Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

ENGINEERING
DESIGN
AND
COMPUTING

Luca Bettermann

Modeling and RBE Computation of Double-Helix Masonry Dome Structures

Semester Thesis

Engineering Design and Computing Laboratory
Swiss Federal Institute of Technology (ETH) Zurich

Supervision

Yu Zhang
Prof. Dr. Kristina Shea

September 2021

Abstract

In the 15th century, sculptor and architect Filippo Brunelleschi developed an ingenious system that was employed for the construction of the worlds largest masonry dome, the Santa Maria del Fiore in Florence. The secret of its stability lies in the arrangement of its bricks, also called the double-helix of masonry. In this thesis, a parametric model of a double helix masonry dome is developed and the rigid body equilibrium computed, resulting in its distribution of internal normal and tangential forces. A measure of infeasibility is introduced, enabling the analysis of infeasible structures by quantifying and visualising tensile forces within the structure. Quadratic programming is used in order to leverage and solve for the internal forces. The resulting force distribution is computed and analyzed for three separate double-helix masonry dome structures with varying geometries. Furthermore, for each of these structures a CAD model and a physical prototype is presented.

Preface

I would like to thank Dr. Tino Stankovic for being my initial point of contact and Prof. Dr. Kristina Shea for the opportunity to carry out this thesis at the EDAC (Engineering Design and Computing) laboratory. Most importantly, I would like to thank my supervisor and Ph.D student Yu Zhang for her patience and time investment, for the many insightful conversations and her exceptional guidance throughout this thesis.

Contents

Chapter 1

Introduction

Brunelleschi's ingenious double-helix pattern, which was immortalized in the Santa Maria del Fiore dome in Florence, is the main subject of this thesis and an analysis of its influence on the statics of masonry domes is conducted. Paris et. al [6] have shown that one of the main characteristics of the double-helix pattern, is that it allows to perform the construction of a dome without the use of scaffolds or flying buttresses, as it is self-supporting at each step during construction. In this thesis however, the focus is on the analysis of completed double-helix masonry dome structures. The main goal is to develop a parametric model coded in python, enabling a detailed analysis of the internal forces and statical properties of the computed dome structure. An extensive understanding of the pattern's influence on the transmission of forces is built up throughout the thesis.

1.1 Double-Helix Masonry Dome Analysis

In order to conduct this analysis, a structural model incorporating the internal forces acting on each brick interface is needed. This objective is split into two separate parts - generating a dome model and conducting a rigid body equilibrium. The rigid body equilibrium's objective is to minimize tension forces, as masonry bricks typically are assumed to have zero tensile strength [7]. Introducing a measure of infeasibility to the problem formulation by relaxing the compression constraint allows for the analysis of infeasible structures, i.e. structures that are subject to tensile forces.

1.1.1 Dome Model

The first step of modeling a Brunelleschi dome is to develop a thorough mathematical understanding of the double-helix pattern and how it adapts to parametric geometrical dome properties. With this understanding, a methodology of how the double-helix is incorporated in a dome structure is developed. Eventually, the code chronologically builds up the model by completing the following steps:

- calculation of dome dimensions
- computation of double-helix pattern
- implementation of masonry bricks in structure
- computation of brick interfaces

After completion of the process detailed above, a double-helix dome model is constructed. In order to decrease computational costs and to simplify the task, the dome's symmetries are exploited. One section of the dome is modeled and its symmetry is used to define the boundary constraints of this section, which simulates a complete dome.

1.1.2 Rigid Body Equilibrium

The rigid body equilibrium, hereby referred to as RBE, computes the forces that are acting on each interface between the bricks that are in contact. It determines the feasibility or infeasibility of the given structure, which depends on the computed equality and inequality constraints. The equality constraint matrix relates the interface forces and ensures every brick to be in equilibrium, if feasible. External forces are introduced here as well, which are usually limited to the bricks self weight. However, as we are modeling only one section of the dome, we do have additional external forces stemming from the symmetric boundary constraints. The inequality or friction constraints enforce the no-slip condition on each brick interface for feasibility. Depending on how the friction constraints are formulated, the relaxed compression constraints allow for structurally infeasible structures to be deemed feasible by the RBE. As mentioned earlier, this allows us to introduce a measure of infeasiblity and analyse infeasible structures as well.

RBE Optimization

For a given feasible structure, there are a large number of solutions that all satisfy the linear system of the RBE. In order to obtain the most realistic solution in terms of mechanical properties of masonry domes, a quadratic optimization method is used to optimize the force equilibrium. The objective function is formulated in a manner to highly penalize tension forces and to encourage symmetric behaviour.

1.2 Practical Applications

In addition to the analytical part of the thesis, the developed code can be used as a tool to create a custom double-helix masonry dome model by adjusting the input parameters and modes. The output to the code consists of a wide range of structural and force distribution plots, information on the dome's feasibilty or infeasability and a quantifiable measure of its extent in case of the latter. Additionally, csv files containing the dome's internal force values and the coordinates of all vertices for each brick of the structure are created. The coordinates csv file can be used to create a CAD model and subsequently a 3D printed prototype of the dome. This output package allows the user to gain a thorough understanding of the customized dome structure by highlighting its structural weak points and its overall force distribution. With the gained knowledge, the structure can be further tweaked by adjusting its parameters and once the dome fulfills the users needs, the output can be used to produce a physical prototype.

Chapter 2

Background

Masonry domes have been a fascination to humankind for centuries, reaching its pinnacle in the 15th century in the form of the Santa Maria del Fiore in Florence. To this day, it is the worlds largest standing masonry dome and is considered to be an engineering marvel [5]. The historical relevance and the to this day unmatched efficiency of Brunelleschi's masonry structure [6] invited researchers to dig deeper on this topic. The subject of static analysis of general masonry structures is a well researched field with a large number of practical applications.

2.1 Related Studies and Adopted Methods

Paris et. al [6] demonstrate how Brunelleschi's double helix pattern permits an equilibrated state throughout all stages of the domes construction. A Limit State Analysis approach was used to study local and global equilibrium states of the dome. The results of this analysis were validated by using discrete element modeling. Additionally, the discrete element modeling shows the existence of plate-band resistance within the pattern, which is responsible for interlocking masonry bricks and preventing sliding and overturning of the masonry dome during all construction stages.
The implementation process of the double helix pattern developed in this thesis is based on the presented graphics in the paper of Paris et al. In this thesis however, a rigid body equilibrium approach is used in order to determine stability, in contrast to the Limit State Analysis used by Paris et. al, which is performed by the graphical editor algorithm Grasshopper for Rhino software.

Whiting et. al [7] introduce structural feasibility into the procedural modeling of buildings. It allows for the analysis of infeasible structures and for more realistic structural models that can be interacted with in physical simulations.
The concept of the measure of infeasiblity has been adopted in this semester thesis. Additionaly, the conventions used for the RBE are based on the work of Whiting et. al.

Whiting et. al [8] present an approach where structural analysis is part of the design process. The paper is focused on the study of how variations of geometry might improve structural stability. A new measure of structural soundness for masonry buildings and cables is presented. The structures closed-form derivative with respect to the displacement of all vertices describing its geometry is derived, which is used for structural optimization.
The penalty formulation deployed by Whiting et. al for the RBE optimisation has inspired the one used in this thesis.

Kao et. al [3] discuss and extend some main features of the RBE method. The contact between blocks is considered to have finite friction capacity which is modeled through a penalty formulation. The penalty formulation widens the standard admissible solution space of compressive-only forces by allowing for tensile forces appearing on potentially unstable regions. A linear and a quadratic objective function is proposed to illustrate the role being played by both the nodal forces and the interface resultants.
The two different methods of enforcing the friction constraints that are introduced by Kao et. al, the friciton-net approach and the friction+ approach, have been adopted in this thesis.

2.2 Research Gaps

Although masonry structures are widely covered in the literature and the Brunelleschi dome is its most prominent example, a mathematical description of the double-helix pattern is missing. The modeling process delivering the desired result according to the geometrical structure of the dome was derived based on graphics and images of said pattern, which required an unexpected and substantial amount of effort. Secondly, a significant part of the literature is focused on vertically stacked masonry bricks, which do not inherit interfaces on the horizontal brick faces. The papers that do stack bricks not only vertically but also horizontally, used simple examples in the documentation typically in the form of pillars or arches. As a result, the literature did not provide clarity on the handling of interfaces on horizontal brick faces.

Chapter 3

Methods

3.1 Overview

A dome structure must provide strength, stiffness and stability [2], in order to iternalise its self weight and naturally occuring external loads. This thesis is only considering the self weight of each brick as the lone external loading that is acting on the dome. Three main assumptions are made for the analysis conducted in this thesis [2]:

1. sliding of the bricks can not occur; and

2. the masonry has no tensile strength; and

3. the masonry has infinite compressive strength.

Similar to an arch, a dome develops internal meridional forces that transfer loads to a support structure at its base [4]. Additionally, domes are able to develop internal hoop forces that act in latitudinal direction as parallel rings, as shown in figure 3.1.

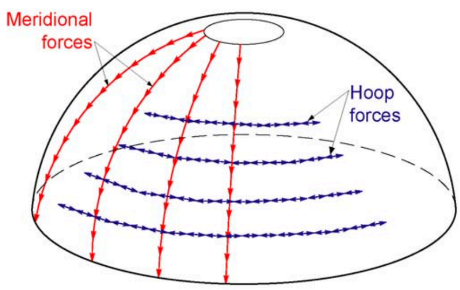

Figure 3.1: Domes develop internal meridional and hoop forces [4]

The main objective of this thesis is to analyse how these forces are transferred throughout the dome if we integrate the Brunelleschi pattern into its structure. The Brunelleschi or the *double-helix* pattern, in its essence, is a simple modification to the stacking of the bricks. Along the pattern, which is wrapped around the dome structure, the bricks are flipped vertically, as shown in figure 3.2. Depending on the source, the pattern is also called *double-loxodrome*, but we will stick with the term *double-helix* as a description for the Brunelleschi pattern in this thesis. In addition to the study of the double-helix pattern, modifications to the domes geometry, e.g. the slope of its curvature, and its influence on the internal force distribution of the structure are being analysed.

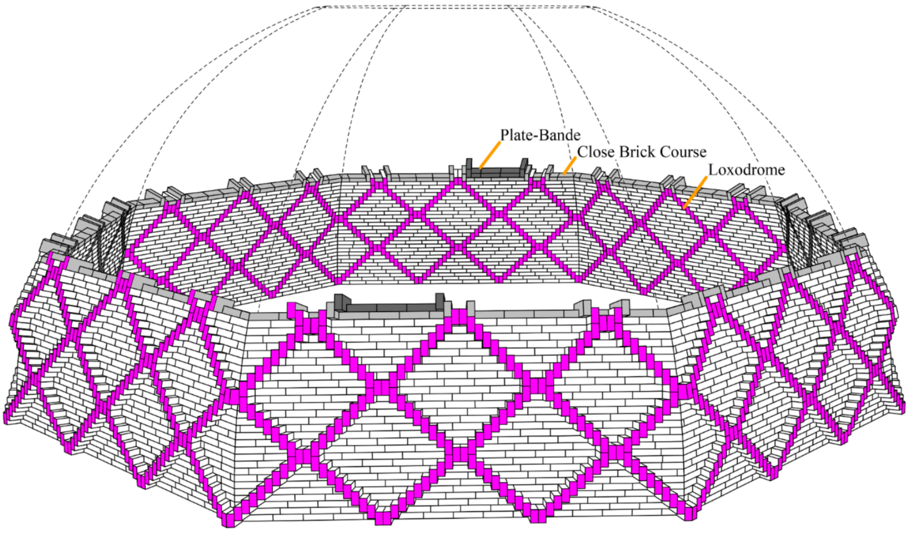

Figure 3.2: Double-helix pattern on an octagonal dome structure [6]

3.1.1 Definitions and Conventions

The double helix masonry dome consists of three different classes of bricks: *regular bricks*, *pattern bricks* and *fill-in bricks*. While the *regular brick* and the *pattern brick* can be seen as the same brick arranged in a different manner, the length of the *fill-in brick* is dynamic.

Table 3.1: Dimensions of brick classes.

Brick Class	Width	Length	Height
Regular	2	4	1
Pattern	4	1	2
Fill-in	2	fill-in	1

The dimensions defined in table 3.1 get multiplied by a unit, which can be customised by the user. For this thesis, in order to have realistic dimensions in terms of *meter*, it is set to

$$unit = 0.1. \tag{3.1}$$

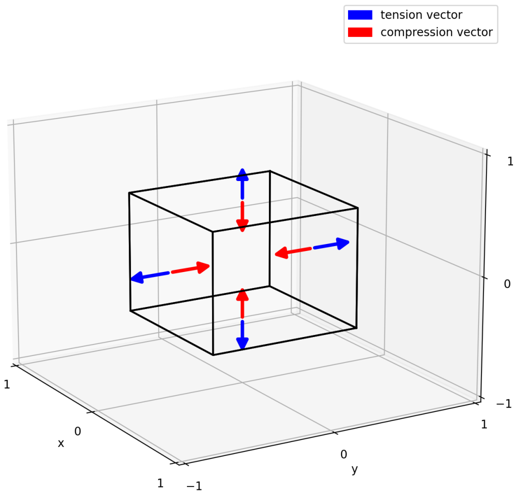

Figure 3.3: Normal forces.

The bricks are subject to normal and tangential forces. Those forces act on the side, upper and bottom faces of the bricks, as those are the only contact faces for a dome structure. The normal forces consist of *compression* and *tension* and are defined as shown in figure 3.3. The tangential forces, on the other hand, are split in *t1* and *t2* forces and are defined as shown in figure 3.4.

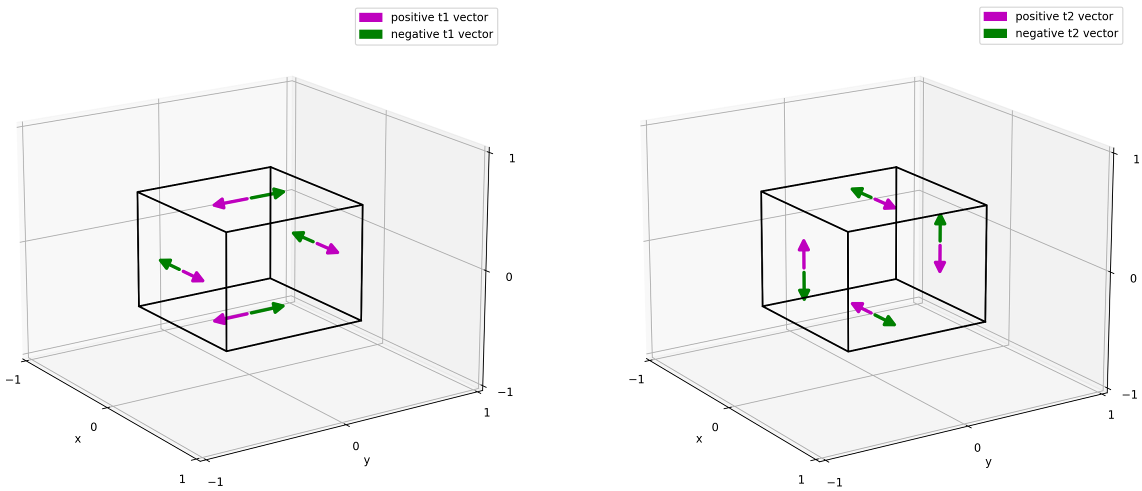

Figure 3.4: Tangential forces.

3.1.2 Input

Modes

The code can be run in different modes that define the kind of structure that is to be modeled and the type of output that is produced. These modes are summarized in table 3.2.

Table 3.2: Modes for model computation and output definition

Mode	True	False
dome	full dome model	section of dome model
fast	reduced computation	full output

Parameters

In order to define the overall geometrical properties of the dome, a set of parameters needs to be set by the user. The meaning and notation of those parameters is defined in table 3.3.

Table 3.3: Parameters that define geometrical properties of dome

Parameter	Definition
stretch	height/radius ratio of dome
n	number of dome walls (e.g. n=6 -> hexagon)
r	number of full rhombi in section (as shown in figure 3.5)
p	number of pattern bricks for each side of initial rhombi (as shown in figure 3.5)
cut	relative height cut-off of dome (e.g. cut=0 -> closed dome)

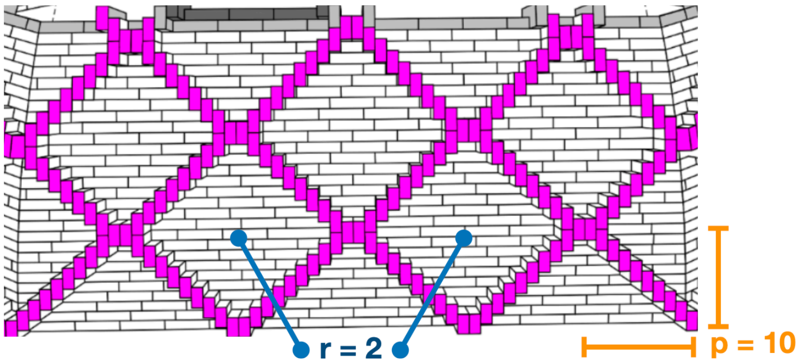

Figure 3.5: Visual description of the rhombi (r) and pattern width (p) parameters.

