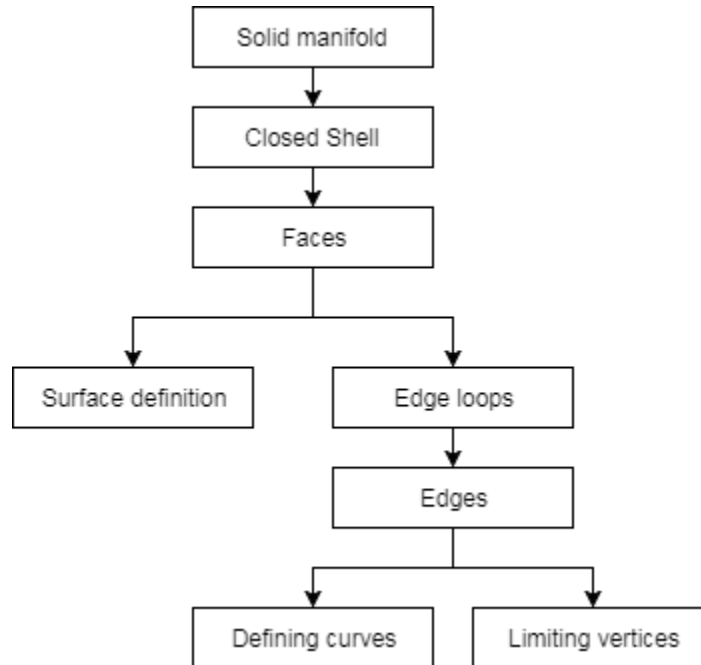


Figure B.1. Hierarchy of a solid manifold in STEP AP 203.

The manifold solid is defined as a closed shell of faces, each with a set of bounds, each of which is a set of curves. However, in order for the solid to be valid, the boundaries must be clearly defined, meaning that the faces making up the shell must reference the same vertices and edge curves where they meet. A geometric model can then consist of a number of solids, sheets, and other geometric entities assigned as items to a geometric representation.

The most important geometric representations which are supported in this work are the following:

- 1) MANIFOLD\_SOLID\_BREP: a boundary representation of a solid, made of faces
- 2) ADVANCED\_FACE: a face, made of a surface definition and a set of boundary loops
- 3) FACE\_BOUND: a face boundary consisting of an edge loop
- 4) EDGE\_LOOP: a loop of oriented edge entities

- 5) **ORIENTED\_EDGE**: an edge curve with an orientation assigned to it, used to define edge loops
- 6) **EDGE\_CURVE**: a curve used to define an edge. Contains curve geometry, start and end vertex, as well as information on the sense in which the curve is defined (from start to end vertex or vice versa)
- 7) **VERTEX\_POINT**: a point used as a vertex. These are used to define edge curves and must be shared by connecting edge curves to define a valid face or solid
- 8) **LINE**: a line, defined by a point and a direction, extends infinitely. Used to define an edge curve
- 9) **B\_SPLINE\_CURVE\_WITH\_KNOTS**: A B-spline curve, defined as discussed previously. Uses Cartesian points to define the control points, knots and multiplicities are defined by vectors. Used by edge curve as the curve definition.
- 10) **B\_SPLINE\_SURFACE\_WITH\_KNOTS**: a B-spline surface, defined as discussed previously. Uses Cartesian points to define the control net, knots and multiplicities are defined by vectors. Used by faces as the surface definition.
- 11) **PLANE**: an infinite plane, defined by a Cartesian point and a normal vector
- 12) **DIRECTION**: a vector defined by a set of x, y and z values. Used to define lines and planes
- 13) **CARTESIAN\_POINT**: a Cartesian point, defined as a set of x, y and z coordinates

A code in MATLAB was developed to read in and interpret the various entities of a STEP file, which stores the relationship between entities as a hierarchical tree structure, developed by Jean-Yves Tinevez (2017). This allows retaining the entity organization of any imported file, which can be later used for surface reconstruction

### **Appendix C: Transitional Text File Format**

The following is a list of indicators used in the transitional text file to indicate the properties of geometric entities for export.

