Analysis and Design of Various Medieval Vaulting Technologies

ENGR 090 Final Report

Kara D. Peterman '09

Swarthmore College
Department of Engineering

ABSTRACT

Finite element modeling methods were used to design models of three medieval vaulting technologies: an ideal quadripartite vault , ideal sexpartite vault, and vault from Beauvais Cathedral. These models were loaded with their own self-weight and wind loads on their horizontal projection, corresponding to realistic scenarios at Beauvais, France. The outward thrusts that would be transmitted to the flying buttress system were as anticipated for all models. Von Mises stresses were analyzed to qualitatively examine the stress distributions and structural efficiency of each model.

Key Words: Vault, Finite Element Analysis, Beauvais Cathedral, Structural Analysis

Table of Contents

Acknowledgements

1 Background and Introduction 5

2 ANSYS 10

2.1 Finite Element Analysis 10

2.2 ANSYS Parametric Design Language 10

3 Model of An Ideal Quadripartite Vault 11

3.1 Geometry 12

4 Model of an Ideal Sexpartite Vault 17

5 Model of a Beauvais Cathedral Vault 20

6 Loading 22

6.1 Dead Load 22

6.2 Wind Load 22

7 Results 23

7.1 External Thrusts 23

7.2 von Mises Stresses 25

8 Model Limitations 27

9 Conclusions 28

10 References 29

11 Appendix Listing 30

Appendix A 31

Appendix B 36

Appendix C 47

Appendix D 50

ACKNOWLEDGEMENTS

This project would have never come to fruition without Art History Professor Michael Cothren's passion for Gothic art and architecture and Marilyn Stokstad's text "History of Western Art." It was here I first encountered a gothic cathedral vault and became fascinated by its geometry and structural efficiency. Professor Robert Mark of Princeton University, with his wisdom and expertise, advised me to switch cathedrals from Chartres to Beauvais and pointed me in the correct direction. Data on Beauvais would not have been attainable if not for Professor Andrew Tallon of Vassar College, who granted me access to a large data source available at the Kycera Family Foundation's architectural documentation project, CyArk.org. Finally, Professor Leonard van Guhlick at Lafayette College spoke with me extensively on similar research being conducted at his institution and provided clarity that I desperately needed.

In the Swarthmore College Engineering Department: Thanks to Holly Castleman for always being a friendly face and to Ed Jaoudi for helping me with various technical "emergencies." While all the professors in the department have had their impact on me, none has shaped my education more than Professor Faruq Siddiqui, with whom I have taken a majority of my engineering classes. You recognized that I work best with a hot fire constantly burning and were never afraid to turn up the heat. At the same time, you respected my decisions to immerse myself in music and the clarinet, something for which I will always be grateful. Thank you for helping me achieve my goal of an engineering major.

Kelsey: We might have earned better grades, gotten more work done, or helped save the world if not for the endless hours we have spent goofing off in Hicks or in each others' company. But it was all still worth it.

To my Mom and Dad: Even though you stared in wonder at photos of your little girl thinking, "What happened such that our daughter wants to be an engineer?", you still supported every success, failure, crazy idea, and interest with guns blazing…

1 BACKGROUND AND INTRODUCTION

Before launching into the background of this particular design project, it is necessary to discuss gothic architecture qualitatively to provide a foundation for the reader. Gothic architecture was largely determined by the theological need for stained glass in cathedrals. While the details of the religious aspects are beyond the scope of this introduction, it is important to consider that 2-dimensional media initially determined cathedral architecture. The pointed arch is the foundation of this architecture **Figure 1**.

Figure 1: Gothic Pointed Arch

Arches commonly occur in other types of medieval architecture. For example, the ribbed vaults that are the ultimate focus of this project are formed via the intersections of pointed arches **Figure 2**.

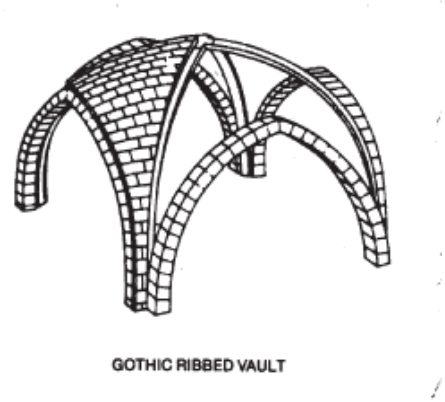

Figure 2: Gothic ribbed vault (four-part) formed by the intersection of pointed arches

Medieval masons had two types of vault available to them in a new construction project: a quadripartite (four-part) vault or a sexpartite (six-part) vault **Figure 2 and Figure 3**. Neither vault was more economic or efficient to build; in fact, cathedral architectural decisions were based largely on trends in gothic architecture rather than realistic constraints. During the high gothic period, which was clearly defined by the early 13th century, most cathedrals were built in the style of Chartres Cathedral (Cathedral of Notre-Dame at Chartres) which was embodied as a codification of gothic trends. After the construction of Chartres and the formulation of the Chartrian model, every cathedral excepting Bourges Cathedral followed the model to the smallest detail. Since the model employed quadripartite vaults, it was a clear choice for medieval masons.

Figure 3: Ribbed sexpartite vault (six-part) in Beauvais Cathedral

Gothic structural systems rely heavily upon a complex system of flying buttresses. First and Second Generation Gothic structures featured thicker walls to better support the weight of the vaults and to resist wind loads. Thinner walls became more desirable as cathedrals began to compete with each other for structural might and general pride. As a result, the flying buttress system was born **Figure 4**.

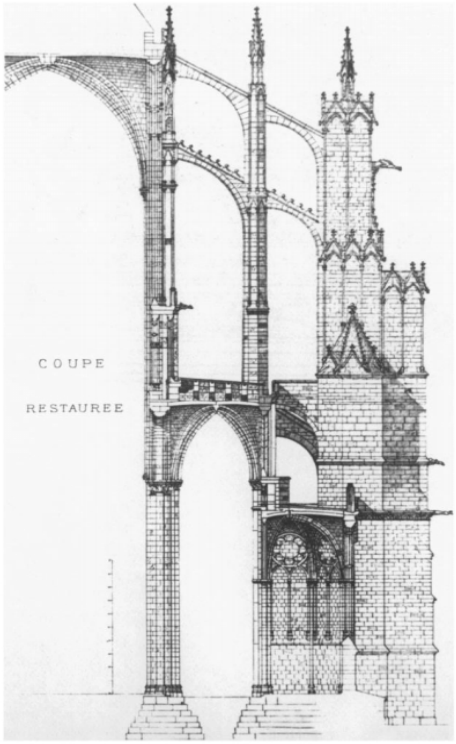

Figure 4: Cross-section of Beauvais demonstrating flying buttress system

The upper choir vaults exert both their own weight and reactions to various types of loading on the buttresses, which in turn transfer these loads to the massive piers of the exterior of the cathedral. The horizontal outward thrust arches exhibit when loaded causes this load transmission. Since vaults are based upon this gothic arch geometry, they transmit load similarly. For this report, the structural function of the other architectural elements of the cross-section is superfluous.

This report places particular significance on Beauvais Cathdral for two main reasons: it is both the tallest gothic monument and the feeblest **Figure 5**. The original quadripartite choir vaults collapsed in 1284, a mere 12 years after their completion, leading architects and masters of the period to replace them with older and less-fashionable sexpartite vaults.

Figure 5: Interior of Beauvais Cathedral, reaching a total height of 157.5ft

There are numerous theories speculating on this counter-intuitive shift in design. E. E. Viollet-le-Duc in his *Dictionnaire raissoné de l'architecture française* and Jacques Heyman in his analysis of Beauvais Cathedral concur that the collapse of the vaults was a result of too-slender colonettes. However, more recent scholarship indicates that this argument cannot be accepted on the basis that in repairing the cathedral, additional measures were taken to redesign the vaults in sexpartite form. Had collapse been caused by inadequately designed colonettes, the only repairs necessary would be to the critical regions (Mark, 1976). Mark and Wolfe argue through research in photo-elastic modeling that wind loads on the intermediate buttresses provided enough force to generate cracking which in turn, resulted in collapse of the vaults.

If this is indeed true, the repairs made to the cathedral reflect this logic. The master made certain to reduce horizontal loads transmitted to the intermediate buttresses, accomplished via the unusual switch to sexpartite vaults. Sexpartite vaults are significantly lighter than quadripartite vaults of the same area and would therefore reduce the horizontal shearing loads transmitted to the wall buttresses. This being said, sexpartite

vaults also create approximately 50% more longitudinal thrust than quadripartite vaults and as a result, additional wall buttresses are necessary to support the transmitted loads (Mark and Taylor, 1982). These additions also aided in reduction of wind loads on the clerestory, ultimately leading to lesser loads on the intermediate buttresses.

While this explains the perceived regression from quadripartite to sexpartite vaults, should sexpartite vaults be condemned as forever obsolete because of their prohibitively large thrusts? Perhaps the medieval engineers did not exhaust the structural possibilities of six-part vaults and abandoned them prematurely in favor of heavy four-part vaults. After all, Beauvais remains standing, albeit feebly, after 700 years in its current sexpartite form. It is this notion that has inspired the research presented here.

2 ANSYS®

2.1 Finite Element Analysis

The finite element method, also referred to as the direct stiffness method, is the basis of analysis of plates and shells in structural software like ANSYS®. As it can be applied to most types of structures, finite element methods are preferable in computer methods as they encompass a majority of scenarios and constraints. Finite element analysis operates under the assumption that the analyzed structure, in this case, some form of a vault, can be divided into numerous basic scenarios. Furthermore, it is assumed that these basic structural scenarios will superimpose to equate to the original structure. Behavior is thus determined by examining smaller units and combining them to construct a larger solution.

Unlike the compatibility equation method, which utilizes unknown redundant forces and flexibility coefficients, the finite element method involves constructing equilibrium equations based upon unknown joint displacements and stiffness coefficients. The forces in a given member can thus be determined via force-displacement relationships. The equilibrium equations take the following form in matrix notation: Equilibrium equations between elements relate the transfer of forces to adjacent elements in proportion to the element stiffness, size, and properties. Internal stresses then superimpose to model the response of the structure under loading.

2.2 ANSYS® Parametric Design Language

The ANSYS® Parametric Design Language (ADPL) was used to create the input for the vault models, rather than the often cumbersome and faulty graphical user interface. Not only does ADPL facilitate changes in the model, but since ADPL can be run from any .txt file, the code is easily stored and transported. Vault parameters can also be altered simply at the beginning of the code such that various geometries could be analyzed easily.

3 MODEL OF AN IDEAL QUADRIPARTITE VAULT

As briefly mentioned in the Introduction of this report, quadripartite vaults are considered the most structurally efficient of medieval vaulting technologies because of how they distribute internal stresses and external thrust to flying buttresses. To account for variance in construction methods and the lack of original drawings or plans for the original four-part vaults at Beauvais, an ideal four-part vault was modeled. This model assumed that the groins of the vault were parabolic and the peaks of the vault were formed by the lines y = x and y = -x. Footprints of Beauvais cathedral were obtained via laser scans collected by Professor Andrew Tallon of Vassar College and made available to the public via the Kycera Family Foundation's documentation project CyArk.org **Figure 6**.