3.2 Dome Modeling

3.2.1 Dome Dimensions

After the input modes and parameters have been set, the dimensions of the dome can be calculated by exploiting the geometrical properties of regular polygons. As a first step, we calculate the bottom *width* of one dome section, which is equivalent to the sidelength of the regular polygon,

$$width = (2(p+1))(r+1)unit, \tag{3.2}$$

where p and r are described by table 3.3. The *circumradius* of the polygonial dome, which we refer to simply as *radius* from here on out, is calculated by again using the geometrical properties of regular polygons,

$$radius = \frac{width}{2sin(\pi/2)}. \tag{3.3}$$

After calculating for these dome properties, the focus now shifts towards modeling the boundaries for the *main section* of the dome. Three parameters a, b and c are introduced and are calculated as follows,

$$a = \frac{width}{2} cotan(\frac{\pi}{n}), \tag{3.4}$$

$$b = \frac{width}{2}, \tag{3.5}$$

$$c = radius \cdot stretch. \tag{3.6}$$

These parameters are used to define the bounding box for the dome section as shown in figure 3.7. Up until now, we assumed to have a closed dome and ignored the *cut* parameter. The actual height of the dome is calculated as shown below,

$$height = c \cdot cut. \tag{3.7}$$

When constructing the dome section, the aim is to have a continuous curved wall on the inside and a non-continous surface on the outside. The non-continuity stems from small gaps between the bricks on consecutive levels, which are present due to the dome's curvature and the fixed shape of the bricks, as displayed in figure 3.6. While the bricks are touching on the inside wall, they diverge vertically on the outside, as the curvature is slightly different for each level. For now, these gaps can be ignored. However, to make sure that all bricks will be stacked correctly, geometrical values are computed and saved in a dataframe, for each level of the section. Those values are:

- $index = l$
- $depth_l$ or x_l, where $x_0 = a$
- $width_l$ or y_l, where $y_0 = b$
- $height_l$ or z_l, where $z_0 = 0$
- $inclination_{reg_l}$ or γ_{reg_l}
- $inclination_{pat_l}$ or γ_{pat_l}

It is worth noting that the pattern bricks have a different inclination angle than regular bricks, as they have different height dimensions. These differences are very subtle, but nevertheless important to consider, in order to construct a smooth dome structure.

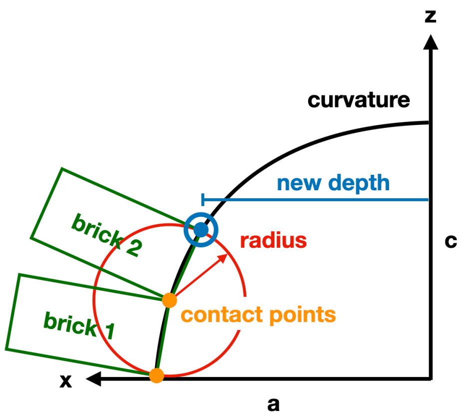

Figure 3.6: Sketch on how the new depth is calculated.

In order to calculate for the *depth* on each level, the last contact point between the bricks is used as a starting point. From there, we use the height dimension of the regular brick, or $dim[z]_{reg}$, as a radius to produce a circle in the xy-plane, as shown in figure 3.6. The circle function is set equal to the function of the curvature,

$$f_{circle}(x_l) = z_{l-1} + \sqrt{dim[z]^2_{reg} - (x_l - x_{l-1})^2} = z_l, \tag{3.8}$$

$$f_{curvature}(x_l) = b\sqrt{1 - \frac{x_l^2}{a^2}} = z_l, \tag{3.9}$$

$$f_{circle}(x_l) = f_{curvature}(x_l). \tag{3.10}$$

With some rearrangements, equation (3.10) results in a quartic function. It is easy to see that there are two intersections, and therefore two real solutions to this equality equation, as shown in figure 3.6. After solving for the two real solutions, the larger x value is eliminated, as that solution corresponds to the lower contact point in relation to the z-axis. The remaining x value is the new $depth_l$. The other values with index l are computed as follows,

$$y_l = x_l \tan(\frac{\pi}{n}), \tag{3.11}$$

$$z_l = c\sqrt{1 - \frac{y_l^2}{b^2}}, \tag{3.12}$$

$$\gamma_{reg_l} = \arctan(\frac{x_l - x_{l-1}}{z_l - z_{l-1}}). \tag{3.13}$$

In order to compute γ_{pat_l}, the radius in figure 3.6 needs to be changed to $dim[z]_{pat}$, the height dimension of the pattern bricks, and the equations 3.12 and 3.13 must be solved again. Once these values have been computed and are saved to a dataframe, the boundaries of the dome section are set and can be visualised in a plot, as shown in figure 3.7.

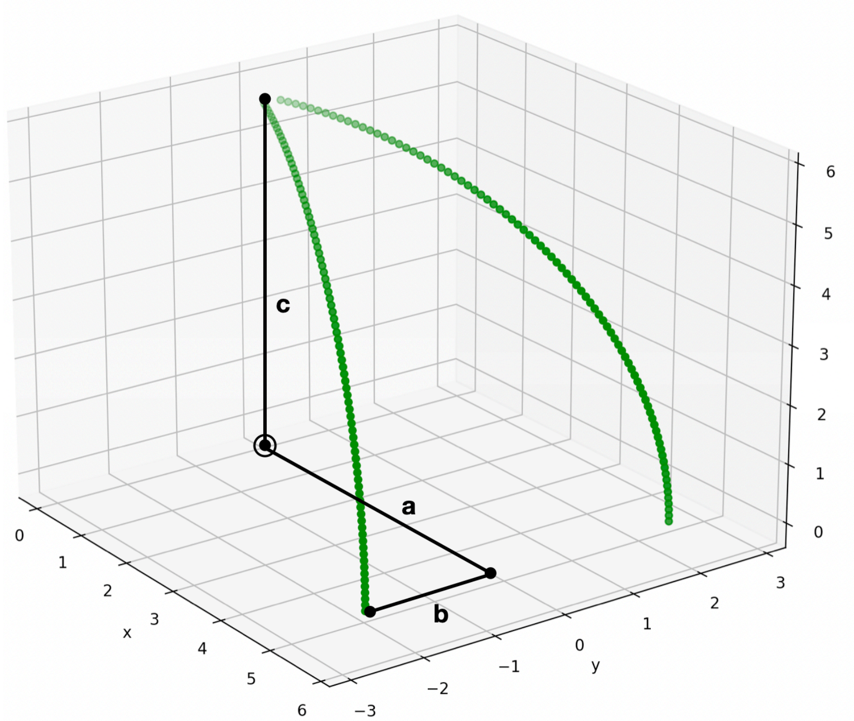

Figure 3.7: Boundaries of the dome section.

3.2.2 Double-Helix Pattern

Although the double-helix pattern seems rather simple when observing it on a straight wall, it becomes more complicated on a dome strucure. As can be seen in figure 3.8, the shape and size of rhombi gradually change in relation to the decreasing width of the dome section. When the pattern is further analysed, one can observe that the rhombi edges that are parallel to each other contain the same amount of pattern bricks. If one starts at the bottom of the structure and works oneself up, the number of pattern bricks get pushed through the structure and are adopted by the corresponding rhombi, as illustrated in figure 3.8. Therefore, the number of pattern bricks for most rhombi is defined by default. The only parameters that need to be defined, or in other words, the only rhombi sides that do not have a set number of pattern bricks, are located on the boundary of the structure and are highlighted in figure 3.8.

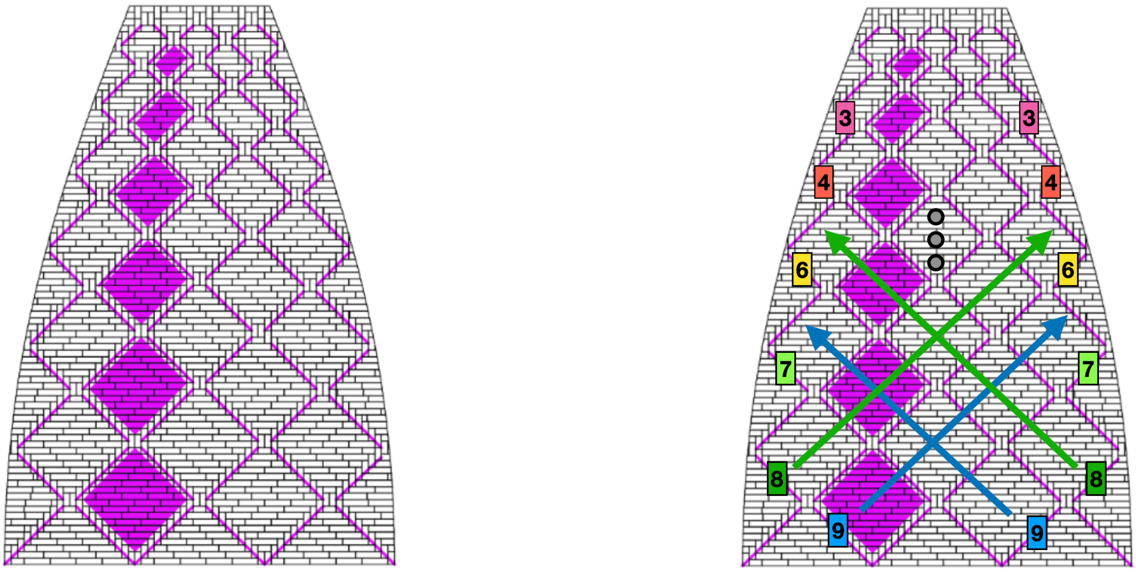

Figure 3.8: Flattened dome section with double-helix pattern [6], undefined rhombi sides highlighted on the right.

Wherever two rhombis are in contact, there are two horizontally stacked pattern bricks next to each other, we call this a *node*. Another observation worth noting, is that the rhombi are always closed at the boundary. In conclusion, the following rules need to be followed in order to construct a double-helix pattern in a polygonial dome structure:

1. parallel rhombi edges contain the same amount of pattern bricks; and
2. all rhombi are closed on the boundary; and
3. rhombi nodes consist of two horizontally stacked pattern bricks.

Patternlines

In order to realise the concept detailed above, a new object is introduced that is called a *patternline*. Patternlines are an abstraction of the double-helix pattern, that allow to split it in separate but related objects and that exploit the symmetry of the double helix pattern. A patternline starts at the bottom of the structure and moves in a zigzag path from node to node. It carries all of the origins for the pattern bricks that will get computed in an ensuing step. For the structure displayed in figure 3.8, there are a total of six patternlines. However, one only needs to compute half of those as the dome section is symmetric. As discussed earlier, the only values that actually need to be computed are the number of pattern bricks at the boundary. The simplified code below details the iterative process to compute the undefined values of the patternlines:

```
def getNumBricks(width, width_list, x_position, current_level):
    # get num of pattern bricks for section of patternline that is undefined

    # pattern bricks count
    num_pattern_bricks = 0

    # iterate until patternline is "out of bounds"
    while x_position < width:

        # move up one level, add one patternbrick and update all values
        current_level += 1
        num_pattern_bricks += 1
        width = width_list[current_level]
        x_position += unit

    # missing value has been found
    return num_pattern_bricks
```

The problem above could have been solved analytically as well, but the discrete nature of it invited an iterative formulation and the computational cost of executing the function is negligible. The results are identical either way. Once the missing parameter is computed, the patternline knows when it needs to change direction and is fully defined. As a last step, all patternlines are mirrored at the mid-axis of the dome section to complete the process, as displayed in figure 3.9.

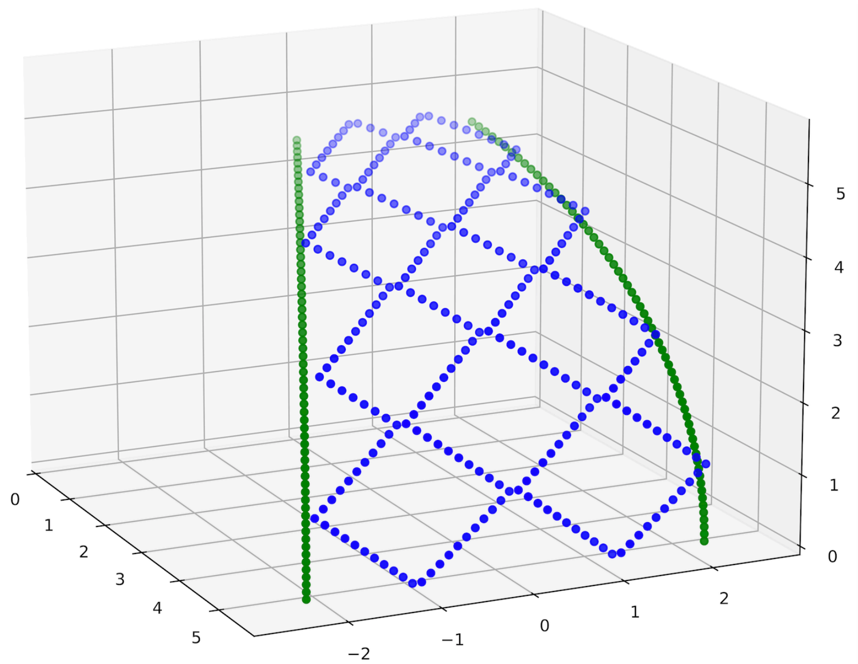

Figure 3.9: Computed patternlines

3.2.3 Masonry Structure

After defining the dome's dimensions and computing the double-helix pattern in the form of patternlines, one can start adding bricks to the structure. The bricks are defined by the following values:

- type
- center
- dimensions
- inclination angle
- list of brick faces

The brick faces are separate objects that are linked to the corresponding brick. They have their own set of values that define them, namely:

- direction in relation to brick
- normal, t1 and t2 basis vectors
- center
- list of vertex coordinates

For each brick that is added to the structure, six faces are computed which in turn each compute the four coordinates of their vertices. This will give the code the necessary tools to compute interfaces and complete the RBE later on.

Pattern Bricks

As the center locations for the pattern bricks are already given by the patternlines, the step of adding pattern bricks to the structure is a rather simple one. The result is displayed in figure 3.10.

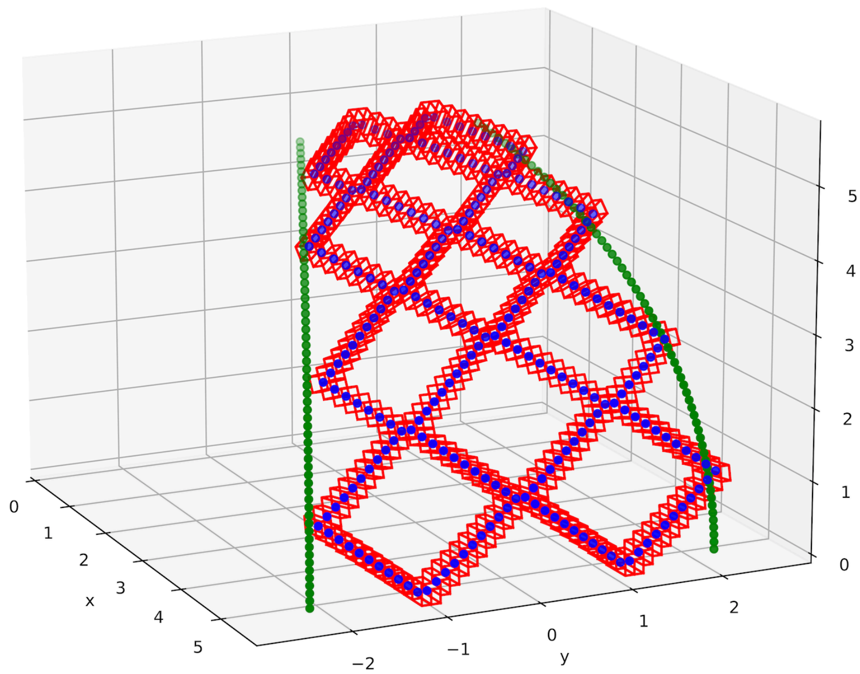

Figure 3.10: Added pattern bricks to the structure

Regular and Fill-in Bricks

In order to complete the structure with regular and fill-in bricks, each level is processed individually. The purpose of the fill-in bricks, as the name suggests, is to fill the open holes that the regular bricks can not fit in. The width of that hole is called the *fill-in* and is the missing variable to complete the dimensions of the fill-in bricks. There is a minimum length constraint to prevent the computation of razorthin masonry bricks. In those cases, a fill-in brick is computed that is slightly larger than the regular brick in order to account for the gap. On each level, we have multiple sequences to fill, which are flanked by the boundaries and divided by the pattern bricks. A heuristic is used to fill the sequence with the correct number of regular bricks with an addition of a fill-in brick if necessary, as detailed below in a simplified manner:

```python
def fillSequence(sequence, dim_regular):

    # initialise brick_list to add all bricks to
    brick_list = []

    # compute number of regular bricks and the excess space
    num_regulars = int(sequence // dim_regular['y'])
    excess_space = sequence % dim_regular['y']

    # fringe case, excess_space to small
    if 0 < excess_space < min_length:
        # compute large fill_in brick to replace a regular brick
        num_regulars -= 1
        fill_in = excess_space + dim_regular['y']

    # regular case
    else:
        fill_in = excess_space

    # check if there is a need to compute a fill-in brick
    if fill_in != 0:
        # compute fill-in brick object and add to brick_list
        reg_brick = FillInBrick(fill_in, ...)
        brick_list.append(reg_brick)

    # iterate over the number of num_regulars we can fit in sequence
    for i in range(num_regulars):
        # compute regular brick object and add to brick_list
        reg_brick = RegularBrick(...)
        brick_list.append(reg_brick)

    # return brick_list that fills sequence
    return brick_list
```

The whole procedure is complicated by the fact that the structure should not stack the bricks identically on consecutive levels, there should always be some kind of offset. This can be reached by adjusting the brick sequence in those cases, e.g. by flipping the fill-in brick to the end of the sequence or by splitting a regular brick into two fill-in bricks. Once all sequences on all levels have been filled, the boundaries and patternlines are removed and the dome section is complete, as displayed in figure 3.11. Additionally, a ground brick is computed that can be regarded as a single fill-in brick where the fill-in value is the total width of the dome section. It represents the solid ground and allows the bricks on the first level to compute interfaces facing the floor, as there are always two brick necessary to create an interface.