# Begin NURBS Export File – indicates the start of the file

# End NURBS Export File – indicated the end of the file

# BEGIN/END FACE – indicates the start/end of information related to a face

# BEGIN/END CONTROL POINTS – indicates the start/end of B-spline control point coordinates. It is organized as a list of x,y,z values, where each row represents a parametric row for the curve/surface being created.

# DEGREE VALUES – signals that the next line contains the curve/surface degree for the curve/surface generated

# KNOTS – signals that the next line contains the knots for the curve/surface

# MULTIPLICITIES – signals that the next line contains multiplicities for the knots of the curve/surface

# BEGIN/END MANIFOLD – indicates the start end of a manifold solid

# BEGIN/END BOUND – indicates the start and end of a face bound

# BEGIN/END CURVE – indicates the start/end of a bound curve

# VERTICES [number of vertices] – indicates the start of vertex coordinates and the total number of vertices in the manifold

# END VERTICES – indicates the end of the vertex list

## Appendix D: Material Properties

Table D.1. Material properties of materials used in models tested.

<b>Material</b>	<b>Young's modulus (ksi)</b>	<b>Relative permittivity</b>	<b>Relative permeability</b>	<b>Bulk conductivity (siemens/m)</b>	<b>Reference</b>
S2 Glass	13000	3.4	1	0	(AZO Materials, 2018; ANSYS, 2014)
FR4 Epoxy	1900	4.4	1	0	(Dielectric Corporation, 2018; ANSYS, 2014)
2024-T3 Aluminum	10600	1	1.000021	38000000	(Aerospace Specification Metals, Inc., 2018; ANSYS, 2014)
Steel stainless	27600	1	1	1100000	(AZO Materials, 2018; ANSYS, 2014)



## Appendix E: Results and Pylon Displacement Simulations

Table E.1 shows the detailed results of the single P3 element deformation

Table E.1. Clamped sheet deflection simulation results.

Maximum tip deflection (% $\lambda$ )	Length (in)	Strain (in/in)	Resonant Frequency (MHz)	Lower Bandwidth Limit (MHz)	Higher Bandwidth Limit (MHz)	Bandwidth (MHz)
0	23.9312992	0	197	188	233	45
0.17	23.9315611	1.09E-05	197	188	231	43
0.33	23.9323467	4.38E-05	197	188	232	44
0.5	23.933656	9.85E-05	197	188	232	44
0.67	23.9354888	1.75E-04	197	189	232	43
0.73	23.9378449	2.74E-04	197	188	232	44
1	23.9407242	3.94E-04	197	188	231	43
2	23.968967	1.57E-03	197	188	231	43
2.5	23.9901175	2.46E-03	196	188	231	43
3	24.0159317	3.54E-03	196	188	231	43
4	24.081461	6.27E-03	196	188	230	42
5	24.1653386	9.78E-03	195	187	230	43
10	24.8487915	3.83E-02	191	183	225	42
15	25.9329431	8.36E-02	184	177	216	39
25	29.0524932	2.14E-01	165	160	195	35
30	30.9698394	2.94E-01	156	152	184	32
35	33.0617003	3.82E-01	146	143	173	30
40	35.294647	4.75E-01	138	136	163	27
45	37.6402209	5.73E-01	130	129	154	25
50	40.0773239	6.75E-01	122	122	145	23

To test the effects of pylon position on element return loss, a pylon was shifted across the length of the antenna element, and the antenna resonance shift was simulated, show in in FigureE.1. A radiation bounding box was used, with a solution frequency of 200 MHz, and an interpolating, direct solution setting in HFSS. The results obtained are plotted in FigureE.2 FigureE.3 and FigureE.4. From these results, it was concluded that the array simulated is very

sensitive to pylon placement. It was also discovered that the shape of the resonance plot would not only vary with pylon placement, but also with the analysis sweep of the simulation in HFSS. For example, the feature in the middle pylon simulation (16 in), seen in Figure E.3, was not present when a sweep of 50-500 MHz was used (Figure E.2), but was present when a smaller sweep, of 150-300 MHz was used. In addition, such resonance features were very sensitive to pylon placement. The pylons in Figure E.4 were offset by 0.35 each in the y direction in from the pylons in Figure E.2, which changed the shape of the plots.