Figure 6: Footprint of Beauvais Cathedral from CyArk

From the footprint data and point clouds, it was determined that the arches were equilateral arches that sprung from a height of approximately 4m **Figure 7**. Since an ideal quadripartite vault is based loosely on the solid formed by two intersection hyperbolic

paraboloids, it was determined that parabolas fit to the curve formed by the arch geometry and the springing point would best model an ideal quadripartite vault in ANSYS®.

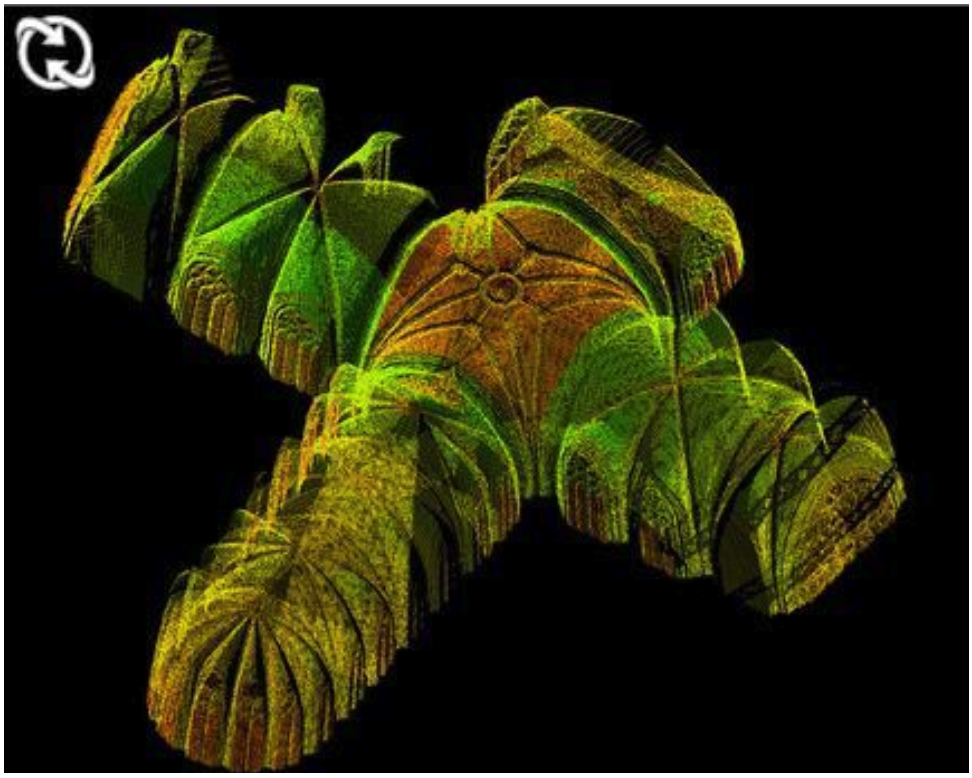

Figure 7: Beauvais choir vaults point cloud laser scan data from CyArk, demonstrating equilateral arch geometries used for modeling ideal quadripartite model.

3.1 GEOMETRY

To construct the vault geometry, ANSYS® was implemented rather than another solid modeling software. Solid models transferred to ANSYS® for finite element analysis often suffer loss of data in the process and as a result, it is preferable to program the geometry directly into the analysis software. Keypoints were constructed to define the vault geometry according to the equations outlined above. The keypoints were then meshed and assigned shell, material, and element properties. This approach is known as the Solid Modeling approach in which models are constructed from the most basic element to the most complex. An alternative in ANSYS® is the Direct Generation method in which elements or volumes are first defined and then assigned automatically generated keypoints,

lines, and areas. However, the direct generation method is only available for shapes within the ANSYS® library of predefined shapes, of which the quadripartite vault is not included.

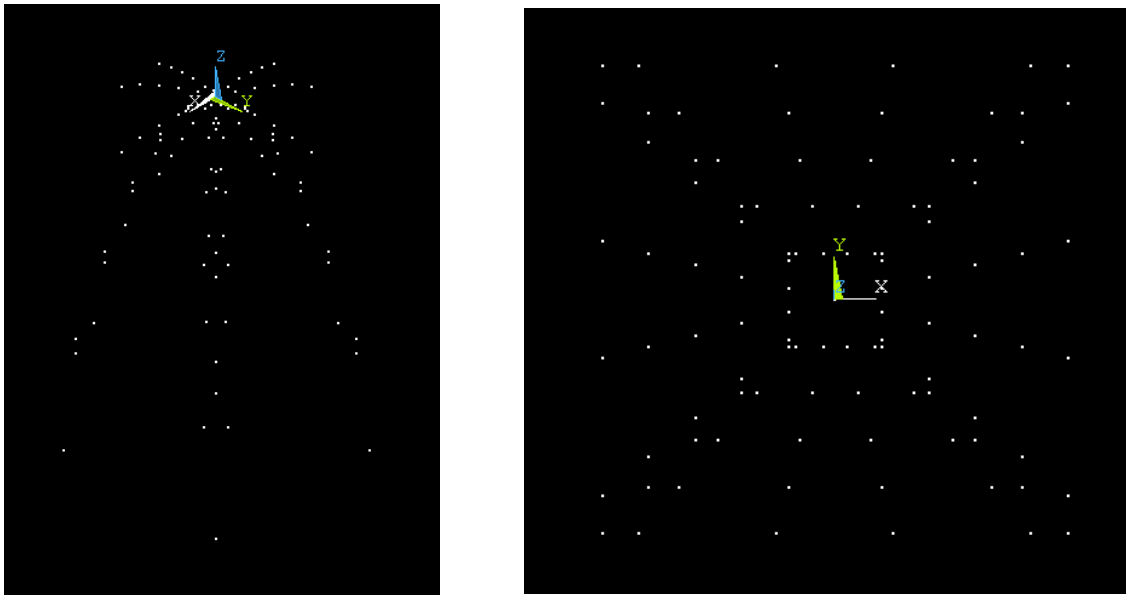

Figure 8: Keypoint map for ideal quadripartite vault model

The shear number of keypoints required for this model demanded a looping code structure. This loop effectively generates a series of parabolas according to the equation set above, whose horizontal projection is a set of lines emanating from an origin point. It is important to note that only six parabolas could be generated in ANSYS® due to the limited capabilities of the SPLINE function, which creates best-fit lines between a maximum of six keypoints. However, the potential increase in precision with creating more than six parabolas is negligible and ANSYS®'s limitations were not considered a hindrance.

The parabolas themselves were similarly fit with the SPLINE function. While technically, parabolas require only three points to define them, ANSYS® did not fit them as well as expected. Thus, six points were plotted per parabola such that the points of the parabolas with maximum curvature were well defined. The line formations below were generated after connecting all keypoints with splines.

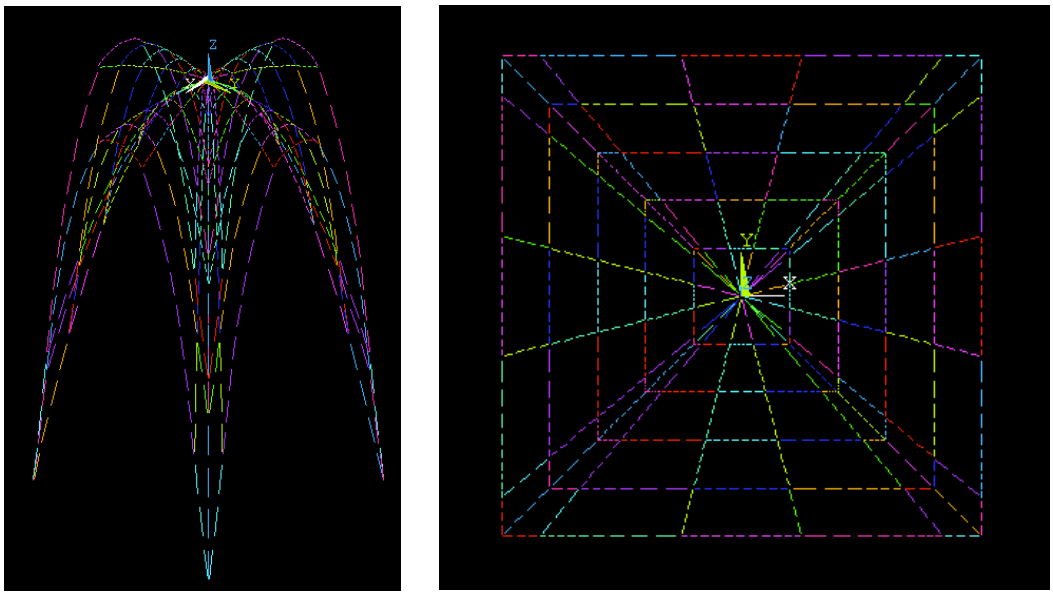

Figure 9: Spline curves for ideal quadripartite model

"Skinning" a non-planar shape over the vault geometry created areas between lines and keypoints **Figure 9**. ANSYS® performs this step automatically by applying a "Coon's Patch" to the figures. To accurately generate this shape, it is necessary to tabulate the keypoint numbers and line numbers to then associate them as defining an area. Once and area is defined, it is not necessary to create volumes—the shell element in ANSYS® receives an input thickness vault, thus creating a volume.

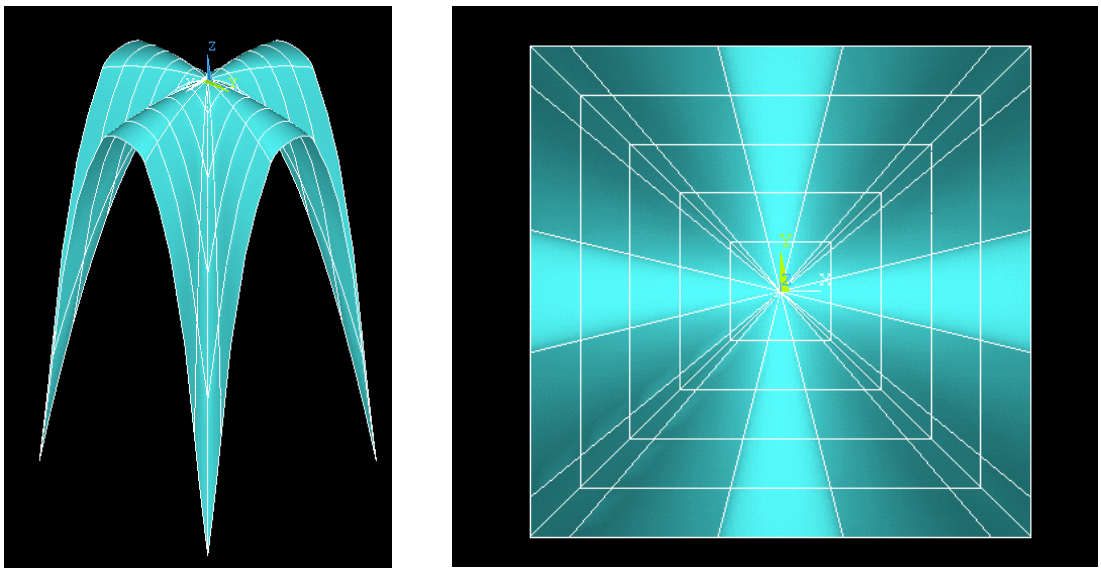

Figure 10: Ideal quadripartite model shown with Coon's Patch applied. Axiometric view on left and vertical projection on right.