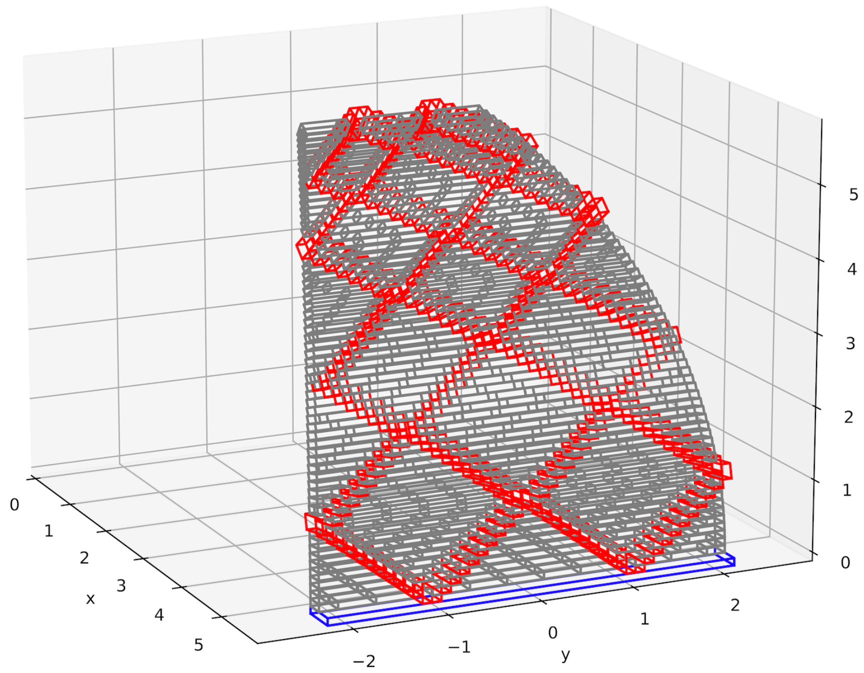

Figure 3.11: Completed dome section

3.2.4 Brick Interfaces

The interface is the medium through which a force is transmitted from one brick to the next and is a necessity in order to compute the RBE later on. If there is no interface on a bricks face, then that face is not subject to any forces. Eventually, the only objects from the dome model that are taken into account in order to compute the force distribution, are the brick interfaces.

Horizontal Interfaces

Up until now, we have ignored the small gaps between the bricks on consecutive levels. In reality, those gaps would prevent the bricks to form any horizontal interfaces, as there is no contact surface area. Typically, those gaps are filled with mortar in order to ensure a full contact area between separate bricks. In this thesis, that problem is simplified by not only ignoring the gaps, but by actively erasing their existence. The procedure to compute the horizontal interfaces goes as follows:

1. For each brick, the level above it is scanned for bricks that overlap vertically with the brick in question.

2. For each pair of bricks that are overlapping vertically, a distinction is made; the lower brick is defined as the *main brick*, the higher brick as the *complementary brick*.

3. The two contact faces, which are not actually in contact because of the gap between them, are singled out.

4. A distinction is made between the two faces; the contact face of the main brick is defined as the *main face*, the contact face of the complementary brick as the *complementary face*.

5. The complementary face is rotated onto the plane defined by the main face, where the rotation axis is defined by the contact edge of the two faces in question.

6. The two faces are now in the same plane and the intersection surface between them is computed.

7. The horizontal interface of the two bricks in question is defined as the computed intersection area, which is in the plane of the main face.

This simplification of the gap problem can also be interpreted in this way: the complementary brick changes its shape in order to be perfectly in contact with the main brick, without *actually* changing its shape. Although one needs to be aware of this abstraction of reality, it is not out of proportion to apply this simplification on the dome model, due to the almost negligible small volume of those gaps.

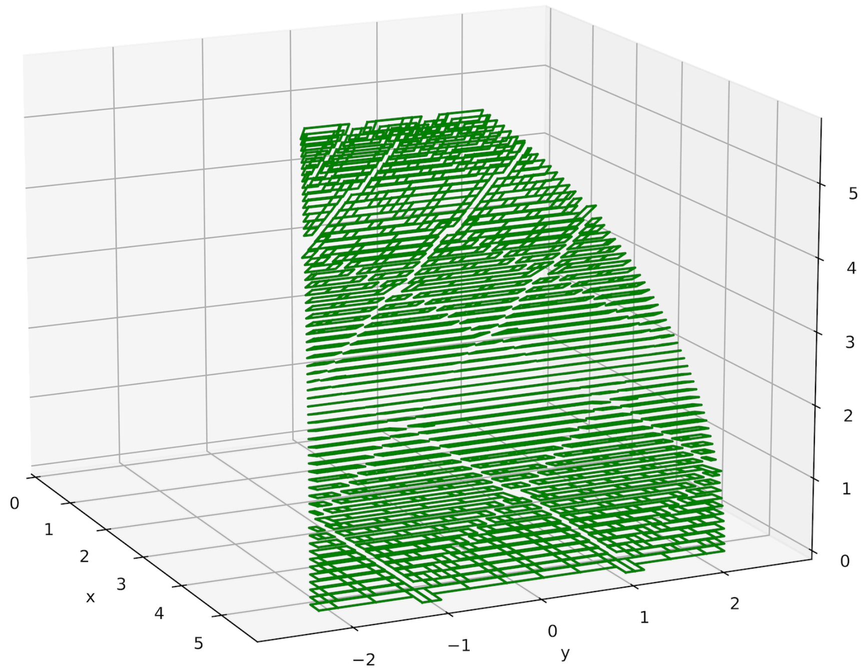

Figure 3.12: Horizontal interfaces, computed by artifically closing the vertical gaps.

Vertical Interfaces

As for the vertical interfaces, there is no gap problem due to the regular polygonial structure of the dome model. However, there is a distinction on the interfaces computation made depending on the dome mode parameter, which has been defined in table 3.2 and is elaborated on in table 3.4.

Table 3.4: Dome mode and boundary constraints

Dome mode	Vertical interfaces	Hoop forces
False	Only compute vertical interfaces between bricks	No
True	Additionally compute vertical interfaces at the boundaries	Yes

If the dome mode is set to *False*, the structure is a free standing dome section. Therefore, there are no hoop forces present and no need to have interfaces on the boundary. On the other hand, if the dome mode is set to *True*, hoop forces will be present and in order to get transmitted through the structure, the model needs interfaces on the horizontal boundaries. The computation of the vertical interfaces is identical to the procedure for the computation of the horizontal interfaces detailed above, with a couple of exceptions:

1. For each brick, its own level is scanned for bricks that are in contact.

2. As there is no gap present between the horizontally stacked bricks, there is no need to rotate the contact faces.

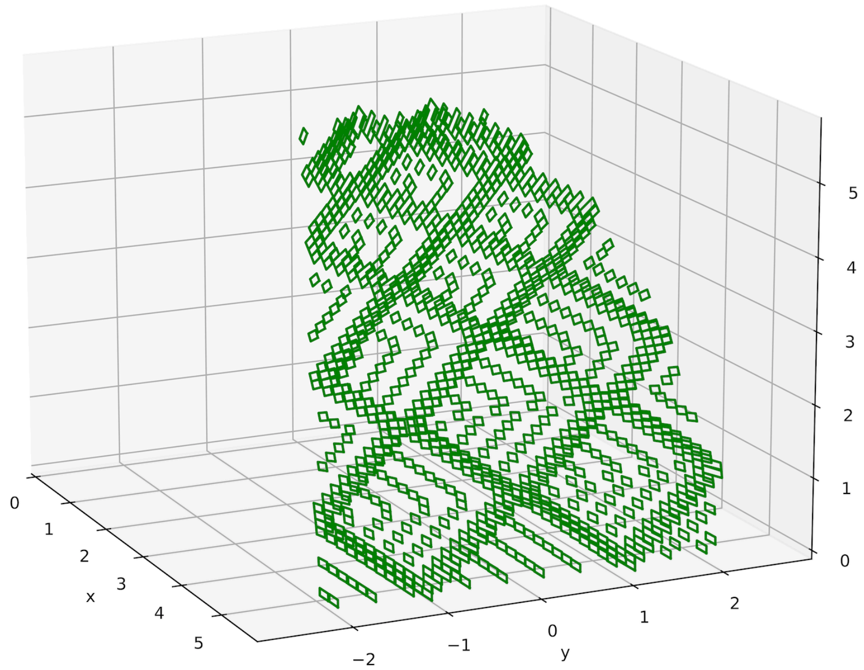

Figure 3.13: Vertical interfaces of a dome section, no interfaces at the boundaries.

3.3 Rigid Body Equilibrium

3.3.1 Equality Constraints

The equality constraints of the RBE are a linear system of equations that enforce the force and moment equilibrium for each brick. If no solution exists in which all bricks are in a equilibrium state, the computation fails and the structure is deemed to be infeasible. The static equilibrium of the enitre masonry brick structure can be generalised and formulated in matrix form:

$$A_{eq}f + f_{ext} = 0, \tag{3.14}$$

where A_{eq} is the matrix of coefficients for the equilibrium equations [7]. It is defined as

$$A_{eq} = \begin{bmatrix} A_{0,0} & A_{0,1} & \dots \\ \vdots & \ddots & \\ A_{n-1,0} & & A_{n-1,n} \end{bmatrix}, \tag{3.15}$$

where $A_{j,k}$ are submatrices that contain coefficients for net force and net torque contribution from *interface* k acting on *block* j. $A_{j,k}$ itself is defined as:

$$A_{j,k} = \begin{bmatrix} a_{k_x} & a_{k_x} & \dots \\ a_{k_y} & a_{k_y} & \dots \\ a_{k_z} & a_{k_z} & \dots \\ b_{i,j,k_x} & b_{i+1,j,k_x} & \dots \\ b_{i,j,k_y} & b_{i+1,j,k_y} & \dots \\ b_{i,j,k_z} & b_{i+1,j,k_z} & \dots \end{bmatrix}, \tag{3.16}$$

where

$$a_{k_x} = \begin{pmatrix} \hat{n}_{k_x} \\ \hat{t1}_{k_x} \\ \hat{t2}_{k_x} \end{pmatrix}, b_{i,j,k_x} = \begin{pmatrix} (\hat{n}_k \times \hat{v}_{i,j})_x \\ (\hat{t1}_k \times \hat{v}_{i,j})_x \\ (\hat{t2}_k \times \hat{v}_{i,j})_x \end{pmatrix}. \tag{3.17}$$

$\hat{n}_{k_x}$, $\hat{t1}_{k_x}$ and $\hat{t2}_{k_x}$ are the normal vector and friction basis vectors for interface k, as shown in figure 3.14.

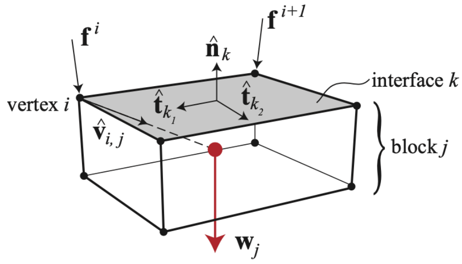

Figure 3.14: Indexing for the RBE [7]

Returning to equation (3.14), the f vector is defined as:

$$f = \begin{pmatrix} f^i \\ f^{i+1} \\ \vdots \end{pmatrix}, \qquad f^i = \begin{pmatrix} f^i_n \\ f^i_{t1} \\ f^i_{t2} \end{pmatrix}. \tag{3.18}$$

It contains all the nodal forces of each interface contained in the masonry dome. Lastly, f_{ext} is a vector containing all the external forces acting on the structure, which consist of the self weight of the bricks and the hoop forces on the boundaries.

Implementation of Hoop Forces

If the dome mode input has been set to *True*, there are boundary conditions present due to the symmetry of the dome structure. The net-normal force and tangential forces acting on a vertical boundary interface are mirrored by the dome section that it is in contact with. From the perspective of the modeled dome section, this can be expressed as an external load, compressing the section from both boundaries. These external forces are called *hoop forces* and were briefly introduced in figure 3.1. Those boundary conditions are simplified by ignoring the mirrored tangential forces and only considering the net-normal force. As the dome has a polygonial shape, there is a certain angle between the two contact sections, which affects the direction of the hoop force acting on the interface. The resulting contribution of the hoop forces in relation to the interface forces is defined as:

$$f_{hoop} = max(\cos\frac{2\pi}{n}, 0) \tag{3.19}$$

Additionally, the sparse matrix A_{hoop} stores all interface forces that contribute to the hoop forces and defines the direction of that force in relation to the brick interface. Finally, the external force vector is defined as

$$f_{external} = f_{weight} + f_{hoop} \cdot A_{hoop}x, \tag{3.20}$$

where A_{hoop} is a zero matrix if the dome mode has been set to *False*.

3.3.2 Measure of Infeasibility

Whiting et. al [7] introduced a penalty formulation that has been adapted in this thesis. The normal force component of the nodal force vector f is split into two separate parts:

$$f_n^i = -f_{n-}^i + f_{n+}^i, \tag{3.21}$$

$$f_{n-}^i, f_{n+}^i \geqslant 0, \tag{3.22}$$

where f_n^{i-} corresponds to *compression* and f_n^{i+} corresponds to *tension*. Equation (3.18) is now adjusted to:

$$f^i = \begin{pmatrix} f_{n-}^i \\ f_{n+}^i \\ f_{t1}^i \\ f_{t2}^i \end{pmatrix} \tag{3.23}$$

Equation (3.17) is adjusted accordingly. The concept of equation (3.23) becomes clearer once we introduce the inequality constraints and set the objective function. In essence, the concept allows tension to be present while being penalized simultaneously, which creates a measure of infeasibility.

3.3.3 Inequality Constraints

In addition to the equality constraints, a friction constraint is applied at all vertices i of each interface j, in order to take into account the sliding phenomena. At each vertice, the two in-plane forces are constrained within the friction cone of the normal force. The resulting inequality constraints can be generalized in matrix form as:

$$A_{fr} f \leqslant 0. \tag{3.24}$$

Since we introduced a penalty information, there are two strategies that can be adopted to model the friction constraint in equation (3.24), as pointed out by Kao et. al [3]. The first approach constrains the tangential forces with compression only and is called *friction+*:

$$\mid f_{t1}^i \mid, \mid f_{t2}^i \mid \leqslant \alpha f_{n-}^i, \tag{3.25}$$

where α denotes the static friction coefficient, typically set to 0.7 for masonry bricks. f_{n+} is not involved in the inequality constraint and therefore is a decoupled parameter. When inspecting equation (3.25) closer, one can conclude the following two statements:

1. It is allowed to increase tension so that $f_{n+} > f_{n-}$ and therefore $f_n > 0$ (see equation (3.22)), i.e. the interface node in question is subject to tension.

2. One can increase f_{n+} and f_{n-} equally in order not to modify the value of f_n, while simultaneously relaxing the constraint detailed in equation (3.25). With this method, $\mid f_{t1}^i \mid$ and $\mid f_{t2}^i \mid$ can be increased to any arbitrary value.

This means, that there are no infeasible solutions for the *friction+* constraints. If we recall one of the three main assumptions made at the beginning of this chapter - the masonry has no tensile strength [2] - it becomes obvious that we have relaxed this assumption with the introduction of the penalty formulation, coupled with the *friction+* constraint. The second approach is called *friction-net* [7] and is defined as follows:

$$\mid f_{t1}^i \mid, \mid f_{t2}^i \mid \leqslant \alpha (f_{n-}^i - f_{n+}^i). \tag{3.26}$$

After further inspection, one can conclude that the statements made above regarding the *friction+* constraints are not applicable to the *friction-net* approach, for the following reasons:

1. According to equation (3.26) a new constraint is introduced: $f_{n+} \leqslant f_{n-}$. Therefore, $f_{n+} > f_{n-}$ violates that constraint and is not feasible.
2. Increasing f_{n+} and f_{n-} equally does not yield any modification in the *friction-net* constraint (see equation (3.26)).