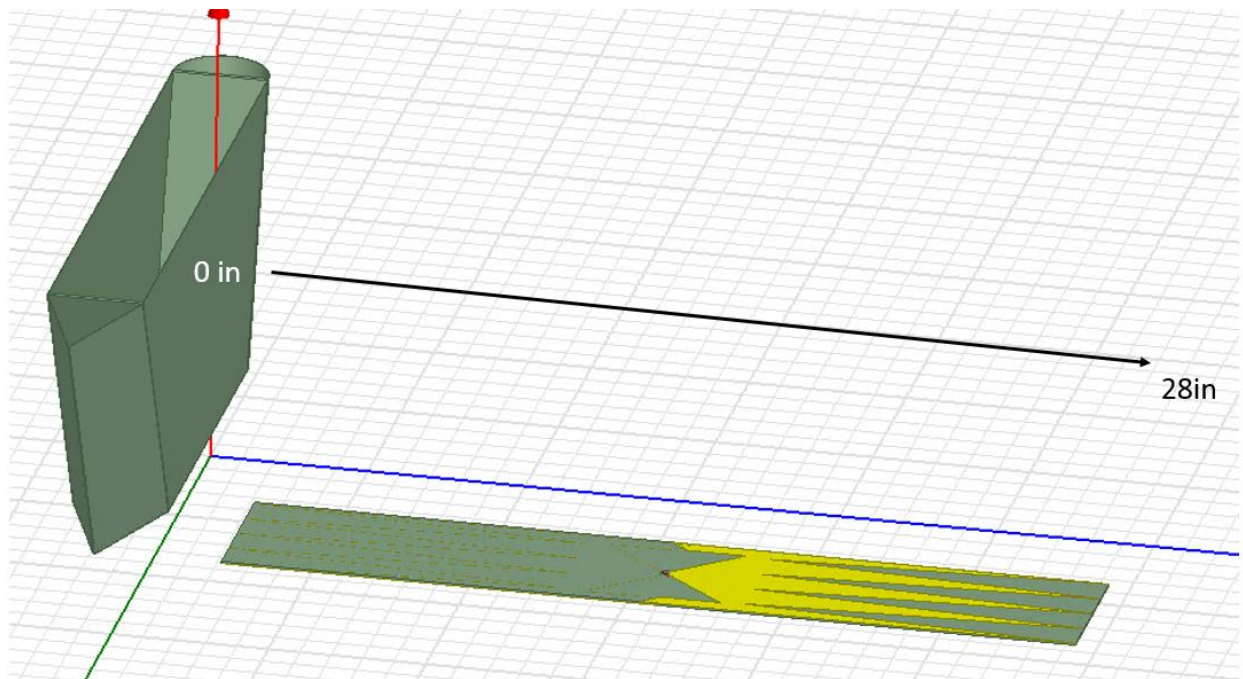


Figure E.1. Set up of the pylon interference simulation.