It is important to note that keypoints, areas, and lines have no structural properties assigned to them. In complex models like the ideal quadripartite vault once geometries are defined, nodes must also be created. Using an automatic node generation based on points of greatest curvature, structural nodes were created to perform the analysis **Figure 11**.

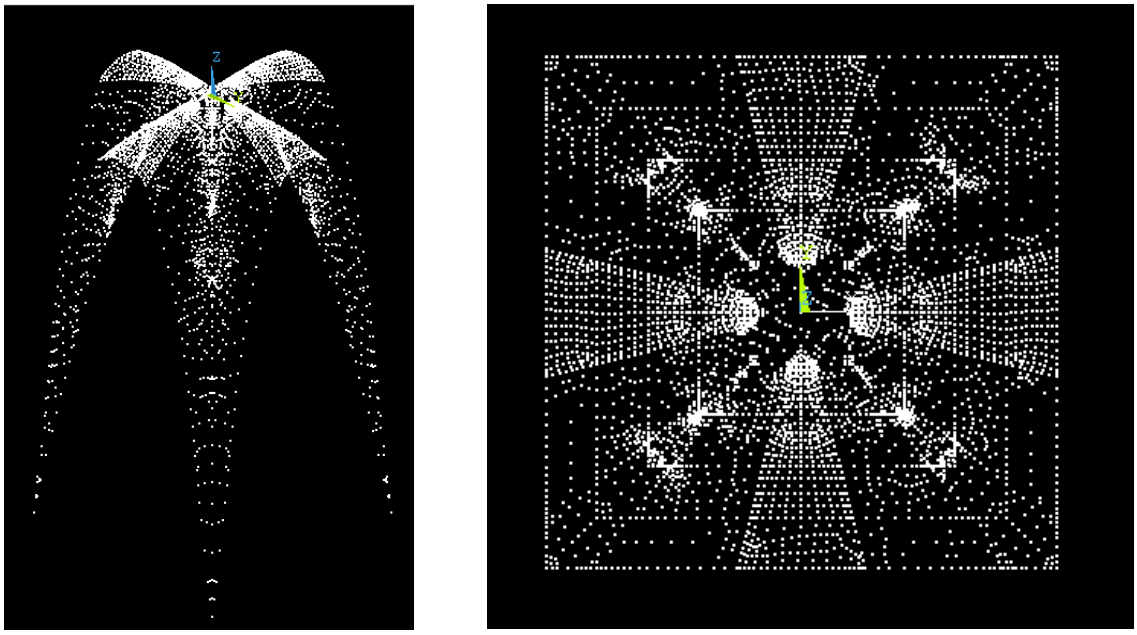

Figure 11: Ideal quadripartite vault model showing automatic node generation. Axiometric on left, vertical projection on right

To analyze the model as a finite element problem, it is necessary to mesh the areas into discrete elements. While this can be accomplished manually, the AMESH command can creat a more thorough mesh via automatic sizing algorithms. Points of greatest curvature receive the smallest mesh while more planar sections receive a less-fine mesh. This enables a more efficient yet precise solution.

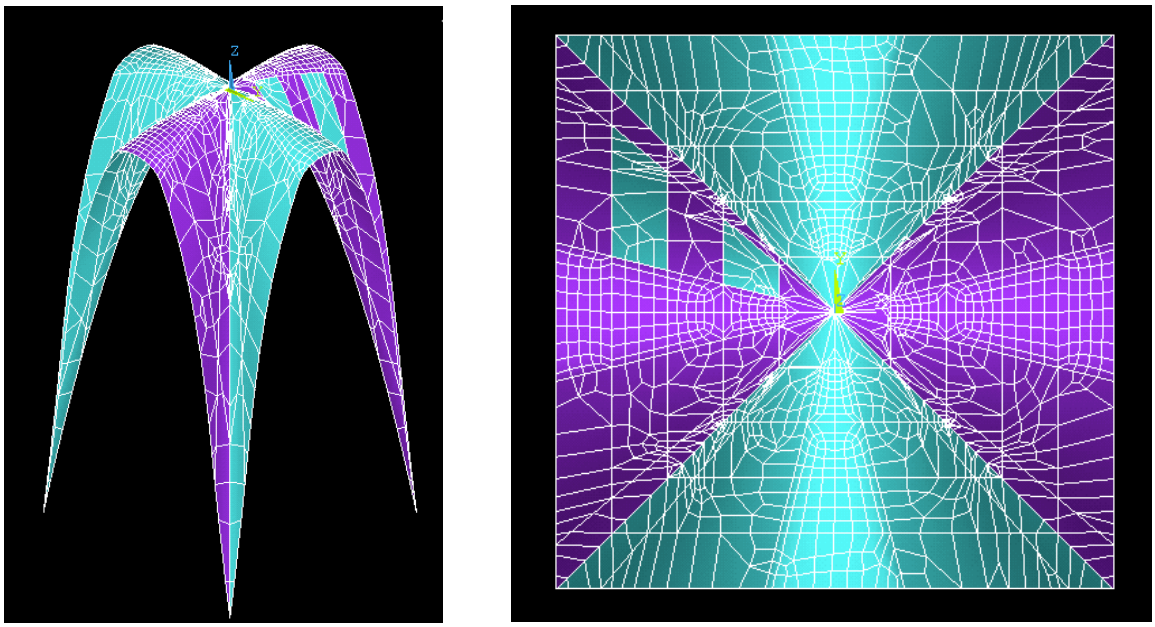

Figure 12: Ideal quadripartite model shown with automatic mesh generation. Axiometric view on left and vertical projection on right.

4 MODEL OF AN IDEAL SEXPARTITE VAULT

To better approach the geometry of a Beauvais Cathedral sexpartite vault, an ideal sexpartite vault was modeled. In this case, "ideal" refers to a vault with the footprint of a regular hexagon. The most significant difference between this model and the ideal quadripartite model is the method in which they are coded into ANSYS®. The quadripartite model was relatively simple: one of the four sections was created and simply mirrored across the x-and y-axes. However, since the sexpartite model is not related to a Cartesian coordinate system, but rather, a polar coordinate system, the model had to be generated via a looping mechanism. The model began in Cartesian coordinates but switched to polar coordinate systems to plot vault segments 60-degrees apart. Within each change to polar coordinate system, there was switch back to Cartesian coordinates so that the parabolas could be best defined. The code for this model is reproduced in Appendix B of this report.

This being said, the theory and practice behind the remaining aspects of the model are identical to that described in chapter 3 of this report. Node and mesh generation were achieved via the same commands as in the quadripartite vault. The sexpartite vault is represented in **Figures 13-16** below.

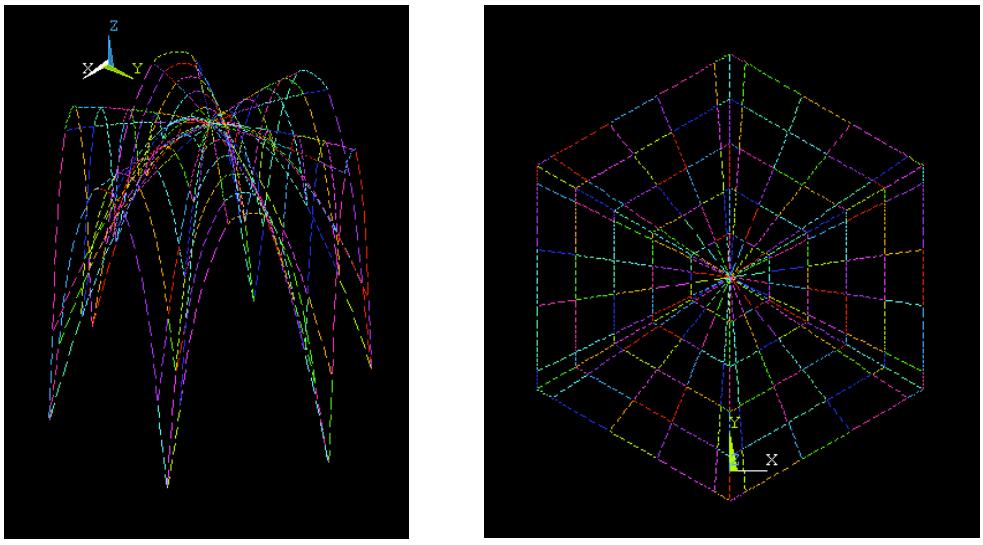

Figure 13: Ideal sexpartite model showing lines used to define areas. Axiometric on left and vertical projection on right.

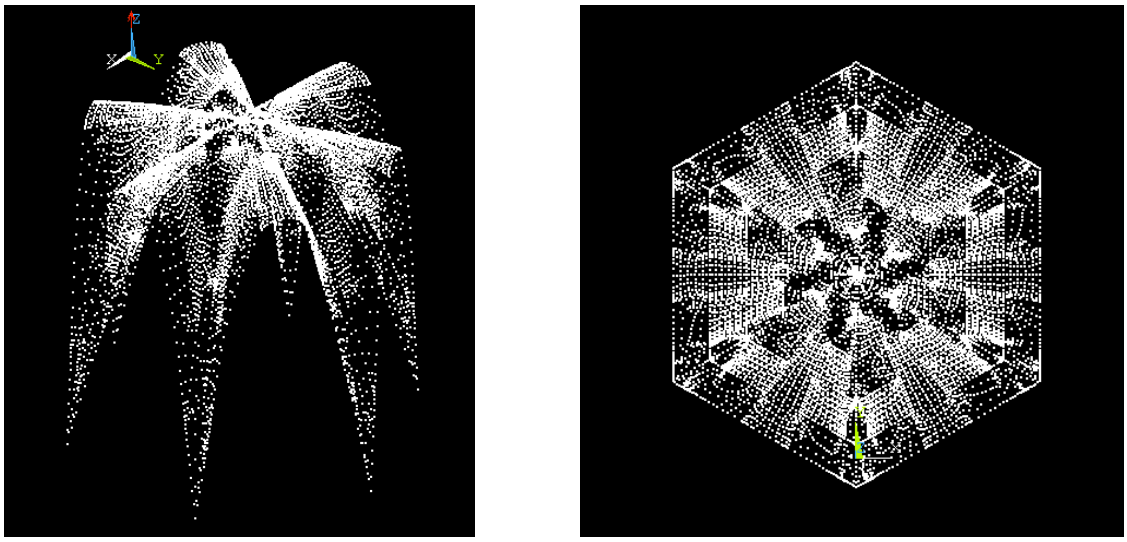

Figure 14: Ideal sexpartite vault showing automatically generated nodes. Axiometric on left and vertical projection on right.