Adding tension to the structure has no relaxing effect on the *friction-net* constraints and, assuming that a tension-only penalty is enforced during the RBE computation, it only increases the objective, which is why tension will not be introduced for feasible structures. Surprisingly, under the assumption made above, the *friction-net* approach reduces the problem back to its original form and negates the measure of infeasibility formulation introduced in equation (3.22). On the other hand, the *friction+* constraint allows for an analysis of infeasible structures, while raising an awareness of the infeasiblity and its magnitude, enabled by the measure of infeasibility. As both methods have their pros and cons, a new mode is introduced that allows to switch between them for different execution cycles of the code.

Table 3.5: Introduction of new friction mode

Mode	True	False
friction	friction-net	friction+

Table 3.5 will get added to the modes detailed in table 3.2. Initially, the code is always run with the friction mode set to *True*. If the structure is deemed to be infeasible, the same structure is re-computed with the friction mode set to *False*. This allows the user to always check feasibility first and if the structure is deemed infeasible, the measure of infeasibility is applied. As detailed in the following section, the friction+ method is not bulletproof when it comes to determining feasibility, while the net-friction approach does not allow bricks to be subject to tension.

3.3.4 RBE as an Optimization Problem

In order to optimise the internal forces subject to the equality and inequality constraints detailed above, the RBE is solved as an energy-minimisation problem, defined as follows:

$$\begin{aligned} \underset{x}{\text{minimize}} \quad & f_{obj}(x) = \frac{1}{2}x^T H x \\ \text{subject to} \quad & A_{eq}x + f_{ext} = 0, \\ & A_{fr}x \leqslant 0, \\ & I_{lb}x \geqslant 0, \end{aligned}$$

where x is identical to f defined in equation (3.18), H is a penalty weighting matrix and I_{lb} an identity matrix to set the lower bounds. A quadratic objective function was favored over a linear formulation, as the results displayed a more balanced and symmetric force distribution. In a quadratic formulation, the solver avoids isolated high forces in relation to the rest of the forces and therefore it reduces outliers. H and I_{lb} from above are defined as:

$$H = \begin{bmatrix} 0.001 & 0 & 0 & 0 & \dots \\ 0 & 1 & 0 & 0 & \\ 0 & 0 & 0.001 & 0 & \\ 0 & 0 & 0 & 0.001 & \\ \vdots & & & & \ddots \end{bmatrix}, \qquad I_{lb} = \begin{bmatrix} 1 & 0 & 0 & 0 & \dots \\ 0 & 1 & 0 & 0 & \\ 0 & 0 & 0 & 0 & \\ 0 & 0 & 0 & 0 & \\ \vdots & & & & \ddots \end{bmatrix}. \tag{3.27}$$

As can be observed in equation (3.27), H is penalising tension overproportionally while the other forces are penalysed by a small margin. After testing different configurations, this setup turned out to deliver the most realistic results, as it focuses on minimizing tension while keeping the overall force magnitudes low. However, there is one significant drawback - by penalising all forces, no matter to which extent, the objective value can not be considered a true measure of infeasibility. Even for feasible structures, the objective function will not be zero, as zero is not attainable. This drawback has been softened by setting the penalty for non-tension forces to 0.001, but it is still present to some degree. As a workaround, the objective function is no longer viewed as the measure of infeasibility. Its only purpose is to optimize the problem formulation. A new function is introduced, f_{MoI}, that replaces the objective function as the measure of infeasibility:

$$f_{MoI}(x) = \Sigma_{j=0}^{n} \Sigma_{i=0}^{4} x_{n+,j}^{i} \tag{3.28}$$

This solution encorporates the best of both worlds, an objective function delivering realistic results, while still being able to measure the infeasibility truthfully. As for I_{lb} in equation (3.27), it enforces the strictly positive constraint introduced in equation (3.22) for the compression and tension forces, while the tangential forces remain unconstrained.

Quadratic Programming

As the optimization problem is formulated as an energy-minimisation problem, quadratic programming is used to find the optimal solution. More precisely, the python-embedded modeling language CVXPY [1] is employed, using the incorporated OSQP solver to conduct the computation.

3.3.5 RBE Visualization

After an optimal solution to the optimization problem has been found, the nodal forces for all interfaces are known. In order to make the result more illustrative and interpretable, the nodal forces are aggregated on each interface and visualised in a customized color spectrum. The output of the code consists of six different force plots, each displaying a certain force type, defined as follows:

$$\begin{aligned}
compression_j &= \Sigma_{i=0}^{4} f_{n-,j}^{i} && (3.29)\\
tension_j &= \Sigma_{i=0}^{4} f_{n+,j}^{i} && (3.30)\\
normal_j &= \Sigma_{i=0}^{4} f_{n+,j}^{i} - f_{n-,j}^{i} && (3.31)\\
t_{1,j} &= \Sigma_{i=0}^{4} f_{t1,j}^{i} && (3.32)\\
t_{2,j} &= \Sigma_{i=0}^{4} f_{t2,j}^{i} && (3.33)\\
friction_j &= \sqrt{\Sigma_{i=0}^{4} {f_{t2,j}^{i}}^{2} + \Sigma_{i=0}^{4} f_{t2,j}^{i}} && (3.34)
\end{aligned}$$

3.3.6 RBE Validation

In order to validate the RBE computation and make sure the results are realistic, two different tests have been performed. The common feature of the tests is that the result expectation is very clearly defined by a simplistic test setting. Therefore, unrealistic results are obvious and the error source is identified with relatively small effort.

Straight Wall Test

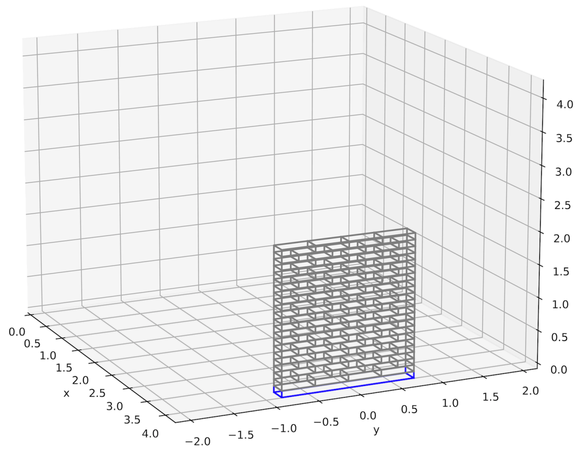

Figure 3.15: Straght wall structure

A third mode in which the code can be run in apart from the binary dome mode parameter defined in table 3.2 has been implemented. The structure for this third mode is a straight wall. As its only purpose is the validation of the code, it has not been introduced as a general input parameter. When bricks are stacked horizontally without any inclination angle or pattern bricks, the results must have the following characteristics in order to pass the straight wall test:

1. The bricks are subject to compression with zero or negligible tension and tangential forces.

2. The compression gradually increases from the highest level to the lowest level.

Friction Test

As the straight wall test focused on a more primitive validation of the RBE, where the tangential forces were not part of the validation other than being zero or negligible in the results. In order to test the mechanics of the friction forces, a dome section with an significant overhang is chosen in this setup which is expected to lead to infeasiblity. Note that the RBE subject to friction+ constraints does not classify the problem as infeasible, but rather as a feasible problem with a structure that is subject to tension and therefore it has a non-zero measure of infeasbility. Primarily, the friction test should reflect this infeasibility in the results, but this is not the actual objective of this test setting. The idea is to compare the measure of infeasibility of identical and infeasible structures that are computed with a varying friction parameter. Tests with the following friction parameters are being conducted:

Table 3.6: Friction parameters to be tested and the expected results.

Friction	Expectation
0.7	Non-zero measure of infeasiblity
0.3	Highest measure of infeasiblity
1.5	Lowest measure of infeasiblity

The following results are expected from the MoI test:

1. The structure with the regular friction parameter of 0.7 is infeasible.

2. A negative correlation between the friction parameter and the MoI is observed.

Combining those two statements, we can also expect the structure that is computed with the friction parameter of 0.3 to be infeasible.

3.4 Output

For some structures, the execution of the dome modeling and RBE computation has taken up to several minutes. Note that the computation time is highly dependent on the computation power of the machine it is run on, therefore this is not a quantitive measure of the overall computational cost when running the code. Nevertheless, it became clear that the results should be saved whenever the code has been executed, in order to minimise unnecessary computation. Therefore, a folder structure has been implenented that stores the output of previous code executions in a methodical manner. The overall folder structure is detailed below:

```
/
└─structure id
   ├─00 parameters.txt
   ├─01 angle
   │  ├─01 pattern bricks.png
   │  ├─02 brick structure.png
   │  └─03 brick interfaces.png
   ├─02 front
   │  └─...
   ├─03 top
   │  └─...
   ├─04 side
   │  └─...
   ├─RBE
   │  ├─00 objective.txt
   │  ├─01 angle
   │  │  ├─00 compression.png
   │  │  ├─01 tension.png
   │  │  ├─02 normal.png
   │  │  ├─03 t1.png
   │  │  ├─04 t2.png
   │  │  └─05 friction.png
   │  ├─02 front
   │  │  └─...
   │  ├─03 top
   │  │  └─...
   │  └─04 side
   │     └─...
   └─locations.csv
```

strucure id is the main folder of the output. Its name is a combination of the input modes and parameters, which ensures that the folder is only overwritten if a structure with the same input is re-computed. *00 parameters.txt* details all the input parameters in a more readable manner. *01 angle*, *02 front*, *03 top* and *04 side* on the main folder level include structural plots of different stages in the dome modeling process. The model has been plotted from different view angles and the plots are saved in the corresponding folders. The *RBE* subfolder contains all files that are related to the optimization problem and its solution vector, as the name is suggesting. *00 objective.txt* summarizes the result from the quadratic programming, including the following metrics:

1. feasibility
2. friction constraint method used
3. value of objective function
4. measure of infeasibility

As earlier, the four folders from different view angles are included within the RBE subfolder, this time including various force plots of the structure. The last file of the output is the *locations.csv* file, which contains the coordinates for all six vertices of each brick. If the fast mode has been set to *True*, which was defined in table 3.2, the output is reduced to only plotting the structure and force plots from one angle with a decreased resolution. Those adaptions reduce the computation time significantly, with its extent also being dependent on the other input parameters.

3.4.1 CAD Models

The dome model encorporates all the necessary information to create a 3D model in a CAD environment. As noted above, the locations.csv file contains all vertex coordinates of each brick of the structure. This file can be used to import the bricks as separate objects into a CAD environment. As the bricks are only touching on one edge and the model does not include any mortar, the CAD environment is not able to create a main body without any additional modifications. Therefore, all coordinates in the locations.csv file are subject to an offset of one *unit*, which was defined in equation (3.1). The direction of the offset is defined by their normal vector which points inward relative to the structure. This allows the CAD model to create a main body, as the bricks are slightly overlapping in certain areas. The results are presented with the CAD Models of the following three structures:

structure	main characteristic	stretch	n	r	p	cut
1	spherical dome	1	8	1	10	0.1
2	stretched dome	2	6	1	8	0.05
3	compressed dome	0.8	4	1	13	0.01

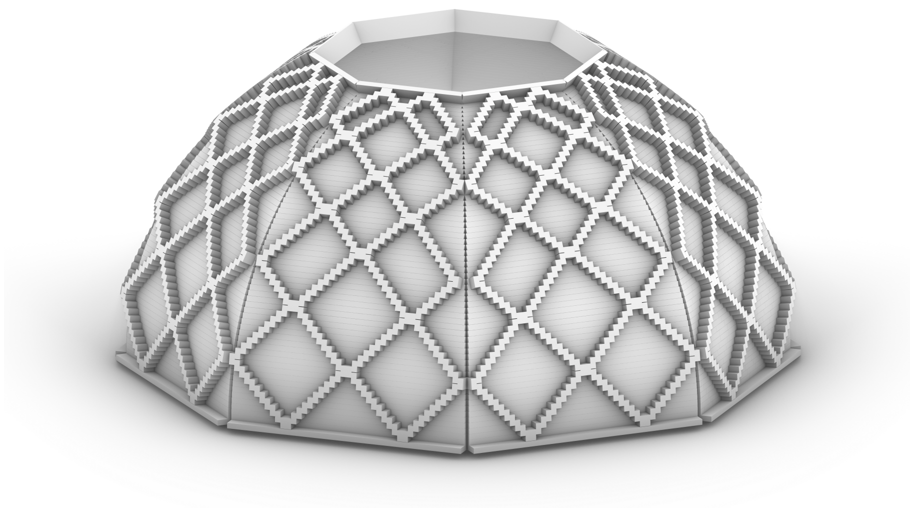

Figure 3.16: Structure 1 in a Rhino environment

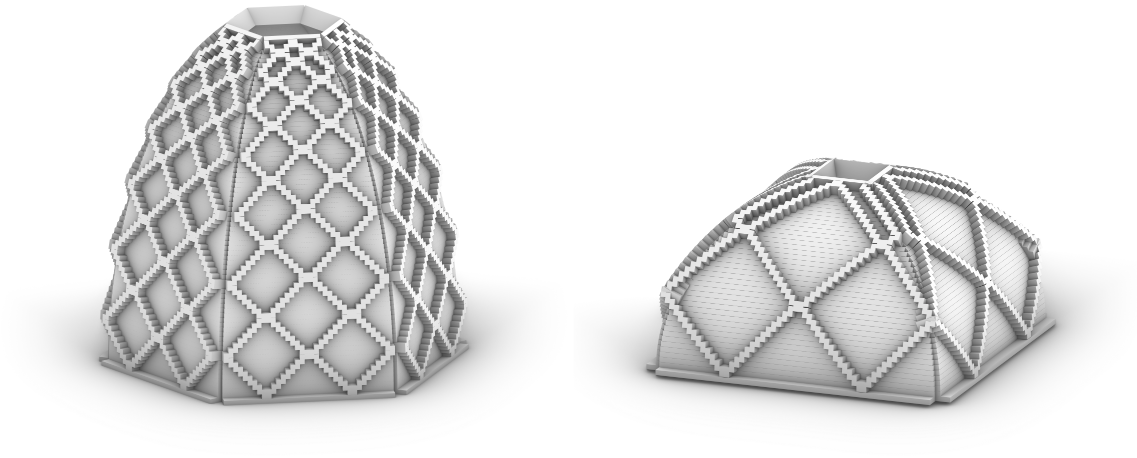

Figure 3.17: Structure 2 (left) and structure 3 in a Rhino environment

3.4.2 Physical Prototypes

In an ensuing step, the dome model can get exported again from the CAD environment in a stl file format. 3D printed, physical prototypes can be created from those files. For the three structures introduced earlier, physical prototypes were created and are presented in figure 3.18.

Figure 3.18: Physical prototypes of the stretched, spherical and compressed dome, from left to right.

Chapter 4

Results

The result of the RBE computation of the three structures introduced in section 3.4.1 are being presented in this chapter, in two separate configurations: as a free standing dome section and as a full dome. For all infeasible structures, the RBE was computed with the friction+ method. The structures in section 3.4.1 have been adjusted as follows:

Table 4.1: Input modes and parameters for the different structures

structure	dome mode	net-friction	stretch	n	r	p	cut
1.1	False	False	1	8	1	10	0.1
2.1	False	False	2	6	1	8	0.05
3.1	False	False	0.8	4	1	13	0.01
1.2	True	True	1	8	1	10	0.1
2.2	True	True	2	6	1	8	0.05
3.2	True	True	0.8	4	1	13	0.01

The results consist of a table that displays some key metrics of the structure and the completed RBE computation, that include: feasibility, objective function value, MoI, weight of structure (in kg) and the MoI factor, which is defined as:

$$MoI_{factor} = \frac{MoI}{weight} \tag{4.1}$$

The MoI factor allows a comparison between structures that do not share the same geometries and number of bricks, as the MoI can be misleading for some configurations. In addition to the metrics listed above, the most relevant force plots are displayed. A full overview of the force plots is documented in the Appendix 1. Note that the following results have all been plotted with individual color spectra, in order to present the force distributions as clear as possible. However, the colors can be misleading if the assigned values next to the color spectra are overlooked. Make sure to consider the numerical values before interpreting the results.