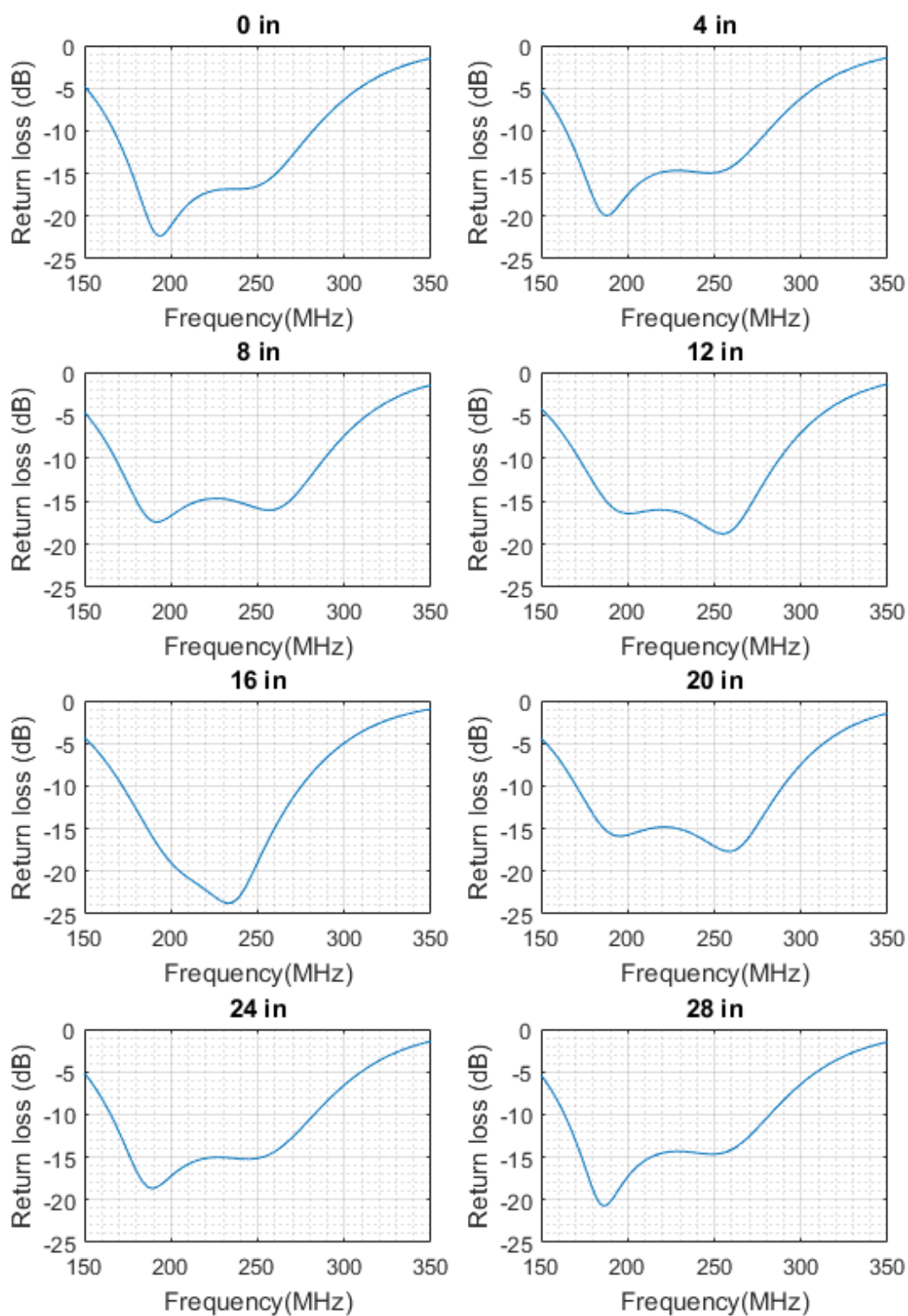


Figure E.2. Return loss plots of MCoRDS single element simulations with the pylon at different locations, using an analysis sweep of 50-500 MHz. Only the range from 150-350 MHz is plotted.