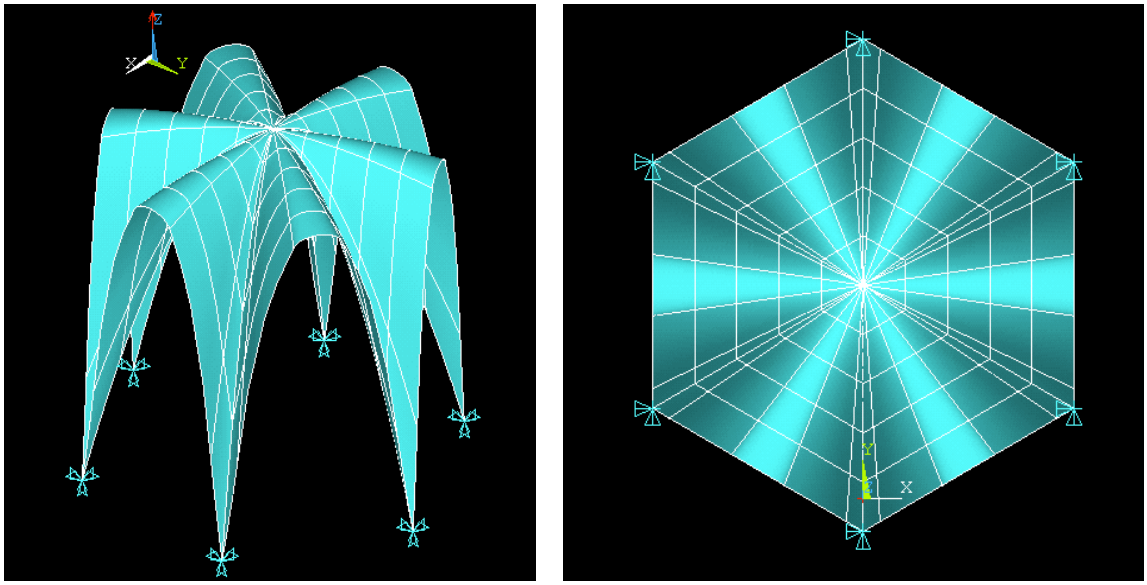

Figure 15: Ideal sexpartite vault showing areas and restraints. Axiometric on left and vertical projection on right.

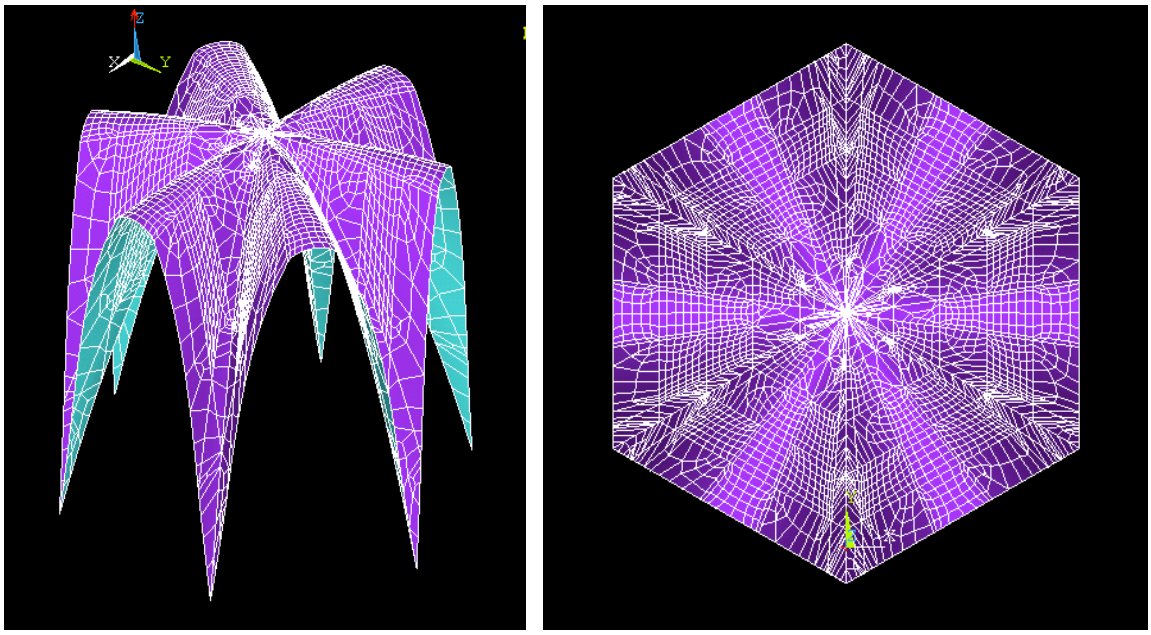

Figure 16: Ideal sexpartite vault showing elements and automatic mesh generation. Axiometric on left and vertical projection on right.

5 MODEL OF BEAVAIS CATHEDRAL VAULT

The model of an actual vault from Beauvais Cathedral was constructed according to the geometries outlines by the point cloud from CyArk **Figure 7**. Since the vaults approximately contain equilateral arches (arches in which an equilateral triangle can be inscribed), it was simple approximate these values with parabolic spline curves. Since this vault, while a six-part vault is perhaps more similar to the quadripartite model, an identical coding process was implemented. One segment of the vault was reflected across the x- and y-axes. Again, the remaining theory behind the finite element modeling tools can be found in section 3 of this report. The code for this vault may be located in Appendix C. Below are images of the Beauvais Cathedral Model **Figures 17-19.**

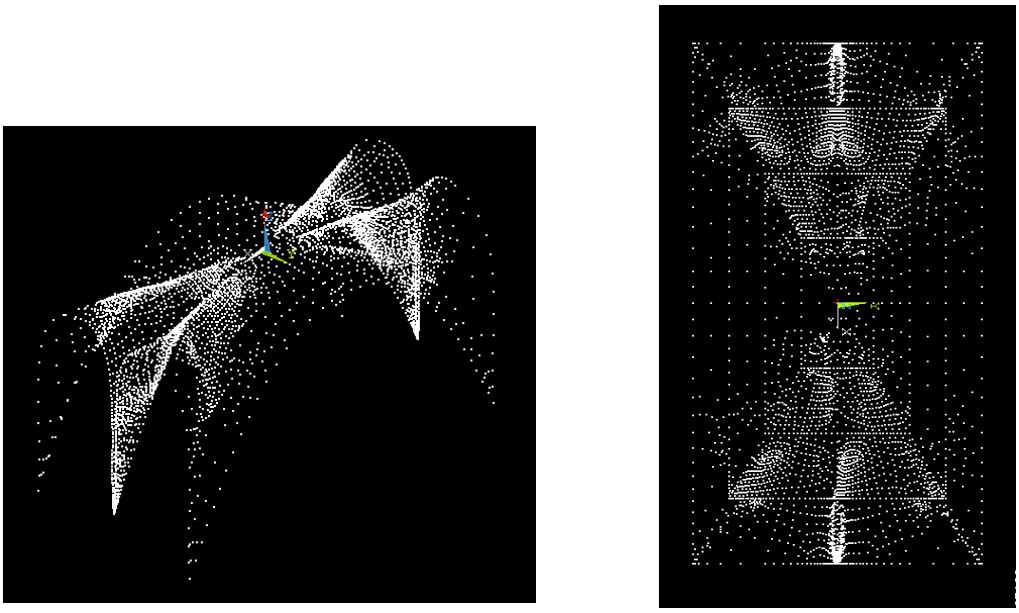

Figure 17: Beauvais Cathedral vault model showing automatically generated nodes. Axiometric on left, vertical projection on right.

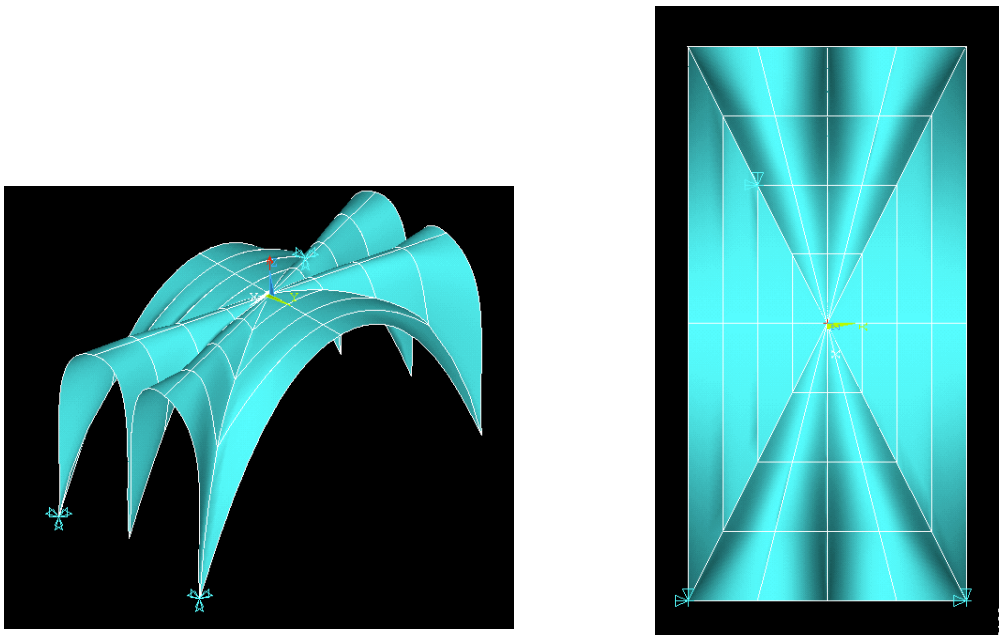

Figure 18: Beauvais Cathedral vault model showing areas. Axiometric on left, vertical projection on right

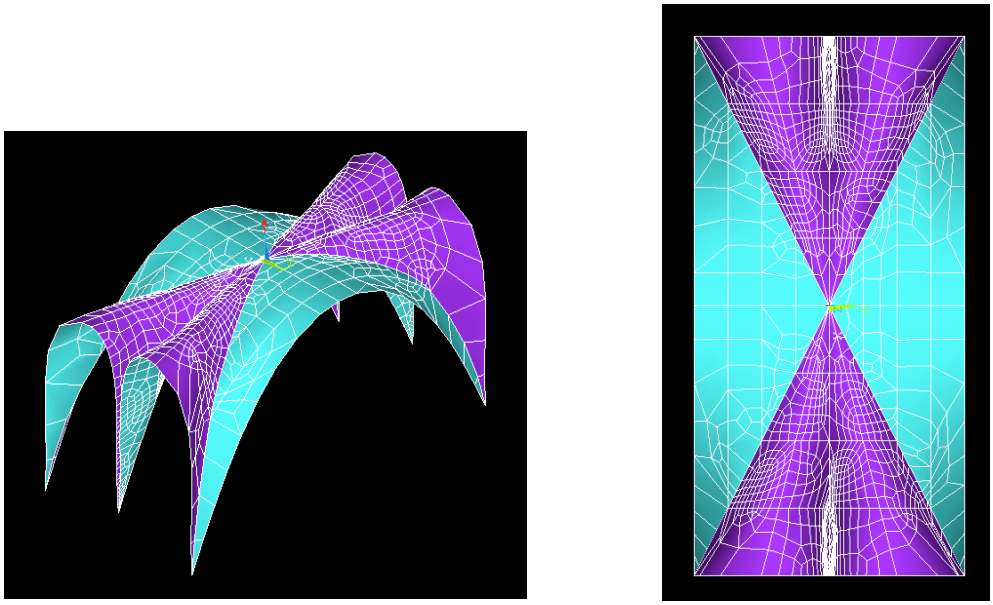

Figure 19: Beauvais Cathedral vault model showing elements and automatically generated mesh. Axiometric on left, vertical projection on right.