4.1 RBE Validation Results

4.1.1 Straight Wall Test Results

Table 4.2: RBE results of straight wall test

feasible	obj. function value	MoI	weight	MoI factor
Yes	$41.7 \cdot 10^3$	~ 0	$12.5 \cdot 10^3$	0

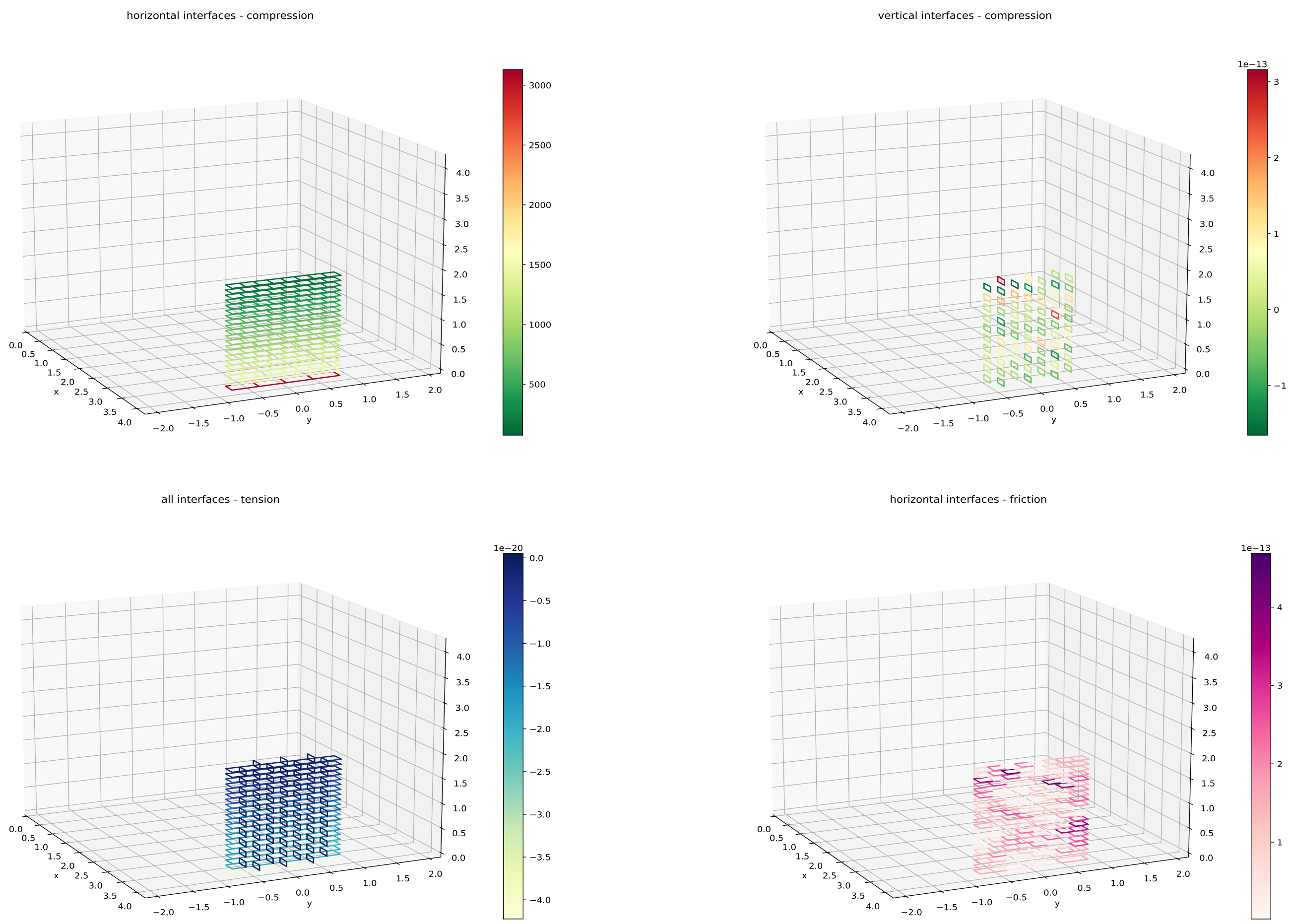

Figure 4.1: Force plots of compression (top), tension (left) and friction (right) for the straight wall test.

4.1.2 Friction Test Results

Table 4.3: Results of friction test conducted with structure 1.1

friction par.	feasible	obj. function value	MoI	weight	MoI factor
0.3	No	$190.8 \cdot 10^6$	$806.6 \cdot 10^3$	$102.6 \cdot 10^3$	7.86
0.7	No	$22.8 \cdot 10^6$	$216.3 \cdot 10^3$	$102.6 \cdot 10^3$	2.11
1.5	No	$3.9 \cdot 10^6$	$7.1 \cdot 10^3$	$102.6 \cdot 10^3$	0.07

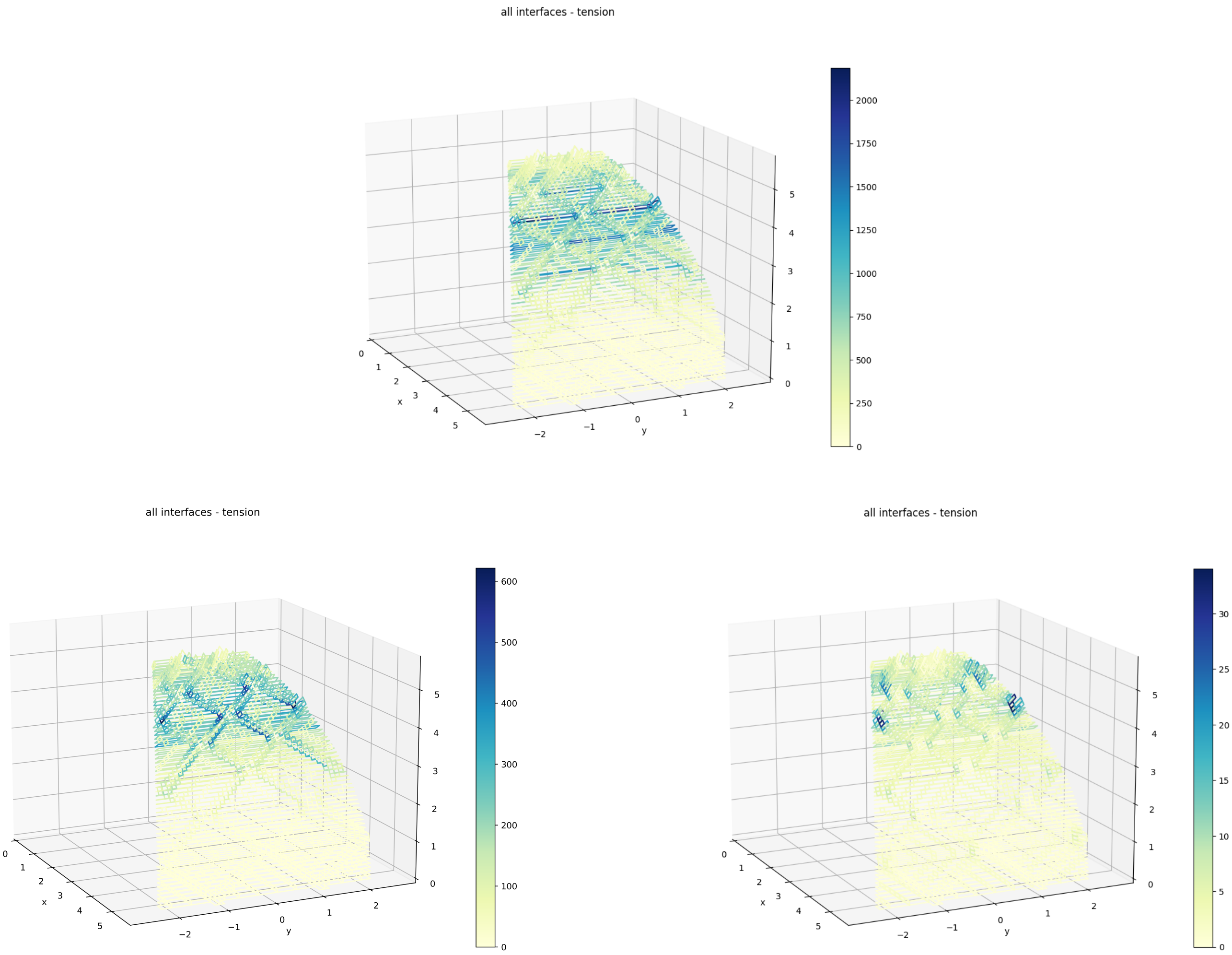

Figure 4.2: Tension plots of structure 1.1 computed with friction parameter 0.3 (top), 0.7 (left) and 1.5 (right).

4.2 RBE Results

4.2.1 Structure 1.1: Spherical Dome Section

Table 4.4: RBE results of structure 1.1

structure	feasible	obj. function value	MoI	weight	MoI factor
1.1	No	$22.8 \cdot 10^6$	$216.3 \cdot 10^3$	$102.6 \cdot 10^3$	2.11

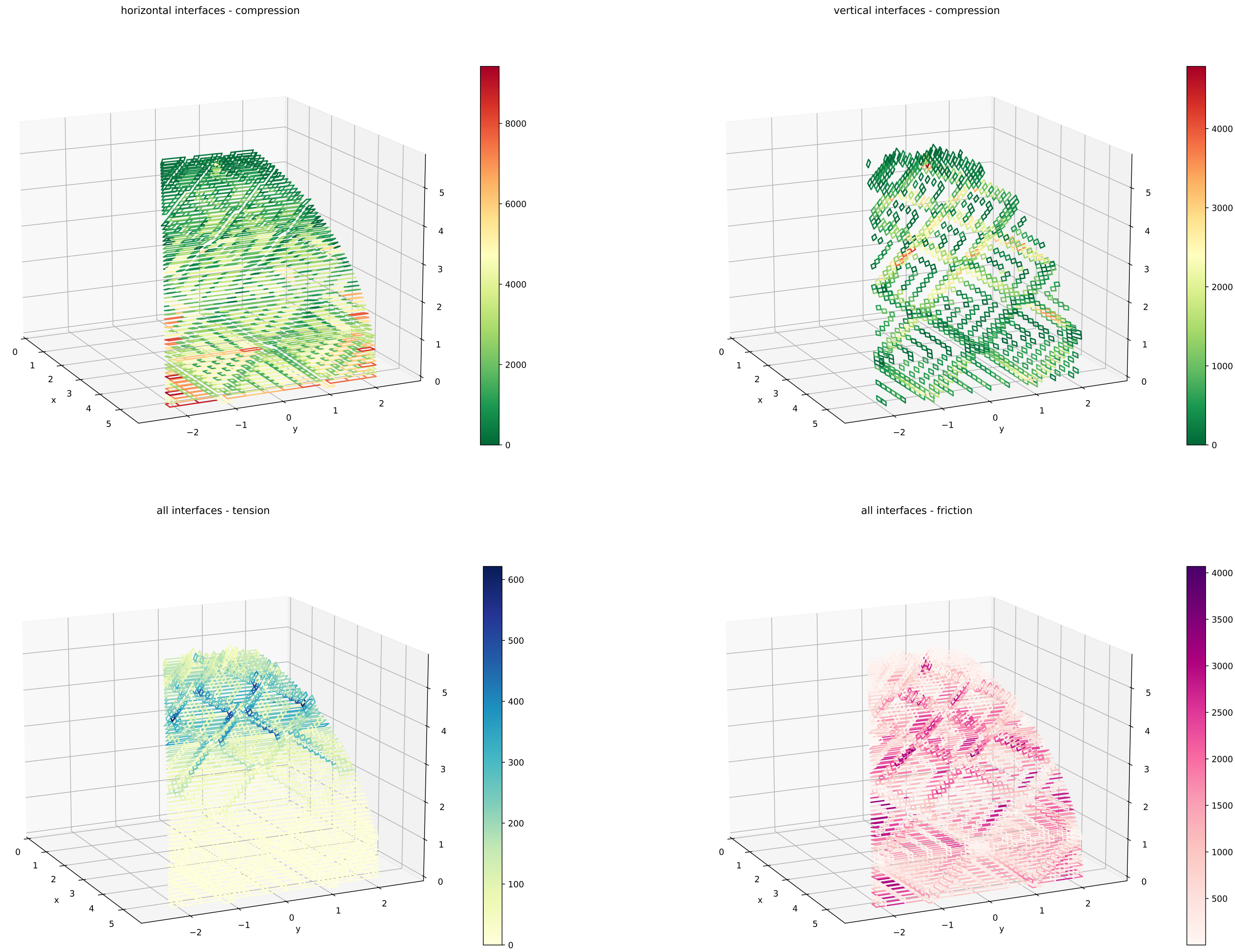

Figure 4.3: Force plots of compression (top), tension (left) and friction (right) for structure 1.1

4.2.2 Structure 2.1: Stretched Dome Section

Table 4.5: RBE results of structure 2.2

structure	feasible	obj. function value	MoI	weight	MoI factor
2.1	Yes	$0.3 \cdot 10^6$	$15.9 \cdot 10^3$	$102.9 \cdot 10^3$	0.155

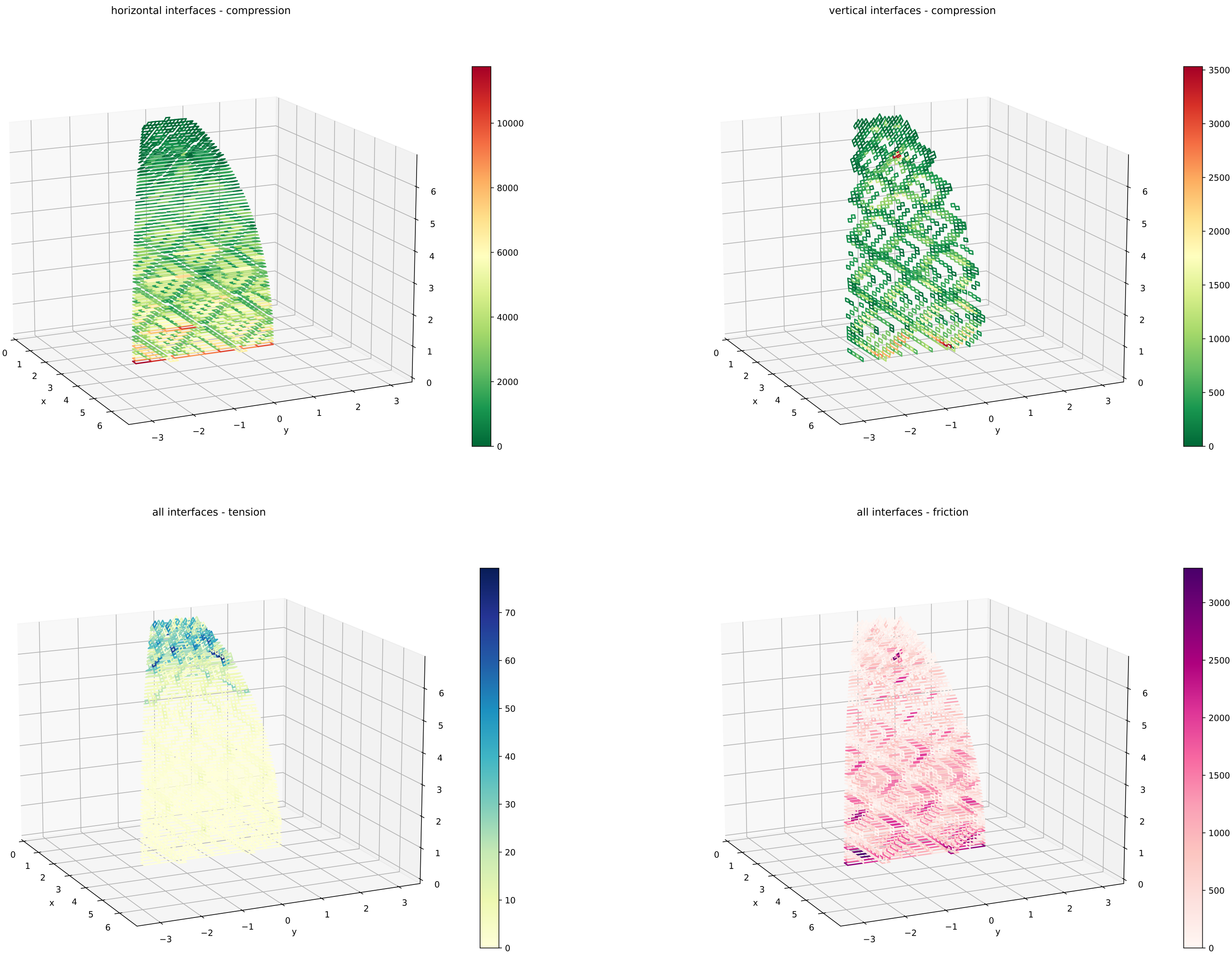

Figure 4.4: Force plots of compression (top), tension (left) and friction (right) for structure 2.1

4.2.3 Structure 3.1: Compressed Dome Section

Table 4.6: RBE results of structure 2.2

structure	feasible	obj. function value	MoI	weight	MoI factor
2.1	Yes	$0.3 \cdot 10^6$	$139.9 \cdot 10^3$	$77.9 \cdot 10^3$	1.795