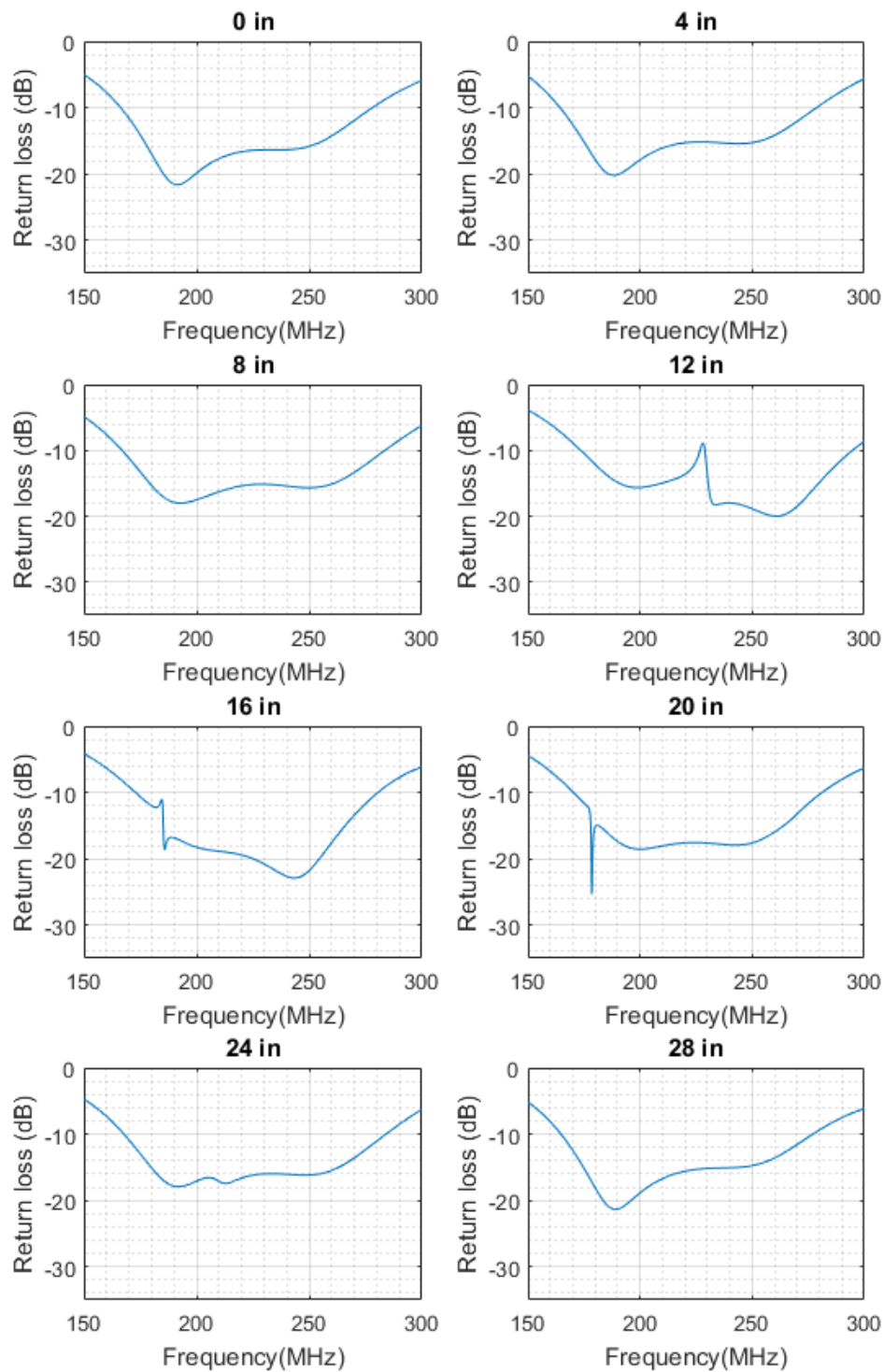


Figure E.3. Return loss plots of MCoRDS single element simulations with the pylon at different locations, using an analysis sweep of 150-300 MHz.