6 LOADING SCENARIOS

6.1 Dead Load

The self-weight of the vault is modeled in ANSYS® as a gravity load of $9.81 m/s^2$ applied at every nodal location on the vaults. ANSYS® assumes an input density of stone masonry and vertical acceleration and automatically applies the load based on the model shape and material properties.

6.2 Wind Load

Wind loads as prescribed by the ASCE code are not applicable to vault-type structures and any attempt to better model wind loading would still not approximate the vaults' actual response to wind. Therefore, a worst-case scenario was assumed: wind loads of 100mph acting directly on the horizontal projection of the vaults. Further analysis of wind loads would be for purely pedagogical purposes—the only experiments worth conduction would be wind tunnel tests on a scale model of a vault.

Wind load not only acts on the horizontal projection, but also creates an uplift pressure that counteracts the gravity load. Values for uplift were difficult to determine given that the vaults are not specified to a given code and are part of a larger structure. Wind will behave entirely differently if the vaults were analyzed as a part of the entire cathedral. Ultimately, a wind pressure of 100mph was applied to the nodes on the horizontal projection and a reasonable value for wind uplift pressure was applied on all nodes. It is important to note the differences in horizontal projections of the vault models: the wind will be applied directly on an edge for both the ideal sexpartite and Beauvais vault models. However, for the ideal sexpartite model, much of the wind will be applied on the groined regions. Thus, for the latter model, point loads were applied on both edge and area of the approximate horizontal projection.

7 RESULTS

Two components of loading results are analyzed in this report: external thrust that would normally be delivered to the flying buttressing system and internal stresses. Since internal stresses for complicated structures like vaults are difficult to characterize, the Von Mises stresses were examined.

7.1 External Thrust

Quadripartite vaults and sexpartite vaults have very different external thrusts in that the sexpartite vaults have a much greater horizontal component, where quadripartite vaults are theoretically equal in both directions **Figure 20**. The external thrusts correspond to the reaction solution of each of the feet of the vault.

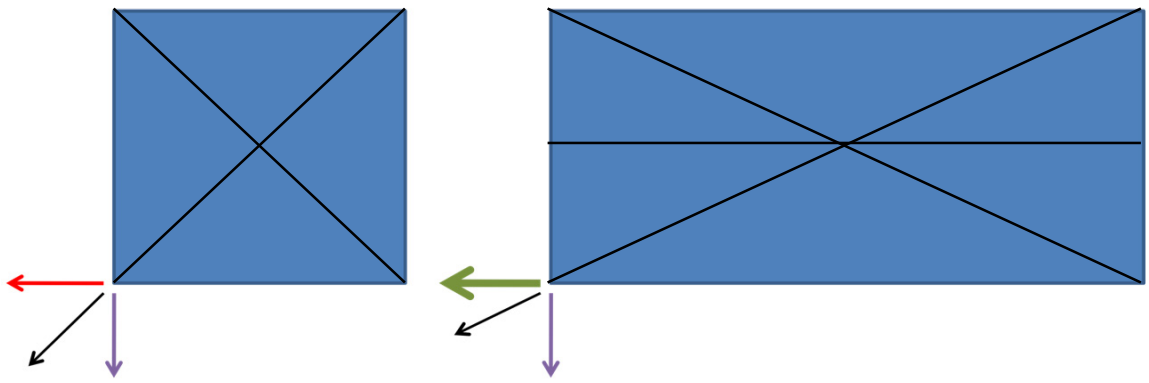

Figure 20: Diagrams of quadripartite and sexpartite footprints demonstrating magnitudes and directions of external thrust to flying buttresses

Table 1: Reaction Solution for Ideal Quadripartite Vault Model (in N)

Node #	Fx	Fy	Fz
1	-5437.9	-5466.0	54104
6	-5426.2	5360.7	53873
32	5444.3	5465.2	54126
42	5419.8	5390.4	53854

Table 2: Reaction Solution for Ideal Sexpartite Vault Model (in N)

Node #	Fx	Fy	Fz
1	-463.74	-295.63	3100.7
6	-429.83	219.75	3012.8
35	-15.084	458.48	2948.9
60	96.919	70.981	776.08
85	481.14	-260.82	3122.6
110	15.223	-576.86	3173.3

Table 3: Reaction Solution for Beauvais Vault Model (in N)

Node #	Fx	Fy	Fz
1	557.65	265.24	1797.4
6	-469.15	301.68	1703.5
32	-247.37	-118.85	602.95
42	175.66	-162.02	544.83

7.2 Von Mises Stresses

Reasonable values for Von Mises stresses were obtained from loading. Stress gradients are depicted below **Figures 21-23**. The quadripartite model clearly demonstrates the lowest internal stresses and distributes them evenly across the structure. With the exception of the feet, there is little stress build-up in the vault. The sexpartite model demonstrates a concentration of stresses along the groins and again, in the feet. The same behavior is reflected in the Beauvais Cathedral model. If these vaults were connected to the remainder of the structure, concentrations of stress in the feet would not be as evident and may not exist at all. This discrepancy is addresses in section 8 of this report. Deflected shapes are presented in Appendix D.

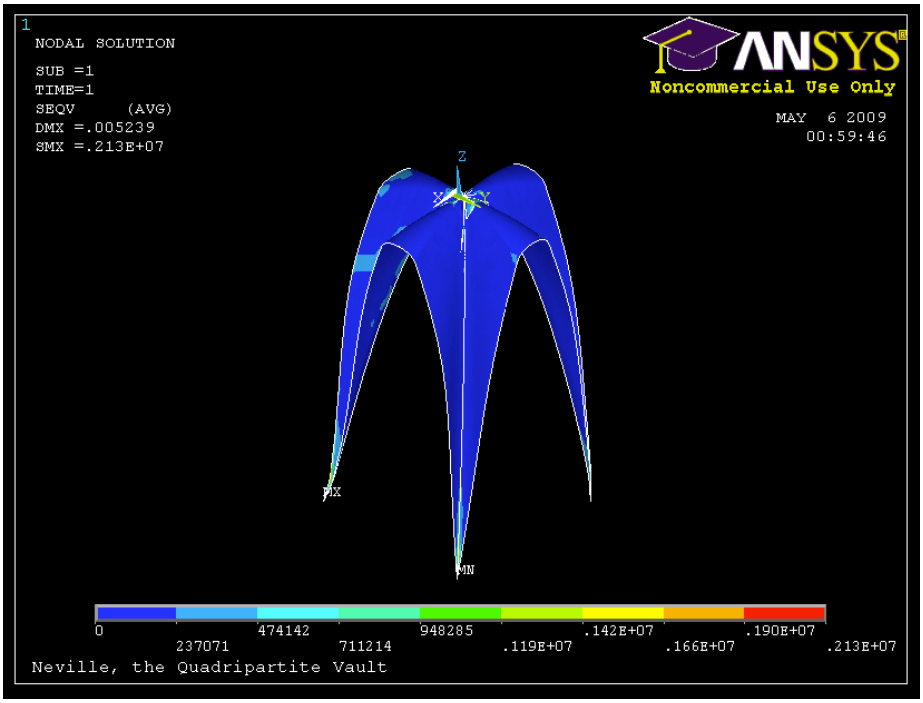

Figure 21: von Mises stresses for ideal quadripartite model. Max deflection is 0.005m and max stress is 237kPa

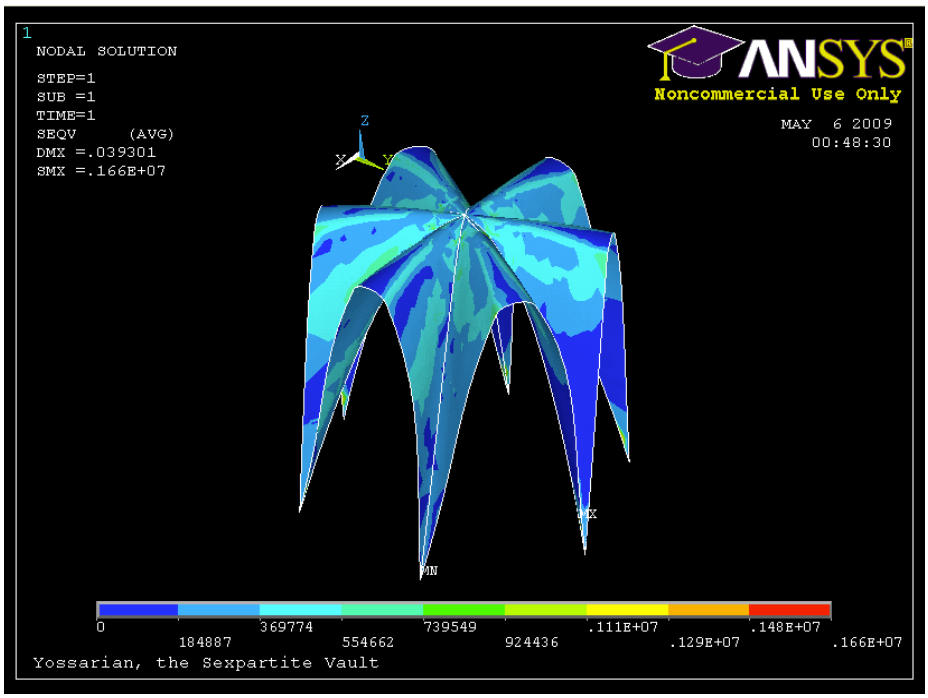

Figure 22: von Mises stresses for ideal sexpartite model. Max deflection in 0.039m and max stress is 166kPa.

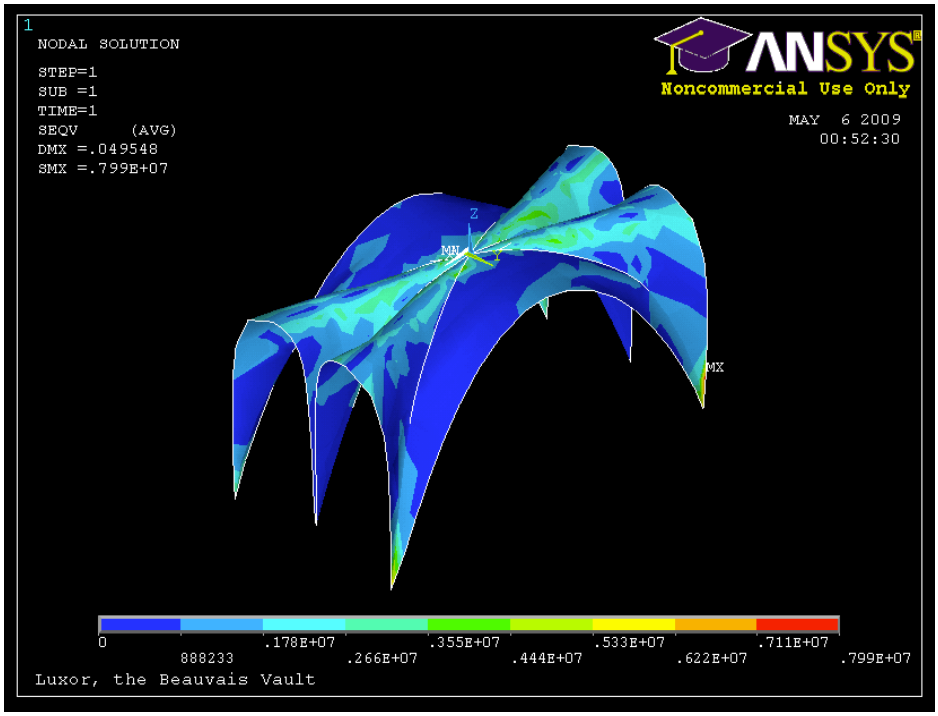

Figure 23: von Mises stresses for Beauvais Cathedral model. Max deflection is 0.049m and max stress is 799 kPa

8 MODELING LIMITATIONS

I feel compelled to briefly discuss limitations in my project and with computer modeling in general. To begin, vaults are impossible to analyze after being plucked from the original structure. A full description of their structural characteristics would depend on their interactions with the entirety of a gothic cathedral. However, modeling the entire cathedral both were out of the scope of this project and my personal interest. Admittedly, it is unfair to simply model one portion of a larger structure without consideration of the remainder of the structure.