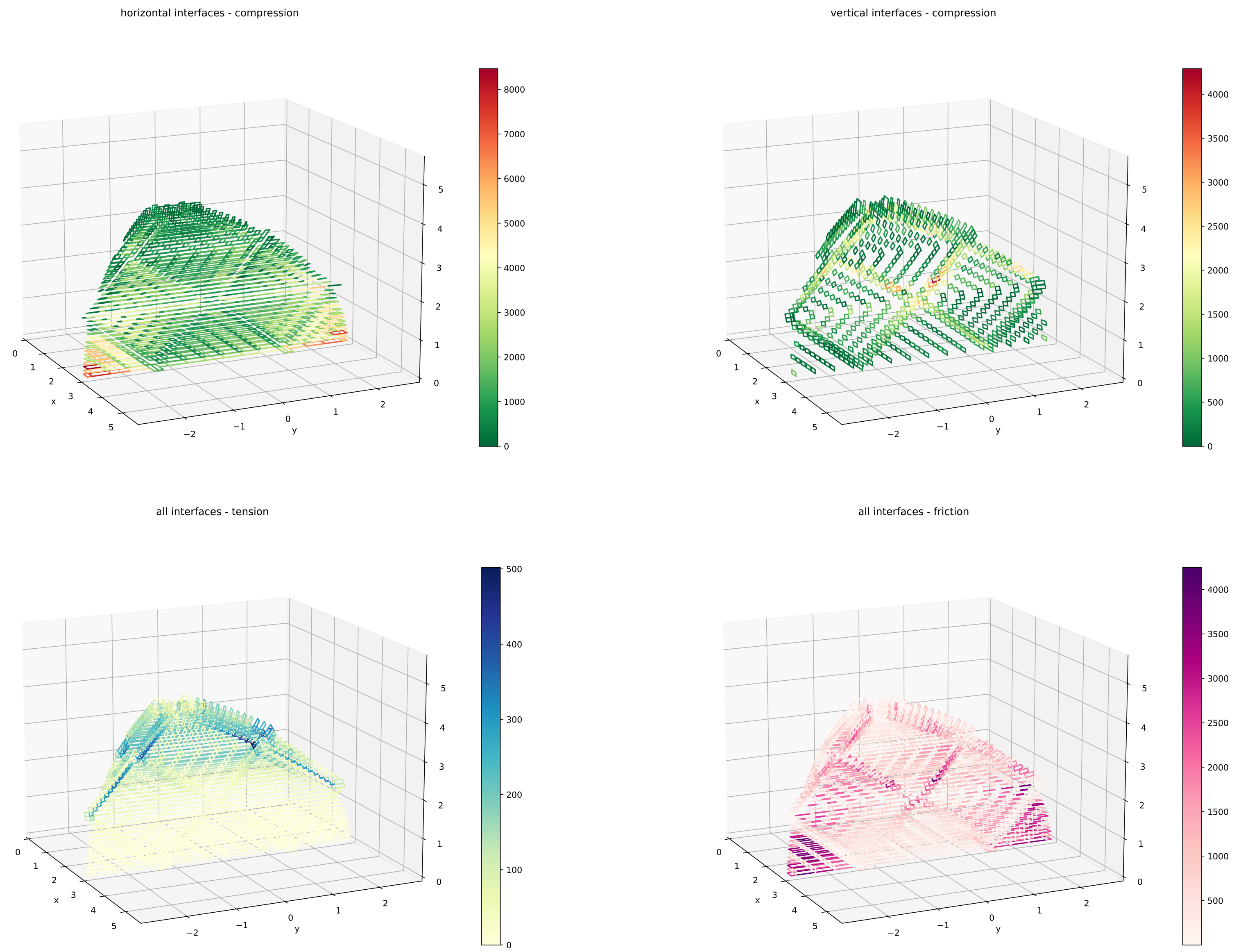

Figure 4.5: Force plots of compression (top), tension (left) and friction (right) for structure 3.1

4.2.4 Structure 1.2: Spherical Dome

Table 4.7: RBE results of structure 1.3

structure	feasible	obj. function value	MoI	weight	MoI factor
1.2	Yes	$0.5 \cdot 10^6$	~ 0	$102.6 \cdot 10^3$	0

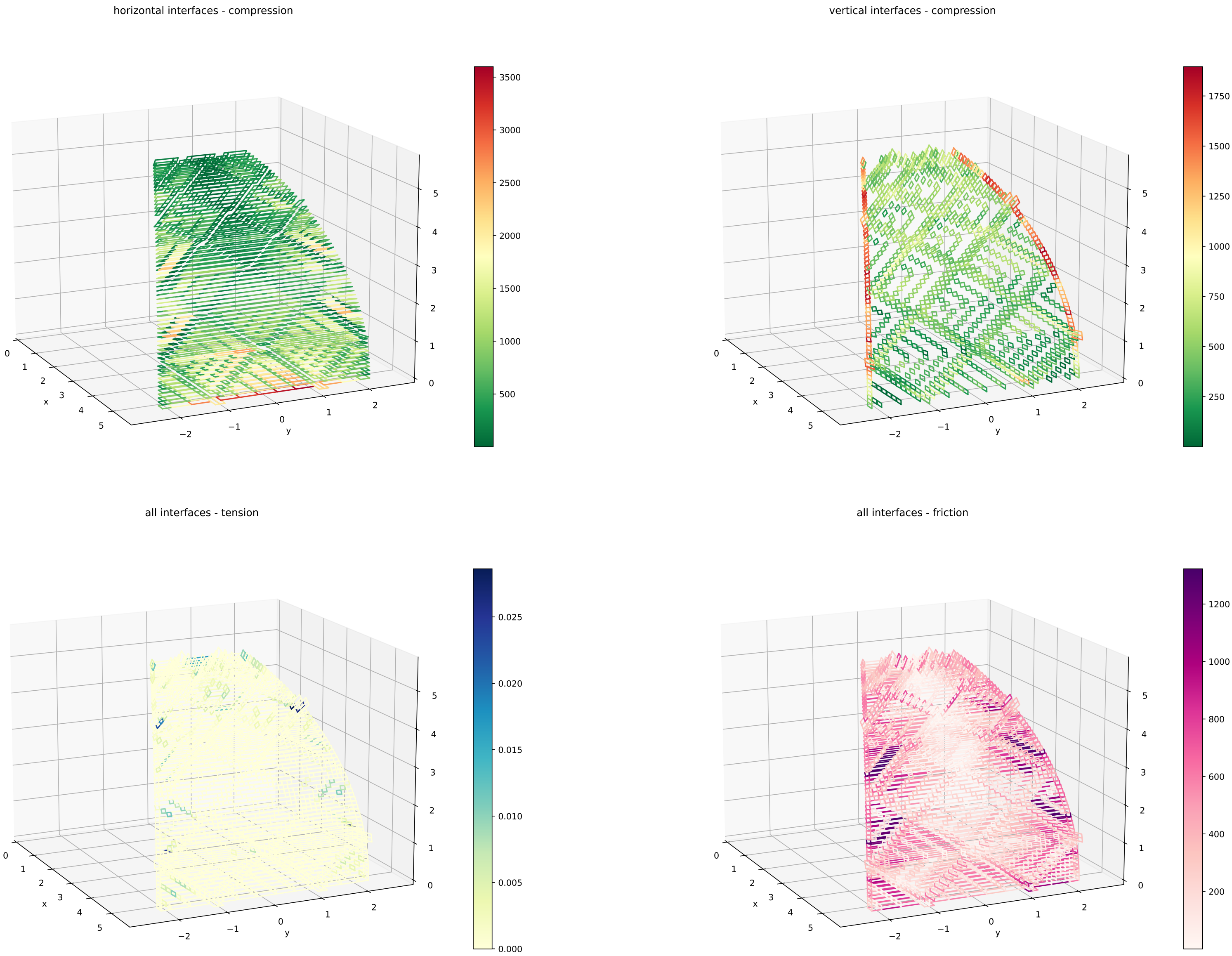

Figure 4.6: Force plots of compression (top), tension (left) and friction (right) for structure 1.2

4.2.5 Structure 2.2: Stretched Dome

Table 4.8: RBE results of structure 2.2

structure	feasible	obj. function value	MoI	weight	MoI factor
2.2	Yes	$0.3 \cdot 10^6$	~ 0	$102.9 \cdot 10^3$	0

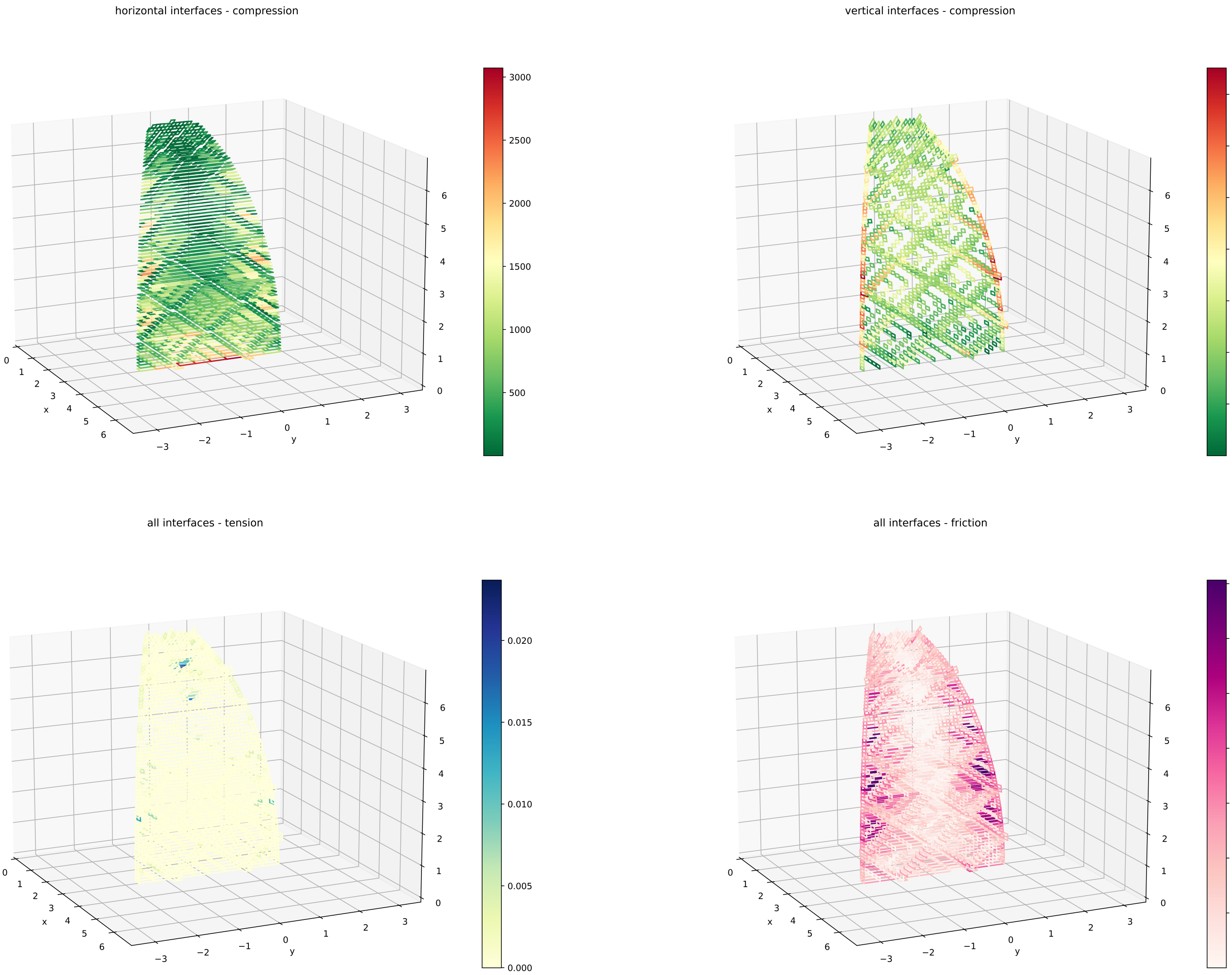

Figure 4.7: Force plots of compression (top), tension (left) and friction (right) for structure 2.2

4.2.6 Structure 3.2: Compressed Dome

Table 4.9: RBE results of structure 2.2

structure	feasible	obj. function value	MoI	weight	MoI factor
3	Yes	$0.3 \cdot 10^6$	~ 0	$77.9 \cdot 10^3$	0

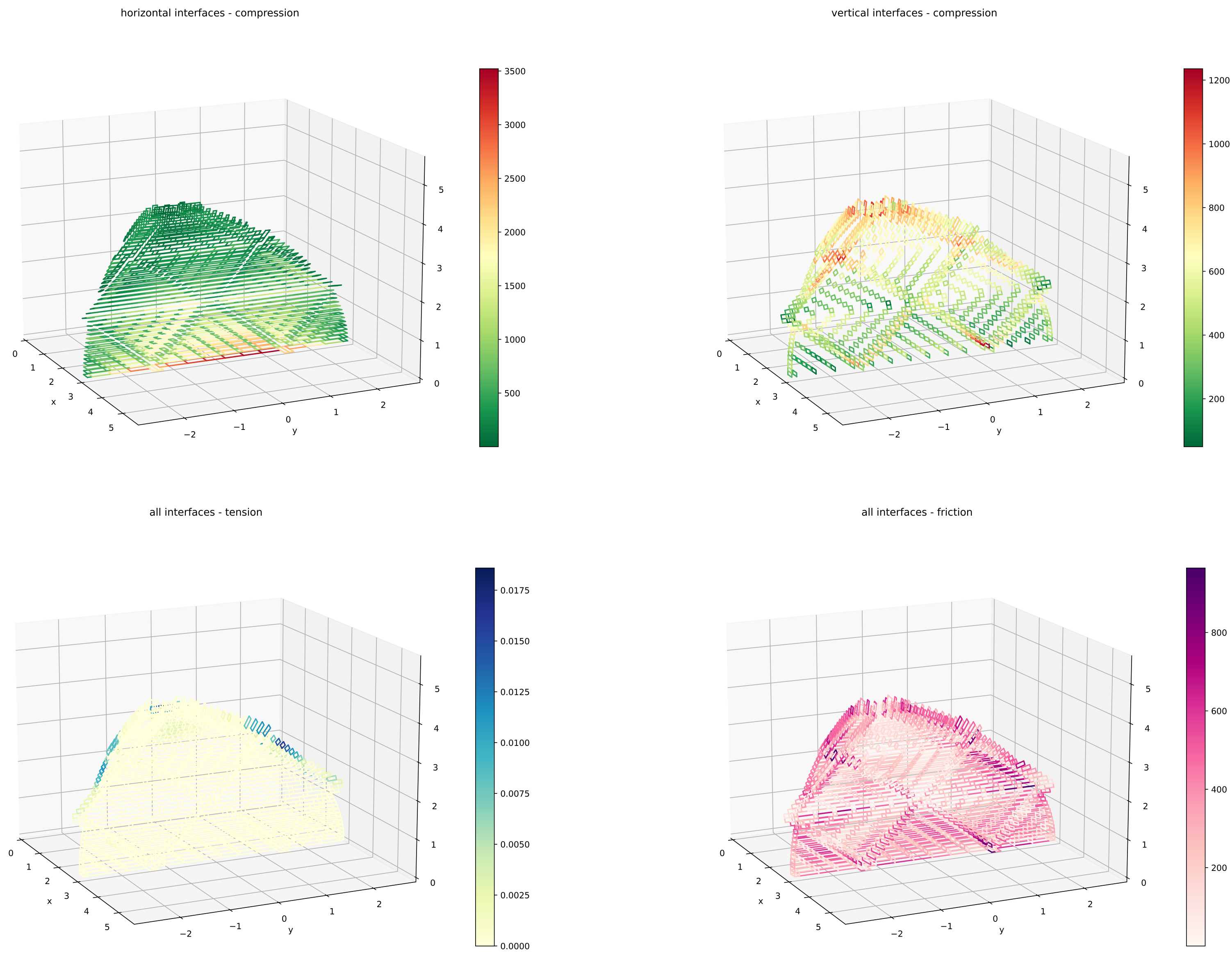

Figure 4.8: Force plots of compression (top), tension (left) and friction (right) for structure 3.2

4.2.7 Full Dome Plots

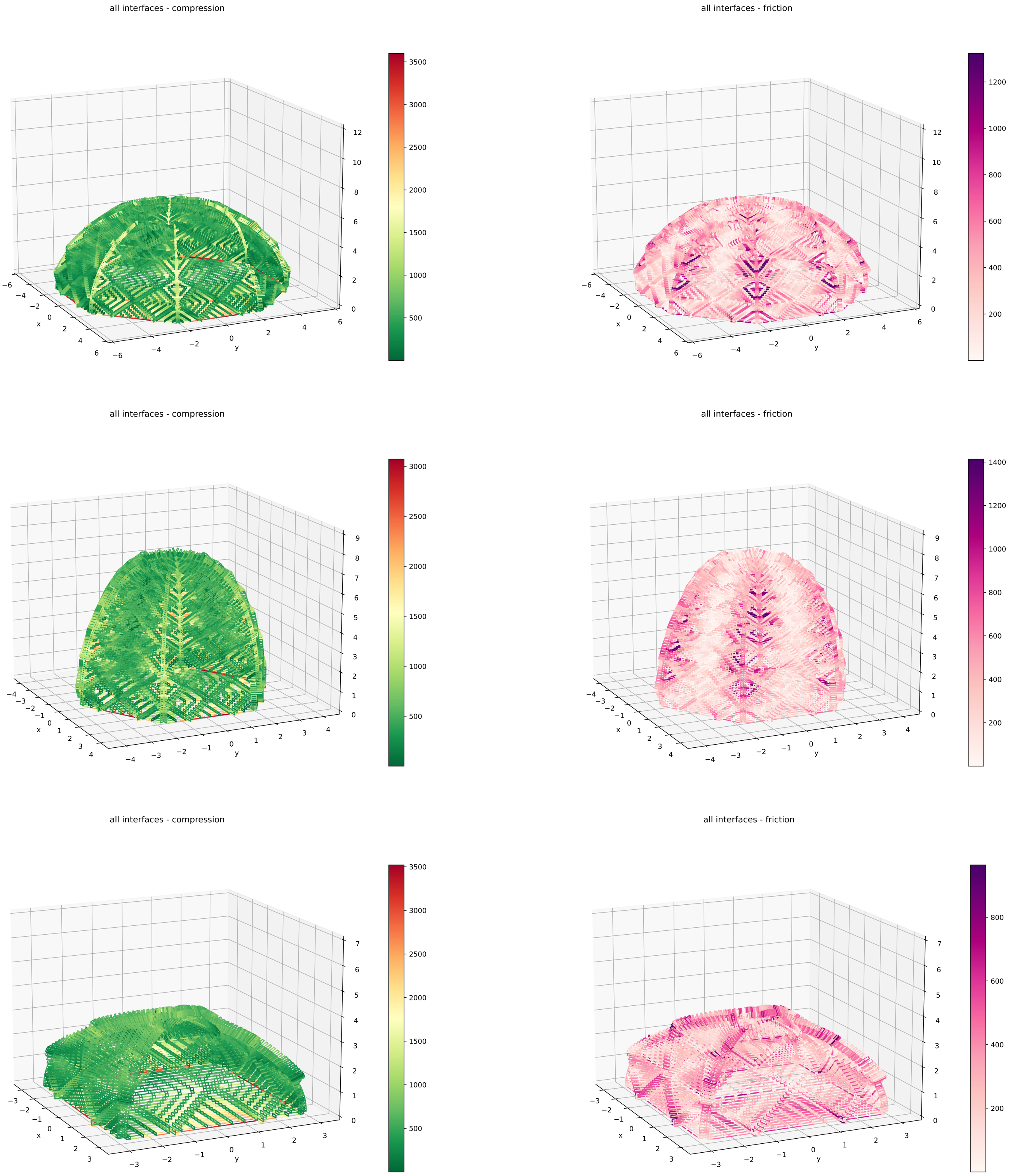

Figure 4.9: Full dome plots for all three structures.