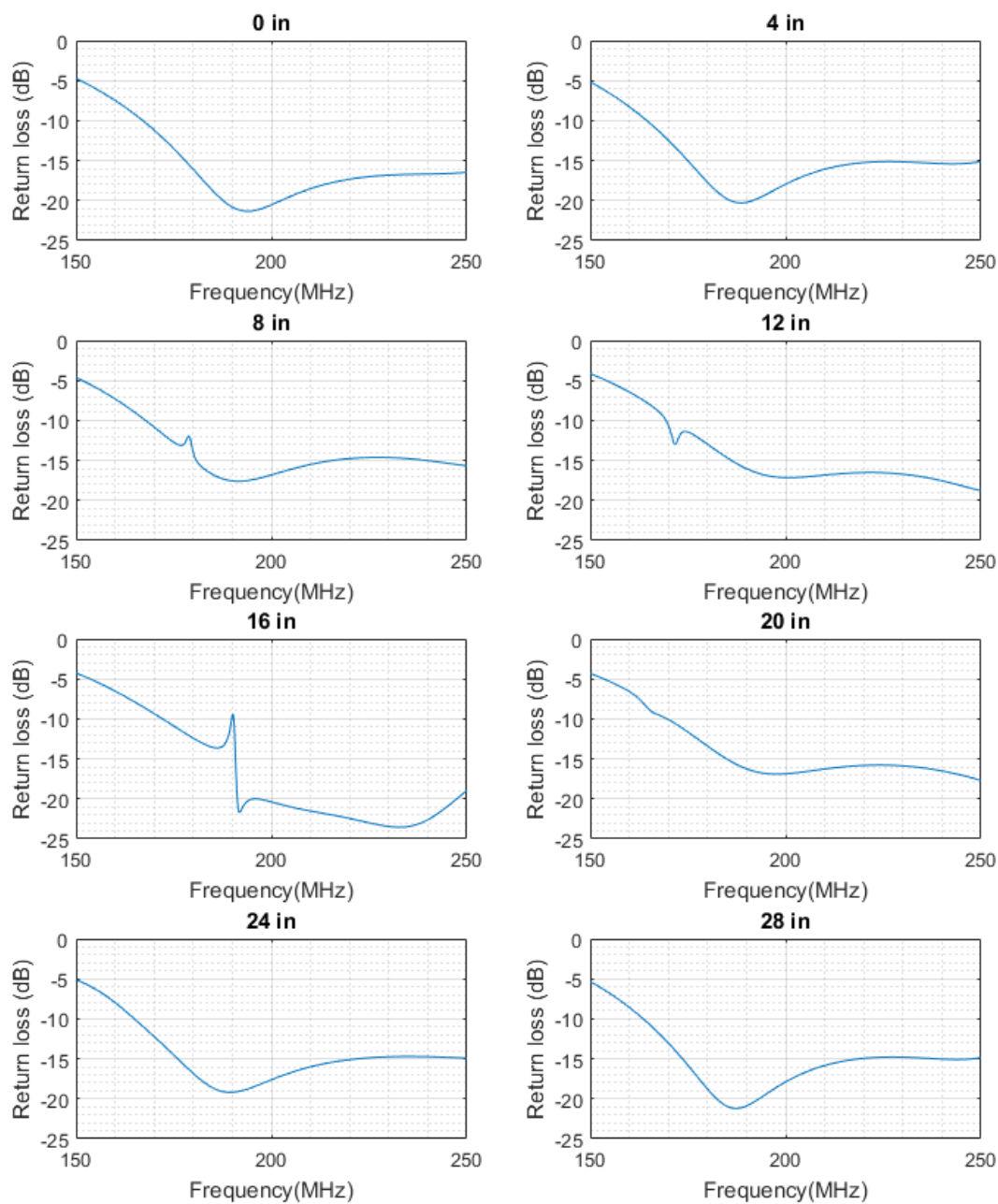


Figure E.4. Return loss plots of MCoRDS single element simulations with the pylon offset 0.35 inches less in the y-direction, using an analysis sweep of 150-250 MHz.