Furthermore, there are limitations in computer modeling. Perhaps the most significant of these is the differences in construction versus a mathematical computer model. Modern construction techniques have difficulties with a computer-precise level of accuracy—medieval construction was even more variable. Within modeling software, there are other assumptions that must be considered. For example, structural properties that programs like ANSYS® assign to nodes cannot be considered completely accurate. While they are indeed excellent approximations, the amalgam of small discrepancies could contribute to significant errors in computer models.

The thickness of the vaults was yet another limitation because the masonry materials have yet to be measured. Models were given a thickness of 0.3125m, corresponding to the thickness Prof. Robert Mark of Princeton University used in his research in the quadripartite vaults in Chartres Cathedral. This value is realistic and an excellent alternative to guessing.

9 CONCLUSIONS

I was able to successfully design models for three vaults: an ideal quadripartite vault, and ideal sexpartite vault, and a vault from Beauvais Cathedral. Models were programmed in ANSYS® using the ANSYS® Parametric Design Language and ultimately loaded with horizontal wind load and gravity load. Results confirmed already-published scholarship and provided insight into medieval vaulting technologies via finite element computer methods. This project was conducted with no cost incurred to the Swarthmore College Engineering Department and satisfies the ABET criteria 3c, which outlines requirements for design projects.

10 REFERENCES

ANSYS. ADPL Programmer's Guide: Release 11.0. Canonsburg, PA. August 2007.

Borg, A. and Mark, R. Chartres Cathedral: A Reinterpretation of Its Structure. The Art Bulletin 55 (3): 367-372.

Mandrick, Jessica. Hyperbolic Paraboloid Thin Shell Structures. Swarthmore College E90 Report, May 2007.

Mark, R. et. AL. 1973. Photoelastic and Finite-element Analysis of a Quadripartite Vault. *Experimental Mechanics*, August 1973: 322-329.

Wolfe, M. and Mark, R. The Collapse of the Vaults in Beauvais Cathedral in 1284. The Medieval Academy of America 51 (3): 462-476.

***Spring of 2009, I took Art History 45: Gothic Art and Architecture with Professor Michael Cothren. Much of the background information used in this project came from class lectures and discussions.*

11 APPENDIX LISTING

A ADPL Code for Ideal Quadripartite Vault Model

B ADPL Code for Ideal Sexpartite Vault Model

C ADPL Code for Beauvais Cathedral Vault Model

D Deflected and Undeflected Shapes for Models

Appendix A: ADPL Code for Ideal Quadripartite Vault Model

```
/FILNAM, Neville
/title, Neville, the Quadripartite Vault
/prep7

!initial values, definition of parameters
      x=0
      y=0
      z=0
      z_one=0
      z_two=0
      z_three=0
      Q=1   !keypoint numbering system


R=4.6188021
y=R

*DOWHILE, y
      x=R
      x=y

      z_one=-((SQRT(3)/2)*(y**2))
      z_two=-((SQRT(3)/2)*((y/4)**2))
      z_three=-((SQRT(3)/2)*((0.84*y)**2))

      K, Q, x, y, z_one

      K, Q+1, x, (0.84*y), z_three

      K, Q+2, x, (y/4), Z_two

      K, Q+3, x, -(y/4), z_two

      K, Q+4, x, -(0.84*y), z_three

      K, Q+5, x, -y, z_one

      SPLINE, Q, Q+1, Q+2, Q+3, Q+4, Q+5,

      Q=Q+6
      y=y-(R/5)
*ENDDO


K, 0, 0, 0


RUNS=6
I=1

*DOWHILE, RUNS
      SPLINE, I, I+6, I+12, I+18, I+24, 31,
      I=I+1
```

```
      RUNS=RUNS-1
*ENDDO


RUNS=5
I=1
J=26

*DOWHILE, RUNS
      AL, I, I+5, J, J+5
      AL, I+1, I+6, J+5, J+10
      AL, I+2, I+7, J+10, J+15
      AL, I+3, I+8, J+15, J+20
      AL, I+4, I+9, J+20, J+25
      I=I+5
      J=J+1
      RUNS=RUNS-1
*ENDDO

RUNS=5
I=21
J=30

*DOWHILE, RUNS
      AL, I, J, J+5
      I=I+1
      J=J+5
      RUNS=RUNS-1
*ENDDO

ARSYM, x, ALL, , , , 1, 0,

!_____________________________________________

R=4.6188021
x=R
Q=63

*DOWHILE, x
      y=R
      y=x
      z_one=-((SQRT(3)/2)*(x**2))
      z_two=-((SQRT(3)/2)*((x/4)**2))
      z_three=-((SQRT(3)/2)*((0.84*x)**2))

      K, Q, x, y, z_one

      K, Q+1, (0.84*x), y, z_three

      K, Q+2, (x/4), y, Z_two

      K, Q+3, -(x/4), y, z_two

      K, Q+4, -(0.84*x), y, z_three

      K, Q+5, -x, y, z_one
```

```
      SPLINE, Q, Q+1, Q+2, Q+3, Q+4, Q+5,

      Q=Q+6
      x=x-(R/5)
*ENDDO

RUNS=6
I=63

*DOWHILE, RUNS
      SPLINE, I, I+6, I+12, I+18, I+24, 31,
      I=I+1
      RUNS=RUNS-1
*ENDDO

RUNS=5
I=111
J=136

*DOWHILE, RUNS
      AL, I, I+5, J, J+5
      AL, I+1, I+6, J+5, J+10
      AL, I+2, I+7, J+10, J+15
      AL, I+3, I+8, J+15, J+20
      AL, I+4, I+9, J+20, J+25
      I=I+5
      J=J+1
      RUNS=RUNS-1
*ENDDO

RUNS=5
I=131
J=140

*DOWHILE, RUNS
      AL, I, J, J+5
      I=I+1
      J=J+5
      RUNS=RUNS-1
*ENDDO

ARSYM, y, 51, 75, 1, , 1, 0,

NUMMRG, KP, , ,

!Restraints at keypoints--each corner of vault
DK, 1, UX, 0
DK, 1, UY, 0
DK, 1, UZ, 0

DK, 6, UX, 0
DK, 6, UY, 0
DK, 6, UZ, 0

DK, 42, UX, 0
DK, 42, UY, 0
DK, 42, UZ, 0
```

```
DK, 32, UX, 0
DK, 32, UY, 0
DK, 32, UZ, 0

!element type and properties
ET, 1, SHELL93

MP, EX, 1, 50E9
MP, EY, 1, 50E9
MP, EX, 1, 50E9
MP, NUXY, 1, 0
MP, PRXY, 1, 0
MP, DENS, 1, 2700/9.81

R, 1, 0.3175, 0.3175, 0.3175, 0.3175, 0, 0,
MAT, 1 $ ETYPE, 1 $ REAL, 1 $ TYPE, 1 $ AATT, 1, 1, 1
MSHKEY, 0
MSHAPE, 0, 2D
SMRTSIZE, 5
AMESH, ALL

ESEL, ALL
ASUM, DEFAULT
*GET, SHELLAREA, AREA, , AREA

!Load Scenarios

!Dead load
ACEL, 0, 0, 9.81

!Maximum regional wind load, uplift on roof, horizontal lift on side
!F, ALL, FZ, 0.1764
FK, 113, FY, 20
FK, 108, FY, 20
FK, 94, FY, 20
FK, 6, FY, 20
FK, 57, FY, 20
FK, 114, FY, 20
FK, 109, FY, 20
FK, 95, FY, 20
FK, 110, FY, 20
FK, 105, FY, 20

F, ALL, FZ, -0.074

/solu
solve

SAVE
FINISH
```