Chapter 5

Discussion

5.1 RBE Verification

5.1.1 Straight Wall Test

Figure 4.1 shows that the meridional compression forces flowing through the horizontal interfaces gradually increase. They increase significantly at the lowest level which can be explained by the large ground floor interface. Tension and friction forces are negligible, as expected. Therefore, the conditions that were defined in section 3.3.6 for the straight wall test have been satisfied. The RBE for straight wall structure seems to be computed correctly.

5.1.2 MoI Test

As the expectation stated in section 3.3.6, the free standing dome section with friction coefficient 0.7 is deemed infeasible. The MoI of the structure that has been computed with friction coefficient 1.5 has decreased drastically, however this result has no physical implications, as friction coefficients above 1 are generally rare. As for the results computed with coefficient 0.3, one can observe a large increase in tensile forces. This does not seem obvious at first glance, but the assigned values to the color spectrum provide some needed context. Overall, a negative correlation between the friction coefficient and the MoI is clearly shown. Therefore, the conditions for the MoI test detailed in section 3.3.6 have been satisfied and the RBE computation delivers the expected results.

5.2 Boundary Conditions

The main takeaway here is that the addition of boundary conditions results in feasibility for all three structures. It seems that the boundary interfaces themselves are the crucial part, as the hoop forces decrease with the number of walls n, if one recalls equation (3.19). As a matter of fact, the hoop forces disappear completely for structure 3.2, displayed in figure 4.8, as the four walls are aligned at right angles. This is evidenced by a decrease in compression for the boundary interfaces from structure 1.2 (4.6) to 3.2 (4.8). Nevertheless, all dome structures are feasible, suggesting that the boundary interfaces play the pivotal role in ensuring an equilibrated state. The boundary conditions have an effect on the meridional forces as well, which are transmitted by the horizontal interfaces. Without any boundary conditions, the largest meridional forces are located at the bottom corners of the dome sections, e.g. structure 1.1 shown in figure 4.3, functioning as an anchor for the structure. With the addition of boundary conditions, this is no longer necessary as the boundaries are now supported and as a result, the highest compression forces are located at the bottom center of the dome structures, e.g. structure 1.2 shown in figure 4.6. In essence, the boundary conditions redirect the meridional force transmissions and stabilise the structure.

5.3 Double-Helix Pattern

For the dome section plots, e.g. figure 4.3 of structure 1.1, one can observe that the meridional forces are of higher magnitude for interfaces within the rhombi compared to interfaces produced by the pattern bricks. This suggests that the double-helix pattern does not transmit meridional forces through the structure. Inspecting the vertical interfaces in 4.6 however, one can clearly observe that the axial forces are mainly transmitted by the double-helix pattern. This effect is expected to increase for the full dome structures with a double-helix pattern that is not interrupted. This expectation does not come to fruition for all structures. In structure 1.2 shown in 4.6, the hoop forces overshadow the influence of the double-helix pattern. With decreasing hoop forces through structures 2.2 and 3.2, the transmission of the axial forces through the double-helix pattern can be observed more clearly, as evidenced by 4.7 and 4.8. This indicates that the boundary conditions, or more precisely the hoop forces, might not be formulated accurately enough. As for the tensile forces in the infeasible dome section structures, e.g. figure 4.3 of structure 1.1, the infeasibility seems to be focused around the double-helix pattern, suggesting that it is the weak link. Considering that the double-helix pattern is designed for dome structures and not free standing, overhanging walls, this observation is not surprising.

5.4 Geometric Variation

One can observe that the meridional force anchors of the dome sections mentioned in section 5.2 are influenced by the stretch factor of the structure. In figure 4.4 of structure 2.1, the meridional forces are not concentrated at the boundaries and distributed more homogeniously throughout the ground floor interfaces, as the lower part of the structure is resemblant of a straight wall. For structure 3.1 displayed in figure 4.5, the opposite is true. There is a large difference of magnitude for the meridional forces between the ground floor interfaces at the boundaries and the ones located in the center. As for the MoI factor, intuitively one would expect a negative correlation with the stretch factor for dome section structures. However, we can see that the MoI factor of structure 3.1 detailed in table 4.6 is slightly lower than the MoI factor of structure 1.1 shown in table 4.4. This can be explained by the decrease of the number of bricks in the infeasible region of structure 3.1. In structure 1.1, there are more bricks in the infeasible region that contribute to the MoI and the MoI factor does not account for this discrepancy. As we can see, the MoI factor does not tell the whole story in terms of infeasibility. The MoI factor of structure 2.1 detailed in table 4.5 behaves as expected, as most of the structure is in the feasible region and lowers the MoI factor. Generally, one can observe similar features in the force plots of all three dome variations, adjusted or scaled according to their shape. One example would be the friction forces that are mostly concentrated at the boundaries for all dome structures, which can be observed in figure 4.9. Overall, the double-helix pattern seems to have a similar impact on the force distribution for all three input variations and their resulting structures.

5.5 Limitations

As discussed in section 5.2, the hoop forces might not be as accurate as initially thought. It would have been interesting to validate the boundary conditions by computing a small but complete dome model and compare it to the dome section with boundary conditions. However, this was not in scope for the semester thesis. Another limitation is the inability to measure how stable the feasible structures are. The MoI is only applicable to infeasible structures, as the name suggests.

5.6 Future Work

As mentioned above, verifying boundary conditions would be a valuable addition to this thesis. A further extension would be the implementation of a measure of feasibility, determined by applying different external loading scenarios. This would allow a quantitative comparison of feasible domes.

Modeling a dome with radial geometry would create a smooth double-helix pattern, not displaying any edges at the separate section boundaries, which could have an interesting influence on the effectiveness of the double-helix pattern. Optimising the pattern itself based on certain criterias would be an intriguing field to explore.

Chapter 6

Conclusion

At the beginning, it was not entirely clear where this subject would lead. The initial focus was on creating an in-depth study of the double-helix pattern and gain valuable insights on its influence on the transmission of forces. During the course of this thesis, it turned out that the parametric model of the masonry dome, which was initially meant simply as an instrument in order to conduct the RBE computation, had value in itself. Apart from the analytical element of the thesis, it became important to create a tool allowing to easily recreate the study and experiment with new structures. This allows future research on the subject to build upon this study efficiently. Over the course of this thesis, this goal has been achieved without losing sight of the initial objective of acquiring an in-depth understanding of the mechanical properties of the Brunelleschi dome.