### Appendix F: B-spline Functions Used for Bowl Deformation

Function form

```
form: 'B-NURBS'
      dim: 4
      number: [4 4]
      coefs: [4x4x4 double]
      knots: {[0 0 0 0 1 1 1 1] [0 0 0 0 1 1 1 1]}
      order: [4 4]
```

10% lambda coefficients:

ans(:,:,1) =

```
-0.7382  68.6517  138.0415  207.4314
-13.4055 -13.4055 -13.4055 -13.4055
   0      0      0      0
 1.0000  1.0000  1.0000  1.0000
```

ans(:,:,2) =

```
-0.7382  68.6517  138.0415  207.4314
64.4292  64.4292  64.4292  64.4292
   0     -11.0000 -11.0000   0
 1.0000  1.0000  1.0000  1.0000
```

ans(:,:,3) =

```
-0.7382  68.6517  138.0415  207.4314
142.2639 142.2639 142.2639 142.2639
   0     -11.0000 -11.0000   0
 1.0000  1.0000  1.0000  1.0000
```

ans(:,:,4) =

```
-0.7382  68.6517  138.0415  207.4314
220.0986 220.0986 220.0986 220.0986
   0      0      0      0
 1.0000  1.0000  1.0000  1.0000
```

12.5% lambda coefficients:

ans(:,:,1) =

```
-0.7382  68.6517  138.0415  207.4314
-13.4055 -13.4055 -13.4055 -13.4055
   0      0      0      0
 1.0000  1.0000  1.0000  1.0000
```

ans(:,:,2) =

```
-0.7382  68.6517  138.0415  207.4314
```

64.4292	64.4292	64.4292	64.4292
0	-15.0000	-15.0000	0
1.0000	1.0000	1.0000	1.0000

ans(:, :, 3) =

-0.7382	68.6517	138.0415	207.4314
142.2639	142.2639	142.2639	142.2639
0	-15.0000	-15.0000	0
1.0000	1.0000	1.0000	1.0000

ans(:, :, 4) =

-0.7382	68.6517	138.0415	207.4314
220.0986	220.0986	220.0986	220.0986
0	0	0	0
1.0000	1.0000	1.0000	1.0000

25% lambda coefficients:

ans(:, :, 1) =

-0.7382	68.6517	138.0415	207.4314
-13.4055	-13.4055	-13.4055	-13.4055
0	0	0	0
1.0000	1.0000	1.0000	1.0000

ans(:, :, 2) =

-0.7382	68.6517	138.0415	207.4314
64.4292	64.4292	64.4292	64.4292
0	-27.0000	-27.0000	0
1.0000	1.0000	1.0000	1.0000

ans(:, :, 3) =

-0.7382	68.6517	138.0415	207.4314
142.2639	142.2639	142.2639	142.2639
0	-27.0000	-27.0000	0
1.0000	1.0000	1.0000	1.0000

ans(:, :, 4) =

-0.7382	68.6517	138.0415	207.4314
220.0986	220.0986	220.0986	220.0986
0	0	0	0
1.0000	1.0000	1.0000	1.0000

50% lambda coefficients

ans(:, :, 1) =

-0.7382	68.6517	138.0415	207.4314
-13.4055	-13.4055	-13.4055	-13.4055
0	0	0	0
1.0000	1.0000	1.0000	1.0000

```
ans(:,:,2) =
```

```
  -0.7382   68.6517  138.0415  207.4314  
  64.4292   64.4292   64.4292   64.4292  
     0    -53.0000  -53.0000     0  
  1.0000   1.0000   1.0000   1.0000
```

```
ans(:,:,3) =
```

```
  -0.7382   68.6517  138.0415  207.4314  
 142.2639  142.2639  142.2639  142.2639  
     0    -53.0000  -53.0000     0  
  1.0000   1.0000   1.0000   1.0000
```

```
ans(:,:,4) =
```

```
  -0.7382   68.6517  138.0415  207.4314  
 220.0986  220.0986  220.0986  220.0986  
     0     0     0     0  
  1.0000   1.0000   1.0000   1.0000
```