Appendix B: ADPL Code for Ideal Sexpartite Vault Model

```
/FILNAM, test3
/title, Yossarian, the Sexpartite Vault
/prep7

!initial values, definition of parameters

      x=0
      y=0
      z=0
      theta=360
      z_one=0
      z_two=0
      z_three=0
      Q=4

!creates point at middle of vault geometry
K, 1, 0, 8, 0,    !keypoint 1: origin
K, 2, 1, 8, 0,    !keypoint 2: defines x axis
K, 3, 0, 9, 0,    !keypoint 3: defines x-y plane

!Change coordinate system to polar.
!CSKP, coordinate system number (>10), cs reference number, origin, x
axis, xy plane,


!create lines emanating from origin (keypoint 1)

theta=360

      y=4.6188021
      R=4.6188021



      !changes coordinate system to polar
      CSKP, 11, 2, 1, 2, 3,


      !creates local cartesian coordinate system
      CLOCAL, 12, 0, 0.0000001, theta, 0,0,0,0

      y=R

      *DOWHILE, y


            x=SQRT(3)*y
            z_one=-((SQRT(3)/2)*(y**2))
            z_two=-((SQRT(3)/2)*((y/4)**2))
            z_three=-((SQRT(3)/2)*((0.84*y)**2))

            K, Q, x, y, z_one

            K, Q+1, x, (0.84*y), z_three
```

```
            K, Q+2, x, (y/4), Z_two

            K, Q+3, x, -(y/4), z_two

            K, Q+4, x, -(0.84*y), z_three

            K, Q+5, x, -y, z_one

            SPLINE, Q, Q+1, Q+2, Q+3, Q+4, Q+5

            Q=Q+6
            y=y-(R/5)

      *ENDDO

!_____________________________________________________________
theta=300


      !changes coordinate system to polar
      CSKP, 11, 2, 1, 2, 3,


      !creates local cartesian coordinate system
      CLOCAL, 12, 0, 0.0000001, theta, 0,0,0,0

      y=R

      *DOWHILE, y


            x=SQRT(3)*y
            z_one=-((SQRT(3)/2)*(y**2))
            z_two=-((SQRT(3)/2)*((y/4)**2))
            z_three=-((SQRT(3)/2)*((0.84*y)**2))

            !K, Q, x, y, z_one

            K, Q, x, (0.84*y), z_three

            K, Q+1, x, (y/4), Z_two

            K, Q+2, x, -(y/4), z_two

            K, Q+3, x, -(0.84*y), z_three

            K, Q+4, x, -y, z_one



            Q=Q+5
            y=y-(R/5)

      *ENDDO
!_____________________________________________________________
theta=240
```

```
      !changes coordinate system to polar
      CSKP, 11, 2, 1, 2, 3,


      !creates local cartesian coordinate system
      CLOCAL, 12, 0, 0.0000001, theta, 0,0,0,0

      y=R

      *DOWHILE, y


            x=SQRT(3)*y
            z_one=-((SQRT(3)/2)*(y**2))
            z_two=-((SQRT(3)/2)*((y/4)**2))
            z_three=-((SQRT(3)/2)*((0.84*y)**2))

            !K, Q, x, y, z_one

            K, Q, x, (0.84*y), z_three

            K, Q+1, x, (y/4), Z_two

            K, Q+2, x, -(y/4), z_two

            K, Q+3, x, -(0.84*y), z_three

            K, Q+4, x, -y, z_one



            Q=Q+5
            y=y-(R/5)

      *ENDDO
!_____________________________________________________________
theta=180


      !changes coordinate system to polar
      CSKP, 11, 2, 1, 2, 3,


      !creates local cartesian coordinate system
      CLOCAL, 12, 0, 0.0000001, theta, 0,0,0,0

      y=R

      *DOWHILE, y


            x=SQRT(3)*y
            z_one=-((SQRT(3)/2)*(y**2))
            z_two=-((SQRT(3)/2)*((y/4)**2))
            z_three=-((SQRT(3)/2)*((0.84*y)**2))
```

```
            !K, Q, x, y, z_one

            K, Q, x, (0.84*y), z_three

            K, Q+1, x, (y/4), Z_two

            K, Q+2, x, -(y/4), z_two

            K, Q+3, x, -(0.84*y), z_three

            K, Q+4, x, -y, z_one



            Q=Q+5
            y=y-(R/5)

      *ENDDO
!_____________________________________________________________
theta=120


      !changes coordinate system to polar
      CSKP, 11, 2, 1, 2, 3,


      !creates local cartesian coordinate system
      CLOCAL, 12, 0, 0.0000001, theta, 0,0,0,0

      y=R

      *DOWHILE, y


            x=SQRT(3)*y
            z_one=-((SQRT(3)/2)*(y**2))
            z_two=-((SQRT(3)/2)*((y/4)**2))
            z_three=-((SQRT(3)/2)*((0.84*y)**2))

            !K, Q, x, y, z_one

            K, Q, x, (0.84*y), z_three

            K, Q+1, x, (y/4), Z_two

            K, Q+2, x, -(y/4), z_two

            K, Q+3, x, -(0.84*y), z_three

            K, Q+4, x, -y, z_one



            Q=Q+5
            y=y-(R/5)

      *ENDDO
```

```
!_____________________________________________________________
theta=60


      !changes coordinate system to polar
      CSKP, 11, 2, 1, 2, 3,


      !creates local cartesian coordinate system
      CLOCAL, 12, 0, 0.0000001, theta, 0,0,0,0

      y=R

      *DOWHILE, y


            x=SQRT(3)*y
            z_one=-((SQRT(3)/2)*(y**2))
            z_two=-((SQRT(3)/2)*((y/4)**2))
            z_three=-((SQRT(3)/2)*((0.84*y)**2))

            !K, Q, x, y, z_one

            K, Q, x, (0.84*y), z_three

            K, Q+1, x, (y/4), Z_two

            K, Q+2, x, -(y/4), z_two

            K, Q+3, x, -(0.84*y), z_three

            !K, Q+4, x, -y, z_one



            Q=Q+4
            y=y-(R/5)

      *ENDDO


SPLINE, 9, 34, 35, 36, 37, 38
SPLINE, 15, 39, 40, 41, 42, 43
SPLINE, 21, 44, 45, 46, 47, 48
SPLINE, 27, 49, 50, 51, 52, 53
SPLINE, 33, 54, 55, 56, 57, 58

SPLINE, 38, 59, 60, 61, 62, 63
SPLINE, 43, 64, 65, 66, 67, 68
SPLINE, 48, 69, 70, 71, 72, 73
SPLINE, 53, 74, 75, 76, 77, 78
SPLINE, 58, 79, 80, 81, 82, 83

SPLINE, 63, 84, 85, 86, 87, 88
SPLINE, 68, 89, 90, 91, 92, 93
SPLINE, 73, 94, 95, 96, 97, 98
SPLINE, 78, 99, 100, 101, 102, 103
```

```
SPLINE, 83, 104, 105, 106, 107, 108

SPLINE, 88, 109, 110, 111, 112, 113
SPLINE, 93, 114, 115, 116, 117, 118
SPLINE, 98, 119, 120, 121, 122, 123
SPLINE, 103, 124, 125, 126, 127, 128
SPLINE, 108, 129, 130, 131, 132, 133

SPLINE, 113, 134, 135, 136, 137, 4
SPLINE, 118, 138, 139, 140, 141, 10
SPLINE, 123, 142, 143, 144, 145, 16
SPLINE, 128, 146, 147, 148, 149, 22
SPLINE, 133, 150, 151, 152, 153, 28

!_____________________________________________________________________________
_______
RUNS=6
I=4

*DOWHILE, RUNS
      SPLINE, I, I+6, I+12, I+18, I+24, 1,
      I=I+1
      RUNS=RUNS-1
*ENDDO
-----
RUNS=5
I=34

*DOWHILE, RUNS
      SPLINE, I, I+5, I+10, I+15, I+20, 1,
      I=I+1
      RUNS=RUNS-1
*ENDDO
-----
RUNS=5
I=59

*DOWHILE, RUNS
      SPLINE, I, I+5, I+10, I+15, I+20, 1,
      I=I+1
      RUNS=RUNS-1
*ENDDO
-----
RUNS=5
I=84

*DOWHILE, RUNS
      SPLINE, I, I+5, I+10, I+15, I+20, 1,
      I=I+1
      RUNS=RUNS-1
*ENDDO
-----
RUNS=5
I=109

*DOWHILE, RUNS
      SPLINE, I, I+5, I+10, I+15, I+20, 1,
```

```
      I=I+1
      RUNS=RUNS-1
*ENDDO
-----
RUNS=4
I=134

*DOWHILE, RUNS
      SPLINE, I, I+4, I+8, I+12, I+16, 1,
      I=I+1
      RUNS=RUNS-1
*ENDDO


---------------

!AL, 1, 6, 296, 151
RUNS=4
I=1
J=151

*DOWHILE, RUNS
      AL, I, I+5, J, J+5
      AL, I+1, I+6, J+5, J+10
      AL, I+2, I+7, J+10, J+15
      AL, I+3, I+8, J+15, J+20
      AL, I+4, I+9, J+20, J+25
      I=I+5
      J=J+1
      RUNS=RUNS-1
*ENDDO
RUNS=5
I=21
J=155

*DOWHILE, RUNS
      AL, I, J, J+5
      I=I+1
      J=J+5
      RUNS=RUNS-1
*ENDDO

-----
RUNS=5
I=26
J=176

AL, 26, 31, 171, 176

*DOWHILE, RUNS
      AL, I, I+5, J, J+5
      AL, I+1, I+6, J+5, J+10
      AL, I+2, I+7, J+10, J+15
      AL, I+3, I+8, J+15, J+20
      AL, I+4, I+9, J+20, J+25

      I=I+5
```

```
      J=J+1
      RUNS=RUNS-1
*ENDDO

RUNS=5
I=46
J=180

*DOWHILE, RUNS
      AL, I, J, J+5
      I=I+1
      J=J+5
      RUNS=RUNS-1
*ENDDO

-----
RUNS=5
I=51
J=201

*DOWHILE, RUNS
      AL, I, I+5, J, J+5
      AL, I+1, I+6, J+5, J+10
      AL, I+2, I+7, J+10, J+15
      AL, I+3, I+8, J+15, J+20
      AL, I+4, I+9, J+20, J+25
      AL, I+5, I+10, J+25, J+30
      I=I+5
      J=J+1
      RUNS=RUNS-1
*ENDDO

RUNS=5
I=71
J=205

*DOWHILE, RUNS
      AL, I, J, J+5
      I=I+1
      J=J+5
      RUNS=RUNS-1
*ENDDO

-----
RUNS=5
I=76
J=226

*DOWHILE, RUNS
      AL, I, I+5, J, J+5
      AL, I+1, I+6, J+5, J+10
      AL, I+2, I+7, J+10, J+15
      AL, I+3, I+8, J+15, J+20
      AL, I+4, I+9, J+20, J+25
      I=I+5
      J=J+1
      RUNS=RUNS-1
```

```
*ENDDO

RUNS=5
I=96
J=230

*DOWHILE, RUNS
      AL, I, J, J+5
      I=I+1
      J=J+5
      RUNS=RUNS-1
*ENDDO

-----
RUNS=5
I=101
J=251

*DOWHILE, RUNS
      AL, I, I+5, J, J+5
      AL, I+1, I+6, J+5, J+10
      AL, I+2, I+7, J+10, J+15
      AL, I+3, I+8, J+15, J+20
      AL, I+4, I+9, J+20, J+25
      I=I+5
      J=J+1
      RUNS=RUNS-1
*ENDDO

RUNS=5
I=121
J=255

*DOWHILE, RUNS
      AL, I, J, J+5
      I=I+1
      J=J+5
      RUNS=RUNS-1
*ENDDO

-----
RUNS=4
I=126
J=276

*DOWHILE, RUNS
      AL, I, I+5, J, J+5
      AL, I+1, I+6, J+5, J+10
      AL, I+2, I+7, J+10, J+15
      AL, I+3, I+8, J+15, J+20
      AL, I+4, I+9, J+20, J+25
      I=I+5
      J=J+1
      RUNS=RUNS-1
*ENDDO

RUNS=4
```

```
I=146
J=280

*DOWHILE, RUNS
      AL, I, J, J+5
      I=I+1
      J=J+5
      RUNS=RUNS-1
*ENDDO

-----

AL, 130, 135, 296, 151
AL, 135, 140, 152, 297
AL, 140, 145, 153, 298
AL, 145, 150, 154, 299
AL, 150, 155, 300

NUMMRG, KP, , ,

!Restraints at keypoints--each corner of vault
DK, 4, UX, 0
DK, 4, UY, 0
DK, 4, UZ, 0

DK, 9, UX, 0
DK, 9, UY, 0
DK, 9, UZ, 0

DK, 38, UX, 0
DK, 38, UY, 0
DK, 38, UZ, 0

DK, 63, UX, 0
DK, 63, UY, 0
DK, 63, UZ, 0

DK, 88, UX, 0
DK, 88, UY, 0
DK, 88, UZ, 0

DK, 113, UX, 0
DK, 113, UY, 0
DK, 113, UZ, 0

!element type and properties
ET, 1, SHELL93

MP, EX, 1, 50E9
MP, EY, 1, 50E9
MP, EX, 1, 50E9
MP, NUXY, 1, 0
MP, PRXY, 1, 0
MP, DENS, 1, 2700/9.81

R, 1, 0.01, 0.01, 0.01, 0.01, 0, 0,
MAT, 1 $ ETYPE, 1 $ REAL, 1 $ TYPE, 1 $ AATT, 1, 1, 1
```

```
MSHKEY, 0
MSHAPE, 0, 2D
SMRTSIZE, 5
AMESH, ALL

ESEL, ALL
ASUM, DEFAULT
*GET, SHELLAREA, AREA, , AREA

!Load Scenarios

!Dead load
ACEL, 0, 0, 9.81

!F, ALL, FZ, 0.1764
FK, 35, FY, 20
FK, 45, FY, 20
FK, 12, FY, 20
FK, 13, FY, 20
FK, 5, FY, 20
FK, 8, FY, 20
FK, 25, FY, 20
FK, 24, FY, 20
FK, 136, FY, 20
FK, 144, FY, 20

F, ALL, FZ, -0.074
/solu
solve

SAVE
FINISH
```