Appendix A

Appendix I

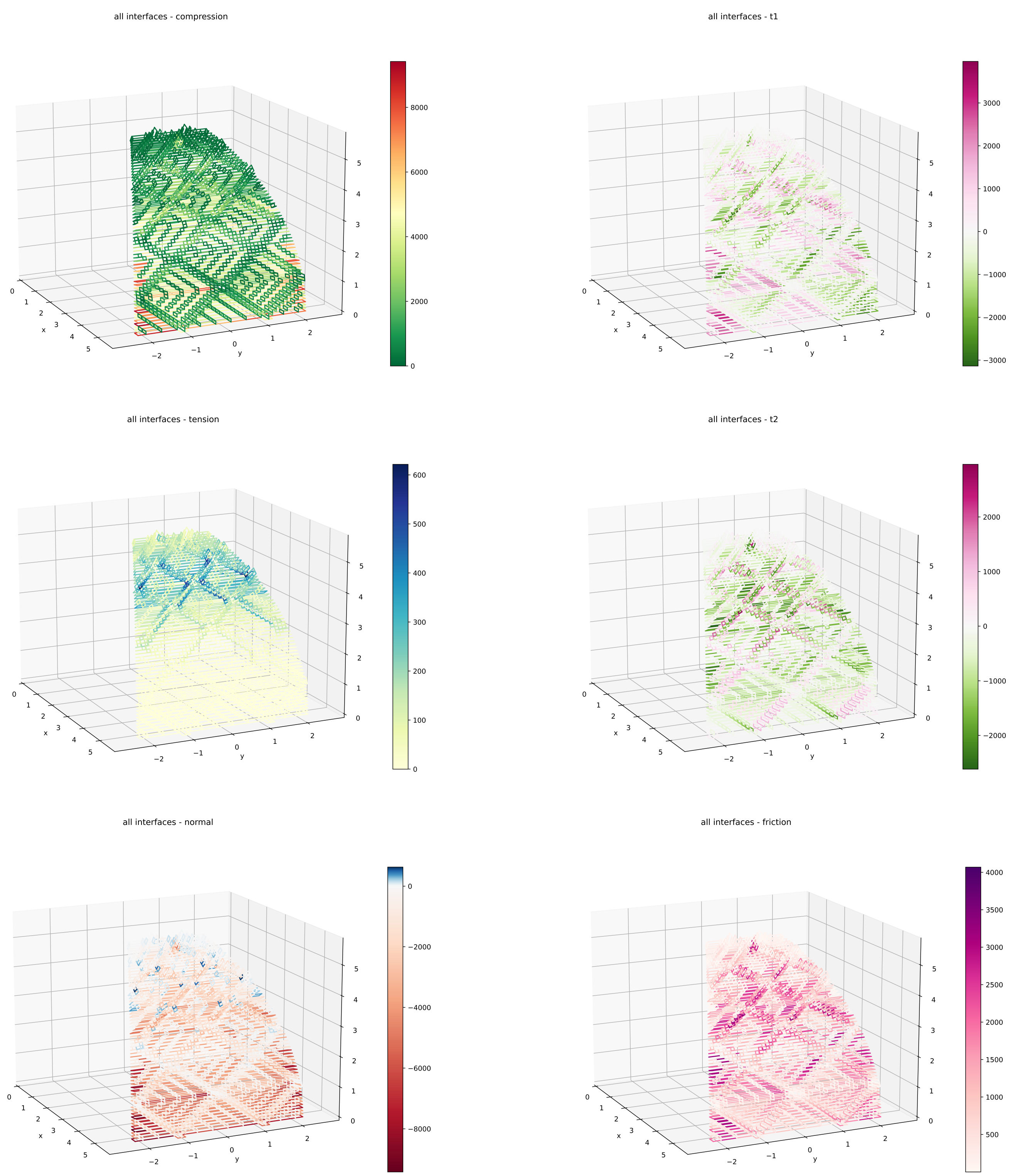

Figure A.1: Structure 1.1 Force Plots

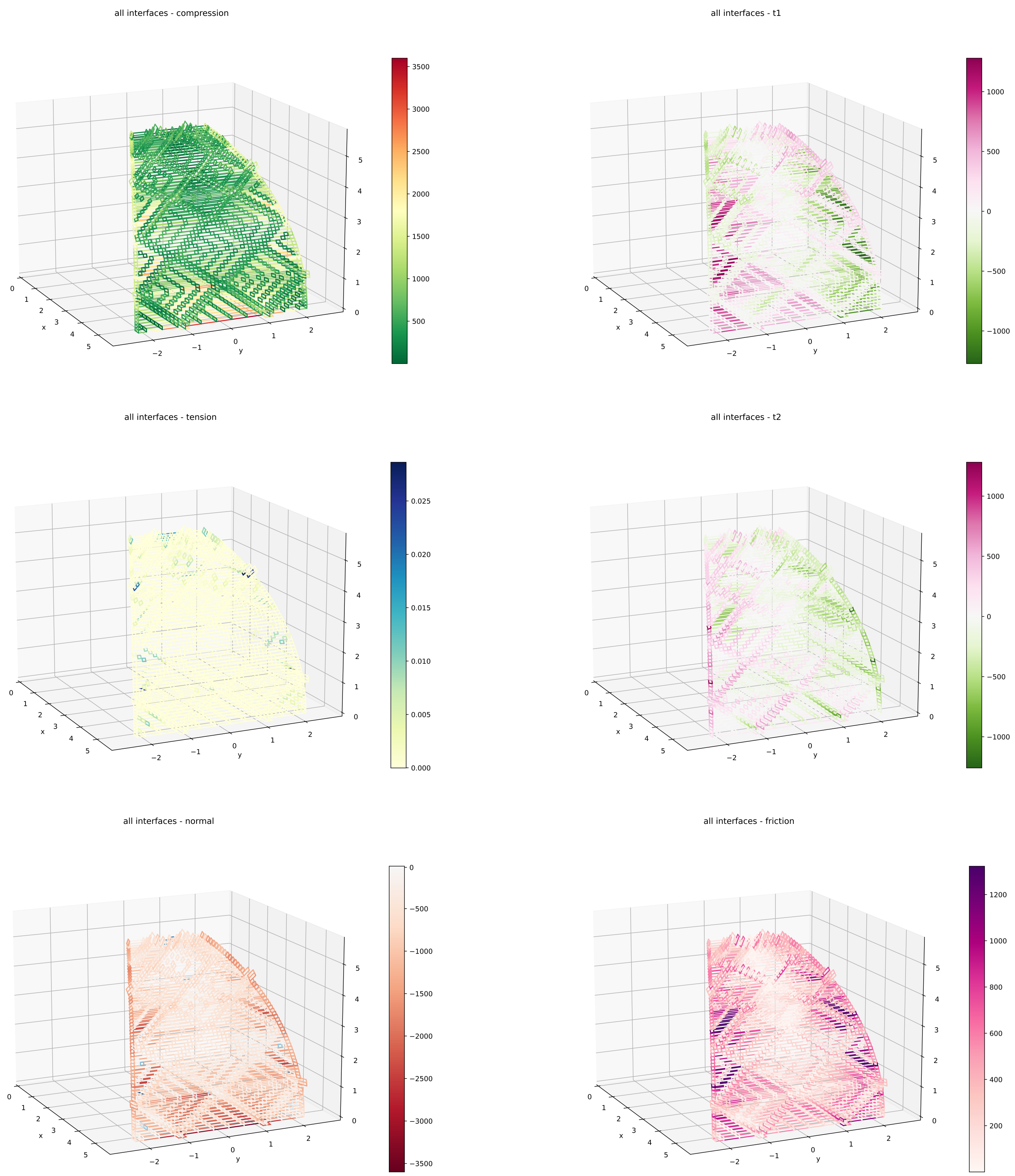

Figure A.2: Structure 1.2 Force Plots

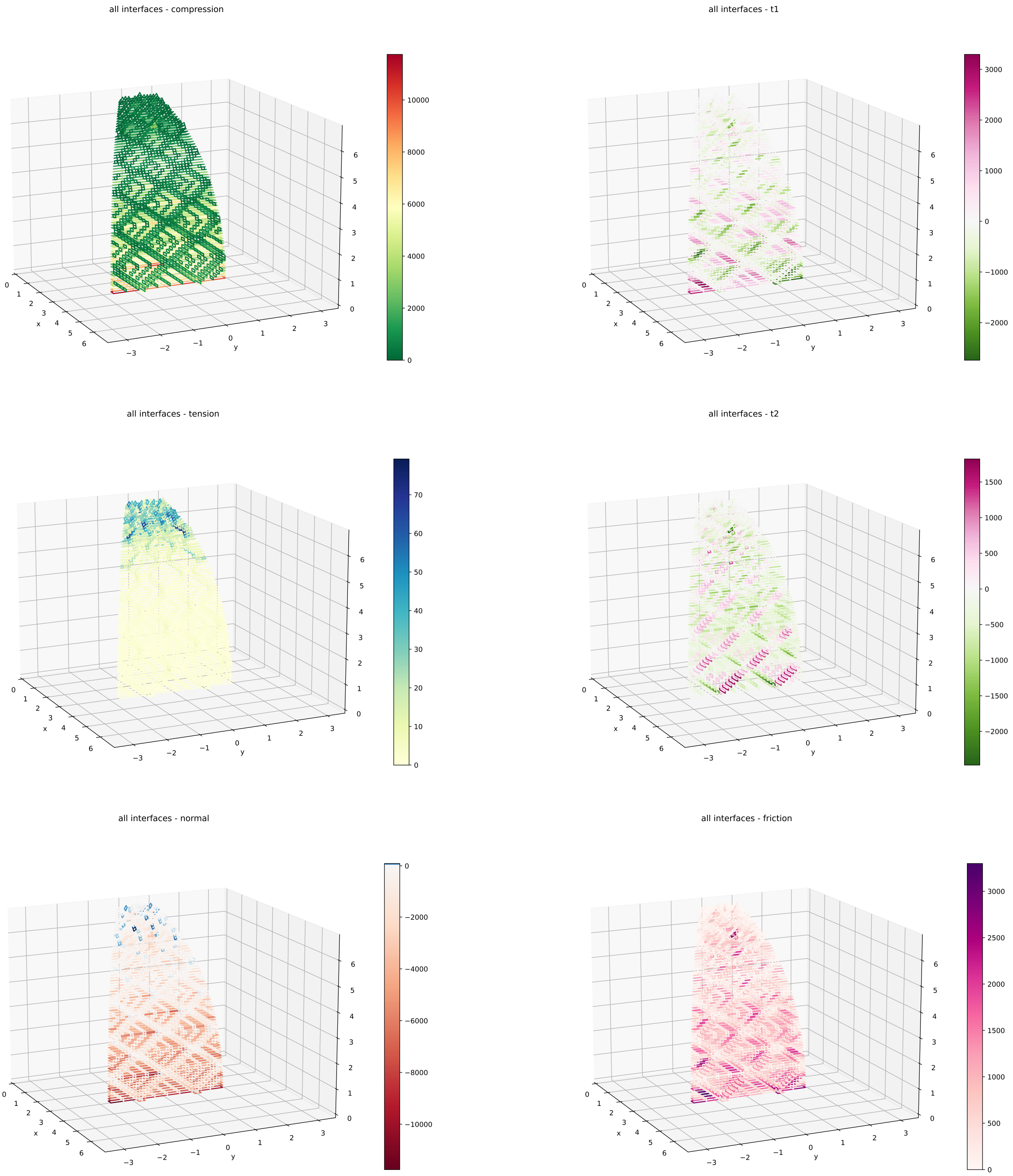

Figure A.3: Structure 2.1 Force Plots

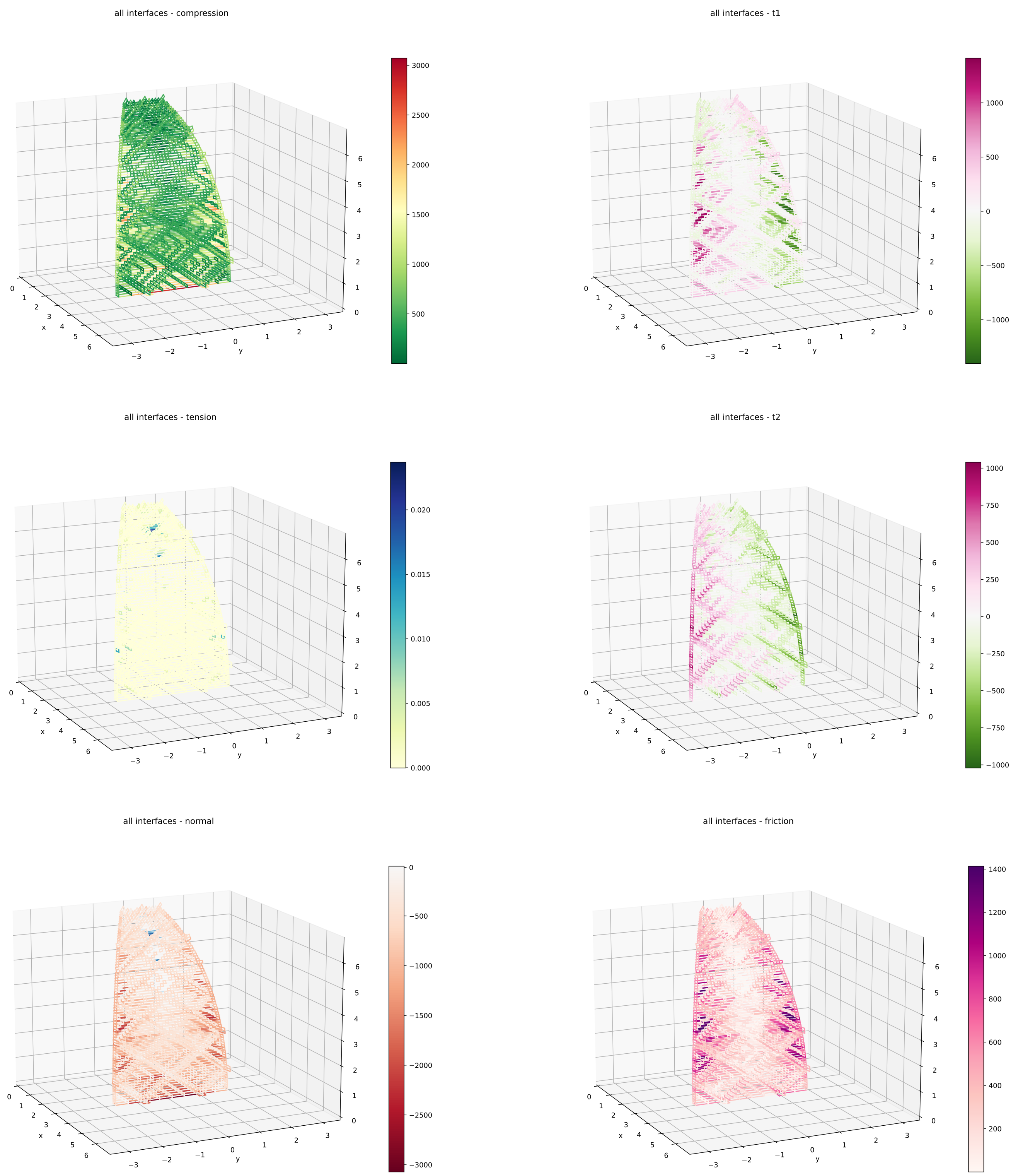

Figure A.4: Structure 2.2 Force Plots

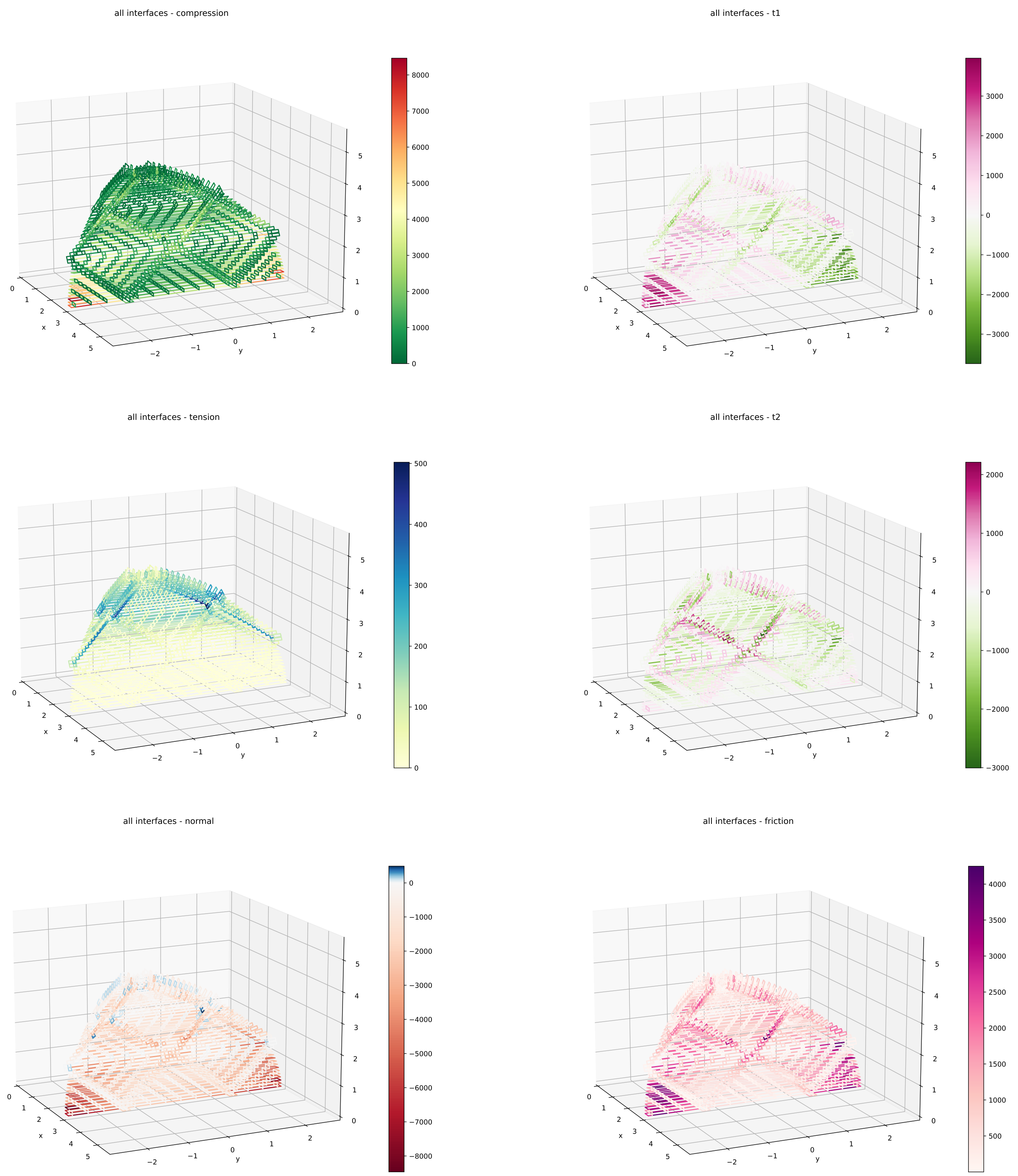

Figure A.5: Structure 3.1 Force Plots

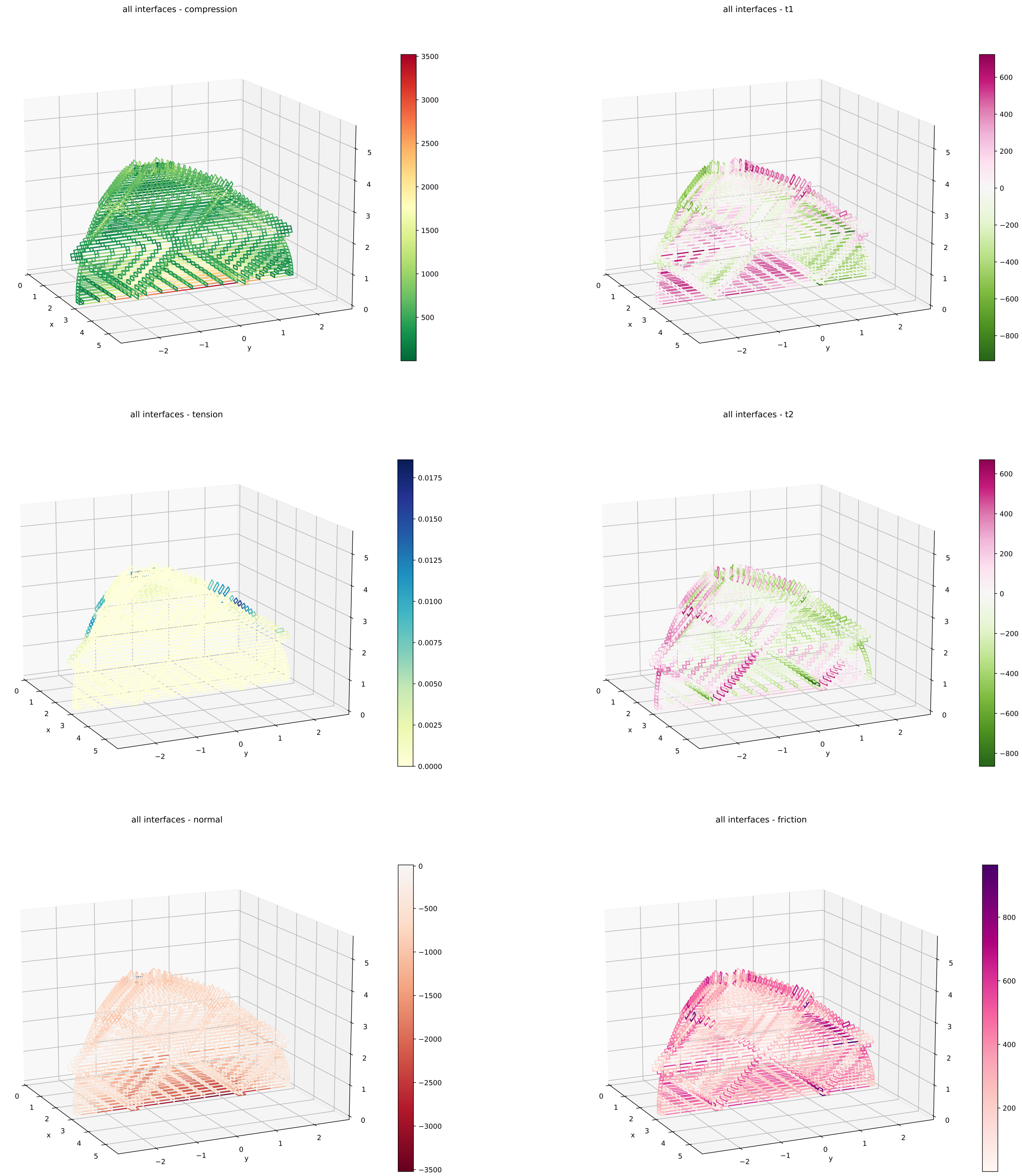

Figure A.6: Structure 3.2 Force Plots

Bibliography

[1] Steven Diamond and Stephen Boyd. Cvxpy: A python-embedded modeling language for convex optimization. *The Journal of Machine Learning Research*, 17(1):2909–2913, 2016.

[2] Jacques Heyman. The stone skeleton. *International Journal of solids and structures*, 2(2):249–279, 1966.

[3] Gene Ting-Chun Kao, Antonino Iannuzzo, Stelian Coros, Tom Van Mele, and Philippe Block. Understanding rigid-block equilibrium method via mathematical programming. *Proceedings of the Institution of Civil Engineers-Engineering and Computational Mechanics*, pages 1–39, 2021.

[4] Wanda W Lau. *Equilibrium analysis of masonry domes.* PhD thesis, Massachusetts Institute of Technology, 2006.

[5] Rowland J Mainstone. Brunelleschi's dome of s. maria del fiore and some related structures. *Transactions of the Newcomen Society*, 42(1):107–126, 1969.

[6] Vittorio Paris, Attilio Pizzigoni, and Sigrid Adriaenssens. Statics of self-balancing masonry domes constructed with a cross-herringbone spiraling pattern. *Engineering Structures*, 215:110–440, 2020.

[7] Emily Whiting, John Ochsendorf, and Frédo Durand. Procedural modeling of structurally-sound masonry buildings. In *ACM SIGGRAPH Asia 2009 papers*, pages 1–9. 2009.

[8] Emily Whiting, Hijung Shin, Robert Wang, John Ochsendorf, and Frédo Durand. Structural optimization of 3d masonry buildings. *ACM Transactions on Graphics (TOG)*, 31(6):1–11, 2012.

Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Engineering Design and Computing Laboratory
Prof. Dr. K. Shea

Title of work:

Modeling and RBE Computation of Double-Helix Masonry Dome Structures

Thesis type and date:

Semester Thesis, September 2021

Supervision:

Yu Zhang
Prof. Dr. Kristina Shea

Student:

Name:	Luca Bettermann
E-mail:	lucabe@student.ethz.ch
Legi-Nr.:	15-918-394
Semester:	5

Statement regarding plagiarism:

By signing this statement, I affirm that I have read and signed the Declaration of Originality, independently produced this paper, and adhered to the general practice of source citation in this subject-area.

Declaration of Originality:

`http://www.ethz.ch/faculty/exams/plagiarism/confirmation_en.pdf`

Zurich, 30. 8. 2022: ____________________