Appendix C: ADPL Code for Beauvais Cathedral Vault Model

```
/FILNAM, Luxor
/title, Luxor, the Beauvais Vault
/prep7

!initial values, definition of parameters

      x=0
      y=0
      z=0

!input keypoints
K, 1, 0, -4, 0,
K, 2, 0, -3, 0,
K, 3, 0, -2, 0,
K, 4, 0, -1, 0,
K, 5, 0, 0, 0,

SPLINE, 1, 2, 3, 4, 5,

K, 6, -8, -4, -SQRT(80),
K, 7, -6, -3, SQRT(35)-SQRT(80),
K, 8, -4, -2, SQRT(60)-SQRT(80),
K, 9, -2, -1, SQRT(75)-SQRT(80),

SPLINE, 6, 7, 8, 9, 5,

K, 10, -8, -2, 0,
K, 11, -6, -1.5, 0,
K, 12, -4, -1, 0,
K, 13, -2, -0.5, 0,

SPLINE, 10, 11, 12, 13, 5,

K, 14, -8, 0, -8,
K, 15, -6, 0, SQRT(18)-8,
K, 16, -4, 0, SQRT(48)-8,
K, 17, -2, 0, SQRT(60)-8,

SPLINE, 14, 15, 16, 17, 5,

K, 18, 8, -4, -SQRT(80),
K, 19, 6, -3, SQRT(35)-SQRT(80),
K, 20, 4, -2, SQRT(60)-SQRT(80),
K, 21, 2, -1, SQRT(75)-SQRT(80),

SPLINE, 18, 19, 20, 21, 5

SPLINE, 6, 1, 18
SPLINE, 7, 2, 19
SPLINE, 8, 3, 20
SPLINE, 9, 4, 21
SPLINE, 6, 10, 14
SPLINE, 7, 11, 15
SPLINE, 8, 12, 16
```

```
SPLINE, 9, 13, 17

AL, 22, 24, 17, 1
AL, 24, 26, 18, 2
AL, 26, 28, 19, 3
AL, 28, 20, 4

AL, 21, 23, 1, 5
AL, 23, 25, 2, 6
AL, 25, 27, 3, 7
AL, 27, 4, 8

AL, 29, 31, 5, 9
AL, 31, 33, 6, 10
AL, 33, 35, 7, 11
AL, 35, 8, 12

AL, 30, 32, 9, 13
AL, 32, 34, 10, 14
AL, 34, 36, 11, 15
AL, 36, 12, 16


ARSYM, x, ALL, , , , 1, 0,
ARSYM, Y, ALL, , , , 1, 0,

NUMMRG, KP, , ,

!Restraints at keypoints--each corner of vault
DK, 6, UX, 0
DK, 6, UY, 0
DK, 6, UZ, 0

DK, 18, UX, 0
DK, 18, UY, 0
DK, 18, UZ, 0

DK, 52, UX, 0
DK, 52, UY, 0
DK, 52, UZ, 0

DK, 44, UX, 0
DK, 44, UY, 0
DK, 44, UZ, 0

!element type and properties
ET, 1, SHELL93

MP, EX, 1, 50E9
MP, EY, 1, 50E9
MP, EX, 1, 50E9
MP, NUXY, 1, 0
MP, PRXY, 1, 0
MP, DENS, 1, 2700/9.81

R, 1, 0.3175, 0.3175, 0.3175, 0.3175, 0, 0,
MAT, 1 $ ETYPE, 1 $ REAL, 1 $ TYPE, 1 $ AATT, 1, 1, 1
```

```
MSHKEY, 0
MSHAPE, 0, 2D
SMRTSIZE, 5
AMESH, ALL

ESEL, ALL
ASUM, DEFAULT
*GET, SHELLAREA, AREA, , AREA

!Load Scenarios
!1197
!Dead load
ACEL, 0, 0, 9.81

!Maximum regional wind load, uplift on roof, horizontal lift on side
!F, ALL, FZ, 0.1764
FK, 7, FY, 20
FK, 8, FY, 20
FK, 10, FY, 20
FK, 12, FY, 20
FK, 56, FY, 20
FK, 58, FY, 20
FK, 52, FY, 20
FK, 6, FY, 20
FK, 14, FY, 20
FK, 53, FY, 20

F, ALL, FZ, -0.074

/solu
solve

SAVE
FINISH
```

Appendix D: Deflected and Undeflected Shapes for Models

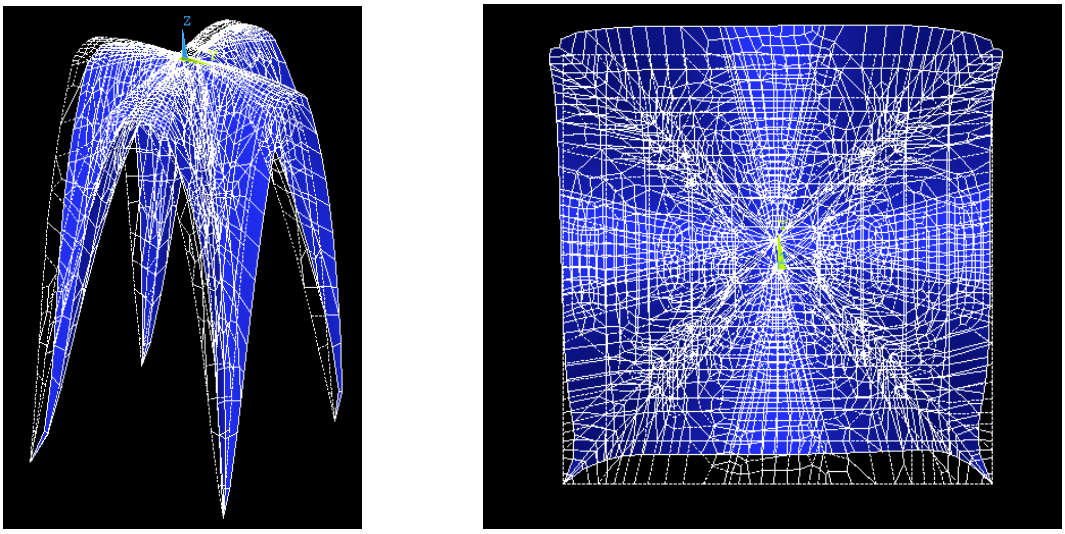

Figure 24: Deflected and undeflected shapes for ideal quadripartite model; white lines represent undeflected shape.

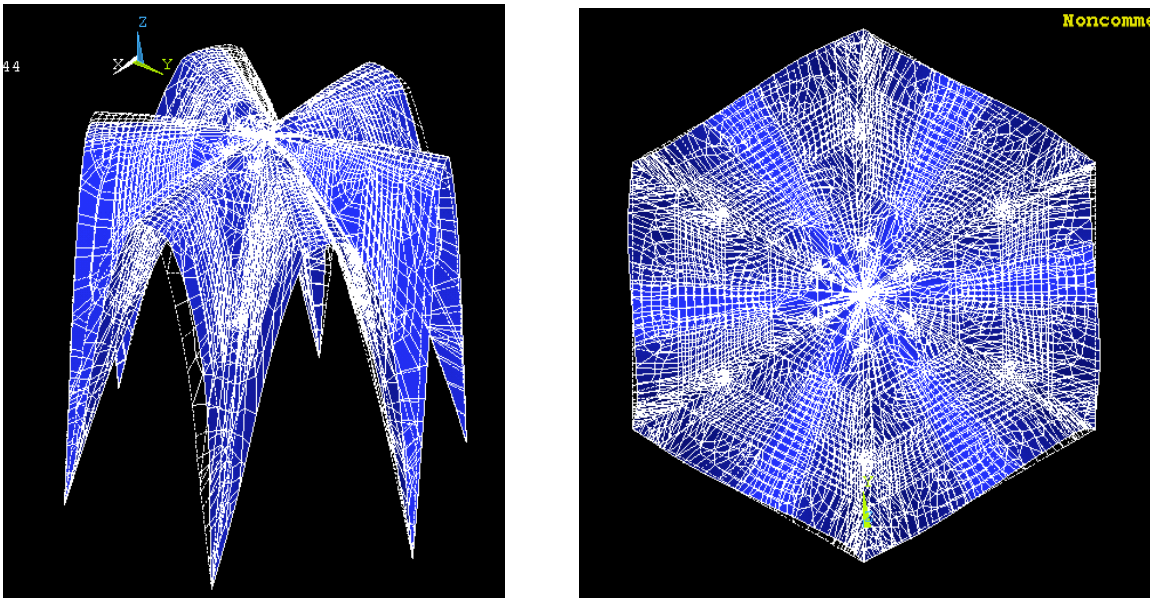

Figure 25: Deflected and undeflected shapes for ideal sexpartite model; white lines represent undeflected shape

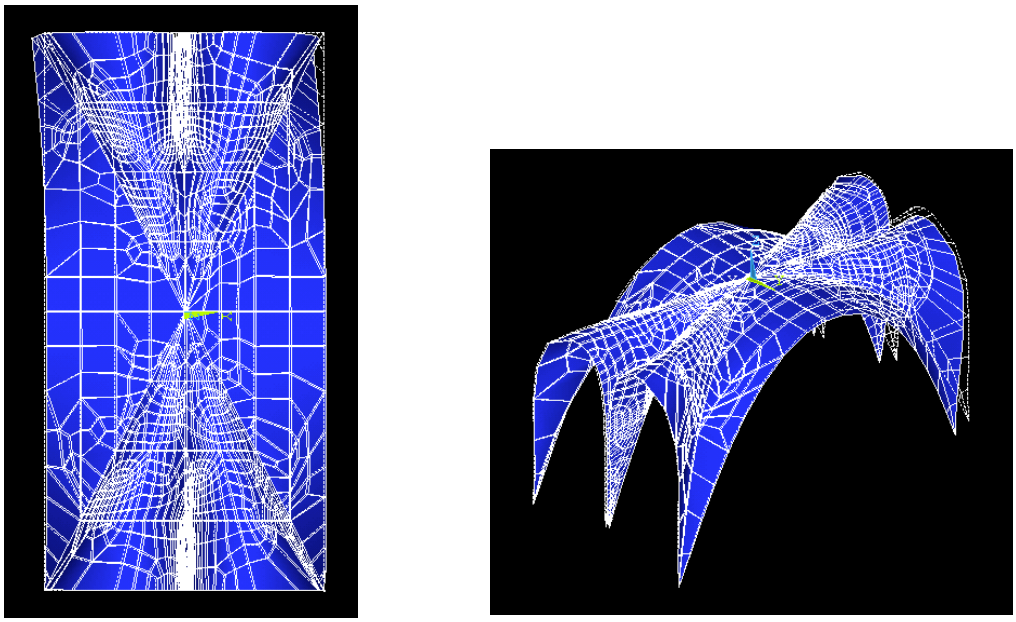

Figure 26: Deflected and undeflected shapes for Beauvais Cathedral vault model. White lines represent undeflected shape